UNIVERSITY OF THE PHILIPPINES MANILA
COLLEGE OF ARTS AND SCIENCES
DEPARTMENT OF PHYSICAL SCIENCES AND MATHEMATICS

TIMEKEEPING ATTENDANCE MONITORING SYSTEM (TAMS) FOR THE AGENCY EMPLOYEES OF SUPERVALUE, INC.

A special problem in partial fulfillment
of the requirements for the degree of
Bachelor of Science in Computer Science

Submitted by:
Marian Felice E. Yarisantos
June 2015

Permission is given for the following people to have access to this SP:

Available to the general public	Yes
Available only after consultation with author/SP adviser	No
Available only to those bound by confidentiality agreement	No

ACCEPTANCE SHEET

The Special Problem entitled "Timekeeping Attendance Monitoring System (TAMS) for the Agency Employees of Supervalue, Inc." prepared and submitted by Marian Felice E. Yarisantos in partial fulfilment of the requirements for the degree of Bachelor of Science in Computer Science has been examined and be recommended for acceptance.

_____

Bernie B. Terrado, M. Sc. ***(candidate)***
Adviser

EXAMINERS:

	Approved	**Disapproved**
1. Gregorio B. Baes, Ph.D. *(candidate)*	_____	_____
2. Avegail D. Carpio, M.Sc.	_____	_____
3. Richard Bryann L. Chua, M.Sc.	_____	_____
4. Perlita E. Gasmen, M.Sc.	_____	_____
5. Ma. Sheila A. Magboo, M.Sc.	_____	_____
6. Vincent Peter C. Magboo, M.D., M.Sc.	_____	_____

Accepted and approved as partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science.

_____

Ma. Sheila A. Magboo, M.Sc.
Unit Head
Mathematical and Computing Sciences Unit
Department of Physical Sciences
and Mathematics

_____

Marcelina B. Lirazan, Ph.D.
Chair
Department of Physical Sciences
and Mathematics

_____

Alex C. Gonzaga, Ph.D., Dr.Eng.
Dean
College of Arts and Science

Abstract

For many years, Supervalue, Inc. or SM Supermarket has been catering extensive line of products and innovative services to what its customers need and want. Since growth is inevitable, there is a need in the usage of technology for business to increase efficiency and accuracy. One application of technology in business is in its attendance monitoring. Without proper computations of the employees' attendance, the company cannot give the right compensation for their employees. This system provides the employees' attendance computations and compensations needed for the employees' evaluation.

Contents

List of Figures

List of Tables

I. Introduction

A. Background of the Study

Growth is inevitable. Since then, there is a need in the usage of technology for business processes to increase accuracy and efficiency. In using automation, not only they can reduce operational costs to fulfil their tasks but also they can increase their productivity rate.[1]

Human error constitutes a large percentage of timekeeping mismanagement. A large company with huge manpower might be concerned about the risks of having incorrect data. This paved way for an automated timekeeping system which lessens the risk of human involved errors.

Attendance monitoring has been a crucial part of any company's management. A part of employee evaluation is his attendance. Inaccuracy in the attendance monitoring can be a tribute to a company's expense. [2] Without proper computation of the employees' attendance, a company cannot give the right compensation for their employees.

Supervalue, Inc. or SM Supermarket is a supermarket chain that caters extensive line of products and innovative service to what its customers need and want. Establishing its first store in 1985 at SM Makati it has now expanded to 38 stores nationwide excluding its standalone store – Savemore Market.[3]

Supervalue, Inc. relies to the agency of providing manpower across its regional warehouses. Manual timekeeping is still used in the attendance monitoring of the agency employees. They still use timecards and Bundy clocks to record their daily clock ins and outs.

B. Statement of the Problem

Over the years, SM Food Retail Group has expanded itself to more branches having the need of software that is customized for their timekeeping process. Although the company's been using an automated attendance monitoring on the regular employees, there are a little less number of third party employees that also needs to be automated on its timekeeping.

Currently, the warehouses still uses manual computation for all the attendance monitoring of their third party employees. Since manual evaluation is still used, it is still prone to human errors, i.e. wrong input, wrong computation. Thus, not delivering it on time for the accounting staff to finalize the actual payroll.

C. Objectives

The aim of the study is to:

- Present to users a friendly user tool that will handle all the attendance transactions that includes:
    a. Total hours worked (regular, over time, rest day, holidays, night differential)
    b. Total hours of tardiness and undertime
    c. Total number of absences
    d. Total hours of overtime
    e. Total number of sick and vacation leaves
    f. Total amount of employee compensation with payslip
    g. Total number of employee violation
- Allow users to input necessary data relevant to the assessment of the employees' attendance. This includes:

a. Employee profile (including the schedule)

b. Daily clock-ins and clock-outs including break times

c. Filed leaves

d. Tardiness and undertime

e. Overtime

f. Change in employee schedule

g. Employee violation

- Provide necessary outputs and reports that includes:

a. Daily Time Record (DTR) per employee

b. Attendance Summary

- By Employee
- By Department

c. Compensation List per employee

d. Payslip

e. Absenteeism Report

- Daily
- Monthly

f. Over Time Report

- Daily
- Monthly

g. Violation Report

- Automatic printing of employees with violations
    - Failure to punch whether IN/OUT

        - Failure to comply with the prescribed 20 minute break time period without valid reason
        - Abuse of the regular break period either by taking longer time than allowable or taking unscheduled break time
        - Failure to register time entry
        - Employees with more than five times tardiness
        - Employees with unauthorized absences
            1. Did not file leave of absences
            2. Absences not approved by the department manager
        - Abandonment of work
    - Automatic printing of employees with sanctions
        - Verbal Warning
        - Written Warning
        - Suspension
        - Dismissal
- Provide users a "Help" feature to aid them in using the system.

D. Significance of the Study

During the manual computation, HR staff take long duration of work hours to compute for all the attendance transactions of the third party employees in the company. Since the company is getting bigger and bigger, number of people hired also increases resulting to longer hours of computation thus delaying the timekeeping reports and the increase of operational costs for the company.

Automation is significant in the evaluation of attendance of employees. It makes the computations with accuracy, can increase productivity and reduce costs.[4]

Accuracy in results is formed through a reliable data. Although the system may not be automated thoroughly, like using biometrics that transfers data automatically, clock ins and outs are encoded much easier lessening the cause of much errors. [5]

In comparison with the manual timekeeping, automation of timekeeping gives less tedious work for the staff. They can make reports in an instant without having to pay lots of time and effort so they can be productive at the same time. Lessening the working hours for overtime pay for the staff means reducing the operational costs of the company.

While the agency of the employees should be the one shouldering their time computations, the company, however makes their own computations as well since they pay the agency with the exact amount of the services the third party employees made.

The timekeeping system that the company uses with their regular employees cannot be altered because of the agreement with the software seller. Thus adding a feature to add the attendance monitoring of the agency employees to that system won't be feasible and may cost them a lot.

E. Scope and Limitation

- This system will use Visual Basic.NET as the interface and MS SQL for its database.
- ITextSharp will be utilized for the results of the processes.

II. Review of Related Literature

Precise timing applications pervade our society. All of mankind may use it effectively and efficiently since the world is evolving into the age where precise timing is at the heart of managing the flow of information.[4]

Without the benefit of computers in the workplace, selecting demographics for identifying and approaching your market would consume an enormous amount of staff time and energy. Computers provide cost-effective functions for production, sales and shipping. Manufacturing companies utilize computers for producing anything from ready-to-wear clothing to automobiles.[6] Computer gives a different feeling about what is happening in the company. Business will be highly competitive and innovative because the computer provides instant information.[7]

For many businesses automating the collecting of employee attendance data when employees begin their shifts is still a manual process. Even business that has automated or computerized processes such as scheduling, inventory, purchasing, general ledger and payroll processing still use manual method such as time cards sheets to collect time and attendance data.[8]

Payroll accounting was the first commercial area to become widely computerized. The calculation of wages and salaries involves a number of variables but common factors which relate to the personal details of each employee, such as gross pay, tax code, insurance premium, other perks, etc, are handled by the computer.[9]

Although Cupido's study does not result in an actual system, it lets us see through the opinions of the employees in the Stellenbosch Municipality. The participants of the change initiative have to experience the sense of belonging and therefore require ownership to be part of their actions. This can only be made possible if employees understand the essence of change, as

well as the direction that will be taken to reach the change destination. The commencement of a transparently operated pilot project to determine the feasibility of a biometric time and attendance clock emphasises the open boundaries of Stellenbosch Municipality and also diminishes the fear that has developed.[10]

Norhidayah Abdullah conducted a study in 2011 that provides a computerized system for Emasjasa Mechanical and Electrical Engineering Consultant that eliminates the manual system of their payroll. It addresses to solve the problems of lack of efficiency, mixed up documentation, increase in cost, and low level of security. Additionally it has an automatic calculation of the employees' net salary.[11]

The Daily Transaction Monitoring System by Energy Link Corporation aims to eliminate their previous manual computation into an automated one. Previously, they manage their attendance through time cards and bundy clocks and then manually computing the hours worked by their employee. In the new system, the attendance is then encoded in the computer by their staff to be computed automatically.[12]

The calculations on the PMS or Payroll Management System are based on the user provided employees details like basic pay, house rent allowance, loan details and so on. Based on these user inputs the system automatically generates pay slip, pay bills, all schedules for debit and credit payments. This system is developed in such a way to suit for both new and old pension schemes of the central government employees of Nigeria. The use of Java Server Pages (JSP) language for system development allows easy modification of the system design. So the system design can be directly implemented to any other central government organization with slight modifications. [13]

Aside from the usual payroll system that only handles the payroll of a company, Patel's The Payroll Management System also handles employee information management. It keeps the records of the functions performed by the individual employee playing a vital role at the time of performance appraisal. Employee management software can carry out many functions like employee data analysis, employee monitoring, centralized employee database, management of the time sheet, etc.[14]

The Time Attendance System developed by Mohammed Fahed Tayfour in 2008, helps in the tracking of employee attendance that consists of three major parts: first, the employee in which he can enter his attendance manually through the timekeeper or automatically through card reader. Second, is the timekeeper who is responsible for recording the employee's attendance manually. The last is the administrator who has the control of the employee information.[15]

Tracking one's attendance can get a bit tricky. The ASMMA or Attendance System using MyKad for Mobile Application can handle the tracking of the employee's attendance. Employees can view their daily time records through their mobile. Automatically, ASMMA records the check ins and outs of the employees and allows to key in their reason for coming late or absent.[16]

In the Design and Development of a Database for Payroll System, it is hoped that the relational database for the payroll system will help the company to fulfil the objectives of saving operational cost, time and effort of the employees continuously and for the long run. The database for payroll system project will look upon the aspects of the ease in generating and accessing user's payroll information through the use of data definition and manipulation languages and other advanced database techniques. [17]

The EARS project or the Employee Attendance Remuneration System integrates two parts of monitoring attendance into a system. The first part is the Employee Attendance System and the

second part is the Employee Remuneration. The recorded time attendance, as the factor to determine the remuneration, is then used to remunerate the employee of the organizations. The integration of two separate systems combines the factors to remunerate the employee. It may decrease the imprecise of recorded time attendance with the real person before the employee remuneration is done. Thus, the fair, precise and accurate remuneration can be generated to the real employee.[18]

Nowadays, biometrics are often used for time keeping to eliminate the possible scenarios in which an employee may lose their time cards or bar codes in their IDs. The new timekeeping and payroll system of the Blanco Family Academy caters the need to eliminate their bundy clocks into fingerprint scanners and LANs. Their time records are automatically transferred through the LAN to compute for their monthly payroll.[5]

III. Theoretical Framework

A. Brief History of Supervalue, Inc.

SM's got it all. It was the tagline everyone knows about them. SM stands proud as an institution, a store, a mall, a bank, a home from a humble beginning of a shoe store back in 1948. [19] Nearly two decades after the success of the shoe store, they evolved to a full-blown department store selling not only footwear but a variety of fashion wear.

In 1985, SM continued to be a successful business conquering to build their very first mall in North Edsa. At the same time they ventured into the food retail market. The SM Food Retail Group consists of the following: Super Shopping Market, Inc. (SM Hypermarket), Supervalue, Inc. (SM Supermarket) and Sanford_Marketing Corp. (Savemore Market).

Since then, Supervalue, Inc. has been operating SM Supermarket establishing their first branch in Ayala Center, Makati. Today it is known as the largest supermarket chain in the Philippines that offers world class shopping experience with over 88 branches nationwide including its standalone store – Savemore Market.[3]

B. Timekeeping Policies

- **Employee Schedule**

  An employee has a schedule that he needs to comply to for his duties. There are 3 punch ins and 3 punch outs that an employee should follow. This includes:

  - 1st time in when an employee reports for work
  - 1st time for the 1 hour break
  - 2nd time in after the 1 hour break

- 2nd time out for the 30 minute break
- 3rd time in after the 30 minute break
- 3rd time out when the employee finishes his duty

An employee has an hour lunch break that is subtracted from his worked hours. The employee may take the break anytime he wants to as long as it doesn't take longer than an hour.

Also, the employee has a free 30 minute coffee break that is not included in the computations. The employee may take the coffee break anytime he wants to as long as it doesn't take longer than 30 minutes.

a. **Regular Work Hours**

An employee must complete an eight hour duty each day depending on his schedule.

b. **Rest Day**

An employee is entitled for a rest day or day off per week.

c. **Night Differential**

The night differential schedule is from 10PM – 6AM.

d. **Over Time**

Any excess hours from the prescribed 8 hour duty.

e. **Under Time**

- Actual 1st time in of employee after the scheduled time in by 60 – 359 minutes
- Actual 3rd time out of employee before the scheduled time out by 60 – 359 minutes

f. **Tardiness/Late**

Actual of 1st time in of employee exceeds the scheduled time in by 1 – 59 minutes

g. **Half Day**

4 working hours with no breaks in between

h. **Absence**

Zero working hours

i. **Leaves**

An employee is entitled for leaves. There is a certain limit on the number of leaves of an employee depending on their positions.

- Sick Leave
- Vacation Leave
- Paternity Leave

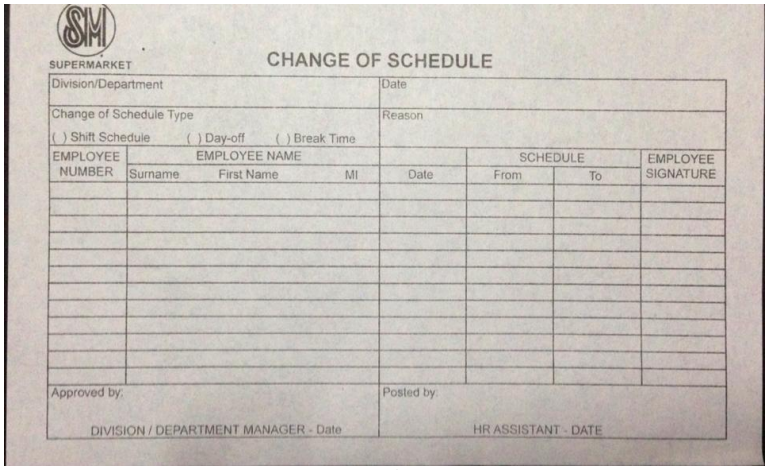
- **Filing**

a. **Change of Schedule**

An employee may change his schedule provided that the information in the form is correct and it is filed accordingly before the date of changed and approved by the department manager.

The change of schedule can be the shift schedule, day off, or break time of an employee. The employee must provide a valid reason for the change of schedule.

The following is a sample of the Change of Schedule form.



SUPERMARKET

CHANGE OF SCHEDULE

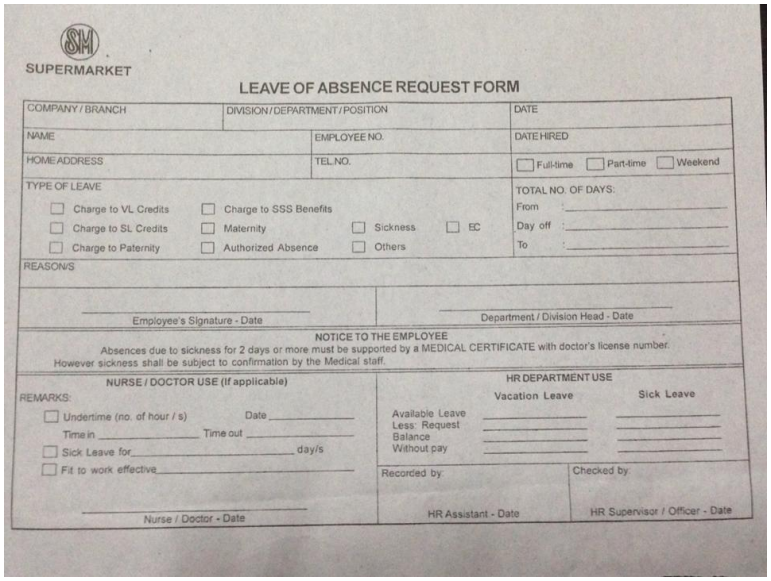
Division/Department				Date		
Change of Schedule Type ( ) Shift Schedule ( ) Day-off ( ) Break Time				Reason		
EMPLOYEE NUMBER	EMPLOYEE NAME				SCHEDULE		EMPLOYEE SIGNATURE
	Surname	First Name	MI	Date	From	To	
Approved by: DIVISION / DEPARTMENT MANAGER - Date				Posted by: HR ASSISTANT - DATE			

Form 1: Change of Schedule Form

b. **Leave of Absence**

An employee is entitled for leave of absences. The employee should file the vacation leave beforehand. For the sick leaves, however, the employee must file it immediately after the leave. The leaves should be first approved by the department manager before the usage (vacation).

The following is a sample form of the Leave of Absence.



SUPERMARKET

LEAVE OF ABSENCE REQUEST FORM

COMPANY / BRANCH	DIVISION / DEPARTMENT / POSITION	DATE
NAME	EMPLOYEE NO.	DATE HIRED
HOME ADDRESS	TEL. NO.	☐ Full-time ☐ Part-time ☐ Weekend
TYPE OF LEAVE ☐ Charge to VL Credits ☐ Charge to SSS Benefits ☐ Charge to SL Credits ☐ Maternity ☐ Sickness ☐ EC ☐ Charge to Paternity ☐ Authorized Absence ☐ Others		TOTAL NO. OF DAYS: From : Day off : To :
REASON/S		
Employee's Signature - Date		Department / Division Head - Date

NOTICE TO THE EMPLOYEE
Absences due to sickness for 2 days or more must be supported by a MEDICAL CERTIFICATE with doctor's license number.
However sickness shall be subject to confirmation by the Medical staff.

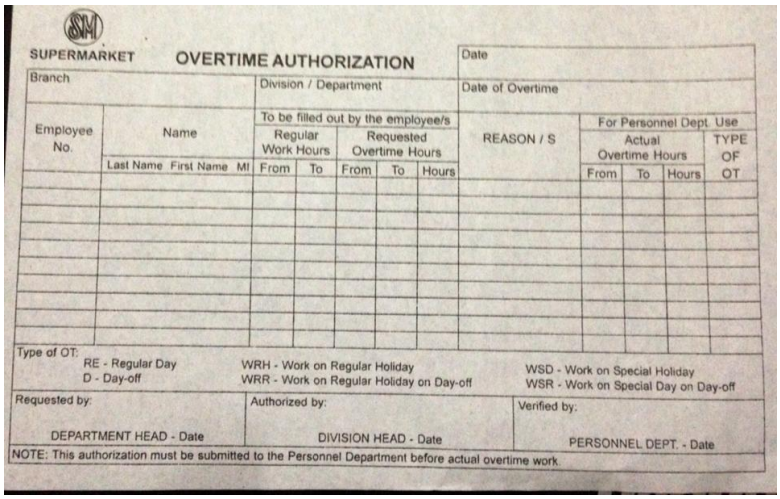
NURSE / DOCTOR USE (If applicable)	HR DEPARTMENT USE		
REMARKS:		Vacation Leave	Sick Leave
☐ Undertime (no. of hour / s) Date _____	Available Leave		
Time in _____ Time out _____	Less: Request		
☐ Sick Leave for _____ day/s	Balance		
☐ Fit to work effective _____	Without pay		
Nurse / Doctor - Date	Recorded by: HR Assistant - Date	Checked by: HR Supervisor / Officer - Date	

Form 2: Leave of Absence Form

c. **Over Time**

Over time hours should also be filed and approved by the department manager before the scheduled over time.

The following figure is an Over Time Authorization Form.



SUPERMARKET **OVERTIME AUTHORIZATION**

Branch	Division / Department	Date of Overtime

Date

Employee No.	Name	To be filled out by the employee/s: Regular Work Hours From	Regular Work Hours To	Requested Overtime Hours From	Requested Overtime Hours To	Requested Overtime Hours Hours	REASON / S	For Personnel Dept. Use: Actual Overtime Hours From	Actual Overtime Hours To	Actual Overtime Hours Hours	TYPE OF OT
	Last Name First Name MI										

Type of OT:
RE - Regular Day
D - Day-off
WRH - Work on Regular Holiday
WRR - Work on Regular Holiday on Day-off
WSD - Work on Special Holiday
WSR - Work on Special Day on Day-off

Requested by:	Authorized by:	Verified by:
DEPARTMENT HEAD - Date	DIVISION HEAD - Date	PERSONNEL DEPT. - Date

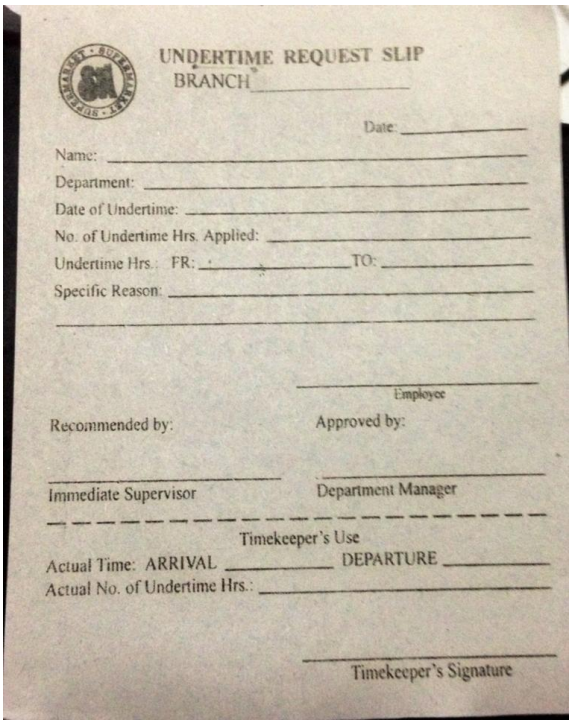
NOTE: This authorization must be submitted to the Personnel Department before actual overtime work.

Form 3: Over Time Authorization Form

d. **Under Time**

Below is a sample of an Under Time Request Slip. Like any other forms, the employee must fill it out with the correct information.

The employee must file an under time request before or right after the date of under time and must be approved by the department manager.



UNDERTIME REQUEST SLIP
BRANCH _____

Date: _____

Name: _____
Department: _____
Date of Undertime: _____
No. of Undertime Hrs. Applied: _____
Undertime Hrs.: FR: _____ TO: _____
Specific Reason: _____
_____

_____
Employee

Recommended by: Approved by:

_____ _____
Immediate Supervisor Department Manager

Timekeeper's Use
Actual Time: ARRIVAL _____ DEPARTURE _____
Actual No. of Undertime Hrs.: _____

_____
Timekeeper's Signature

Form 4: Under Time Request Slip

- **Offenses Against Company Interest**

	Penalties					
Negligence	**1st**	**2nd**	**3rd**	**4th**	**5th**	**6th**
Careless or double punching/swiping of time device. *Prescription period is one calendar month.*	VW	WW	S 3 days	S 6 days	S 12 days	D
Failure to comply with the prescribed 20-minute break time period without valid reason. *Prescription period is one calendar month.*	VW	WW	S 3 days	*S 6 days	*S 12 days	D
Failure to register time entry. *Prescription period is one calendar month.*	WW	S 3 days	S 6 days	S 12 Days	D	
Incurrence of tardiness exceeding four (4) allowable and excusable instances either at reporting time or after any given break periods between working time	5th T - WW	6th T – S 6 days	7th T – S 12 days	8th T – D		
Abuse of the regular break period either by taking longer time than allowable (i.e. returning to work after more than 30 or 15 minutes after each scheduled lunch or snack break respectively) or taking unscheduled break time. *Prescription period is one calendar month.*	WW	S 3 days	S 6 days	S 12 days	Dismissal	

Absence from work for less than five (5) consecutive working days. *Prescription period is one calendar month.*	WW	S 6 days	S 12 days	Dismissal
Absence from work for more than five (5) consecutive working days. *Prescription period is one year from date of last occurrence.*	S 6 days	*S 12 days	Dismissal	
Abandonment of work. Failure to report for work without notice or authorization from supervisor for five (5) or more consecutive working days despite notice to report back for work within specified time.	Dismissal			

VW – Verbal Warning

WW – Written Warning

S – Suspension

D – Dismissal

T – Tardiness

* **-** with a warning of dismissal

C. Computations

* **Adjustment Types**

1. Regular
2. ND – Night Differential
3. OT – Overtime
4. OTND – Overtime Work with Night Differential
5. WRD – Work on Rest Day
6. WRDND – Work on Rest Day with Night Differential
7. OTRD – Overtime Work on Rest Day
8. OTRDND – Overtime Work on Rest Day with Night Differential
9. SH – Work on Special Holiday
10. SHND – Work on Special Holiday with Night Differential

11. SHOT – Overtime Work on Special Holiday

12. SHOTND – Overtime Work on Special Holiday with Night Differential

13. WRH – Work on Regular/Legal Holiday

14. WRHND – Work on Regular Holiday with Night Differential

15. OTRH – Overtime Work on Regular Holiday

16. OTRHND – Overtime Work on Regular Holiday with Night Differential

17. WSR – Work on Special Holiday Falling on Rest Day

18. WSRND – Work on Special Holiday Falling on Rest Day with Night Differential

19. OTSR – Overtime Work on Special Holiday Falling on Rest Day

20. OTRSND – Overtime Work on Special Holiday Falling on Rest Day with Night Differential

21. WRR – Work on Regular Holiday Falling on Rest Day

22. WRRND – Work on Regular Holiday Falling on Rest Day with Night Differential

23. OTRR – Overtime Work on Regular Holiday Falling on Rest Day

24. OTRRND – Overtime Work on Regular Holiday Falling on Rest Day with Night Differential

- **Time Computations**

S_{IN} – Scheduled 1st time in

S_{OUT} – Scheduled 3rd or last time out

IN_1 – Actual 1st time in of employee

IN_2 – Actual 2nd time in of employee after lunch break

IN_3 – Actual 3rd time in of employee after coffee break

OUT_1 – Actual 1st time out of employee for lunch break

OUT_2 – Actual 2nd time out of employee for coffee break

OUT_3 – Actual 3rd or last time out of employee for his shift

ND_{IN} – Start of night differential shift

ND_{OUT} – End of night differential shift

WH – Total number of working hours

WRD – Total number of working hours on a rest day

OT – Total number of overtime hours

OTRD – Total number of overtime hours on a rest day

NDH – Total number of night differential hours

OTND – Total number of overtime with night differential hours

Late – Total number of late/tardy minutes

UT – Total number of undertime minutes

8 HOUR WORK

*(applicable to **Regular, SH, WRH**)*

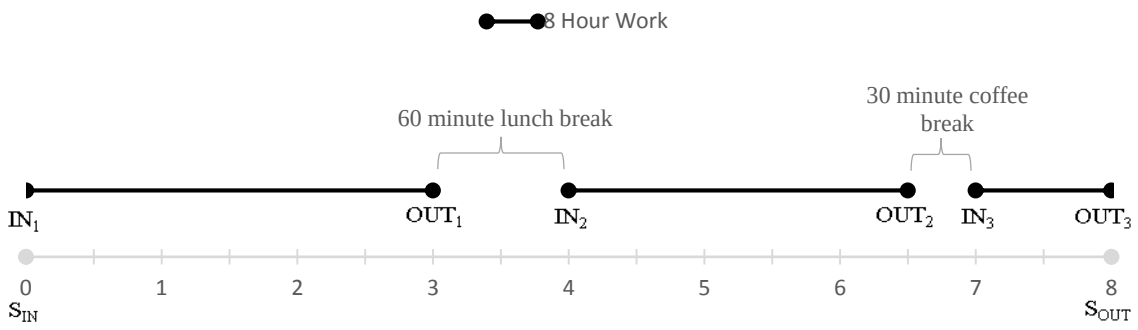


Figure 1: 8 Hour Work timeline

1st: IN_1 must not exceed the S_{IN}.

$$S_{IN} \geq IN_1$$

2nd: Lunch break must not exceed 60 minutes

$$IN_2 - OUT_1 \leq 60$$

3rd: Coffee break must not exceed 30 minutes

$$IN_3 - OUT_2 \leq 30$$

4th: OUT_3 must not be before S_{OUT}

$$S_{OUT} \leq OUT_3$$

If the above conditions are satisfied,

$$\boldsymbol{WH = 8.00}$$

LATE/TARDY

IN_1 exceeded S_{IN} by 1 – 59 minutes

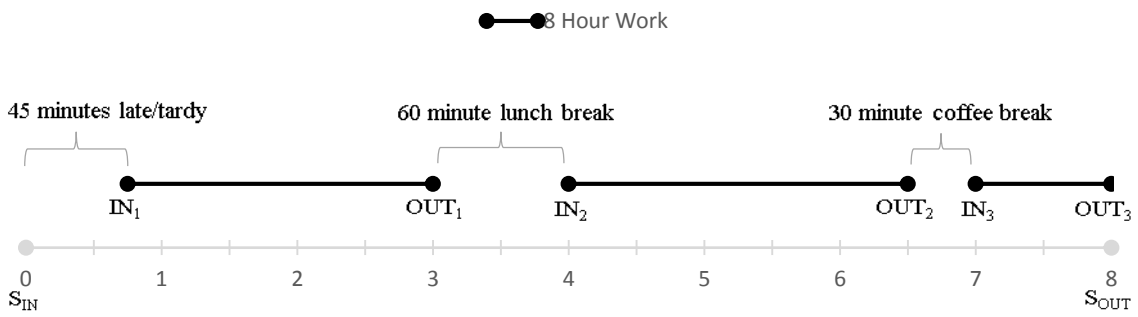


Figure 2: Late/Tardy timeline

$$Late = IN_1 - S_{IN}$$

$$\boldsymbol{WH} = 8.00 - \frac{Late}{60.0}$$

UNDERTIME

IN_1 exceeded S_{IN} by 60 – 239 minutes

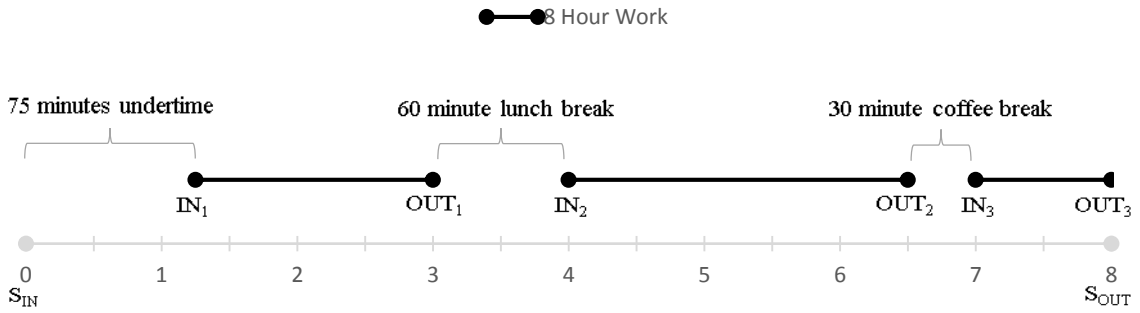


Figure 3: Undertime A timeline

$$UT = IN_1 - S_{IN}$$

$$\boldsymbol{WH} = 8.00 - \frac{UT}{60.0}$$

OR

OUT_3 is before S_{OUT}

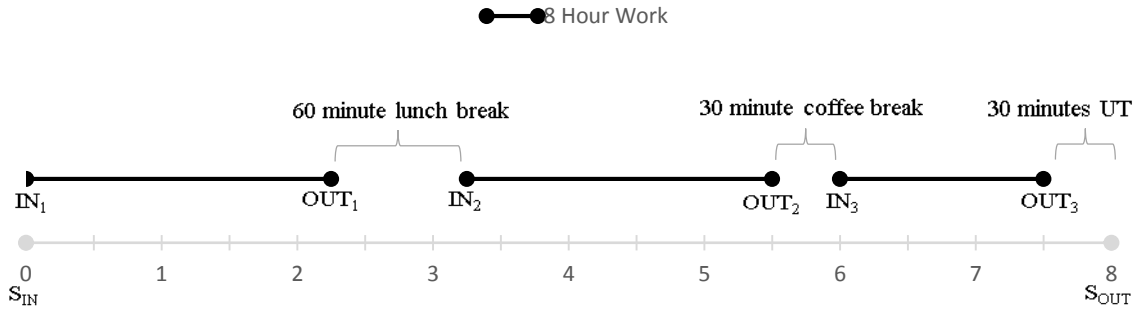


Figure 4: Undertime B timeline

$$UT = S_{OUT} - OUT_3$$

$$\boldsymbol{WH} = 8.00 - \frac{UT}{60.0}$$

OR

Excess minutes of lunch break

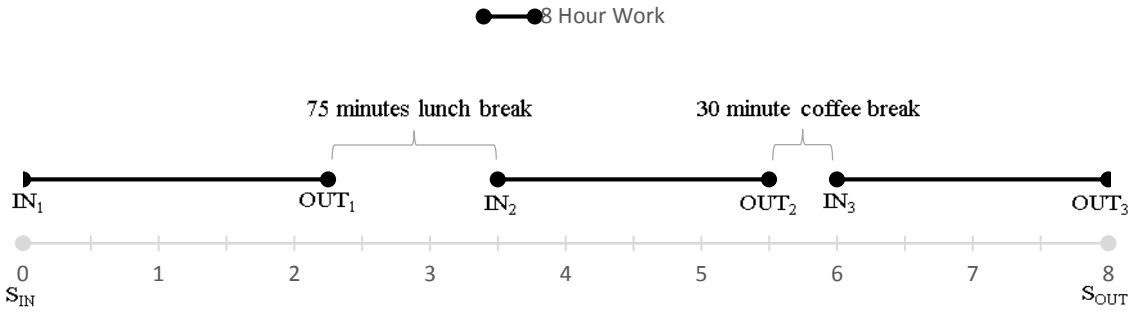


Figure 5: Undertime C timeline

$$UT = (IN_2 - OUT_1) - 60$$

$$\boldsymbol{WH} = 8.00 - \frac{UT}{60.0}$$

OR

Excess minutes of coffee break

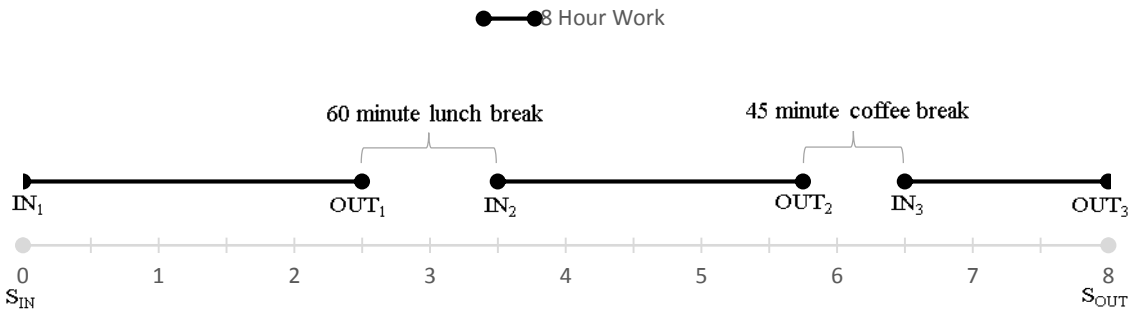


Figure 6: Undertime D timeline

$$UT = (IN_3 - OUT_2) - 30$$

$$\boldsymbol{WH} = 8.00 - \frac{UT}{60.0}$$

HALF DAY

IN_1 exceeded S_{IN} by 240 minutes.

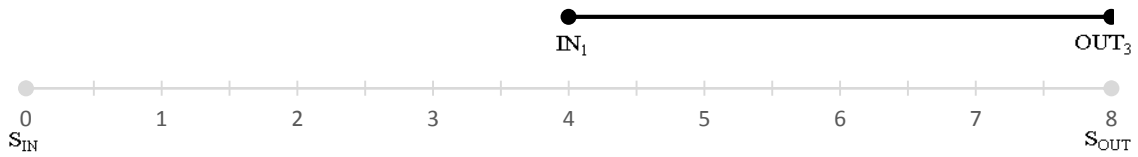


Figure 7: Halfday A timeline

No lunch or coffee break for half day

$$\boldsymbol{WH} = 4.00$$

OR

OUT_3 must be 240 minutes before S_{OUT}

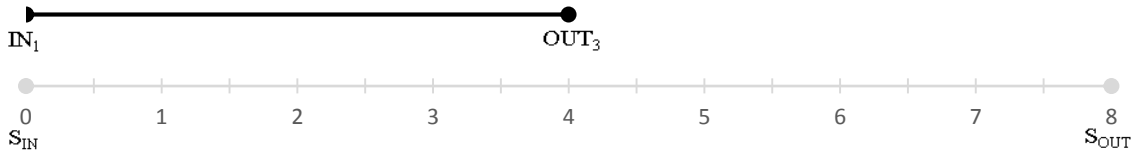


Figure 8: Halfday B timeline

No lunch or coffee break for half day

$$\boldsymbol{WH} = 4.00$$

ABSENT

- Working Hours is zero.
- No time ins and time outs.
- If absent before a regular holiday, the regular holiday has no credit.

$$\boldsymbol{WH} = 0$$

OVERTIME

*(applicable to **OT, SHOT, OTRH**)*

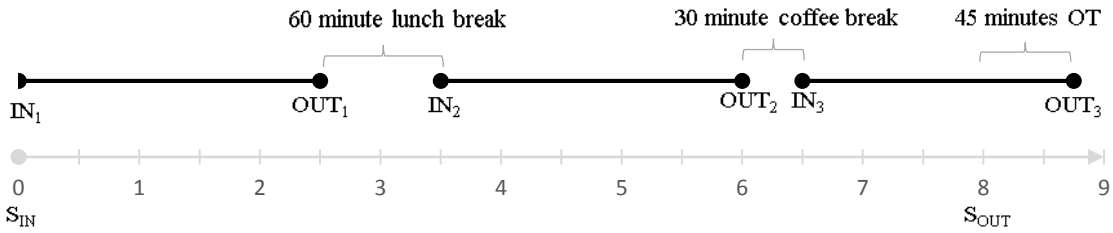


Figure 9: Overtime timeline

$$OT = OUT_3 - S_{OUT}$$

NIGHT DIFFERENTIAL

*(applicable to **ND, SHND, WRHND, WRDND, WSRND, WRRND**)*

- For **ND, SHND** and **WRHND**
  a. If $IN_1 \leq S_{IN}$ then change IN_1 to $\boldsymbol{S_{IN}}$
  b. If $OUT_3 \geq S_{OUT}$ then change OUT_3 to $\boldsymbol{S_{OUT}}$

Case 1:
IN_1 *on or before* $ND_{IN,}$
OUT_3 *between* ND_{IN} *and* ND_{OUT}

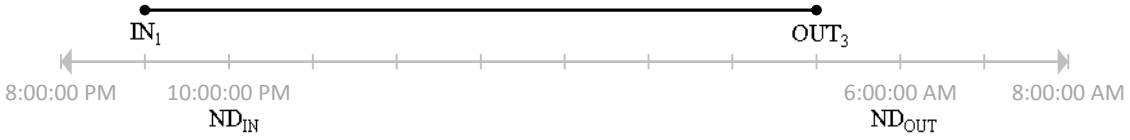


Figure 10: Night Differential Case 1 timeline

Case 1 - Break A:
OUT_1 *between* ND_{IN} *and* $ND_{OUT,}$
IN_2 *between* ND_{IN} *and* ND_{OUT}

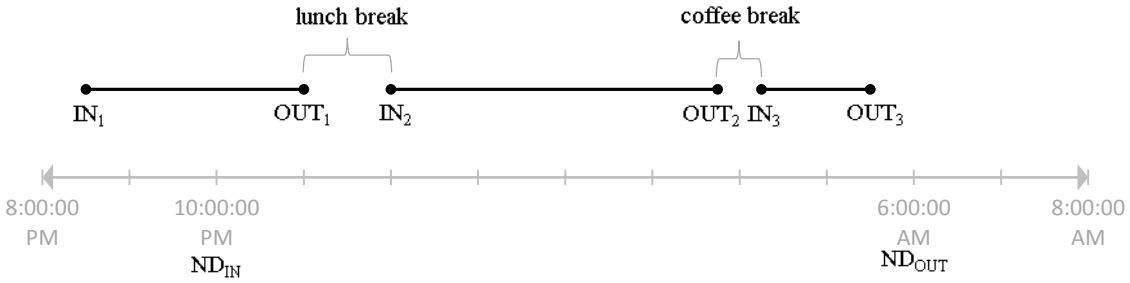


Figure 11: Night Differential Case 1 – Break A timeline

- If lunch break does not exceed 60 minutes and coffee break does not exceed 30 minutes

$$NDH = (OUT_3 - ND_{IN}) - 1.0$$

- If lunch break does not exceed 60 minutes and coffee break exceeds 30 minutes

$$NDH = [(OUT_3 - ND_{IN}) - 1.0] - [(IN_3 - OUT_2) + 0.5]$$

- If lunch break exceeds 60 minutes and coffee break does not exceed 30 minutes

$$NDH = (OUT_3 - IN_2) + (OUT_2 - ND_{IN})$$

- If lunch break exceeds 60 minutes and coffee break exceeds 30 minutes

$$NDH = [(OUT_3 - IN_2) + (OUT_2 - ND_{IN})] - [(IN_3 - OUT_2) + 0.5]$$

Case 1 - Break B:
OUT_1 *on or before* $ND_{IN,}$
IN_2 *between* ND_{IN} *and* ND_{OUT}

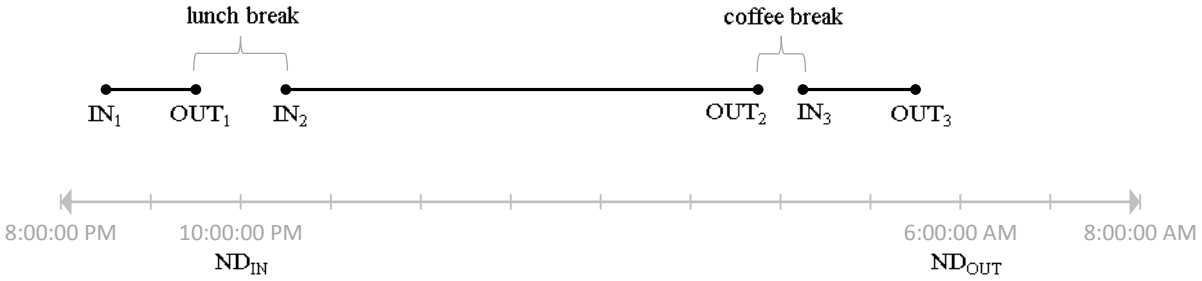


Figure 12: Night Differential Case 1 – Break B timeline

- If lunch break does not exceed 60 minutes and coffee break does not exceed 30 minutes

$$NDH = (OUT_3 - OUT_1) - 1.0$$

- If lunch break does not exceed 60 minutes and coffee break exceeds 30 minutes

$$NDH = [(OUT_3 - OUT_1) - 1.0] - [(IN_3 - OUT_2) + 0.5]$$

- If lunch break exceeds 60 minutes and coffee break does not exceed 30 minutes

$$NDH = (OUT_3 - IN_2)$$

- If lunch break exceeds 60 minutes and coffee break exceeds 30 minutes

$$NDH = [(OUT_3 - IN_2)] - [(IN_3 - OUT_2) + 0.5]$$

Case 1 - Break C:
OUT_1 *before* $ND_{IN,}$
IN_2 *before* ND_{IN}

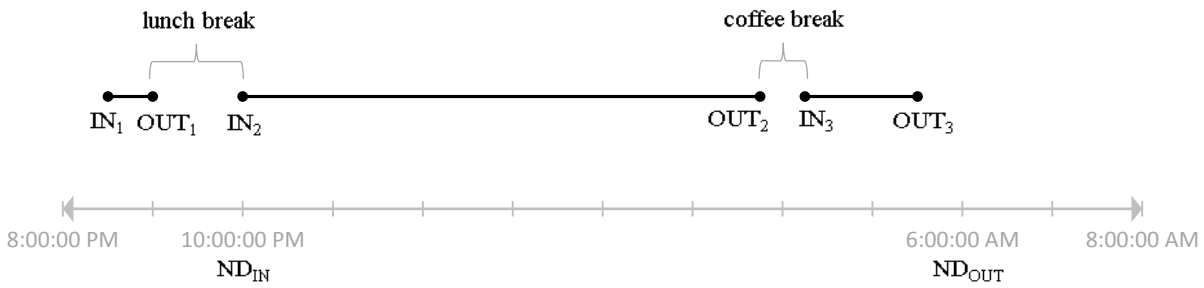


Figure 13: Night Differential Case 1 – Break C timeline

- If coffee break does not exceed 30 minutes

$$NDH = OUT_3 - ND_{IN}$$

- If coffee break exceeds 30 minutes

$$NDH = (OUT_3 - ND_{IN}) \ - [(IN_3 - OUT_2) + 0.5]$$

Case 2:
IN_1 *between* ND_{IN} *and* $ND_{OUT,}$
OUT_3 *after* ND_{OUT}

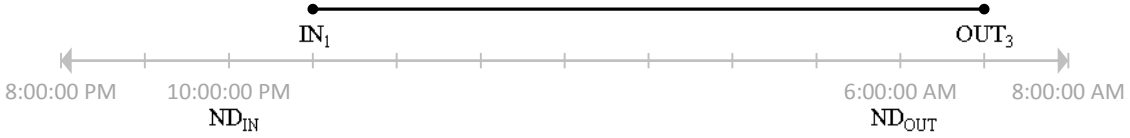


Figure 14: Night Differential Case 2 timeline

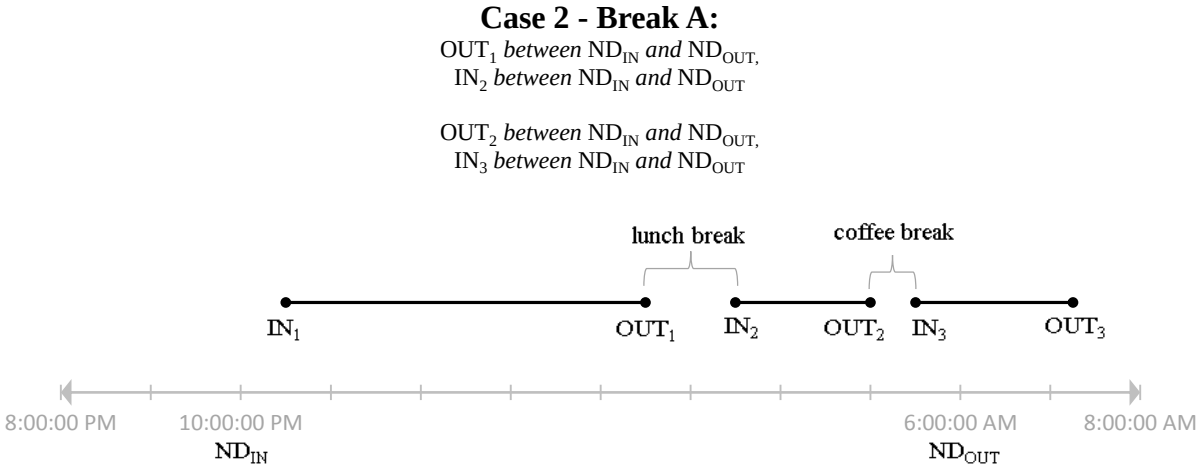


Figure 15: Night Differential Case 2 – Break A timeline

- If lunch break does not exceed 60 minutes and coffee break does not exceed 30 minutes

$$NDH = (ND_{OUT} - IN_1) - 1.0$$

- If lunch break does not exceed 60 minutes and coffee break exceeds 30 minutes

$$NDH = [(ND_{OUT} - IN_1) - 1.0] - [(IN_3 - OUT_2) + 0.5]$$

- If lunch break exceeds 60 minutes and coffee break does not exceed 30 minutes

$$NDH = (ND_{OUT} - IN_2) + (OUT_1 - IN_1)$$

- If lunch break exceeds 60 minutes and coffee break exceeds 30 minutes

$$NDH = [(ND_{OUT} - IN_2) + (OUT_1 - IN_1)] - [(IN_3 - OUT_2) + 0.5]$$

Case 2 - Break B:

OUT_1 *between* ND_{IN} *and* $ND_{OUT,}$
IN_2 *between* ND_{IN} *and* ND_{OUT}

OUT_2 *between* ND_{IN} *and* $ND_{OUT,}$
IN_3 *after* ND_{OUT}

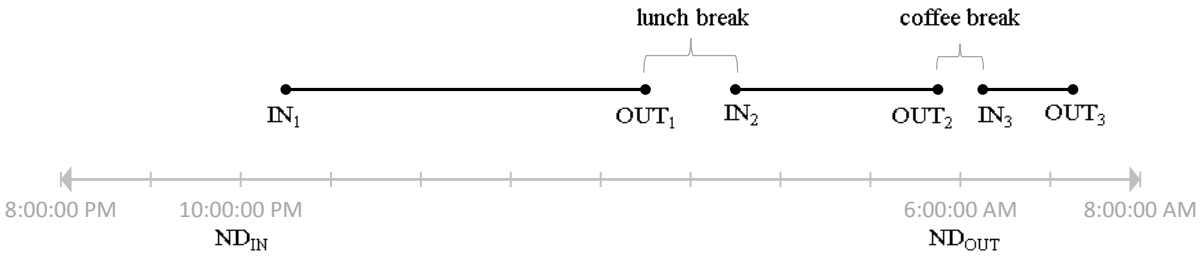


Figure 16: Night Differential Case 2 – Break B timeline

- If lunch break does not exceed 60 minutes

$$NDH = (OUT_2 - IN_1) - 1.0$$

- If lunch break exceeds 60 minutes

$$NDH = (OUT_2 - IN_2) + (OUT_1 - IN_1)$$

Case 2 - Break C:

OUT_1 *between* ND_{IN} *and* $ND_{OUT,}$
IN_2 *after* ND_{OUT}

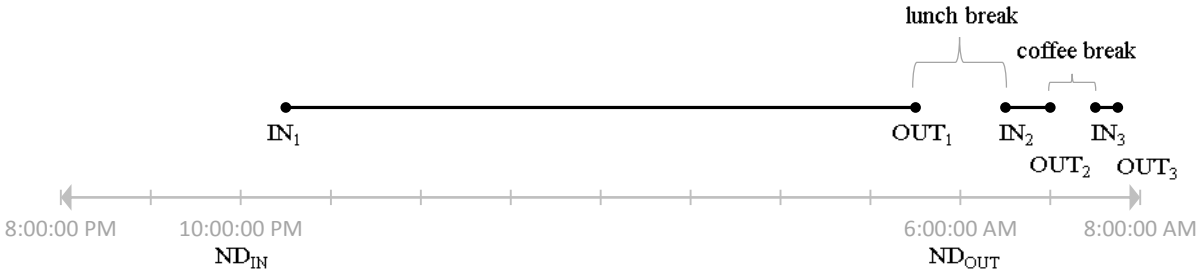


Figure 17: Night Differential Case 2 – Break C timeline

$$NDH = OUT_1 - IN_1$$

Case 2 - Break D:
OUT_1 *after* $ND_{OUT,}$
IN_2 *after* ND_{OUT}

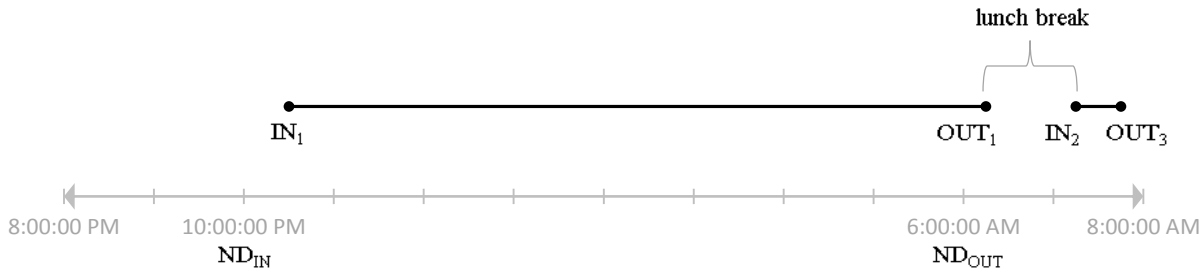


Figure 18: Night Differential Case 2 – Break D timeline

$$NDH = ND_{OUT} - IN_1$$

Case 3:
OUT_1 *between* ND_{IN} *and* $ND_{OUT,}$
IN_2 *between* ND_{IN} *and* ND_{OUT}

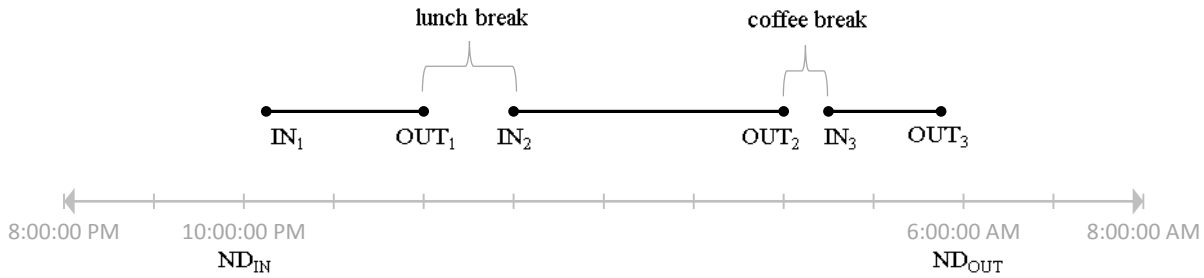


Figure 19: Night Differential Case 3 timeline

- If lunch break does not exceed 60 minutes and coffee break does not exceed 30 minutes

$$NDH = (OUT_3 - IN_1) - 1.0$$

- If lunch break does not exceed 60 minutes and coffee break exceeds 30 minutes

$$NDH = [(OUT_3 - IN_1) - 60] - [(IN_3 - OUT_2) + 0.5]$$

- If lunch break exceeds 60 minutes and coffee break does not exceed 30 minutes

$$NDH = (OUT_1 - IN_1) + (OUT_3 - IN_2)$$

- If lunch break exceeds 60 minutes and coffee break exceeds 30 minutes

$$NDH = [(OUT_1 - IN_1) + (OUT_3 - IN_2)] - [(IN_3 - OUT_2) + 0.5]$$

Case 4:
IN_1 *before* $ND_{IN,}$
OUT_3 *after* ND_{OUT}

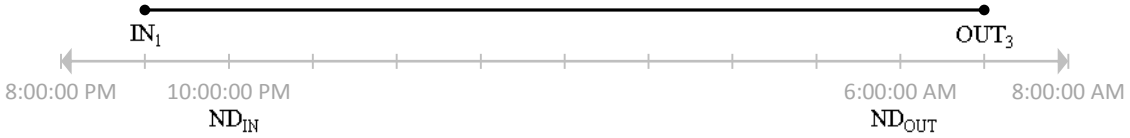


Figure 20: Night Differential Case 4 timeline

Case 4 - Break A:
OUT_1 *between* ND_{IN} *and* $ND_{OUT,}$
IN_2 *between* ND_{IN} *and* ND_{OUT}

OUT_2 *between* ND_{IN} *and* $ND_{OUT,}$
IN_3 *between* ND_{IN} *and* ND_{OUT}

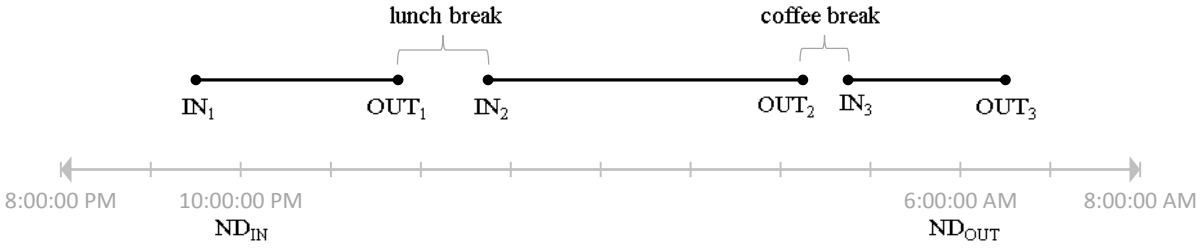


Figure 21: Night Differential Case 4 – Break A timeline

- If lunch break does not exceed 60 minutes and coffee break does not exceed 30 minutes

$$NDH = (ND_{OUT} - ND_{IN}) - 1.0$$

- If lunch break does not exceed 60 minutes and coffee break exceeds 30 minutes

$$NDH = [(ND_{OUT} - ND_{IN}) - 1.0] - [(IN_3 - OUT_2) + 0.5]$$

- If lunch break exceeds 60 minutes and coffee break does not exceed 30 minutes

$$NDH = (OUT_1 - ND_{IN}) + (ND_{OUT} - IN_2)$$

- If lunch break exceeds 60 minutes and coffee break exceeds 30 minutes

$$NDH = [(OUT_1 - ND_{IN}) + (ND_{OUT} - IN_2)] - [(IN_3 - OUT_2) + 0.5]$$

Case 4 - Break B:
OUT_1 *between* ND_{IN} *and* ND_{OUT},
IN_2 *between* ND_{IN} *and* ND_{OUT}

OUT_2 *between* ND_{IN} *and* ND_{OUT},
IN_3 *after* ND_{OUT}

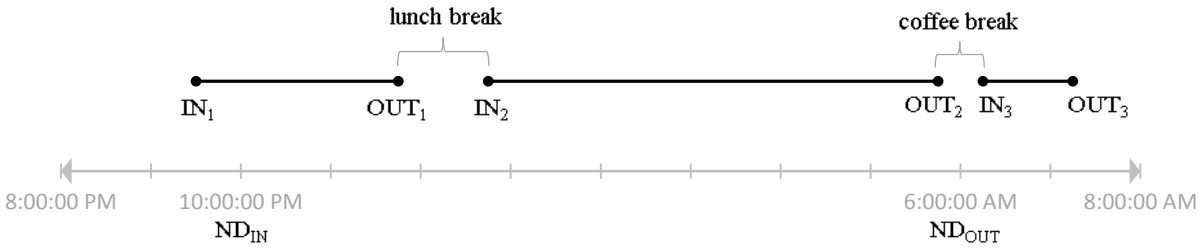


Figure 22: Night Differential Case 4 – Break B timeline

- If lunch break does not exceed 60 minutes

$$NDH = (OUT_2 - ND_{IN}) - 1.0$$

- If lunch break exceeds 60 minutes

$$NDH = (OUT_1 - ND_{IN}) + (OUT_2 - IN_2)$$

Case 4 - Break C:

OUT_1 *before* $ND_{IN,}$
IN_2 *between* ND_{IN} *and* ND_{OUT}

OUT_2 *between* ND_{IN} *and* $ND_{OUT,}$
IN_3 *between* ND_{IN} *and* ND_{OUT}

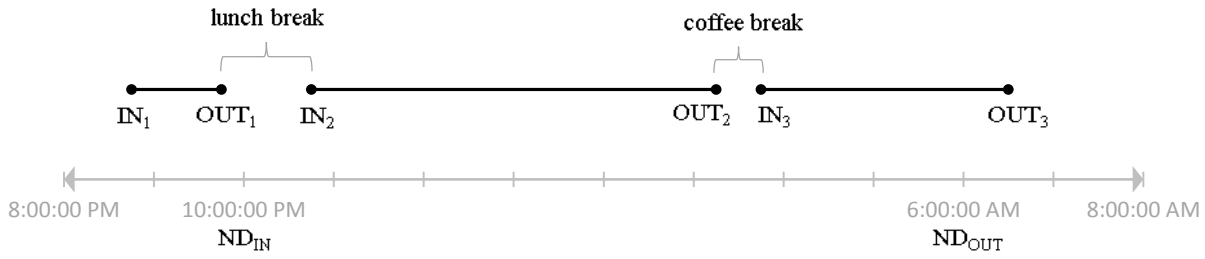


Figure 23: Night Differential Case 4 – Break C timeline

- If coffee break does not exceed 30 minutes

$$NDH = ND_{OUT} - IN_2$$

- If coffee break exceeds 30 minutes

$$NDH = (ND_{OUT} - IN_2) - [(IN_3 - OUT_2) + 0.5]$$

Case 4 - Break D:

OUT_1 *before* $ND_{IN,}$
IN_2 *between* ND_{IN} *and* ND_{OUT}

OUT_2 *between* ND_{IN} *and* $ND_{OUT,}$
IN_3 *after* ND_{OUT}

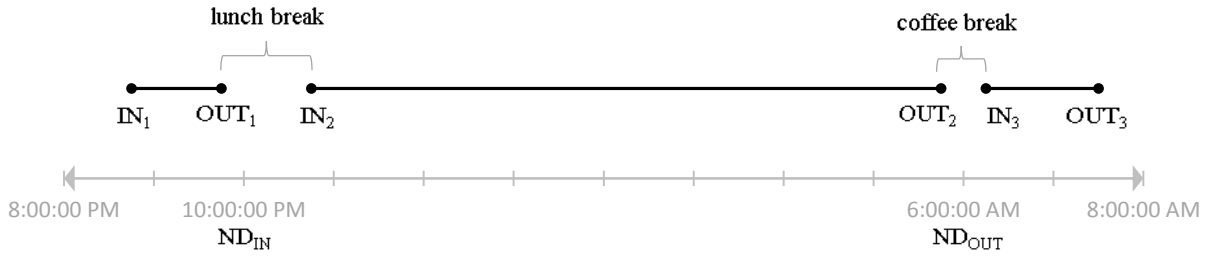


Figure 24: Night Differential Case 4 – Break D timeline

$$NDH = OUT_2 - IN_2$$

Case 4 - Break E:

OUT_1 *before* ND_{IN},
IN_2 *between* ND_{IN} *and* ND_{OUT}

OUT_2 *after* ND_{OUT},
IN_3 *after* ND_{OUT}

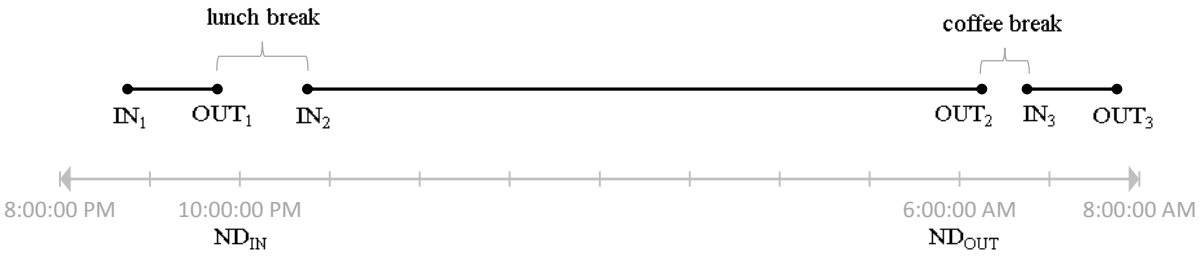


Figure 25: Night Differential Case 4 – Break E timeline

$$NDH = ND_{OUT} - IN_2$$

Case 4 - Break F:

OUT_1 *between* ND_{IN} *and* ND_{OUT},
IN_2 *after* ND_{OUT}

** coffee break always after* ND_{OUT}

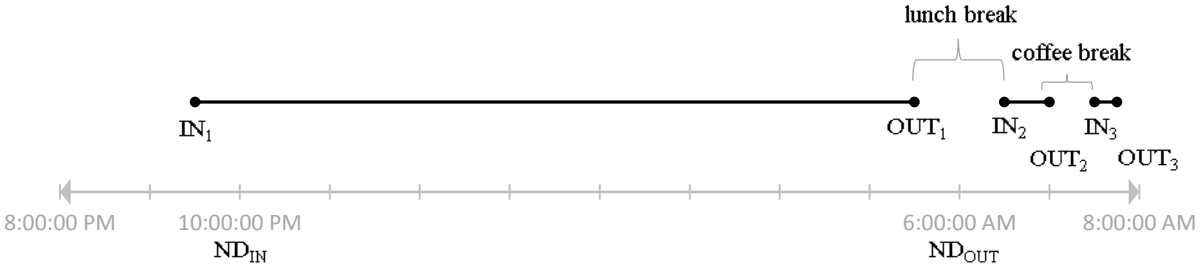


Figure 26: Night Differential Case 4 – Break F timeline

$$NDH = OUT_1 - ND_{IN}$$

Case 4 - Break G:
OUT_1 *before* $ND_{IN,}$
IN_2 *before* ND_{IN}

** any position of coffee break*

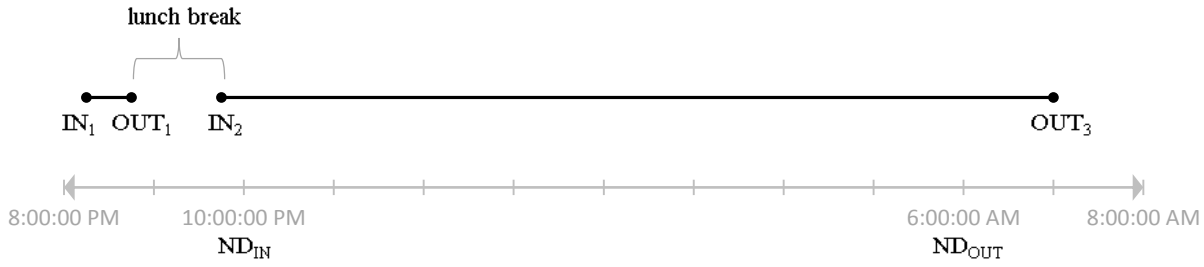


Figure 27: Night Differential Case 4 – Break G timeline

- If coffee break does not exceed 30 minutes

$$NDH = ND_{OUT} - ND_{IN}$$

- If coffee break exceeds 30 minutes

$$NDH = (ND_{OUT} - ND_{IN}) - [(IN_3 - OUT_2) + 0.5]$$

Case 4 - Break H:
OUT_1 *after* $ND_{OUT,}$
IN_2 *after* ND_{OUT}

** coffee break always after* ND_{OUT}

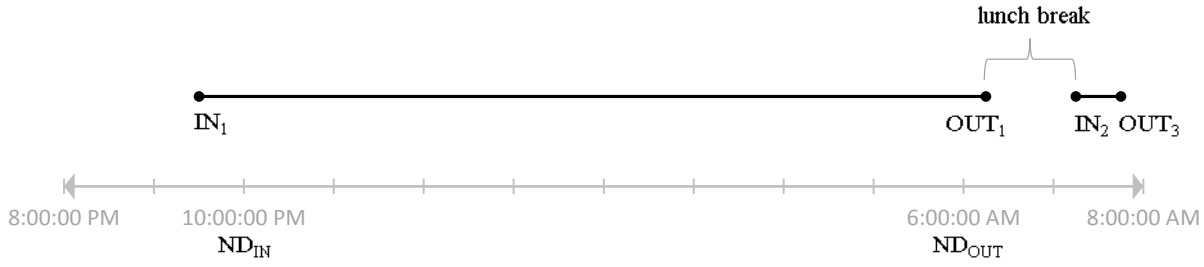


Figure 28: Night Differential Case 4 – Break H timeline

$$NDH = ND_{OUT} - ND_{IN}$$

OVERTIME WITH NIGHT DIFFERENTIAL

*(applicable to **OTND, OTRDND, SHOTND, OTRHND, OTSRND, OTRRND**)*

Case 1:

S_{OUT} *before* ND_{IN},
OUT_3 *between* ND_{IN} *and* ND_{OUT}

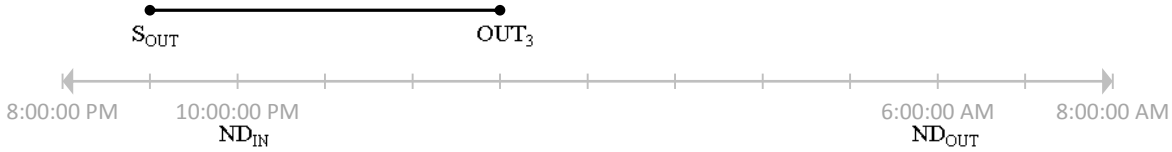


Figure 29: Overtime with Night Differential Case 1 timeline

$$OTND = OUT_3 - ND_{IN}$$

Case 2:

S_{OUT} *between* ND_{IN} *and* ND_{OUT},
OUT_3 *after* ND_{OUT}

Figure 30: Overtime with Night Differential Case 2 timeline

$$OTND = ND_{OUT} - S_{OUT}$$

Case 3:

S_{OUT} *before* ND_{IN},
OUT_3 *after* ND_{OUT}

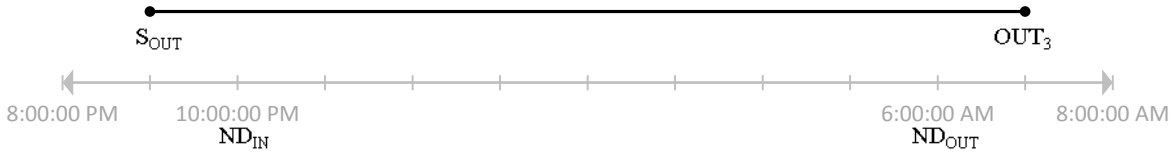


Figure 31: Overtime with Night Differential Case 3 timeline

$$OTND = ND_{OUT} - ND_{IN}$$

WORK ON A REST DAY

*(applicable to **WRD, WSR, WRR**)*

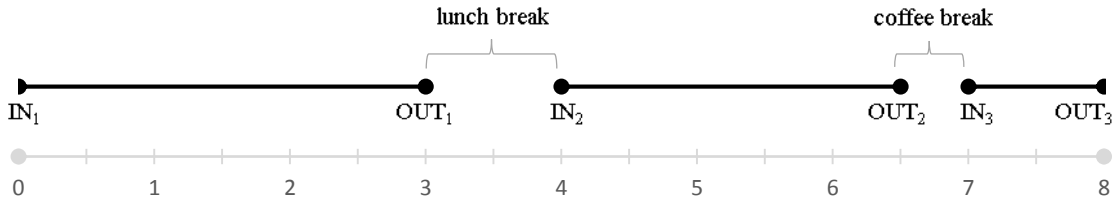


Figure 32: Work on a Rest Day timeline

- If lunch break does not exceed 60 minutes and coffee break does not exceed 30 minutes

$$WRD = (OUT_3 - IN_1) - 1.0$$

- If lunch break does not exceed 60 minutes and coffee break exceeds 30 minutes

$$WRD = [(OUT_3 - IN_1) - 1.0] - [(IN_3 - OUT_2) + 0.5]$$

- If lunch break exceeds 60 minutes and coffee break does not exceed 30 minutes

$$WRD = (OUT_3 - IN_2) + (OUT_1 - IN_1)$$

- If lunch break exceeds 60 minutes and coffee break exceeds 30 minutes

$$WRD = [(OUT_3 - IN_2) + (OUT_1 - IN_1)] - [(IN_3 - OUT_2) + 0.5]$$

- Else,

$$WRD = OUT_3 - IN_1$$

$*$ *If* $WRD \geq 8.00$ *then*

$$\boldsymbol{WRD = 8.00}$$

OVERTIME WORK ON A REST DAY

*(applicable to **OTRD, OTSR, OTRR**)*

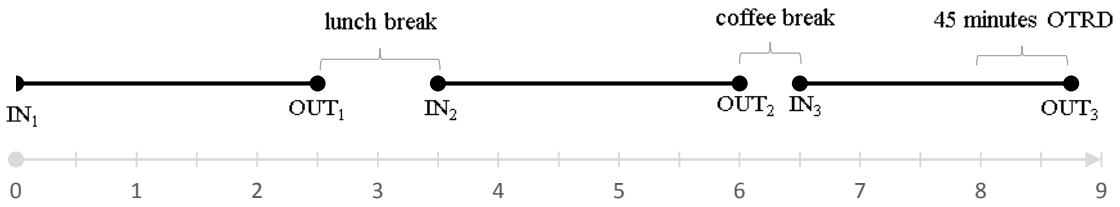


Figure 33: Overtime Work on a Rest Day timeline

$$\textit{Only if: } WRD > 8.00$$

$$OTRD = WRD - 8.00$$

* **Employee Compensation**
  - Regular

$$\frac{Basic + COLA}{8} \times \#of\ Regular\ hours$$

  - ND

$$\frac{Basic}{8} \times 10\% \times \#\ of\ ND\ hours$$

  - OT

$$\frac{Basic}{8} \times 125\% \times \#\ of\ OT\ hours$$

  - OTND

$$\frac{Basic}{8} \times 125\% \times 10\% \times \#\ of\ OTND\ hours$$

  - WRD

$$\frac{Basic + COLA}{8} \times 130\% \times \#\ of\ WRD\ hours$$

- WRDND

$$\frac{Basic}{8} \times 130\% \times 10\% \times \#\ of\ WRDND\ hours$$

- OTRD

$$\frac{Basic}{8} \times 130\% \times 130\% \times \#\ of\ OTRD\ hours$$

- OTRDND

$$\frac{Basic}{8} \times 130\% \times 130\% \times 10\% \times \#\ of\ OTRDND\ hours$$

- SH

$$\frac{Basic + COLA}{8} \times 130\% \times \#\ of\ SH\ hours$$

- SHND

$$\frac{Basic}{8} \times 130\% \times 10\% \times \#\ of\ SHND\ hours$$

- SHOT

$$\frac{Basic}{8} \times 130\% \times 130\% \times \#\ of\ SHOT\ hours$$

- SHOTND

$$\frac{Basic}{8} \times 130\% \times 130\% \times 10\% \times \#\ of\ SHOTND\ hours$$

- WRH

$$\frac{Basic + COLA}{8} \times 200\% \times \#\ of\ WRH\ hours$$

- WRHND

$$\frac{Basic}{8} \times 200\% \times 10\% \times \#\ of\ WRHND\ hours$$

- OTRH

$$\frac{Basic}{8} \times 200\% \times 130\% \times \#\ of\ OTRH\ hours$$

- OTRHND

$$\frac{Basic}{8} \times 200\% \times 130\% \times 10\% \times \#\ of\ OTRHND\ hours$$

- WSR

$$\frac{Basic}{8} \times 150\% \times \#\ of\ WSR\ hours$$

- WSRND

$$\frac{Basic}{8} \times 150\% \times 10\% \times \#\ of\ WSRND\ hours$$

- OTSR

$$\frac{Basic}{8} \times 150\% \times 130\% \times \#\ of\ OTSR\ hours$$

- OTSRND

$$\frac{Basic}{8} \times 150\% \times 130\% \times 10\% \times \#\ of\ OTSRND\ hours$$

- WRR

$$\frac{Basic}{8} \times 260\% \times \#\ of\ WRR\ hours$$

- WRRND

$$\frac{Basic}{8} \times 260\% \times 10\% \times \#\ of\ WRRND\ hours$$

- OTRR

$$\frac{Basic}{8} \times 260\% \times 130\% \times \#\ of\ OTRR\ hours$$

- OTRRND

$$\frac{Basic}{8} \times 260\% \times 130\% \times 10\% \times \#\ of\ OTRRND\ hours$$

EXAMPLES:

c. Regular Work Day **(Regular),** Regular Work Day with Night Differential **(ND),** Overtime Work on a Regular Day **(OT)** and Overtime on Regular Work Day with Night Differential **(OTND)**

Date: February 10, 2014 - Monday
Schedule: 19:00 – 4:00

IN_1	OUT_1	IN_2	OUT_2	IN_3	OUT_3
18:32	22:27	23:25	2:10	2:33	6:45

Table 2: In and Out Example 1

1st: $19{:}00 \geq 18{:}32$ ✓

2nd: $(23{:}25 - 22{:}27) \leq 60$

$58 \leq 60$ ✓

3rd: $(2{:}33 - 2{:}10) \leq 30$

$23 \leq 30$ ✓

4th: $6{:}00 \leq 6{:}12$ ✓

$$\boldsymbol{WH = 8.00}$$

Night Differential Case 1 – Break A

- *Lunch break did not exceed 60 minutes and coffee break did not exceed 30 minutes*

$$ND = (S_{OUT} - ND_{IN}) - 1.0$$
$$ND = (4{:}00 - 22{:}00) - 1.0$$
$$\boldsymbol{ND = 5.00}$$

Overtime

$$OT = OUT_3 - S_{OUT}$$
$$OT = 6{:}45 - 4{:}00$$
$$\boldsymbol{OT = 2.45}$$

Overtime with Night Differential Case 2

$$OTND = (ND_{OUT} - S_{OUT})$$
$$OTND = 6{:}00 - 4{:}00$$
$$\boldsymbol{OTND = 2.00}$$

Compensation

- *Regular*

$$\frac{₱500.00 + ₱25.00}{8} \times 8.00 = \mathbf{₱525.00}$$

- *ND*

$$\frac{₱500.00}{8} \times 0.10 \times 5.00 = \mathbf{₱31.25}$$

- *OT*

$$\frac{₱500.00}{8} \times 1.25 \times 2.45 = \mathbf{₱191.41}$$

- *OTND*

$$\frac{₱500.00}{8} \times 1.25 \times 0.10 \times 2.00 = \mathbf{₱15.63}$$

- ***Total***

$$₱525.00 + ₱31.25 + ₱191.41 + ₱15.63 = \mathbf{₱763.29}$$

d. Work on Rest Day (**WRD**), Work on Rest Day with Night Differential (**WRDND**), Overtime Work on Rest Day (**OTRD**) and Overtime Work on Rest Day with Night Differential (**OTRDND**)

Date: February 9, 2014 – Sunday

Schedule: Rest Day

IN_1	**OUT_1**	**IN_2**	**OUT_2**	**IN_3**	**OUT_3**
18:32	22:27	23:25	3:10	3:33	6:12

Table 3: In and Out Example 2

∗ *Lunch break did not exceed 60 minutes and coffee break did not exceed 30 minutes*

$$WRD = (OUT_3 - IN_1) - 1$$

$$WRD = (6{:}12 - 18{:}32) - 1$$

$$WRD = (11.67) - 1$$

$$WRD = 10.67$$

∗ $10.67 \geq 8.00$ then,

$$\boldsymbol{WRD} = 8.00$$

WRDND Case 4 – Break 1

- *Lunch break did not exceed 60 minutes and coffee break did not exceed 30 minutes*

$$WRDND = (ND_{OUT} - ND_{IN}) - 60$$

$$WRDND = (6{:}00 - 22{:}00) - 60$$

$$\boldsymbol{WRDND} = 5.50$$

Any excess to 8 hours is regarded as overtime.

OTRD

OTRD starts at 3:30

$$OTRD = 10.67 - 8$$

$$\boldsymbol{OTRD} = 2.67$$

OTRDND Case 4

$$OTRDND = 6{:}00 - 3{:}30$$

$$\boldsymbol{OTRDND} = 2.50$$

Compensation

- *WRD*

$$\frac{₱500.00 + ₱25.00}{8} \times 1.30 \times 8.00 = \boldsymbol{₱682.50}$$

- *WRDND*

$$\frac{₱500.00}{8} \times 1.30 \times 0.10 \times 5.50 = \boldsymbol{₱44.69}$$

- *OTRD*

$$\frac{₱500.00}{8} \times 1.30 \times 1.30 \times 2.67 = \boldsymbol{₱282.02}$$

- *OTRDND*

$$\frac{₱500.00}{8} \times 1.30 \times 1.30 \times 0.10 \times 2.50 = \boldsymbol{₱26.41}$$

- ***Total***

$$₱682.50 + ₱44.69 + ₱282.02 + ₱26.41 = \boldsymbol{₱1,035.62}$$

a. Work on Special Holiday **(SH),** Work on Special Holiday with Night Differential **(SHND),** Overtime Work on Special Holiday **(SHOT)** and Overtime Work on Special Holiday with Night Differential **(SHOTND)**

Date: April 19, 2014 – Saturday (Black Saturday)

Schedule: 19:00 – 4:00

IN_1	OUT_1	IN_2	OUT_2	IN_3	OUT_3
18:32	22:27	23:25	2:10	2:33	6:45

Table 4: In and Out Example 3

1st: $19{:}00 \geq 18{:}32$ ✓

2nd: $(23{:}25 - 22{:}27) \leq 60$

$58 \leq 60$ ✓

3rd: $(2{:}33 - 2{:}10) \leq 30$

$23 \leq 30$ ✓

4th: $6{:}00 \leq 6{:}12$ ✓

$$\boldsymbol{SH = 8.00}$$

Night Differential Case 1 – Break A

- *Lunch break did not exceed 60 minutes and coffee break did not exceed 30 minutes*

$$SHND = (S_{OUT} - ND_{IN}) - 1.0$$

$$SHND = (4{:}00 - 22{:}00) - 1.0$$

$$\boldsymbol{SHND = 5.00}$$

Overtime

$$SHOT = OUT_3 - S_{OUT}$$

$$SHOT = 6{:}45 - 4{:}00$$

$$\boldsymbol{SHOT = 2.45}$$

Overtime with Night Differential Case 2

$$SHOTND = (ND_{OUT} - S_{OUT})$$

$$SHOTND = 6{:}00 - 4{:}00$$

$$\boldsymbol{SHOTND = 2.00}$$

Compensation

- *SH*

$$\frac{₱500.00 + ₱25.00}{8} \times 1.30 \times 8.00 = \boldsymbol{₱682.50}$$

- *SHND*

$$\frac{₱500.00}{8} \times 1.30 \times 0.10 \times 5.00 = ₱\mathbf{40.63}$$

- *SHOT*

$$\frac{₱500.00}{8} \times 1.30 \times 1.30 \times 2.45 = ₱\mathbf{258.78}$$

- *SHOTND*

$$\frac{₱500.00}{8} \times 1.30 \times 1.30 \times 0.10 \times 2.00 = ₱\mathbf{21.13}$$

- ***Total***

$$₱682.50 + ₱40.63 + ₱258.78 + ₱21.13 = ₱\mathbf{1,003.04}$$

b. Work on Regular Holiday **(WRH),** Work on Regular Holiday with Night Differential **(WRHND),** Overtime Work on Regular Holiday **(OTRH)** and Overtime Work on Regular Holiday with Night Differential **(OTRHND)**

Date: December 25, 2013 – Wednesday (Christmas Day)
Schedule: 19:00 – 4:00

IN_1	OUT_1	IN_2	OUT_2	IN_3	OUT_3
18:32	22:27	23:25	2:10	2:33	6:45

Table 5: In and Out Example 4

1st: $19{:}00 \geq 18{:}32$ ✓

2nd: $(23{:}25 - 22{:}27) \leq 60$

$58 \leq 60$ ✓

3rd: $(2{:}33 - 2{:}10) \leq 30$

$23 \leq 30$ ✓

4th: $6{:}00 \leq 6{:}12$ ✓

$$\boldsymbol{WRH = 8.00}$$

Night Differential Case 1 – Break A

- *Lunch break did not exceed 60 minutes and coffee break did not exceed 30 minutes*

$$WRHND = (S_{OUT} - ND_{IN}) - 1.0$$

$$WRHND = (4{:}00 - 22{:}00) - 1.0$$

$$\boldsymbol{WRHND = 5.00}$$

Overtime

$$OTRH = OUT_3 - S_{OUT}$$

$$OTRH = 6{:}45 - 4{:}00$$

$$\boldsymbol{OTRH = 2.45}$$

Overtime with Night Differential Case 2

$$OTRHND = (ND_{OUT} - S_{OUT})$$

$$OTRHND = 6{:}00 - 4{:}00$$

$$\boldsymbol{OTRHND = 2.00}$$

Compensation

- *WRH*

$$\frac{₱500.00 + ₱25.00}{8} \times 2.00 \times 8.00 = \boldsymbol{₱1,050.00}$$

- *WRHND*

$$\frac{₱500.00}{8} \times 2.00 \times 0.10 \times 5.00 = \boldsymbol{₱62.50}$$

- *OTRH*

$$\frac{₱500.00}{8} \times 2.00 \times 1.30 \times 2.45 = \boldsymbol{₱398.13}$$

- *OTRHND*

$$\frac{₱500.00}{8} \times 2.00 \times 1.30 \times 0.10 \times 2.00 = \boldsymbol{₱32.50}$$

- ***Total***

$$₱1{,}050.00 + ₱62.50 + ₱398.13 + ₱32.50 = \boldsymbol{₱1,543.13}$$

c. Work on Special Holiday Falling on Rest Day (**WSR**), Work on Special Holiday Falling on Rest Day with Night Differential (**WSRND**), Overtime Work on Special Holiday Falling on Rest Day (**OTSR**) and Overtime Work on Special Holiday Falling on Rest Day with Night Differential (**OTSRND**)

Date: April 19, 2014 – Saturday (Black Saturday)

Schedule: Rest Day

IN_1	OUT_1	IN_2	OUT_2	IN_3	OUT_3
18:32	22:27	23:25	3:10	3:33	6:12

Table 6: In and Out Example 5

∗ *Lunch break did not exceed 60 minutes and coffee break did not exceed 30 minutes*

$$WSR = (OUT_3 - IN_1) - 1$$

$$WSR = (6{:}12 - 18{:}32) - 1$$

$$WSR = (11.67) - 1$$

$$WSR = 10.67$$

∗ $10.67 \geq 8.00$ then,

$$\boldsymbol{WSR} = 8.00$$

WSRND Case 4 – Break 1

- *Lunch break did not exceed 60 minutes and coffee break did not exceed 30 minutes*

$$WSRND = (ND_{OUT} - ND_{IN}) - 60$$

$$WSRND = (6{:}00 - 22{:}00) - 60$$

$$\boldsymbol{WSRND} = 5.50$$

Any excess to 8 hours is regarded as overtime.

OTSR

OTSR starts at 3:30

$$OTSR = 10.67 - 8$$

$$\boldsymbol{OTSR} = 2.67$$

OTSRND Case 4

$$OTSRND = 6{:}00 - 3{:}30$$

$$\boldsymbol{OTSRND} = \mathbf{2.50}$$

Compensation

- *WSR*

$$\frac{₱500.00}{8} \times 1.50 \times 8.00 = \mathbf{₱750.00}$$

- *WSRND*

$$\frac{₱500.00}{8} \times 1.50 \times 0.10 \times 5.50 = \mathbf{₱51.56}$$

- *OTSR*

$$\frac{₱500.00}{8} \times 1.50 \times 1.30 \times 2.67 = \mathbf{₱325.41}$$

- *OTSRND*

$$\frac{₱500.00}{8} \times 1.50 \times 1.30 \times 0.10 \times 2.50 = ₱\mathbf{30.47}$$

- ***Total***

$$₱750.00 + ₱51.56 + ₱325.41 + ₱30.47 = ₱\mathbf{1,157.44}$$

a. Work on Regular Holiday Falling on Rest Day (**WRR**), Work on Regular Holiday Falling on Rest Day with Night Differential (**WRRND**), Overtime Work on Regular Holiday Falling on Rest Day (**OTRR**) and Overtime Work on Regular Holiday Falling on Rest Day with Night Differential (**OTRRND**)

Date: December 25, 2013 – Wednesday (Christmas Day)

Schedule: Rest Day

IN_1	**OUT_1**	**IN_2**	**OUT_2**	**IN_3**	**OUT_3**
18:32	22:27	23:25	3:10	3:33	6:12

Table 7: In and Out Example 6

∗ *Lunch break did not exceed 60 minutes and coffee break did not exceed 30 minutes*

$$WRR = (OUT_3 - IN_1) - 1$$

$$WRR = (6{:}12 - 18{:}32) - 1$$

$$WRR = (11.67) - 1$$

$$WRR = 10.67$$

∗ $10.67 \geq 8.00$ then,

$$\boldsymbol{WRR} = 8.00$$

WRRND Case 4 – Break 1

- *Lunch break did not exceed 60 minutes and coffee break did not exceed 30 minutes*

$$WRRND = (ND_{OUT} - ND_{IN}) - 60$$

$$WRRRND = (6{:}00 - 22{:}00) - 60$$

$$\boldsymbol{WRRND} = 5.50$$

Any excess to 8 hours is regarded as overtime.

OTRR

OTRR starts at 3:30

$$OTRR = 10.67 - 8$$

$$\boldsymbol{OTRR} = 2.67$$

OTRRND Case 4

$$OTRRND = 6{:}00 - 3{:}30$$

$$\boldsymbol{OTRRND} = 2.50$$

Compensation

- *WRR*

$$\frac{₱500.00}{8} \times 2.60 \times 8.00 = \boldsymbol{₱1,300.00}$$

- *WRRND*

$$\frac{₱500.00}{8} \times 2.60 \times 0.10 \times 5.50 = \boldsymbol{₱89.38}$$

- *OTRR*

$$\frac{₱500.00}{8} \times 2.60 \times 1.30 \times 2.67 = \boldsymbol{₱564.04}$$

- *OTRRND*

$$\frac{₱500.00}{8} \times 2.60 \times 1.30 \times 0.10 \times 2.50 = \boldsymbol{₱52.81}$$

- ***Total***

$$₱1{,}300.00 + ₱89.38 + ₱564.04 + ₱52.81 = \boldsymbol{₱2,006.23}$$

D. Database Management System

The storage, computation, query, update and administration of the database lies in the database management system. As a data storage and retrieval system that permits data to be stored, it facilitates the creation and maintenance of a computerized database.[23]

Microsoft SQL Server 2008 R2 was developed by Microsoft and was released last April 21, 2010.[24] It includes Management Studio which enables administrators to manage and monitor multiple database applications and servers.[25]

E. iTextSharp

Reporting is the final part of the system as the results will come out of this. The outcome will be on a report format that can be saved as a PDF file.

Originally in Java, iText is an open source library for the creation and manipulation of PDFs. It is then ported in the .NET Framework under the name iTextSharp written in C#.[26] Its features include PDF generation as well as manipulation (stamping with watermarks, splitting and merging), PDF form filling and digital signatures.[27]

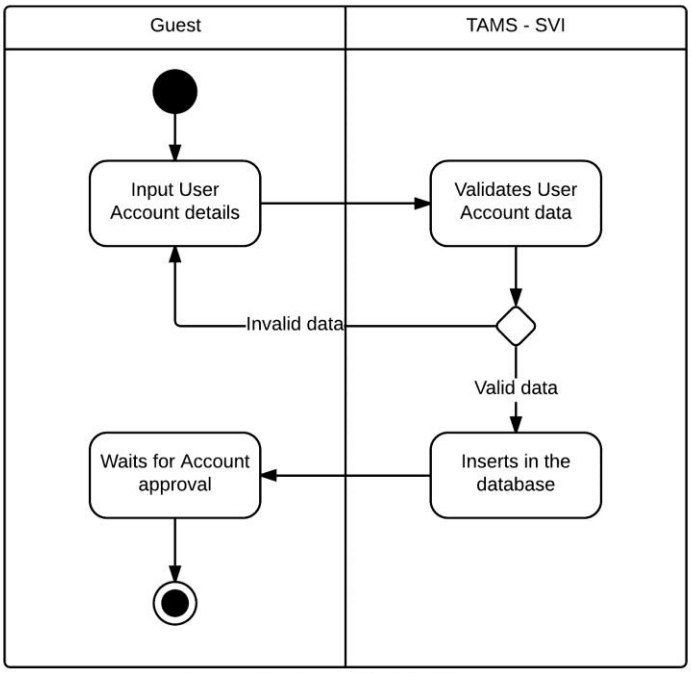
IV. Design and Implementation

A. Use Case and Activity Diagram

- **Request for Account**

To use the system, one must have an account. Guests have to request accounts to the system administrator before using the system. They must provide basic information such as last name, first name, middle name (optional), username and password. After the input of details are done, the system verifies and confirms the data before adding it in the database. The guest must wait for the approval of the system administrator for an account.

The following is the activity diagram for requesting a user account.



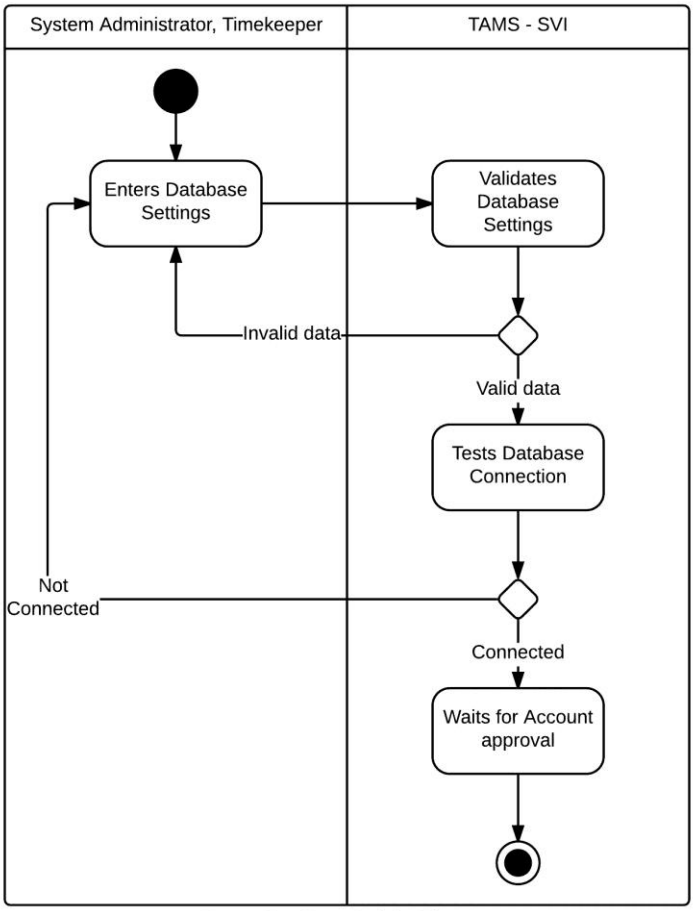
Request for Account Activity Diagram

- **Set-up Database**

Setting up the database is also a requirement in using the system.

The user enters the details of the database. The system then confirms the data for validation. Then the system tests the connection of the database and saves the settings.

The following is the activity diagram of setting the database.

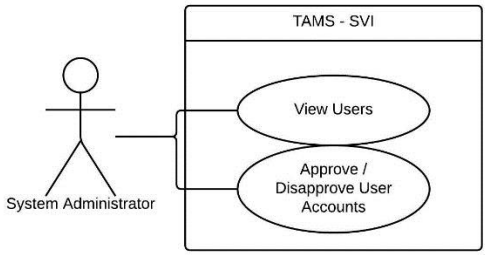


Set-up Database Activity Diagram

- **User Accounts**

The system administrator handles the user accounts maintenance. He can view all the users of the system and approve/disapprove of user accounts.

The following is the use case diagram of the User Accounts.

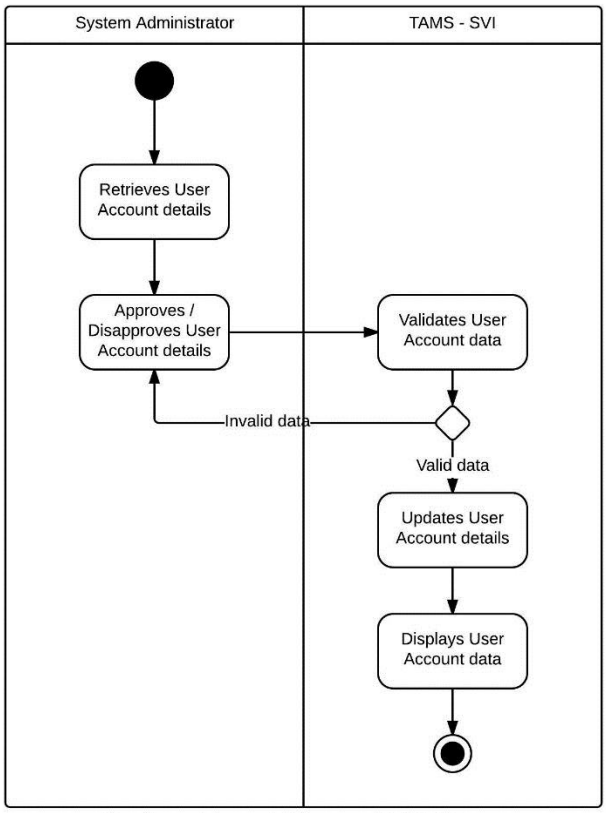


User Accounts Use Case Diagram

- **Approve/Disapprove User Accounts**

The system administrator has the power to approve or disapprove a user account. The system administrator retrieves the user data he has to approve then makes the decision. The system then verifies and confirms the decision before updating it in the database. Then the system will display the user data.

The following illustrates the activity diagram of the approval or disapproval of a user account by the system administrator.

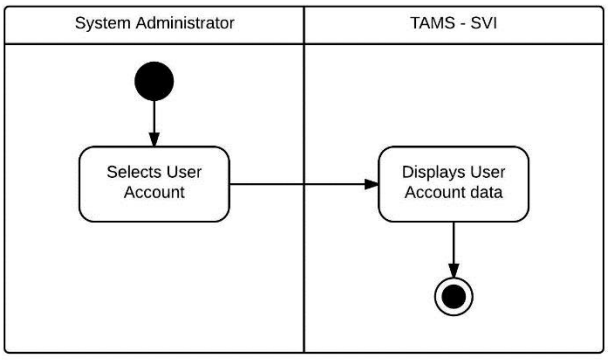


User Account Approval / Disapproval Activity Diagram

- **View Users**

The system administrator also has the privilege to view all the users and its details in the system. When the system administrator selects a user, the system will display the user data.

The following is an illustration of the activity diagram of viewing a user data.

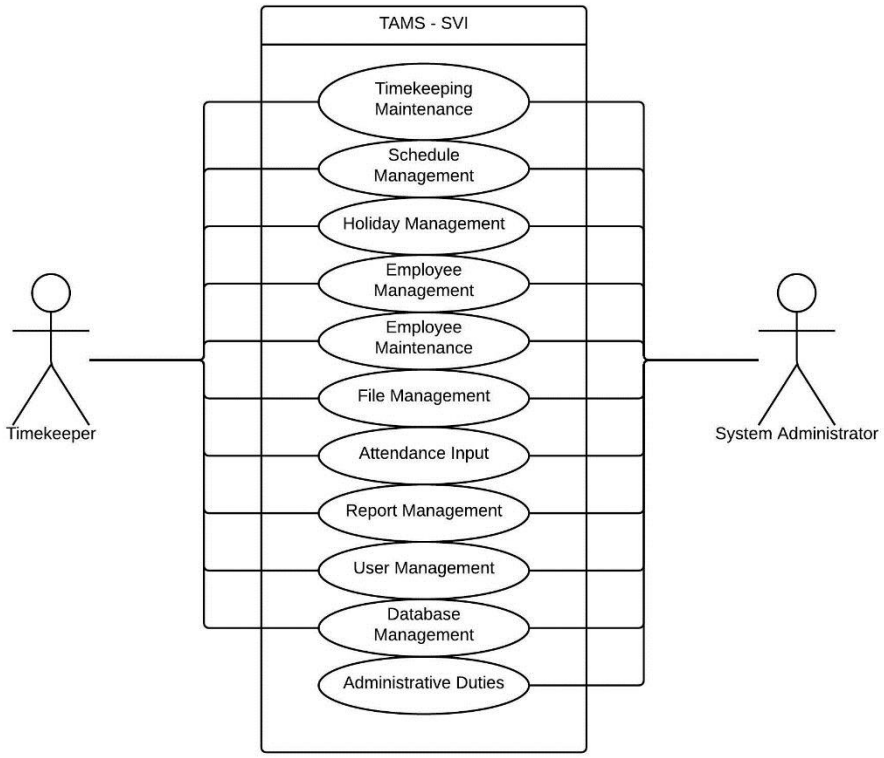


View User Account Activity Diagram

- **Timekeeping and Attendance Monitoring System**

There are two kinds of users in the system, the System Administrator and the Timekeeper. The System Administrator account handles the all over maintenance of the system as well as the tasks of a timekeeper. The timekeeper account primarily handles the timekeeping process in the system.

The following illustrates the top level use case diagram of the system.

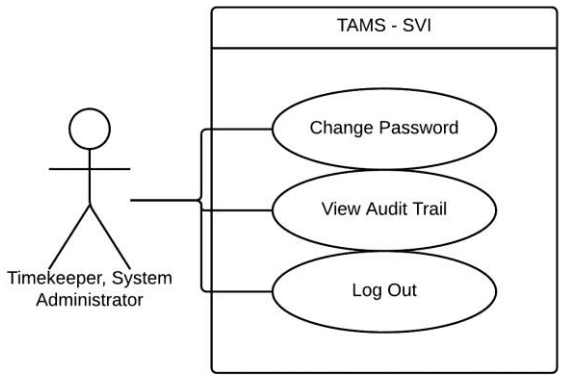


TAMS - SVI Top Level Use Case Diagram

- **User Management**

Users of the system has the capability of changing his log in password as well as the viewing of his own audit trail (the administrator account can view all the users' trails).

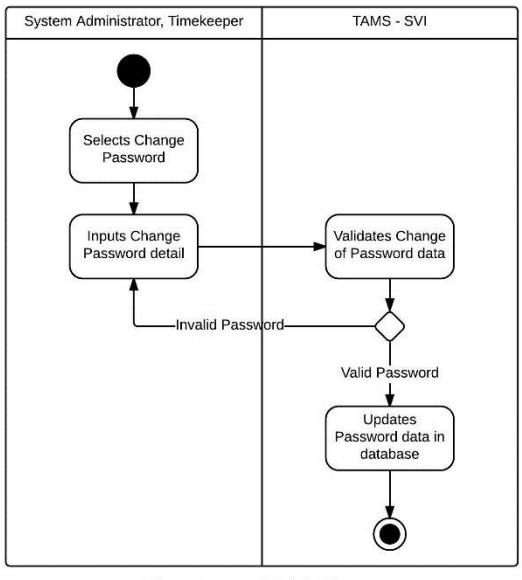
The following illustrates the User Management Use Case Diagram.



User Management Use Case Diagram

- **Change Password**

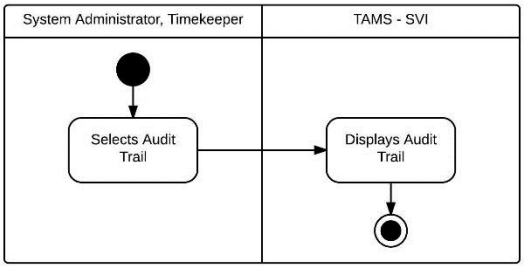
To change the password of log in details, the user must select the Change Password and type in another password. The system then validates and confirms the change in password before updating the data in the database. The following is an illustration of the Change Password Activity Diagram.



Change Password Activity Diagram

- **View Audit Trail**

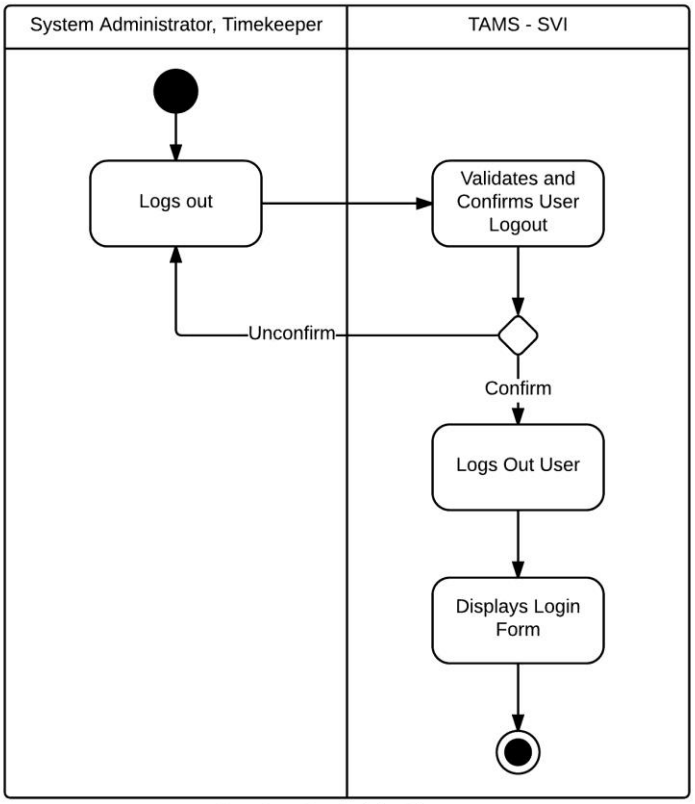
To view the audit trail, the user must select the Audit Trail option. The system then displays the user's own audit trail. As said above, the system administrator can view all the users' audit trail. The following illustrates the View Audit Trail Activity Diagram.



View Audit Trail Activity Diagram

- **Log Out**

The system confirms the log out of the user upon its selection. The login form is shown upon the log out of the user.

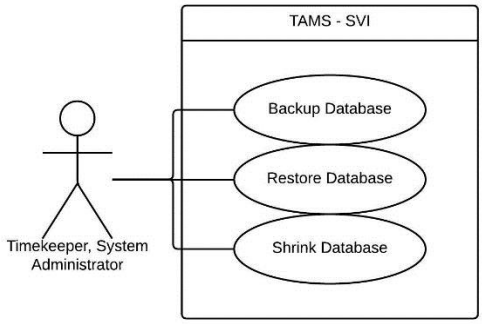


User Log Out Activity Diagram

- **Database Management**

All the accounts in the system has the capability to manage the database. This includes the backup, restoring and shrinking of the database. To back up is to archive data for later use. To restore is to return data to a former condition. To shrink is to lessen the file size of data without changing its contents.

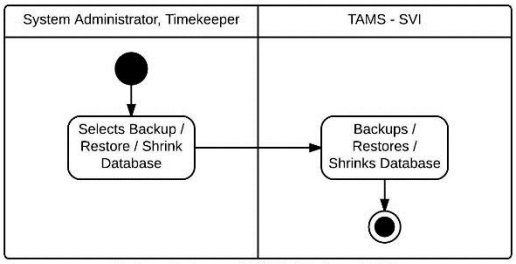
The following is the use case diagram of the Database Management.



Database Management Use Case Diagram

- **Back Up/Restore/Shrink Database**

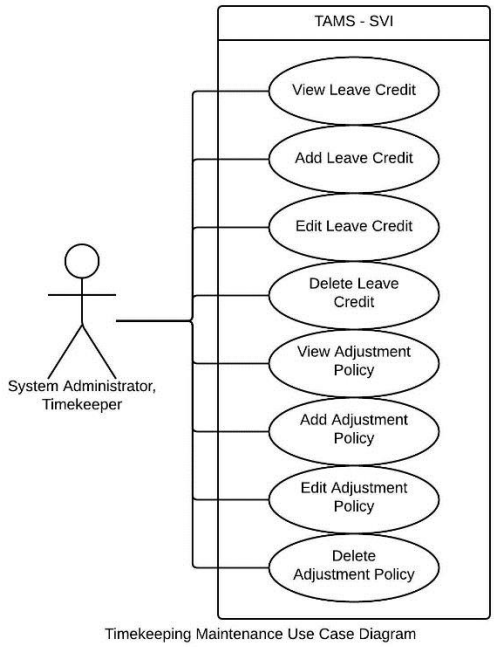
To backup, restore or shrink the database, the user must select any one of the tasks then the system completes it. Below is the activity diagram of the said task.



Backup, Restore and Shrink Database Activity Diagram

- **Timekeeping Maintenance**

The Timekeeping Maintenance contains the adjustment policies and the leave credit types needed in timekeeping task. The following is the use case diagram of the Timekeeping Maintenance.



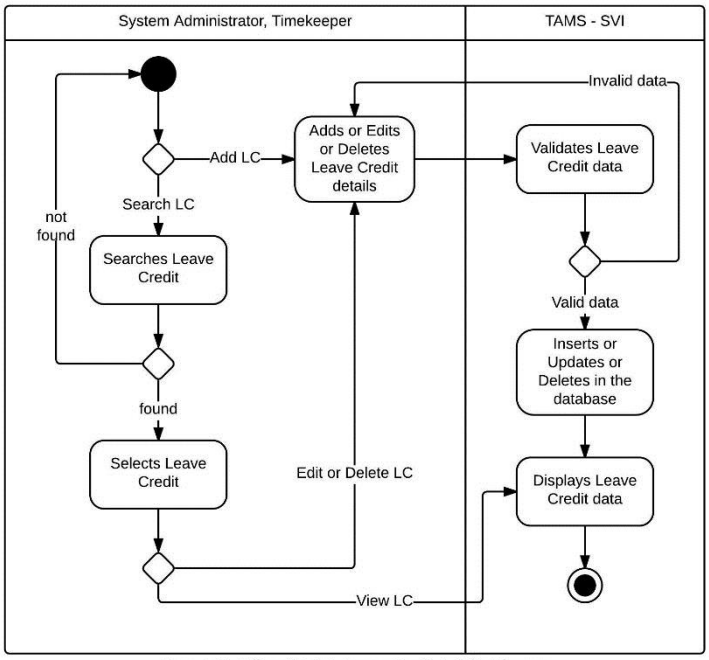
Timekeeping Maintenance Use Case Diagram

- **View, Add, Edit and Delete Leave Credit**

Leave credit determines the types of leave in the system. There are three types: Vacation Leave, Sick Leave and Paternity Leave.

To add a leave credit, the user must provide its details (name and code). Then the system validates and confirms the data before adding it into the database and displays the data. To view a credit, the user first searches all the leave credits in the database before making a selection. Upon selecting a leave credit, the system displays the data of the credit. To edit or delete a leave credit, the user first searches for the credit then selects and edit or delete the details. The system validates and confirms the changes before updating or deleting it in the database then displays the data.

The following is the illustration of the View, Add, Edit and Delete Leave Credit Activity Diagram.



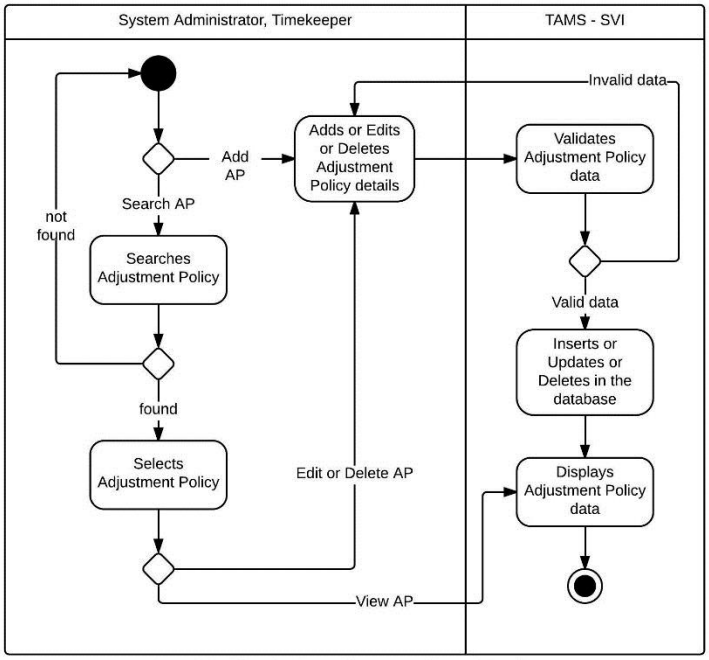
View, Add, Edit and Delete Leave Credit Activity Diagram

- **View, Add, Edit and Delete Adjustment Policy**

The adjustment policy determines the rate at which an employee's work hours are multiplied to. A few examples are 'Regular' which has the rate 100%, 'Night Differential' which has the additional rate of 10%, and 'Overtime work' which has the rate of 125%.

To add an adjustment policy, the user should provide its details (name, code and rate). The system then validates and confirms the data before adding it in the database. The system will display the policy data. To view, edit and delete a policy, the user first searches in the database. If found, the user will select a policy to view or edit or delete. For viewing, the system will display the details of the policy. For editing, the user provides the necessary changes in the policy. For deleting, the user deletes the details of the policy. The system will validate and confirm the changes before updating or deleting in the database then displays the edited adjustment policy.

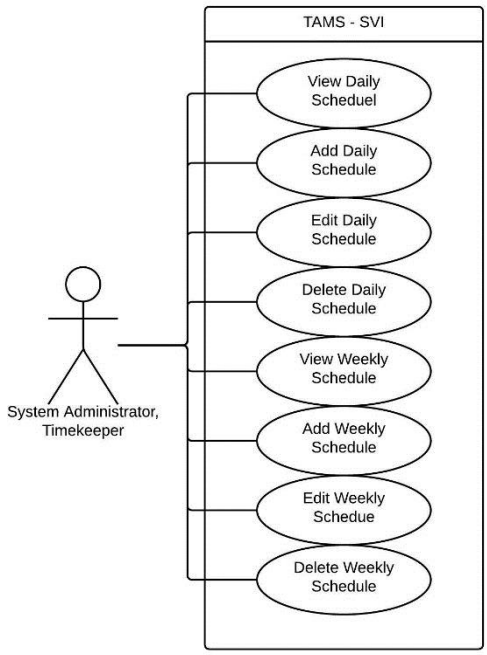
Below is the View, Add, Edit and Delete Adjustment Policy Activity Diagram.



View, Add, Edit and Delete Adjustment Policy Activity Diagram

- **Schedule Management**

The Schedule Management contains all the schedules available in the system. The following is its Use Case Diagram.



Schedule Management Use Case Diagram

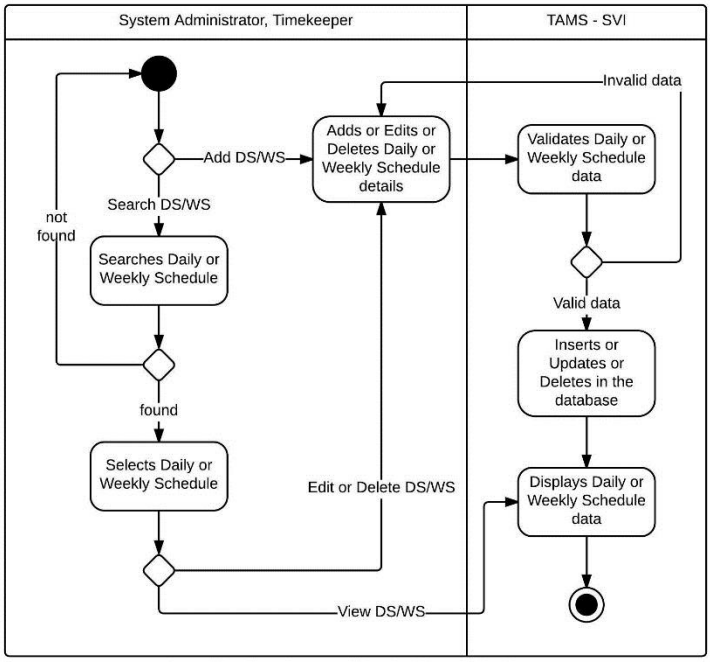
- **View, Add, Edit and Delete Daily/Weekly Schedule**

The daily schedule determines in what time the employee must report for work and get out of work. The weekly schedule determines in which days the employee must work within the week.

To add a daily or weekly schedule, the user should provide its details (daily schedule: name, 1st time in, 1st time out, 2nd time in, 2nd time out, 3rd time in, 3rd time out and duration; weekly schedule: name and work schedule per day). The system then validates and confirms the data before adding it in the database. The system will display the schedule data. To view, edit and delete a schedule, the user first searches in the database. If found, the user will select a schedule to view or edit or delete. For viewing, the system will display the details of the schedule. For editing, the user provides the necessary changes in the schedule. For deleting, the user deletes the details of the schedule. The

system will validate and confirm the changes before updating or deleting in the database then displays the edited daily or weekly schedule.

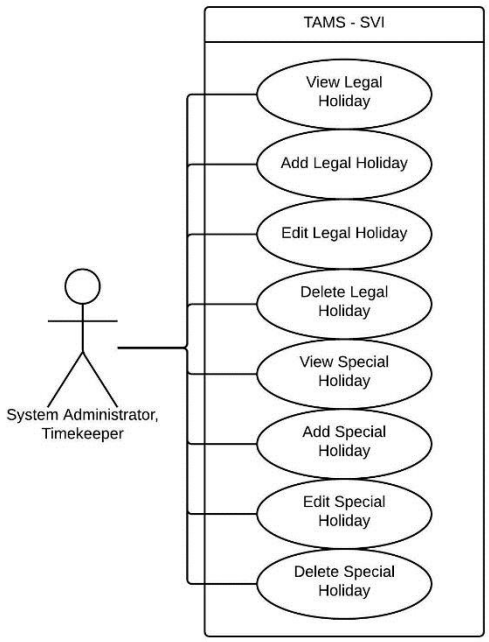
The following is the activity diagram of viewing, adding, editing and deleting of daily or weekly schedule.



View, Add, Edit Daily or Weekly Schedule Activity Diagram

- **Holiday Management**

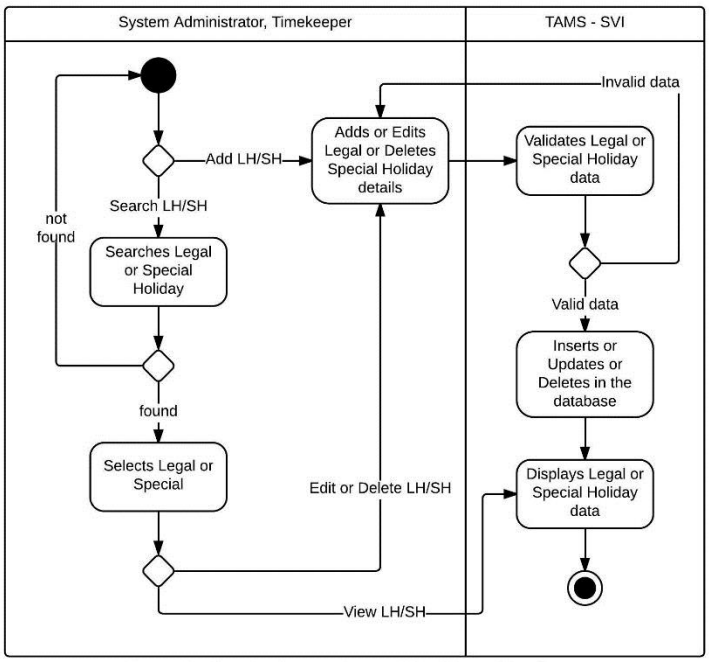
Holiday is also a factor in the timekeeping and payroll process. There are two types of holidays: the legal (regular) and the special holiday. The legal holiday is implemented to all branches while the special holiday is only implemented to certain branches. The Holiday Management contains the holidays in the system.



Holiday Management Use Case Diagram

- **View, Add, Edit and Delete Legal/Special Holiday**

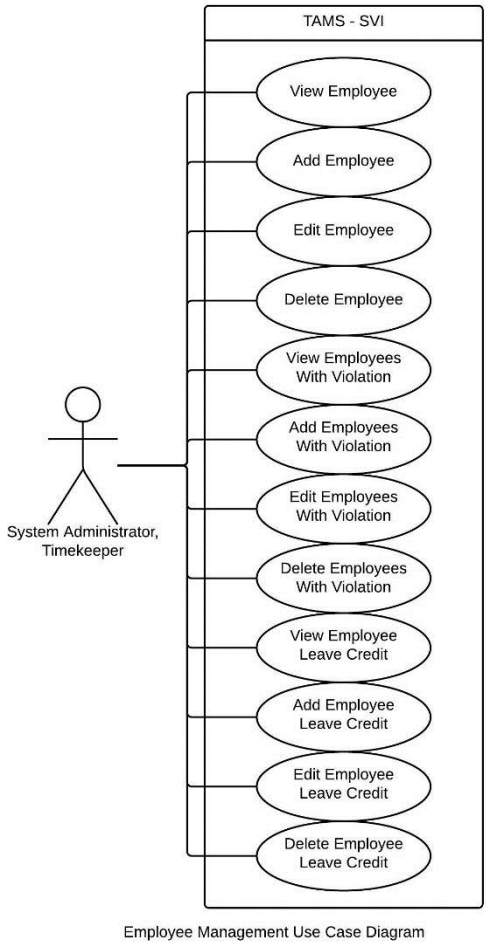
To add a legal or special holiday, the user should provide its details (legal holiday: name and date; special holiday: name, date, location). The system then validates and confirms the data before adding it in the database. The system will display the holiday data. To view, edit and delete a holiday, the user first searches in the database. If found, the user will select a holiday to view or edit or delete. For viewing, the system will display the details of the holiday. For editing, the user provides the necessary changes in the holiday. For deleting, the user deletes the details of the holiday. The system will validate and confirm the changes before updating or deleting in the database then displays the edited legal or special holiday.



View, Add, Edit and Delete Legal or Special Holiday Activity Diagram

- **Employee Management**

Employee Management contains employee profile, employees with violation and the employee leave remaining credit. Its use case diagram is the following illustration.



Employee Management Use Case Diagram

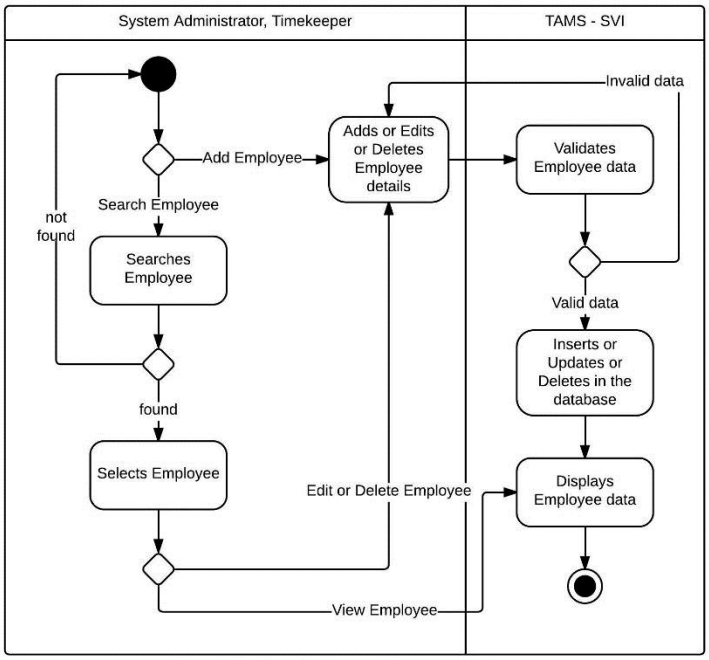
- **View, Add, Edit and Delete Employee**

Employee profile contains the details of an employee: last name, first name, middle name, employee number, branch, department, designation, employee status and employee weekly schedule.

To add an employee profile, the user should provide its details. The system then validates and confirms the data before adding it in the database. The system will display the profile data. To view, edit and delete a profile, the user first searches in the database. If found, the user will select a profile to view or edit or delete. For viewing, the system will display the details of the profile. For editing, the user provides the necessary changes in

the profile. For deleting, the user deletes the details of the profile. The system will validate and confirm the changes before updating or deleting in the database then displays the edited employee profile.

The following illustrates the activity diagram for viewing, adding, editing and deleting of employee profile.



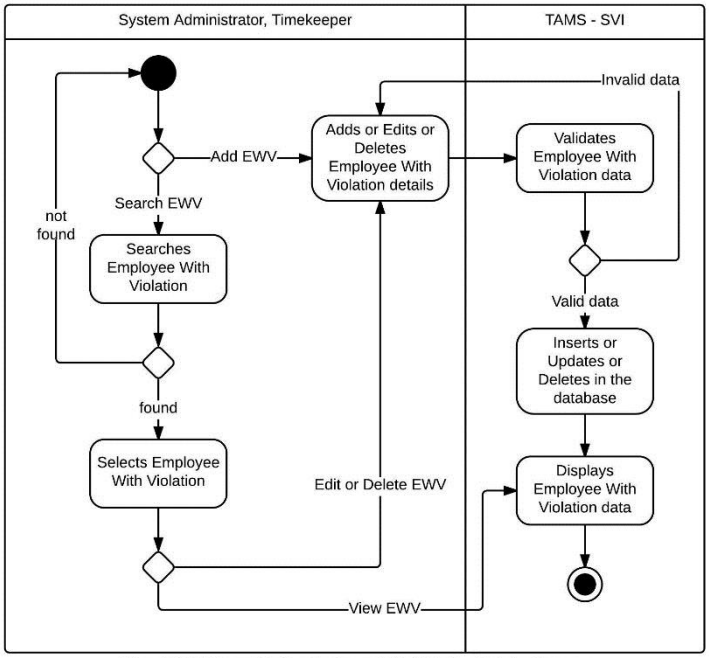
View, Add, Edit and Delete Employee Activity Diagram

- **View, Add, Edit and Delete Employee With Violation**

The Employee With Violation contains the list of the employees who committed violations against the company.

To add an employee with violation, the user should provide its details (employee number, employee name, date, violation, offense rank and penalty). The system then validates and confirms the data before adding it in the database. The system will display the employee data. To view, edit and delete an employee, the user first searches in the database. If found, the user will select an employee to view or edit or delete. For viewing,

the system will display the details of the employee. For editing, the user provides the necessary changes in the employee. For deleting, the user deletes the details of the employee with violation. The system will validate and confirm the changes before updating or deleting in the database then displays the edited employee with violation.



View, Add, Edit and Delete Employee With Violation Activity Diagram

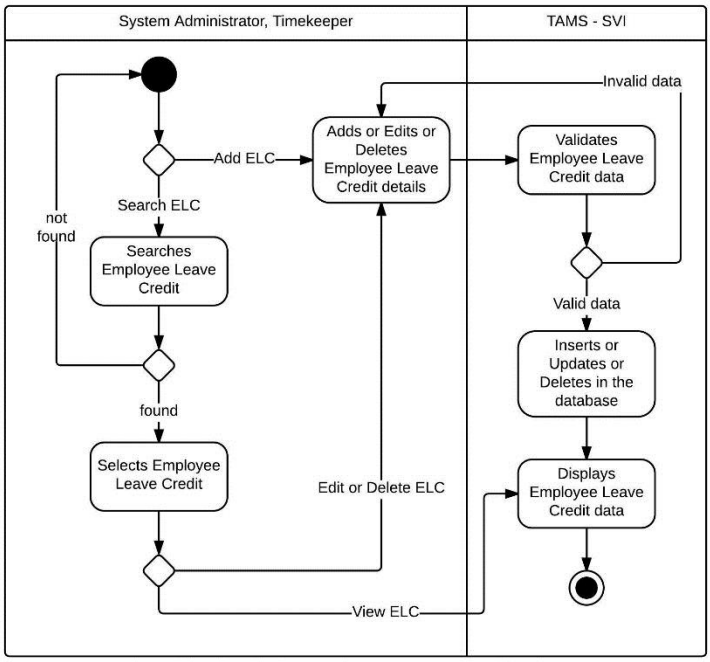
- **View, Add, Edit and Delete Employee Leave Credit**

The Employee Leave Credit contains the list of the remaining credits of an employee per leave type.

To add an employee leave remaining credit, the user should provide its details (employee name, leave credit code/type, total number of leave). The system then validates and confirms the data before adding it in the database. The system will display the remaining credit data. To view, edit and delete a remaining credit, the user first searches in the database. If found, the user will select a remaining credit to view or edit or delete. For viewing, the system will display the details of the remaining credit. For editing, the user

provides the necessary changes in the remaining credit. For deleting, the user deletes the details of the employee leave remaining credit. The system will validate and confirm the changes before updating or deleting in the database then displays the employee leave remaining credit.

The following illustrates the activity diagram of the employee leave remaining credit.

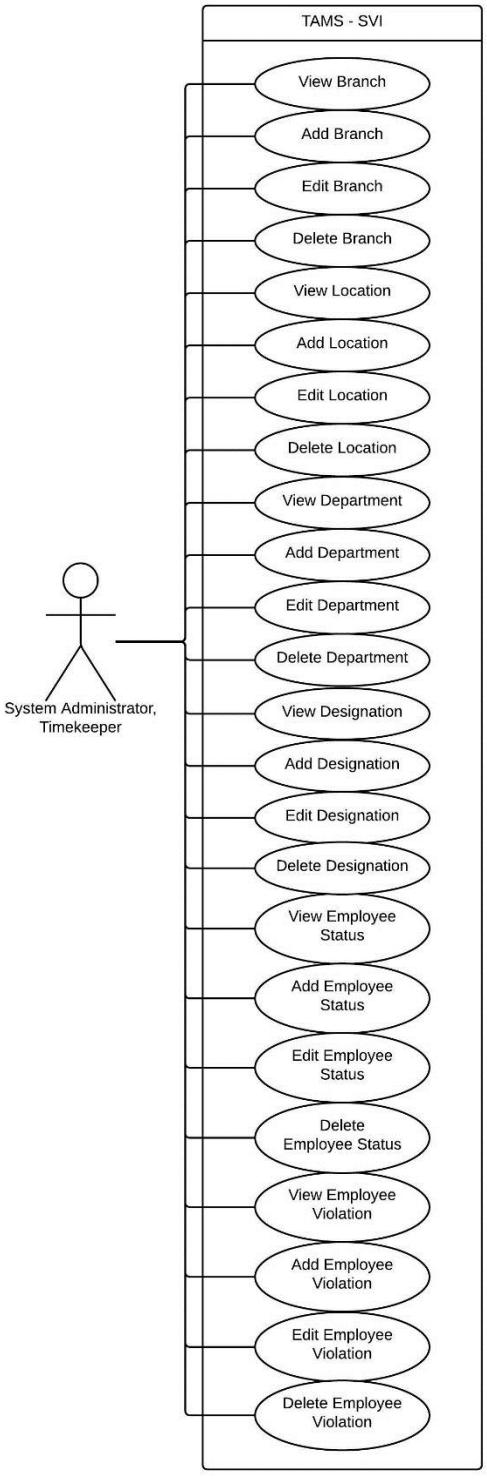


View, Add, Edit and Delete Employee Leave Credit Activity Diagram

- **Employee Maintenance**

The Employee Maintenance contains the maintenance of the employee profile. As stated above, the employee profile consists of branch (with its location), department, designation, employee status and employee violation.

The following illustrates the use case diagram of the Employee Maintenance.



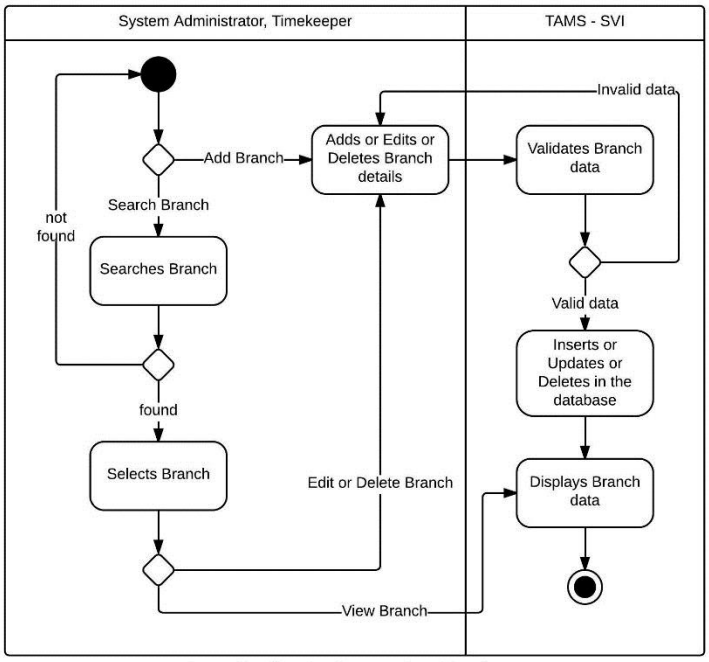
Employee Maintenance Use Case Diagram

- **View, Add, Edit and Delete Branch**

Branch determines where the employees will be assigned to work.

To add a branch, the user should provide its details (branch name and location). The system then validates and confirms the data before adding it in the database. The system will display the branch data. To view, edit and delete a branch, the user first searches in the database. If found, the user will select a branch to view or edit or delete. For viewing, the system will display the details of the branch. For editing, the user provides the necessary changes in the branch. For deleting, the user deletes the details of the branch. The system will validate and confirm the changes before updating or deleting in the database then displays the branch.

Below is the activity diagram of viewing, adding, editing and deleting of branch.



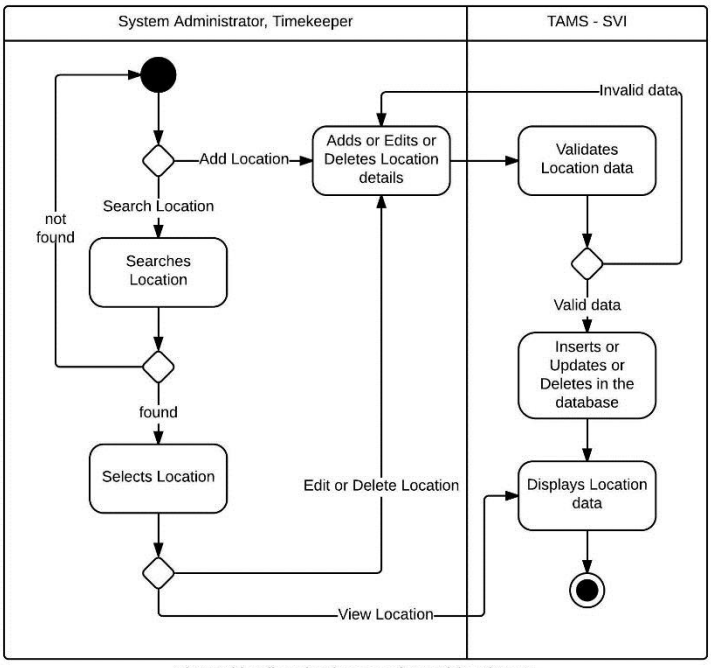
View, Add, Edit and Delete Branch Activity Diagram

- **View, Add, Edit and Delete Location**

Location refers to the geographical place where the branch is erected. Location is also used in the special holiday where it takes place.

To add a location, the user should provide its details (city name). The system then validates and confirms the data before adding it in the database. The system will display the location data. To view, edit and delete a location, the user first searches in the database. If found, the user will select a location to view or edit or delete. For viewing, the system will display the details of the location. For editing, the user provides the necessary changes in the location. For deleting, the user deletes the details of the location. The system will validate and confirm the changes before updating or deleting in the database then displays the location.

The following is an illustration of the activity diagram of viewing, adding, editing and deleting a location.



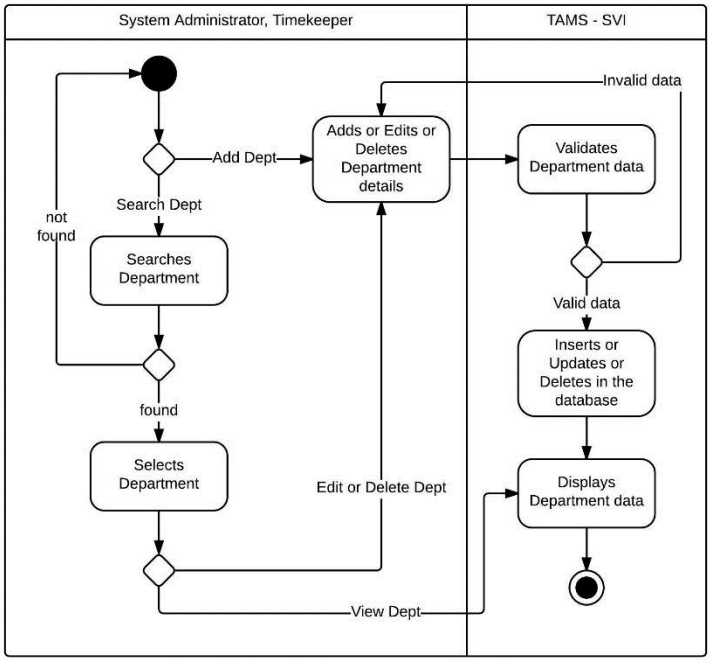
View, Add, Edit and Delete Location Activity Diagram

- **View, Add, Edit and Delete Department**

Department is a section in the company where the employee will be assigned.

To add a department, the user should provide its details (department name). The system then validates and confirms the data before adding it in the database. The system will display the department data. To view, edit and delete a department, the user first searches in the database. If found, the user will select a department to view or edit or delete. For viewing, the system will display the details of the department. For editing, the user provides the necessary changes in the department. For deleting, the user deletes the department details. The system will validate and confirm the changes before updating or deleting in the database then displays the department.

The following illustrates the activity diagram of viewing, adding, editing and deleting a department.



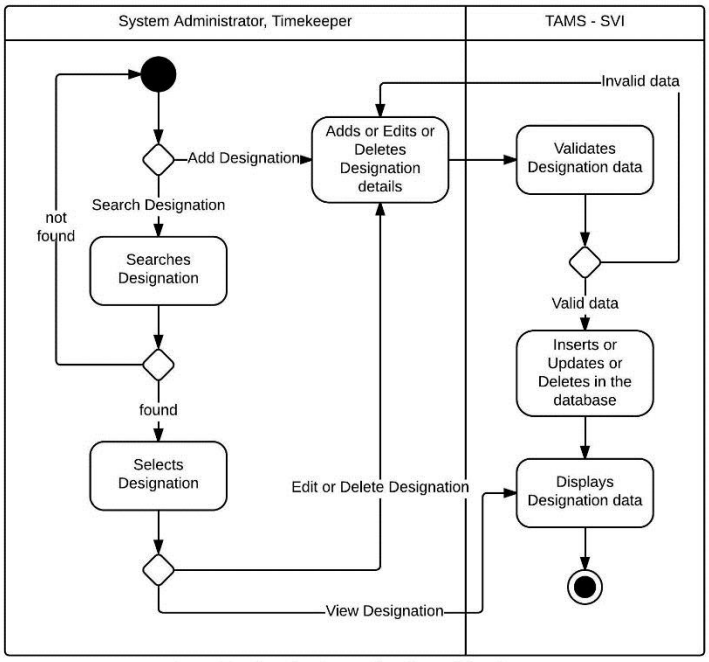
View, Add, Edit and Delete Department Activity Diagram

- **View, Add, Edit and Delete Designation**

Designation classifies the employees' responsibilities in the company.

To add a designation, the user should provide its details (designation name, basic pay and COLA). The system then validates and confirms the data before adding it in the database. The system will display the designation data. To view, edit and delete a designation, the user first searches in the database. If found, the user will select a designation to view or edit or delete. For viewing, the system will display the details of the designation. For editing, the user provides the necessary changes in the designation. For deleting, the user deletes the designation details. The system will validate and confirm the changes before updating or deleting in the database then displays the designation.

The following illustrates the activity diagram of viewing, adding, editing and deleting of designation.



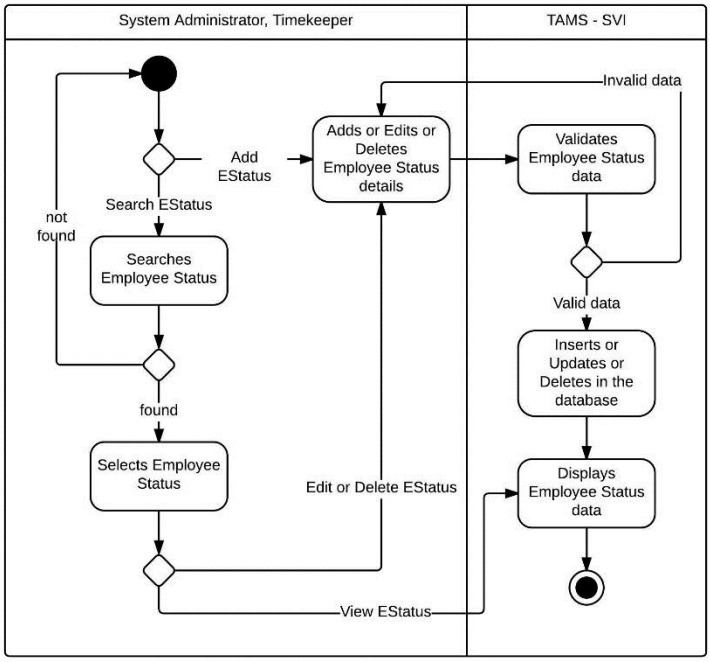
View, Add, Edit and Delete Designation Activity Diagram

- **View, Add, Edit and Delete Employee Status**

An employee has a status. It can be regular, contractual, part time or any other classification.

To add an employee status, the user should provide its details (employee status name). The system then validates and confirms the data before adding it in the database. The system will display the employee status data. To view, edit and delete an employee status, the user first searches in the database. If found, the user will select an employee status to view or edit or delete. For viewing, the system will display the details of the employee status. For editing, the user provides the necessary changes in the status. For deleting, the user deletes the employee status details. The system will validate and confirm the changes before updating or deleting in the database then displays the employee status.

Below is the activity diagram illustration of viewing, adding, editing and deleting an employee status.



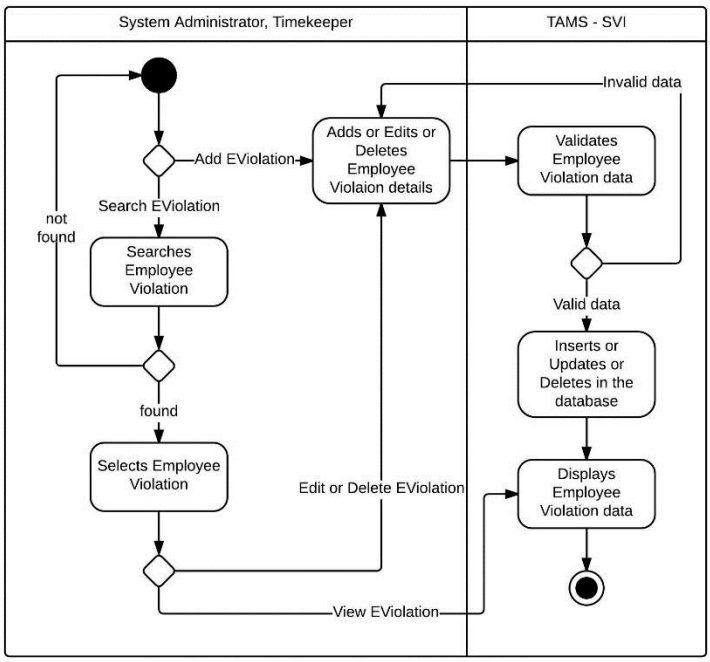
View, Add, Edit and Delete Employee Status Activity Diagram

- **View, Add, Edit and Delete Employee Violation**

Employee Violation contains the list of the offenses against the company with its corresponding penalty.

To add an employee violation, the user should provide its details (employee violation name, description, offense rank and penalty). The system then validates and confirms the data before adding it in the database. The system will display the employee violation data. To view, edit and delete an employee violation, the user first searches in the database. If found, the user will select an employee violation to view or edit or delete. For viewing, the system will display the details of the employee violation. For editing, the user provides the necessary changes in the violation. For deleting, the user deletes the employee violation details. The system will validate and confirm the changes before updating or deleting in the database then displays the employee violation.

The following illustrates the activity diagram of viewing, adding, editing and deleting an employee violation.

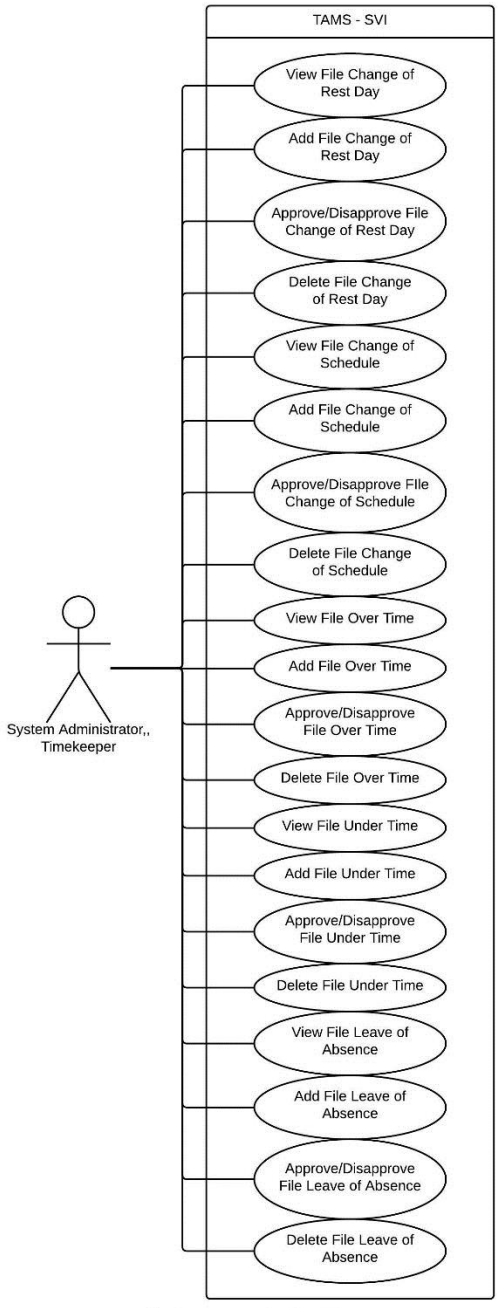


View, Add, Edit and Delete Employee Violation Activity Diagram

- **File Management**

File Management contains all the filed data (filed undertimes, filed over times, filed changes of rest day, filed changes in shift schedule, filed leave of absences) of the employees.

The following is the Use Case Diagram of the File Management.



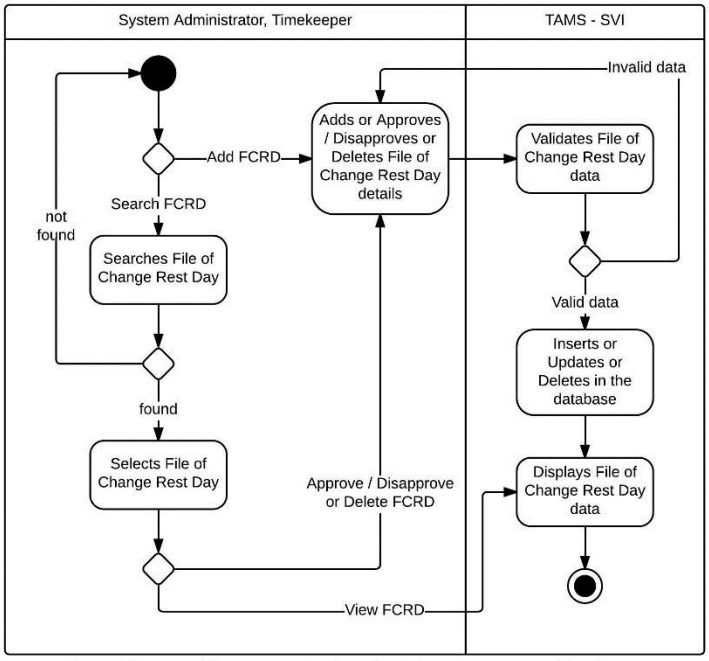
File Management Use Case Diagram

- **View, Add, Edit and Delete File Change of Rest Day**

Employees may change their rest days provided that they have a valid reason and the approval of their supervisors and department head. File of Change of Rest Day contains the list of the filed changes in the rest days of the employees.

To add a file of change in rest day, the user should provide its details (employee number, employee name, date, reason and approval/disapproval). The system then validates and confirms the data before adding it in the database. The system will display the file of change in rest day data. To view, approve/disapprove and delete a file of change in rest day, the user first searches in the database. If found, the user will select a file of change in rest day to view or approve/disapprove or delete. For viewing, the system will display the details of the file of change in rest day. For approval or disapproval, the user provides the necessary changes in the file of change in rest day. For deleting, the user deletes the data of the file of change in rest day. The system will validate and confirm the changes before updating or deleting in the database then displays the file of change in rest day.

The following illustrates the activity diagram of viewing, adding, approval/disapproval and deleting of file of change in rest day.



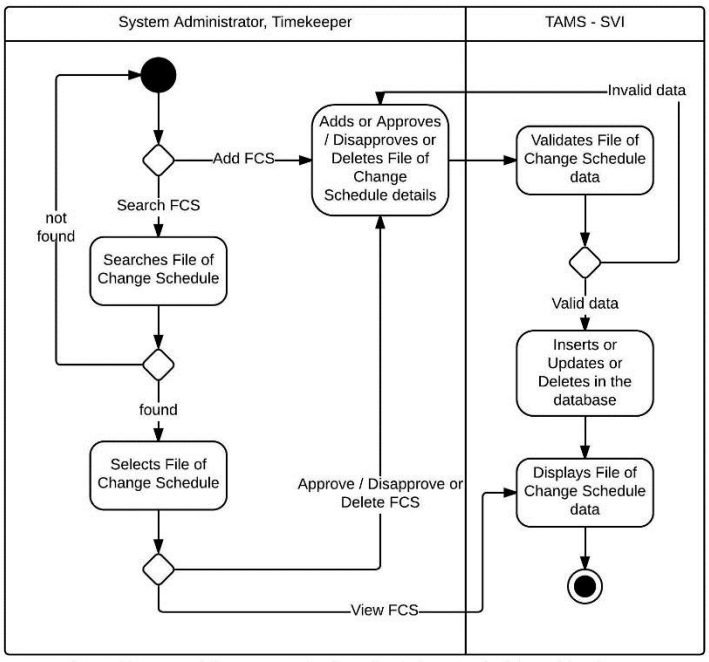
View, Add, Approve/Disapprove and Delete File of Change Rest Day Activity Diagram

- **File Change of Shift Schedule**

Changes in the shift schedule are also allowed with a valid reason and the approval of the employee's immediate supervisor and department head.

To add a file of change in shift schedule, the user should provide its details (employee number, employee name, date, original schedule, change schedule, reason and approval/disapproval). The system then validates and confirms the data before adding it in the database. The system will display the file of change in shift schedule data. To view, approve/disapprove and delete a file of change in shift schedule, the user first searches in the database. If found, the user will select a file of change in shift schedule to view or approve/disapprove or delete. For viewing, the system will display the details of the file of change in shift schedule. For approval or disapproval, the user provides the necessary changes in the file of change in shift schedule. For deleting, the user deletes the file of change rest day data. The system will validate and confirm the changes before updating or deleting in the database then displays the file of change in shift schedule.

The following is an illustration of the activity diagram of the viewing, adding, approval/disapproval and deleting of file of change in shift schedule.



View, Add, Approve/Disapprove and Delete File of Change Schedule Activity Diagram

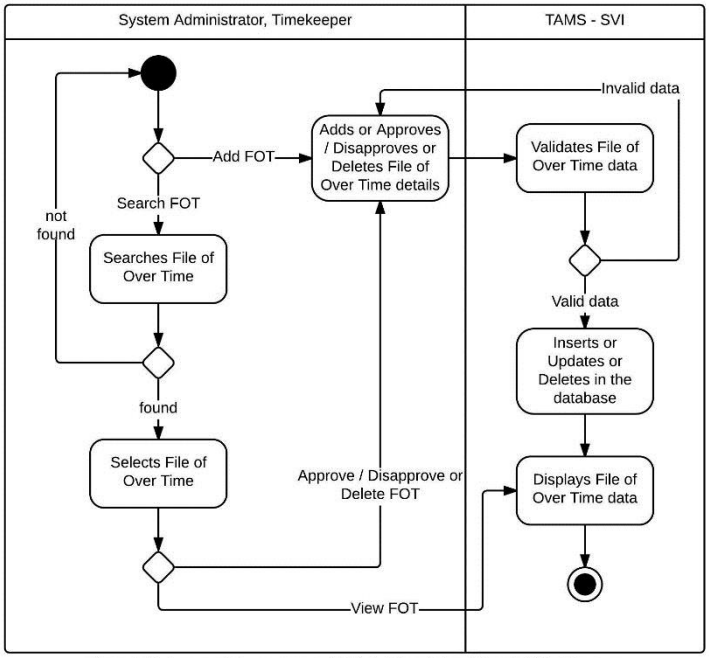
- **View, Add, Edit and Delete File of Overtime**

Over time in work happens when an employee exceeds his work duration for the day. However, over times are not automatically computed with the regular hours worked. It also has to be filed with a valid reason and the approval of the immediate supervisor and department head.

To add a filed overtime, the user should provide its details (employee number, employee name, date, reason and approval/disapproval). The system then validates and confirms the data before adding it in the database. The system will display the filed overtime data. To view, approve/disapprove and delete a filed overtime, the user first searches in the database. If found, the user will select a filed overtime to view or approve/disapprove or delete. For viewing, the system will display the details of the filed overtime. For approval or disapproval, the user provides the necessary changes in the filed overtime. For deleting, the user deletes the data of the filed overtime. The system will

validate and confirm the changes before updating or deleting in the database then displays the filed overtime.

The following illustrates the activity diagram of viewing, adding, approval/disapproval and deleting of filed overtime.



View, Add, Approve/Disapprove and Delete File of Over Time Activity Diagram

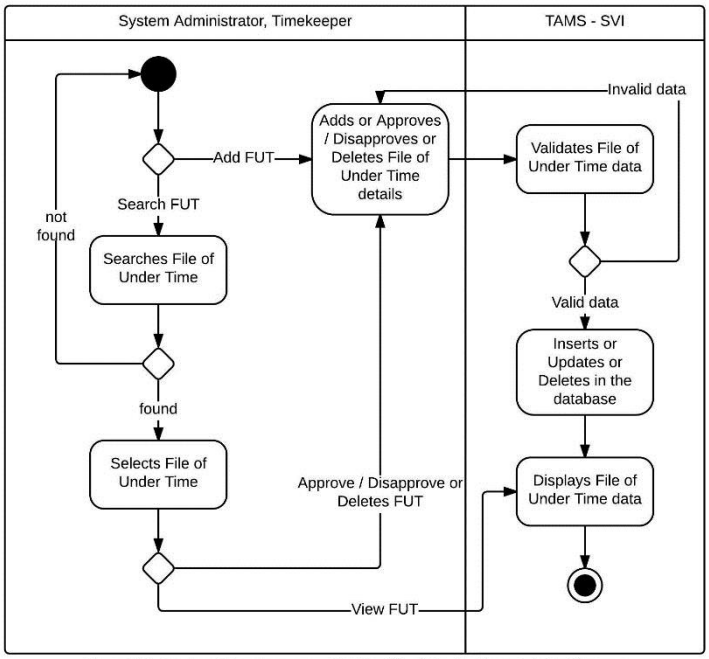
- **View, Add, Edit and Delete File of Undertime**

Not only overtime work has to be filed, but also the under time work. Under time work happens if the employee does not complete his work duration for the day.

To add a filed undertime, the user should provide its details (employee number, employee name, date, reason and approval/disapproval). The system then validates and confirms the data before adding it in the database. The system will display the filed undertime data. To view, approve/disapprove and delete a filed undertime, the user first searches in the database. If found, the user will select a filed undertime to view or approve/disapprove or delete. For viewing, the system will display the details of the filed

undertime. For approval/disapproval, the user provides the necessary changes in the filed undertime. For deleting, the user deletes the data of the filed undertime. The system will validate and confirm the changes before updating the database then displays the filed undertime.

The following illustrates the activity diagram of viewing, adding, approval/disapproval and deleting of filed undertime.



View, Add, Approve/Disapprove and Delete File of Under Time Activity Diagram

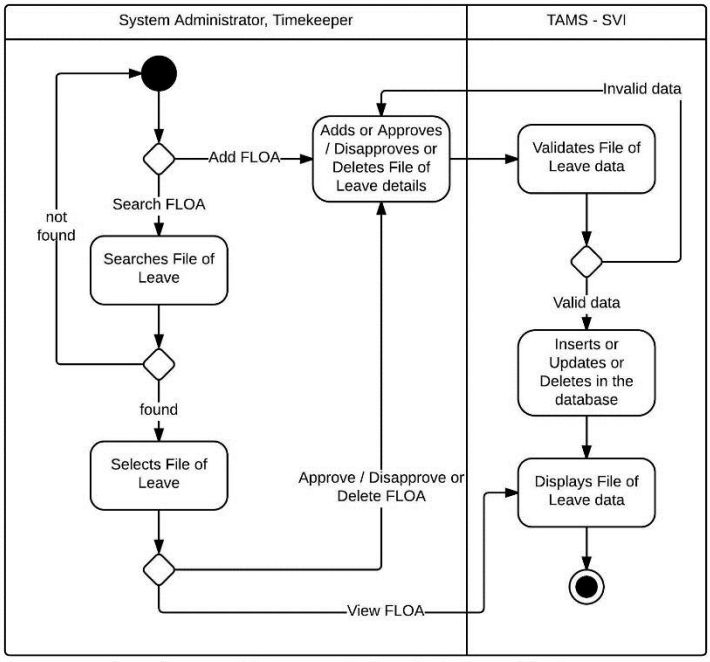
- **View, Add, Edit and Delete File Leave of Absence**

Employees are entitled to leaves of absences. A leave of absence has to be filed with a valid reason and approval of immediate supervisor and department head.

To add a filed leave of absence, the user should provide its details (employee number, employee name, filing date, leave type, start date, end date, reason and approval/disapproval). The system then validates and confirms the data before adding it in

the database. The system will display the filed leave of absence data. To view, approve/disapprove and delete a filed leave of absence, the user first searches in the database. If found, the user will select a filed leave of absence to view or approve/disapprove or delete. For viewing, the system will display the details of the filed leave of absence. For approval/disapproval, the user provides the necessary changes in the filed leave of absence. For deleting, the user deletes the filed leave of absence data. The system will validate and confirm the changes before updating or deleting in the database then displays the filed leave of absence.

The following is an illustration of the activity diagram for viewing, adding, approval/disapproval and deleting of filed leave of absence.

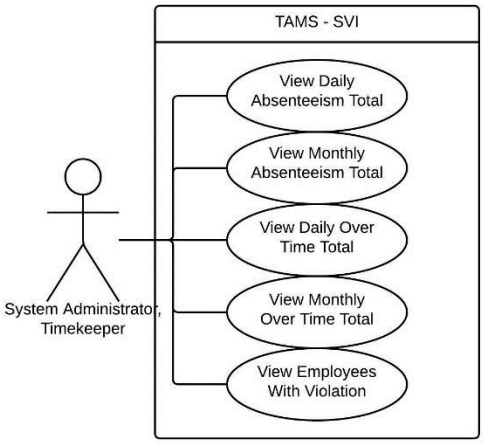


View, Add, Approve/Disapprove and Delete File of Leave Activity Diagram

- **Report Management**

For easier viewing of the total absences and total overtimes on a daily and monthly basis, a Report Management is made.

The following is the use case diagram of the Report Management.



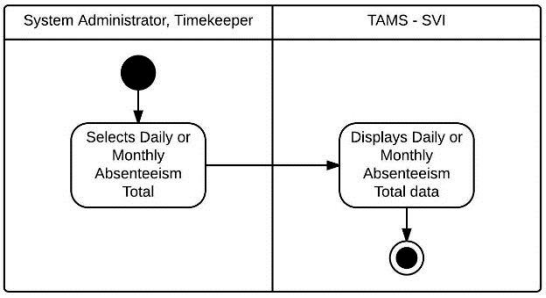
Report Management Use Case Diagram

- **View Daily/Monthly Absenteeism Total**

The Absenteeism Total contains the list of employees with absences in a certain date or month.

The daily or monthly absenteeism total can be viewed by selecting the option in the treeview. Then the system displays the daily or monthly absenteeism total data.

The following is the activity diagram of viewing the daily or monthly absenteeism total.



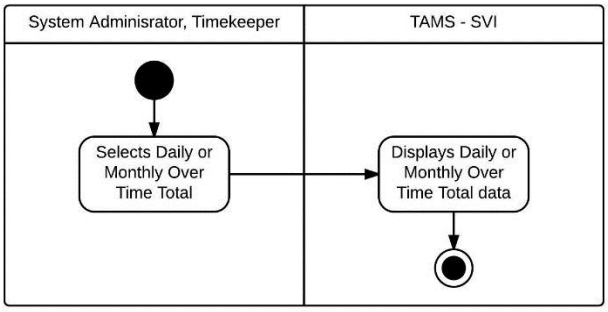
View Absenteeism Total Activity Diagram

- **View Daily/Monthly Overtime Total**

The Overtime Total contains the list of employees who has overtime work in a certain date or month.

The daily or monthly overtime total can be viewed by selecting the option in the treeview. Then the system displays the daily or monthly overtime total data.

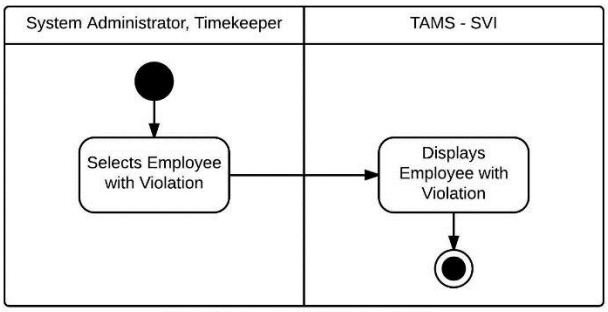
The following is the activity diagram of viewing the daily or monthly overtime total.



View Over Time Total Activity Diagram

- **View Employees With Violation**

By selecting the Employees with Violation, the user can view the list of employees who committed a violation on a certain day with its corresponding offense rank and penalty.

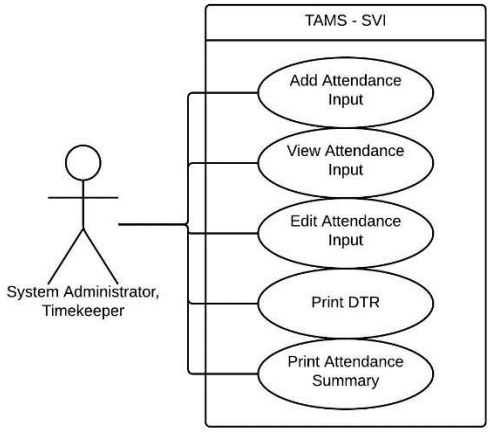


View Employee with Violation Activity Diagram

- **Attendance Input**

The most essential part of this timekeeping system is the Attendance Input. Users of the system can input the attendance of the employees with ease. The results that will be shown is in Crystal Reports that can be exported as PDFs.

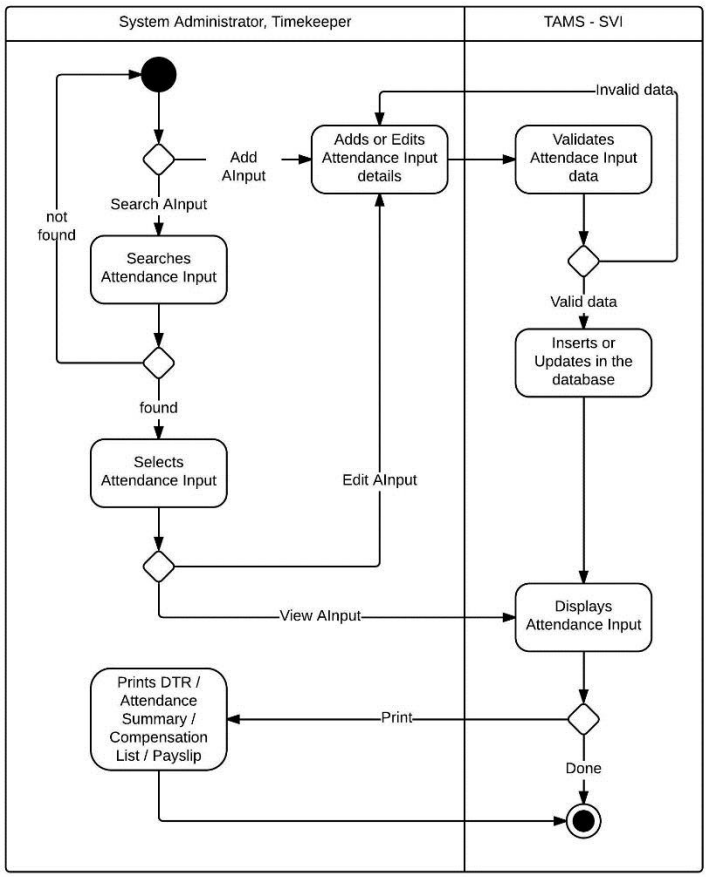
The following is the use case diagram of the Attendance Input.



Attendance Input Use Case Diagram

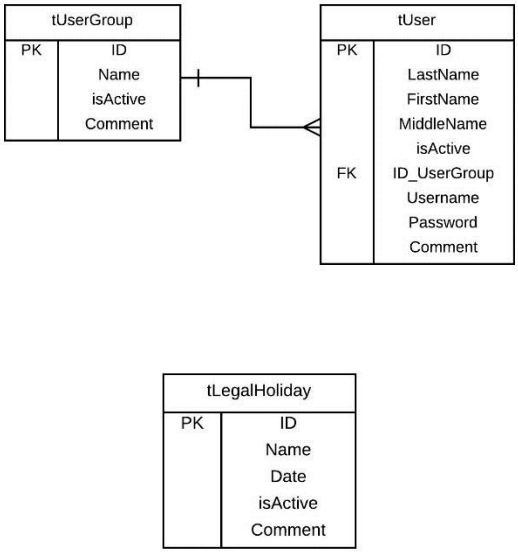
To add an attendance, the user should specify the details (date: from start to end, branch, department and designation). The input window will appear with all the employees' details (employee number, employee name, date, schedule, ins and outs). After the input of the attendance details, the user can save data. If the input data is already in the database then the system will just display the data stored in the database. The system then displays the data. The user may have the option to print the DTR (Daily Time Record), Attendance Summary, Compensation List and the Payslip of the employees.

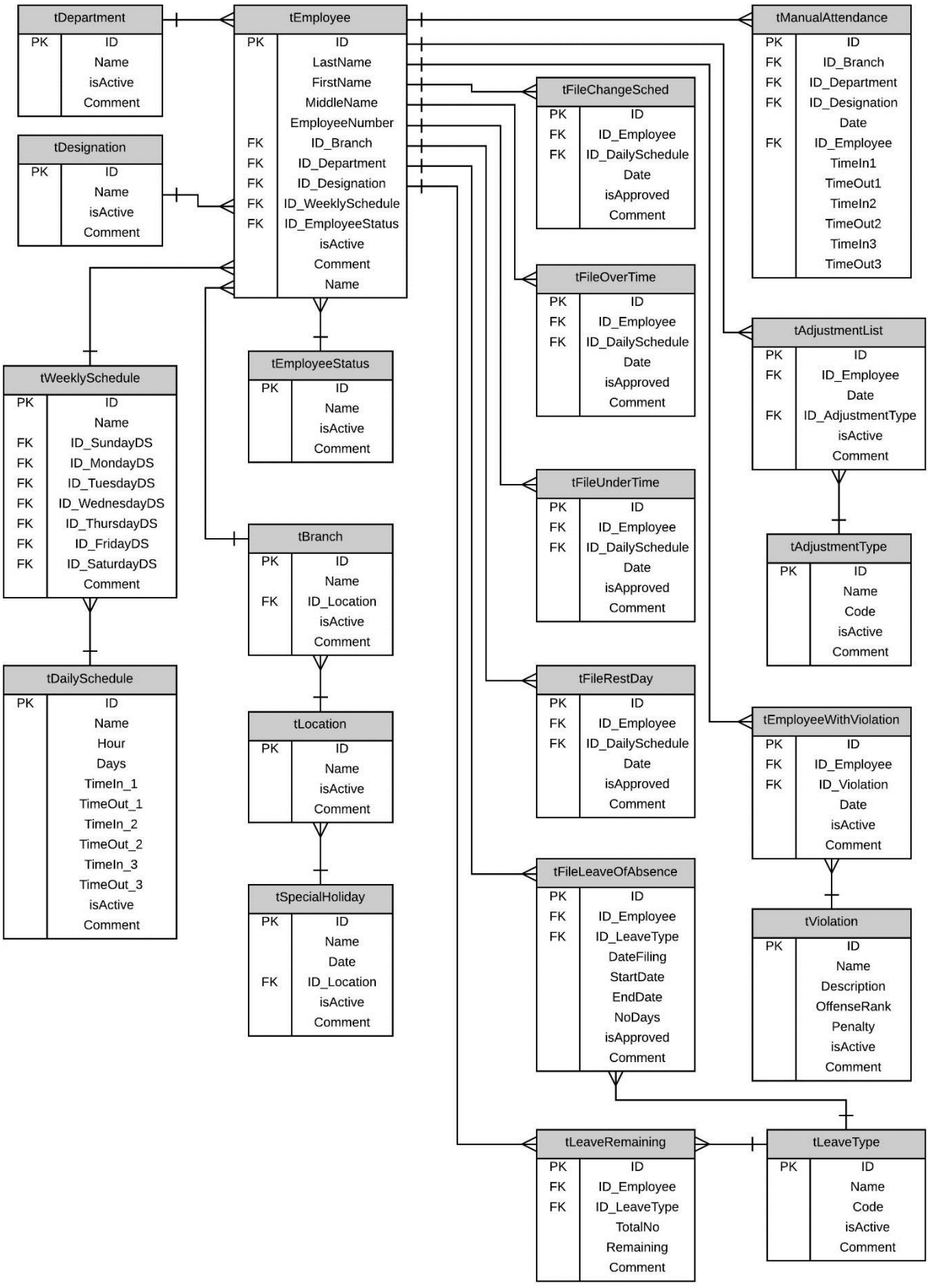
The following illustrates the activity diagram of the Attendance Input.



View, Add, Edit, Print Attendance Input Activity Diagram

B. ERD


tUserGroup
PK
ID
Name
isActive
Comment
tUser
PK
ID
LastName
FirstName
MiddleName
isActive
FK
ID_UserGroup
Username
Password
Comment
tLegalHoliday
PK
ID
Name
Date
isActive
Comment


tDepartment
PK ID
Name
isActive
Comment
tDesignation
PK ID
Name
isActive
Comment
tEmployee
PK ID
LastName
FirstName
MiddleName
EmployeeNumber
FK ID_Branch
FK ID_Department
FK ID_Designation
FK ID_WeeklySchedule
FK ID_EmployeeStatus
isActive
Comment
Name
tManualAttendance
PK ID
FK ID_Branch
FK ID_Department
FK ID_Designation
Date
FK ID_Employee
TimeIn1
TimeOut1
TimeIn2
TimeOut2
TimeIn3
TimeOut3
tFileChangeSched
PK ID
FK ID_Employee
FK ID_DailySchedule
Date
isApproved
Comment
tFileOverTime
PK ID
FK ID_Employee
FK ID_DailySchedule
Date
isApproved
Comment
tAdjustmentList
PK ID
FK ID_Employee
Date
FK ID_AdjustmentType
isActive
Comment
tEmployeeStatus
PK ID
Name
isActive
Comment
tWeeklySchedule
PK ID
Name
FK ID_SundayDS
FK ID_MondayDS
FK ID_TuesdayDS
FK ID_WednesdayDS
FK ID_ThursdayDS
FK ID_FridayDS
FK ID_SaturdayDS
Comment
tFileUnderTime
PK ID
FK ID_Employee
FK ID_DailySchedule
Date
isApproved
Comment
tAdjustmentType
PK ID
Name
Code
isActive
Comment
tBranch
PK ID
Name
FK ID_Location
isActive
Comment
tFileRestDay
PK ID
FK ID_Employee
FK ID_DailySchedule
Date
isApproved
Comment
tEmployeeWithViolation
PK ID
FK ID_Employee
FK ID_Violation
Date
isActive
Comment
tDailySchedule
PK ID
Name
Hour
Days
TimeIn_1
TimeOut_1
TimeIn_2
TimeOut_2
TimeIn_3
TimeOut_3
isActive
Comment
tLocation
PK ID
Name
isActive
Comment
tFileLeaveOfAbsence
PK ID
FK ID_Employee
FK ID_LeaveType
DateFiling
StartDate
EndDate
NoDays
isApproved
Comment
tViolation
PK ID
Name
Description
OffenseRank
Penalty
isActive
Comment
tSpecialHoliday
PK ID
Name
Date
FK ID_Location
isActive
Comment
tLeaveRemaining
PK ID
FK ID_Employee
FK ID_LeaveType
TotalNo
Remaining
Comment
tLeaveType
PK ID
Name
Code
isActive
Comment

C. Data Dictionary

ENTITY NAME: tAdjustmentPolicy – contains the type of the adjustment policy

ATTRIBUTE	DESCRIPTION	DATA TYPE
ID	Unique number that identifies the adjustment policy	int
Name	Name of the adjustment policy	varchar(255)
Code	Adjustment policy code	varchar(20)
Rate	Rate of the adjustment policy used in the computation of employee compensation	decimal(18,3)
isActive	Marks the adjustment policy as active or inactive	bit
Comment	Other remarks of the adjustment policy	varchar(8000)

ENTITY NAME: tAttendanceComputation – contains the ins and outs of the employee for the day and their corresponding time computations

ATTRIBUTE	DESCRIPTION	DATA TYPE
ID	Unique number that identifies the attendance computation	int
ID_Employee	ID of the employee	int
Date	Date of the attendance	datetime
ID_DS	ID of the daily schedule of the employee for that date	int
S_In1	Scheduled Time In 1 of the employee	datetime
A_In1	Actual Time In 1 of the employee	datetime
In1diff	Difference of the Scheduled Time In 1 from the Actual Time In 1	Computed, int
A_Out1	Actual Time Out 1 of the employee	datetime
A_In2	Actual Time In 2 of the employee	datetime
LunchDuration	Lunch duration of the employee from the actual Time In 2 and Time Out 1	Computed, int
A_Out2	Actual Time Out 2 of the employee	datetime
A_In3	Actual Time In 3 of the employee	datetime
SnackDuration	Snack duration of the employee from the actual Time In 3 and Time Out 2	Computed, int
S_Out3	Scheduled Time Out 3 of the employee	datetime
A_Out3	Actual Time Out 3 of the employee	datetime
ND_Out	Scheduled Out of night differential	datetime
ND_In	Scheduled In of night differential	datetime
Out3diff	Difference of the Scheduled Time Out 3 from the Actual Time Out 3	Computed, int
RegularHours	Computed Regular Working Hours worked of the employee in minutes	Computed, int
ActualDuration	Computed actual duration of working hours of the employee in minutes	Computed, int
OT	Computed Overtime hours of the employee	Computed, numeric(17,6)
ND	Computed Night Differential hours of the employee	Computed, numeric(17,6)
OTND	Computed Overtime with Night Differential hours of the employee	Computed, numeric(17,6)
WRD	Computed Work on Rest Day hours of the employee	Computed, numeric(17,6)
WRDND	Computed Work on Rest Day with Night Differential hours of the employee	Computed, numeric(17,6)
OTRD	Computed Overtime Work on Rest Day hours of the employee	Computed, numeric(17,6)
OTRDND	Computed Overtime Work on Rest Day with Night Differential hours of the employee	Computed, numeric(17,6)
SH	Computed Work on Special Holiday hours of the employee	Computed, numeric(17,6)
SHND	Computed Work on Special Holiday with Night Differential hours of the employee	Computed, numeric(17,6)
SHOT	Computed Overtime Work on Special Holiday hours of the employee	Computed, numeric(17,6)

SHOTND	Computed Overtime Work on Special Holiday with Night Differential hours of the employee	Computed, numeric(17,6)
WRH	Computed Work on Regular Holiday hours of the employee	Computed, numeric(17,6)
WRHND	Computed Work on Regular Holiday with Night Differential hours of the employee	Computed, numeric(17,6)
OTRH	Computed Overtime Work on Regular Holiday hours of the employee	Computed, numeric(17,6)
OTRHND	Computed Overtime Work on Regular Holiday with Night Differential hours of the employee	Computed, numeric(17,6)
WSR	Computed Work on Special Holiday Falling on Rest Day hours of the employee	Computed, numeric(17,6)
WSRND	Computed Work on Special Holiday Falling on Rest Day with Night Differential hours of the employee	Computed, numeric(17,6)
OTSR	Computed Overtime Work on Special Holiday hours of the employee	Computed, numeric(17,6)
OTSRND	Computed Overtime Work on Special Holiday with Night Differential hours of the employee	Computed, numeric(17,6)
WRR	Computed Work on Regular Holiday Falling on Rest Day hours of the employee	Computed, numeric(17,6)
WRRND	Computed Work on Regular Holiday Falling on Rest Day with Night Differential hours of the employee	Computed, numeric(17,6)
OTRR	Computed Overtime Work on Regular Holiday hours of the employee	Computed, numeric(17,6)
OTRRND	Computed Overtime Work on Regular Holiday with Night Differential hours of the employee	Computed, numeric(17,6)
Late	Computed late minutes of the employee	Computed, int
Absent	Marks the employee as absent or not	bit
HalfDay	Computed halfday work of the employee	Computed, int
isRestDay	Marks the employee if on rest day or not	bit
isLeave	Marks the employee if on leave or not	bit
isSpecial	Marks the date as special holiday or not	bit
isLegal	Marks the date as regular holiday or not	bit
isOvertime	Marks the employee if he has a filed overtime for that day	bit
isRegular	Marks the date as a regular working day or not	bit
UT	Computed undertime minutes of the employee	Computed, int
VL	Marks the employee if he is in Vacation Leave or not	bit
SL	Marks the employee if he is in Sick Leave or not	bit
PL	Marks the employee if he is in Paternity Leave or not	bit

ENTITY NAME: tAuditTrail – contains the trail of the system

ATTRIBUTE	DESCRIPTION	DATA TYPE
ID	Unique number that identifies the trail	int
ID_Username	ID of the username involved in the changes in the system	int
Action	Action made by the user	varchar(8000)
DateModified	Date and time of the modification in the system	datetime

ENTITY NAME: tBranch – contains the branches of the company

ATTRIBUTE	DESCRIPTION	DATA TYPE
ID	Unique number that identifies the branch of an employee	int
Name	Name of the branch of an employee	varchar(255)
ID_Location	Location ID of the branch	int
isActive	Marks the branch as active or inactive	bit
Comment	Other remarks of the branch	varchar(8000)

ENTITY NAME: tCompensationList – contains the compensation list of the employees

ATTRIBUTE	DESCRIPTION	DATA TYPE
ID	Unique number that identifies the compensation of the employee	int
ID_Employee	ID of the employee	int
ID_Designation	ID of the designation of the employee	int
BasicPay	Basic Pay of the employee	decimal(18,2)
COLA	COLA of the employee	decimal(18,2)
Date	Date of the compensation of the employee	datetime
Duration	Duration of the compensation of employee	int
ID_AttendanceComputation	ID of the attendance computation of the employee	int
ID_AdjPolicy	ID of the adjustment policy involved	int
Rate	Rate of the adjustment policy	decimal(18,2)
Total	Total compensation of the employee	Computed, decimal(38,6)

ENTITY NAME: tDailySchedule – contains the types of daily schedules

ATTRIBUTE	DESCRIPTION	DATA TYPE
ID	Unique number that identifies the daily schedule of an employee	int
Name	Name of the daily schedule of an employee	varchar(255)
Hour	Total hours' work of an employee for the day	decimal(18,2)
TimeIn_1	First time in of the employee	time(7)
TimeOut_1	Time out of the employee for lunch break	time(7)
TimeIn_2	Time in of the employee after lunch break	time(7)
TimeOut_2	Time out of the employee for coffee break	time(7)
TimeIn_3	Time in of the employee after coffee break	time(7)
TimeOut_3	Last time out of the employee for the day	time(7)
isActive	Marks the daily schedule as active or inactive	bit
Comment	Other remarks of the daily schedule	varchar(8000)

ENTITY NAME: tDepartment – contains the departments of the company

ATTRIBUTE	DESCRIPTION	DATA TYPE
ID	Unique number that identifies the department of the employee	int
Name	Name of the department of the employee	varchar(255)
isActive	Marks the department as active or inactive	bit
Comment	Other remarks of the department	varchar(8000)

ENTITY NAME: tDesignation – contains the positions in the company

ATTRIBUTE	DESCRIPTION	DATA TYPE
ID	Unique number that identifies the designation of the employee	int
Name	Name of the assigned designation to the employee	varchar(255)
isActive	Marks the designation as active or inactive	bit
BasicPay	Basic pay of the designation	decimal(18,2)
COLA	COLA of the designation	decimal(18,2)
Comment	Other remarks of the designation	varchar(8000)

ENTITY NAME: tEmployee – contains the employee details

ATTRIBUTE	DESCRIPTION	DATA TYPE
ID	Unique number that identifies the employee in the database	int
LastName	Last name of the employee	varchar(255)
FirstName	First name of the employee	varchar(255)
MiddleName	Middle name of the employee	varchar(255)
EmployeeNumber	Unique identifier of the employee in the company or the employee number	varchar(100)
ID_Branch	Branch ID where the employee is assigned	int
ID_Department	Department ID where the employee is assigned	int
ID_Designation	Designation ID which the employee is assigned	int
ID_WeeklySchedule	Weekly Schedule ID of the employee	int
ID_EmployeeStatus	ID of the employee's status	int
isActive	Marks the employee as active or inactive	bit
Comment	Other remarks about the employee	varchar(8000)
Name	Computed name of the employee (Last name, First name, Middle name)	varchar(768)

ENTITY NAME: tEmployeeStatus – contains the types of employees

ATTRIBUTE	DESCRIPTION	DATA TYPE
ID	Unique number that identifies the employee status	int
Name	Name of the employee status	varchar(255)
isActive	Marks the employee status as active or inactive	bit
Comment	Other remarks of the employee status	varchar(8000)

ENTITY NAME: tEmployeeWithViolation – contains the list of the employees with violations

ATTRIBUTE	DESCRIPTION	DATA TYPE
ID	Unique number that identifies the employee with violation list	int
ID_Employee	Employee ID with violation	int
ID_Violation	Violation ID committed	int
Date	Date of the violation	datetime
Comment	Other remarks of the employee with violation	varchar(8000)

ENTITY NAME: tFileChangeRestDay – contains the filed rest days

ATTRIBUTE	DESCRIPTION	DATA TYPE
ID	Unique identifier for filed rest days	int
ID_Employee	Unique identifier of employee in a database	int
Date	Date of the filed rest day	datetime
isApproved	Approval of the filed rest day	bit
Reason	Reason of the change in rest day	varchar(8000)
Comment	Other remarks of the filed rest day	varchar(8000)

ENTITY NAME: tFileChangeShiftSched – contains the filed change of schedule

ATTRIBUTE	DESCRIPTION	DATA TYPE
ID	Unique file ID number of change of schedule made by employee	int
ID_Employee	ID number of the employee requesting for change of schedule	int
ID_DS	ID number of the daily schedule to be changed	int
Date	Date of the schedule to be changed	datetime

isApproved	Approval of the change of the schedule	bit
Reason	Reason of the change of shift schedule	varchar(8000)
Comment	Other remarks of the filed change of schedule	varchar(8000)

ENTITY NAME: tFileLeaveOfAbsence – contains the filed leaves of the employee

ATTRIBUTE	DESCRIPTION	DATA TYPE
ID	Unique number that identifies the filed leaves	int
ID_Employee	Employee ID requesting the leave	int
ID_LeaveType	ID of the leave type	int
DateFiling	Date of filing the leave of absence	datetime
StartDate	Start date of the leave	datetime
EndDate	End date of the leave	datetime
Duration	Total number of days of leave	int
isApproved	Approval of the leave	bit
Reason	Reason of the leave	varchar(8000)
Comment	Comments of the LOA	varchar(8000)

ENTITY NAME: tFileOverTime – contains the filed overtimes

ATTRIBUTE	DESCRIPTION	DATA ENTRY
ID	Unique file ID number of the overtime	int
ID_Employee	ID number of the employee filing his/her overtime	int
Date	Date of the filed overtime	datetime
isApproved	Approval of the filed overtime	bit
Reason	Reason of the overtime	varchar(8000)
Comment	Other remarks of the filed overtime	varchar(8000)

ENTITY NAME: tFileUndertime – contains the filed under times

ATTRIBUTE	DESCRIPTION	DATA TYPE
ID	Unique identifier for filed under time	int
ID_Employee	Unique identifier of employee in a database	int
Date	Date of the filed under time	datetime
isApproved	Approval of the filed under time	bit
Reason	Reason of the undertime	varcchar(8000)
Comment	Other remarks of the filed under time	varchar(8000)

ENTITY NAME: tLeaveRemaining – contains the remaining number of leaves of employee

ATTRIBUTE	DESCRIPTION	DATA TYPE
ID	Unique identifier for the leave credits	int
ID_Employee	Employee ID	int
TotalNumber	Total number of leaves	int
Remaining	Total number of remaining leaves	int
ID_LeaveType	Leave Type ID	int
Comment	Other remarks of the leave credits	varchar(8000)

ENTITY NAME: tLeaveType – contains the type of leaves

ATTRIBUTE	DESCRIPTION	DATA TYPE
ID	Unique identifier for the leave type	int
Name	Name of the leave type	varchar(255)
Code	Code of the leave type	varchar(10)
isActive	Marks the leave type as active or inactive	bit
Comment	Other remarks of the leave type	varchar(8000)

ENTITY NAME: tLegalHoliday – contains the legal holidays

ATTRIBUTE	DESCRIPTION	DATA TYPE
ID	Unique number that identifies the legal holiday	int
Name	Name of the legal holiday	varchar(255)
Date	Date of the legal holiday	datetime
isActive	Marks the legal holiday as active or inactive	bit
Comment	Other remark/s of the legal holiday	varchar(8000)

ENTITY NAME: tLocation – contains the location of all the branches

ATTRIBUTE	DESCRIPTION	DATA TYPE
ID	Unique number that identifies the location	int
City	Name of the location	varchar(255)
isActive	Marks the legal holiday as active or inactive	bit
Comment	Other remark/s of the legal holiday	varchar(8000)

ENTITY NAME: tSpecialHoliday – contains the special holiday

ATTRIBUTE	DESCRIPTION	DATA TYPE
ID	Unique number that identifies the special holiday	int
Name	Name of the special holiday	varchar(255)
Date	Date of the special holiday	datetime
ID_Location	Location ID of where the holiday is implemented	int
isActive	Marks the special holiday as active or inactive	bit
Comment	Other remark/s of the special holiday	varchar(8000)

ENTITY NAME: tUser – contains the user accounts

ATTRIBUTE	DESCRIPTION	DATA TYPE
ID	Unique number that identifies the user of the system	int
LastName	Last name of the user	varchar(255)
FirstName	First name of the user	varchar(255)
MiddleName	Middle name of the user	varchar(255)
isActive	Marks the user as active or inactive	bit
isAdmin	Marks the user as an admin or not	bit
Username	Username of the user	varchar(255)
Password	Password of the user	varchar(255)
Comment	Other remarks of the user	varchar(8000)
isApproved	Marks the user as approved or not	bit
Name	Computed name of the user	Computed, varchar(768)

ENTITY NAME: tViolation – violations in the company

ATTRIBUTE	DESCRIPTION	DATA TYPE

ID	Unique number that identifies the violation	int
Name	Name of the violation	varchar(255)
Description	Description of the violation	varchar(8000)
OffenseRank	Rank of the offense	int
Penalty	Penalty of the violation	varchar(255)
isActive	Marks the violation as active or inactive	bit
Comment	Other remark/s of the weekly schedule	varchar(8000)

ENTITY NAME: tWeeklySchedule – contains the weekly schedule

ATTRIBUTE	DESCRIPTION	DATA TYPE
ID	Unique number that identifies the weekly schedule	int
Name	Name of the weekly schedule	varchar(255)
ID_MondayDS	Daily Schedule ID that will be used for Monday	int
ID_TuesdayDS	Daily Schedule ID that will be used for Tuesday	int
ID_WednesdayDS	Daily Schedule ID that will be used for Wednesday	int
ID_ThursdayDS	Daily Schedule ID that will be used for Thursday	int
ID_FridayDS	Daily Schedule ID that will be used for Friday	int
ID_SaturdayDS	Daily Schedule ID that will be used for Saturday	int
ID_SundayDS	Daily Schedule ID that will be used for Sunday	int
isActive	Marks the weekly schedule as active or inactive	bit
Comment	Other remark/s of the weekly schedule	varchar(8000)

D. Technical Requirements

Minimum Hardware Requirements:

Memory: 2GB

Hard Disk Space: 50GB

Software Requirements:

OS: Windows XP

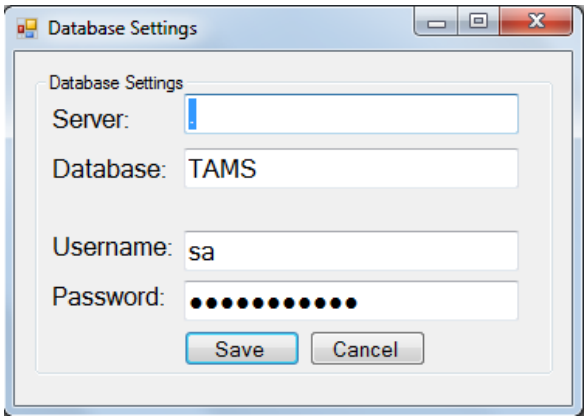
Microsoft SQL Server 2008 R2

.NET Micro Framework 4.1 SDK

V. RESULTS

A. Set up database

Before anyone can use the system, the database should be set up first that includes the server and database name as well as the username and password of the database.

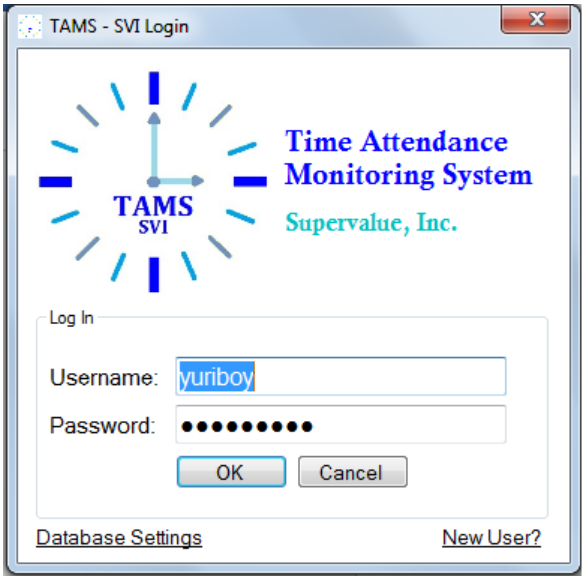


B. Login and registration into the system

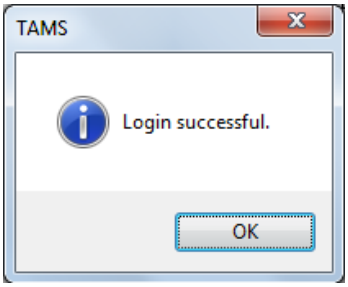
Upon the execution of the application, it launches the login form wherein users can input their account credentials to use the system. The form contains options for the unregistered users to create an account.

1. Log In

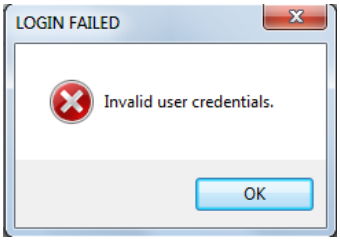
Users who have accounts must enter their credentials to be able to use the system. System administrators and timekeepers should input their credentials to be able to access the system with their respective privileges.



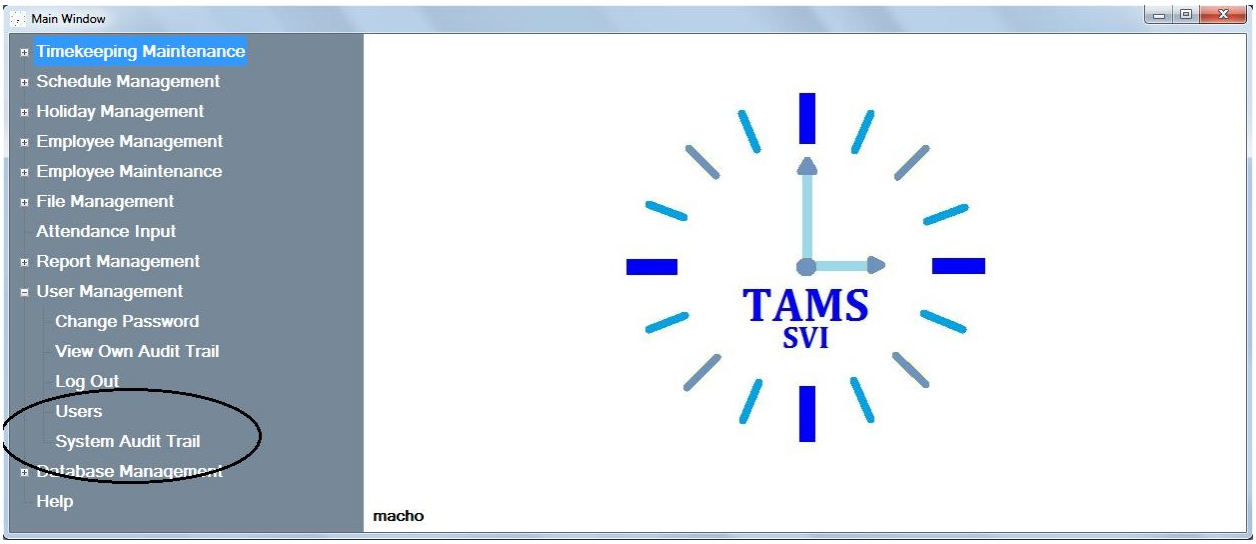
If log in is successful, a message box will appear notifying the successful log in.



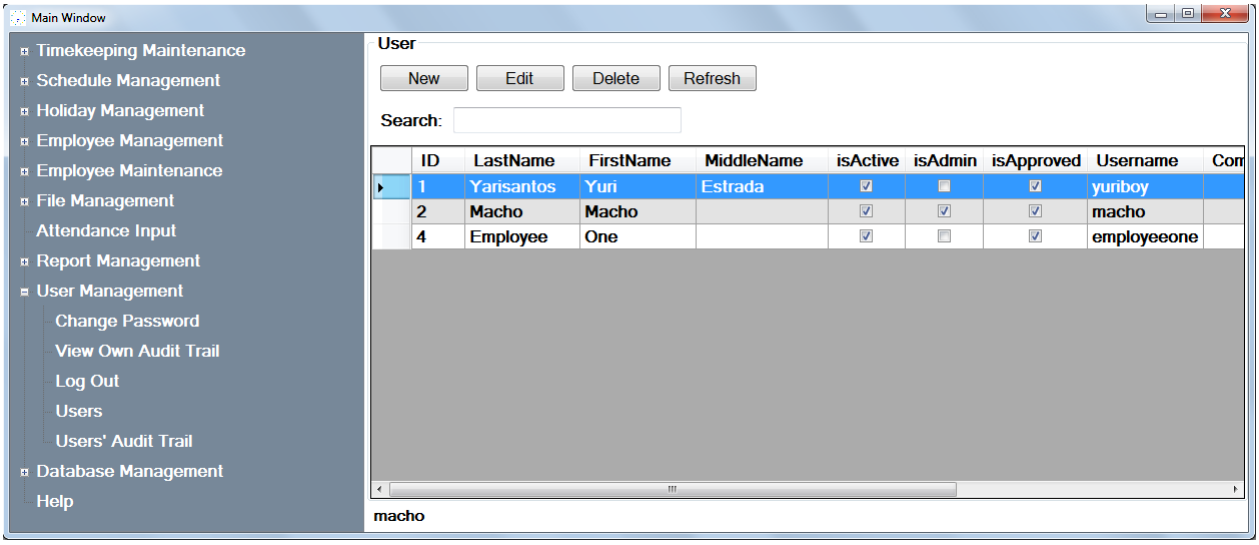
Otherwise, the message box will appear notifying wrong credential input.



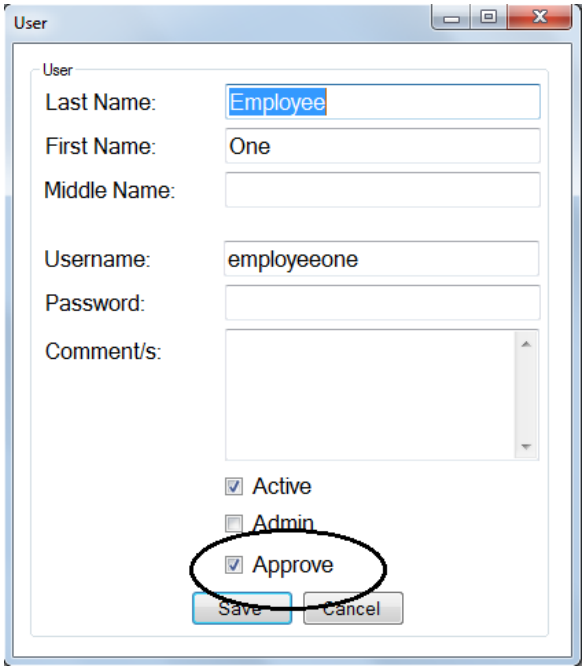
The administrator account has privileges that the timekeeper account does not have. These are adding, editing, deleting of user accounts as well as its approval. The administrator account can also view the system audit trail.



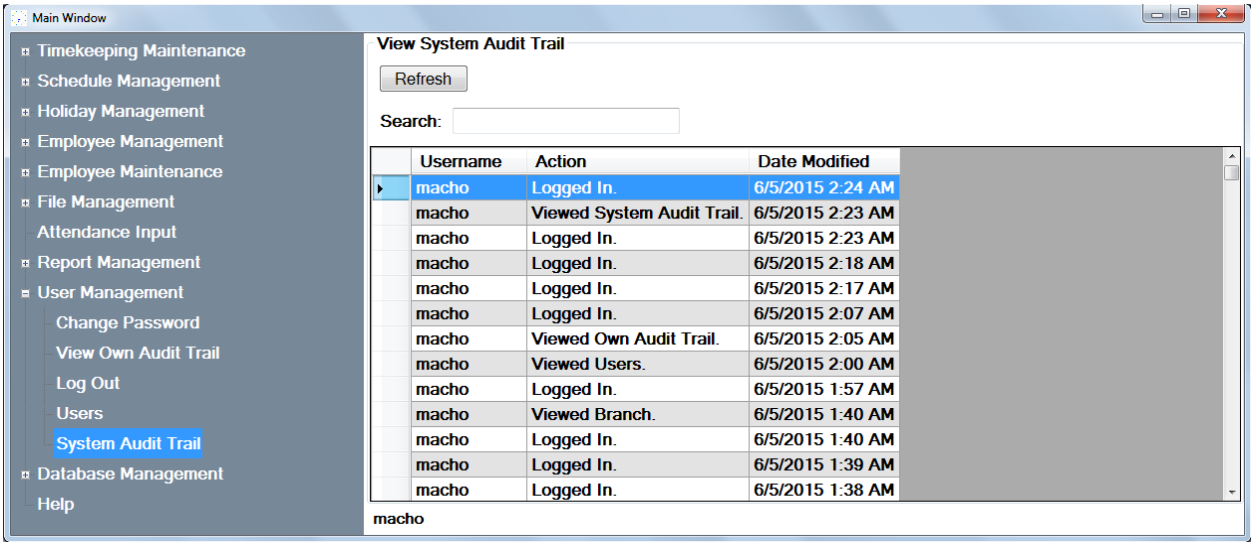
Administrator account can add new users, approve users and delete users.



The approval of the users will be on the edit button.

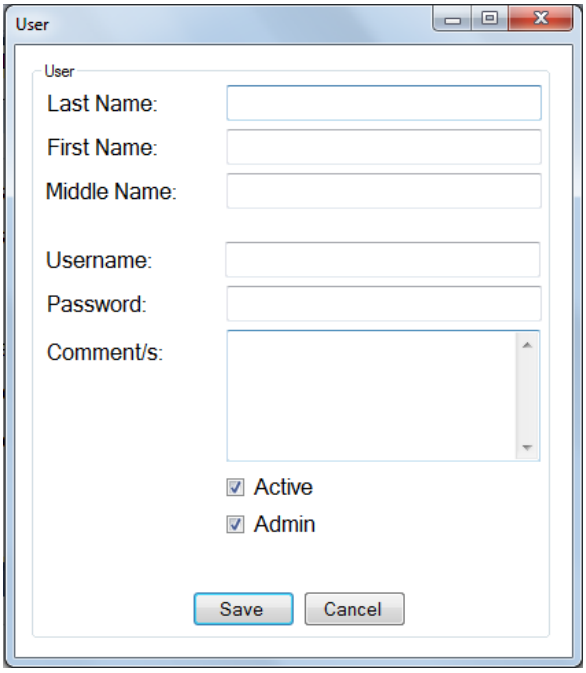


System Audit Trail



2. Register

Unregistered users cannot access the system. An account will be created upon the request of the user provided with their credentials and the approval of the system administrator.

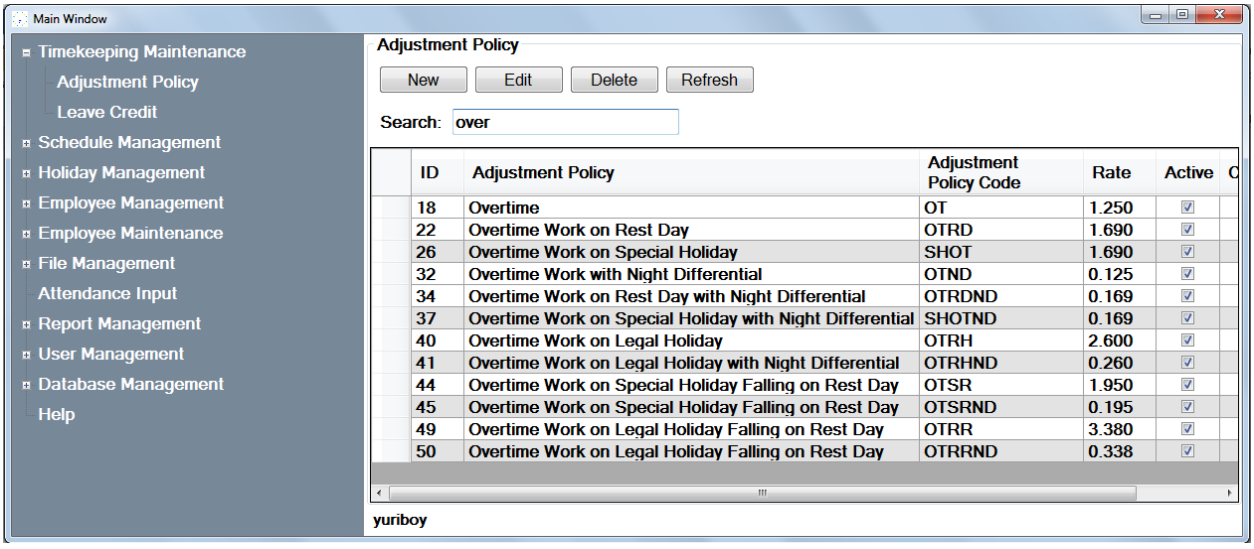


C. Search

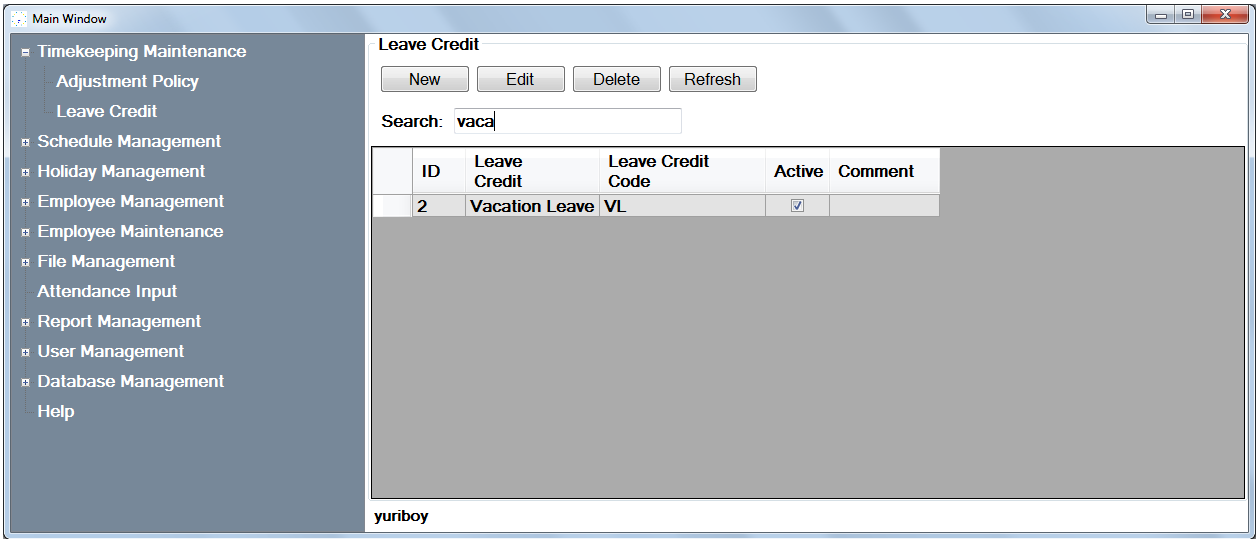
Search function enables the user to easily find things that needs to be viewed. The details of the selected item will be displayed after its selection.

1. Timekeeping Maintenance

a. Adjustment Policy

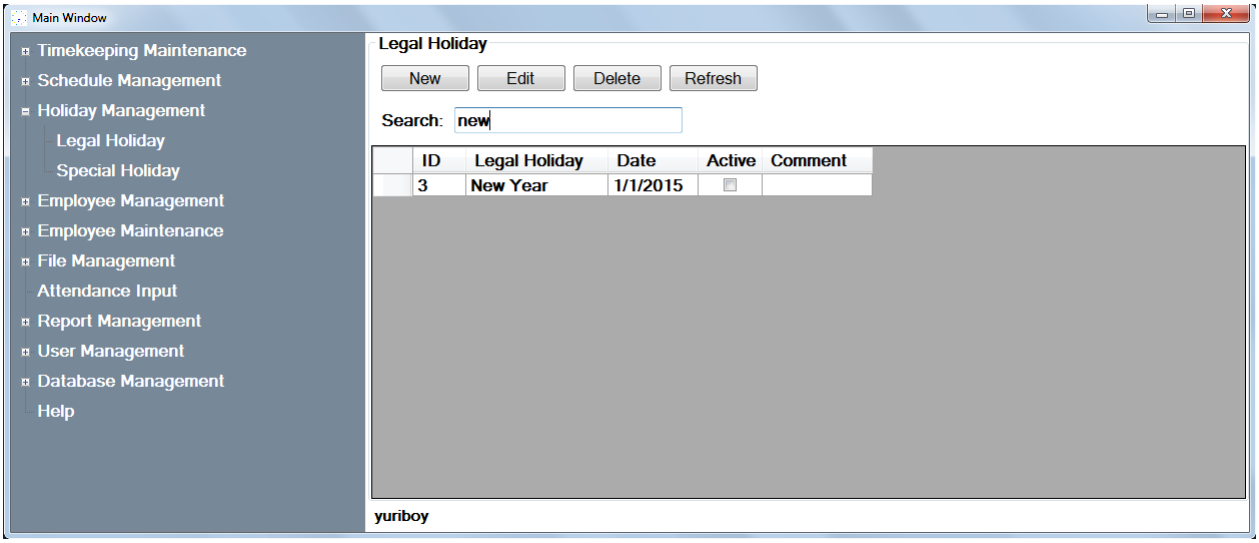


b. Leave Credit

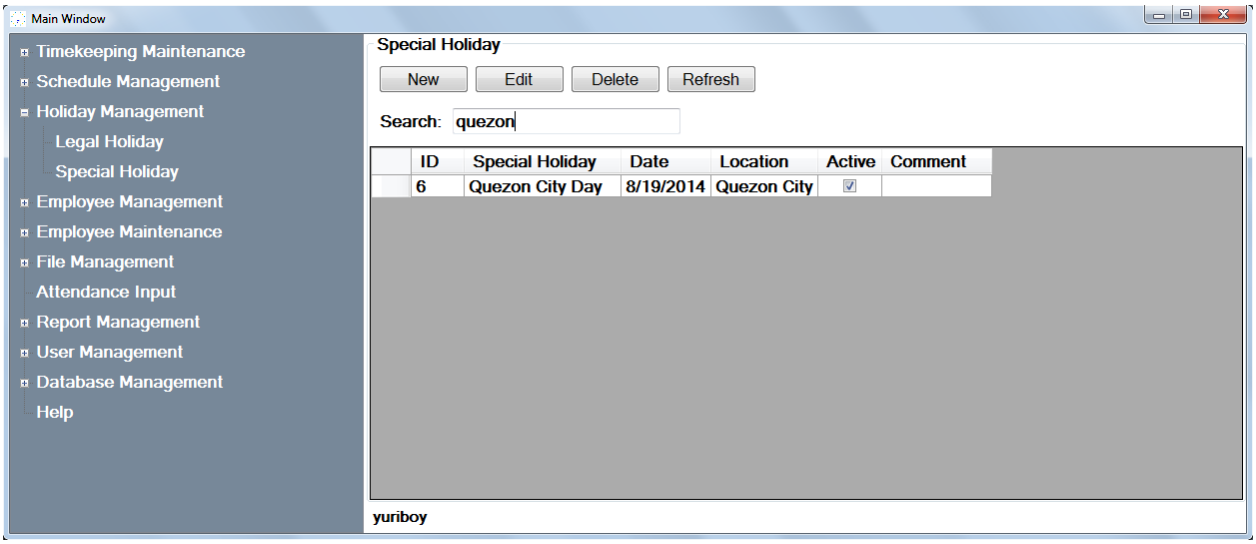


2. Holiday Management

a. Legal Holiday

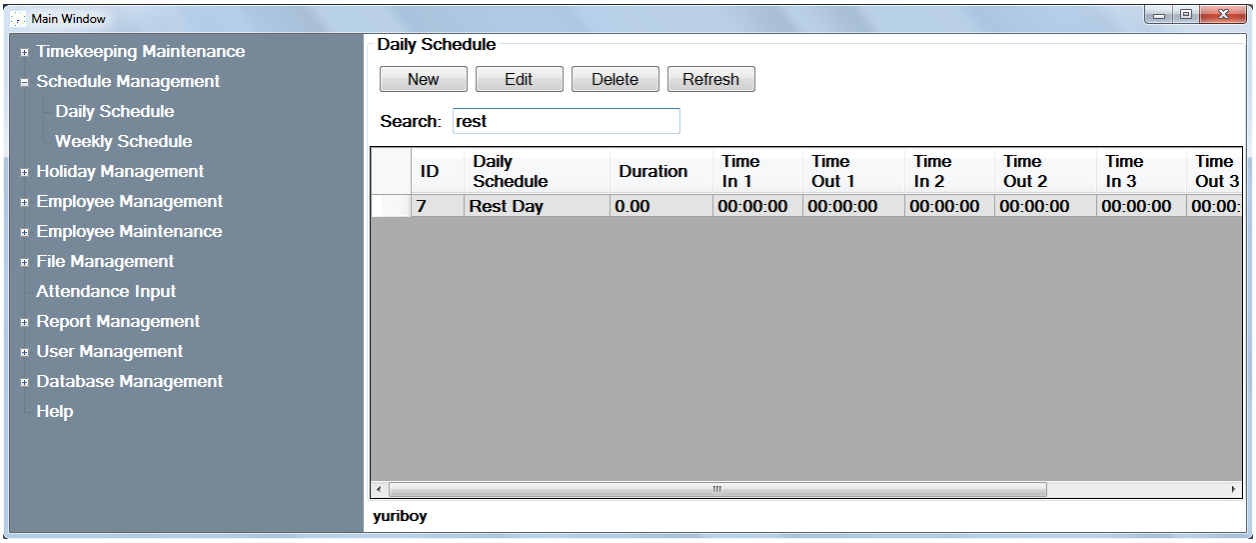


b. Special Holiday

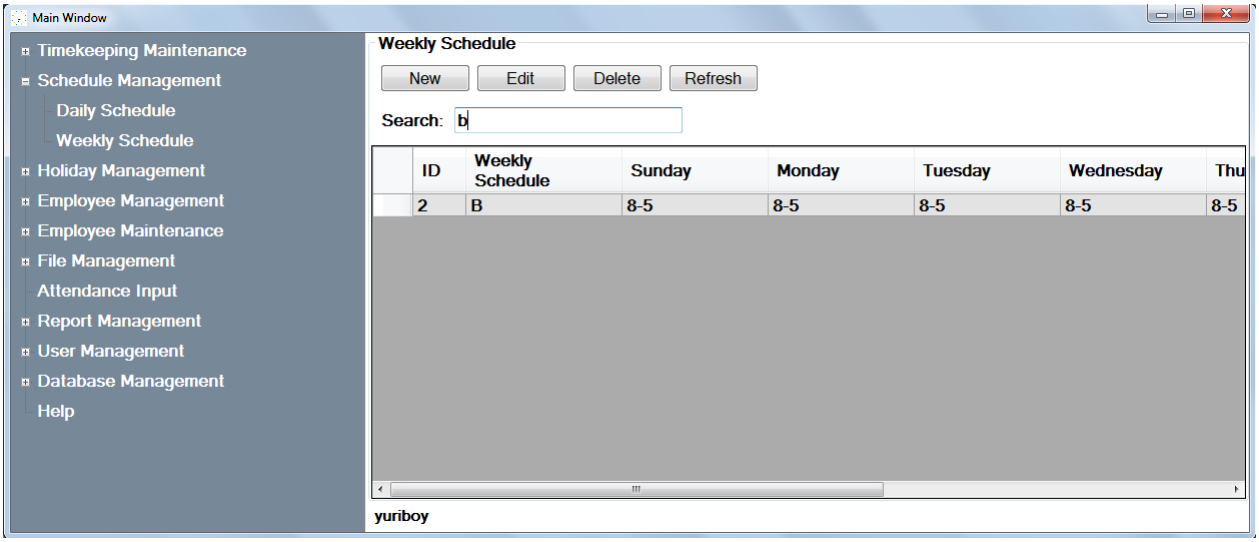


3. Schedule Management

a. Daily Schedule

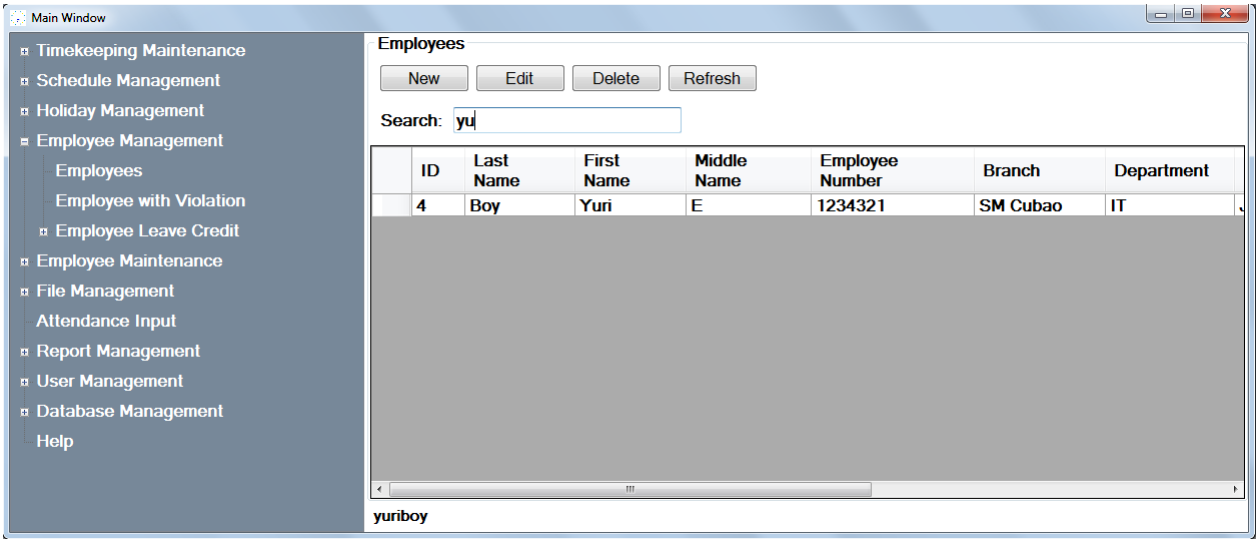


b. Weekly Schedule

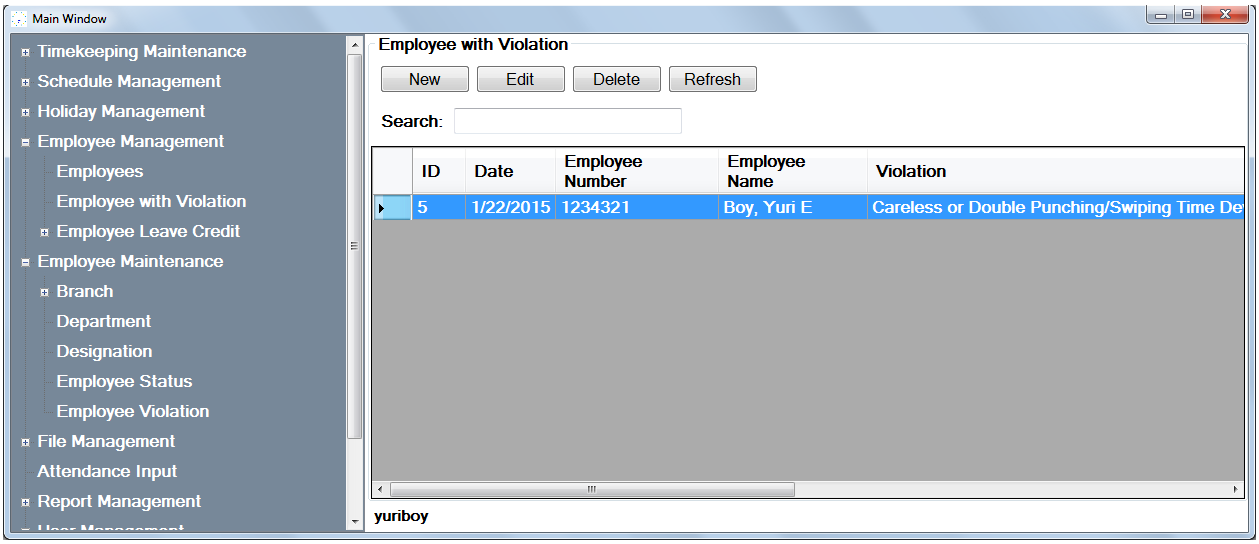


4. Employee Management

a. Employees

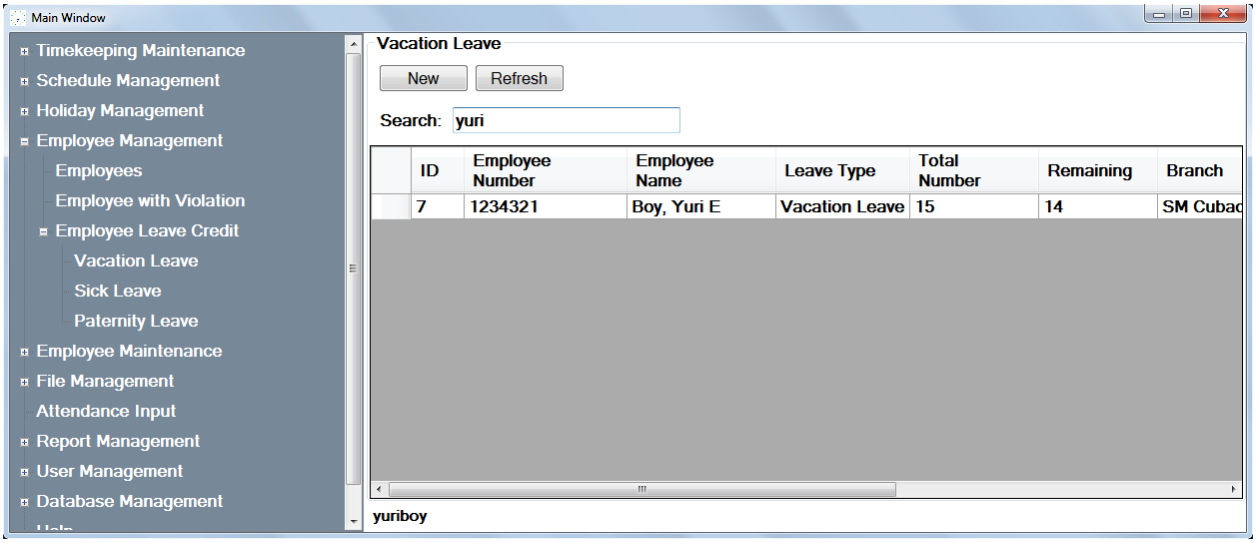


b. Employee with Violation

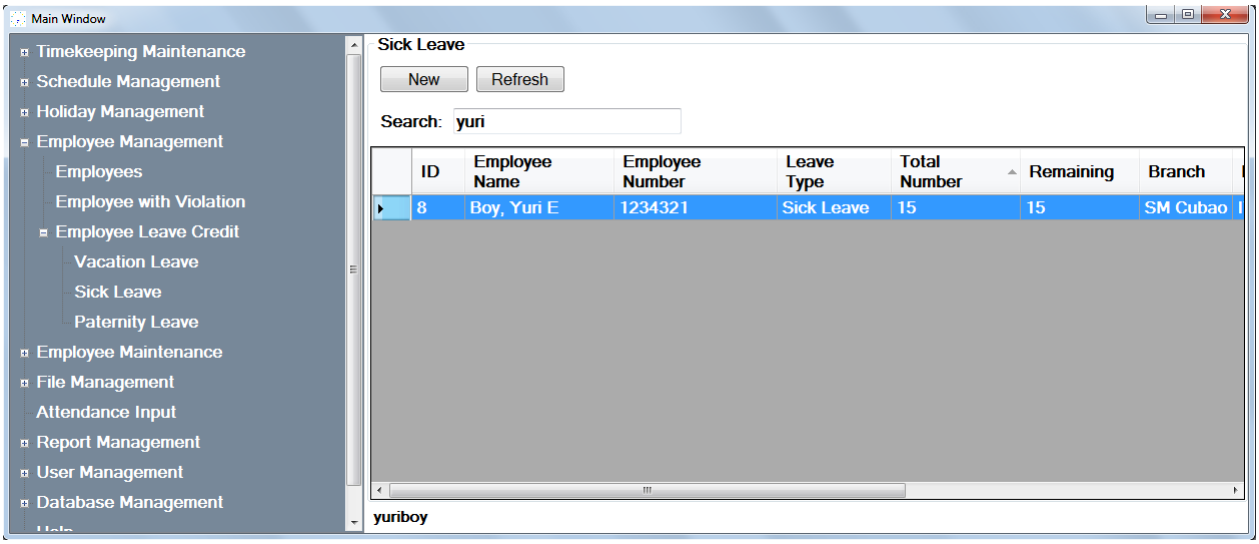


c. Employee Leave Credit

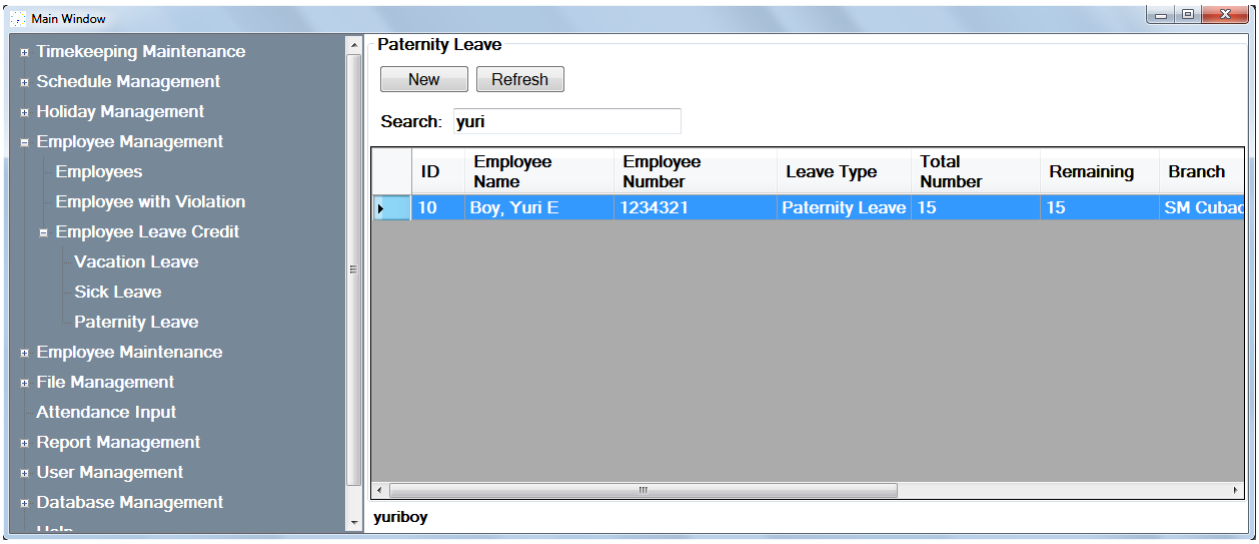
- **Vacation Leave**



- **Sick Leave**

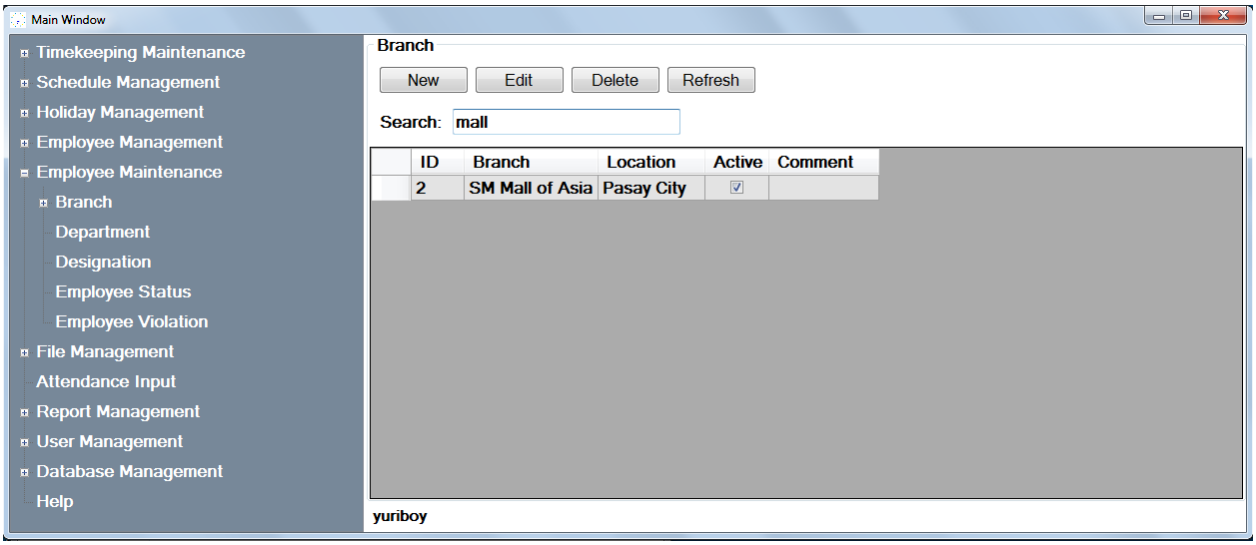


- **Paternity Leave**

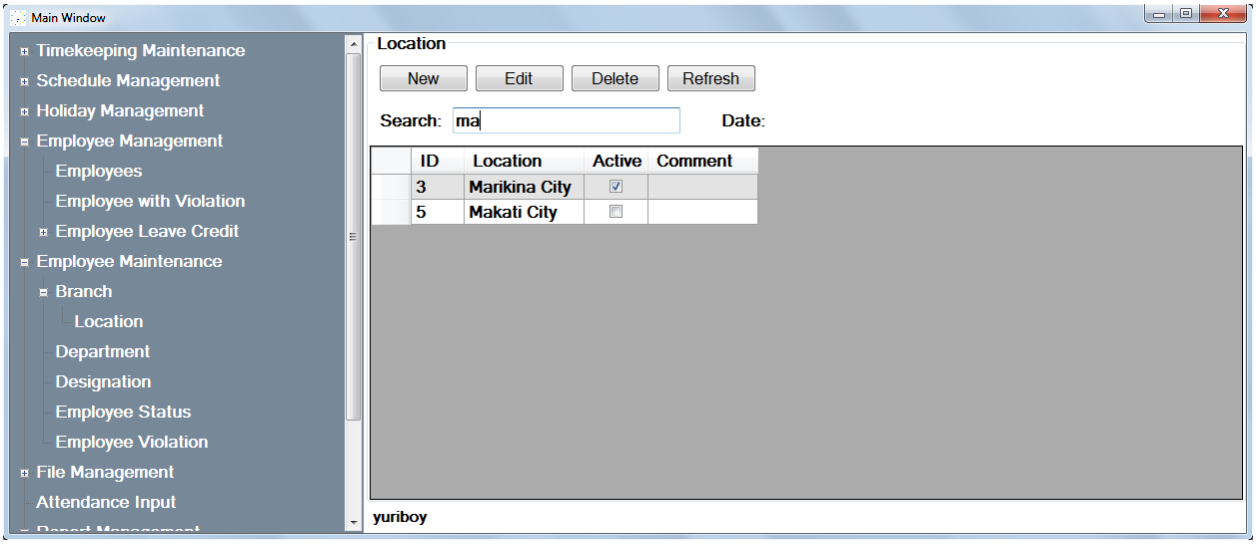


5. Employee Maintenance

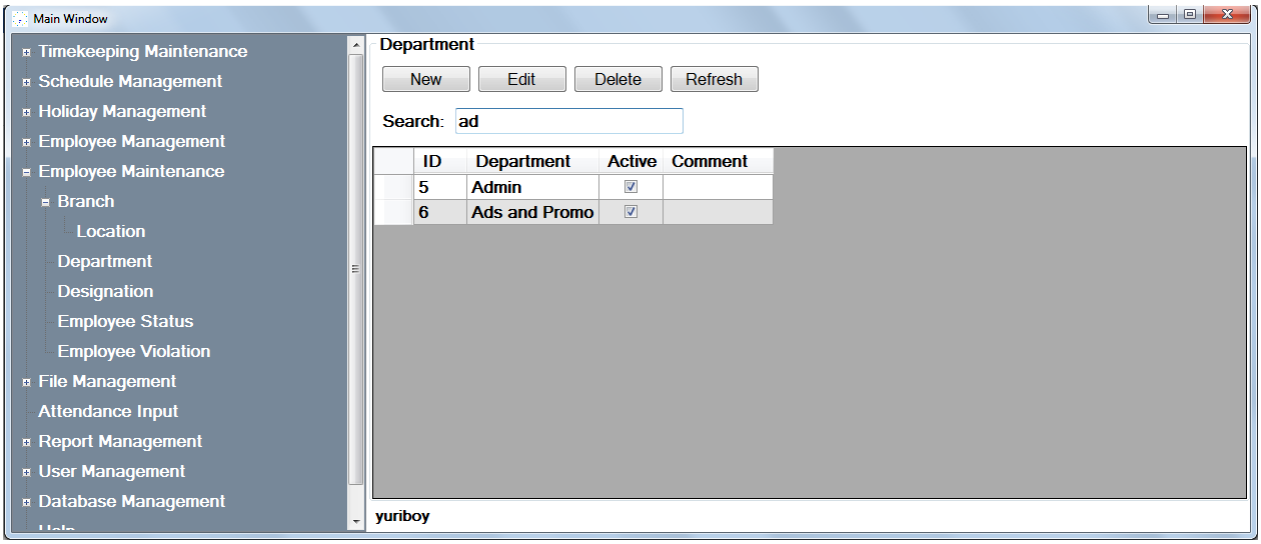
a. Branch



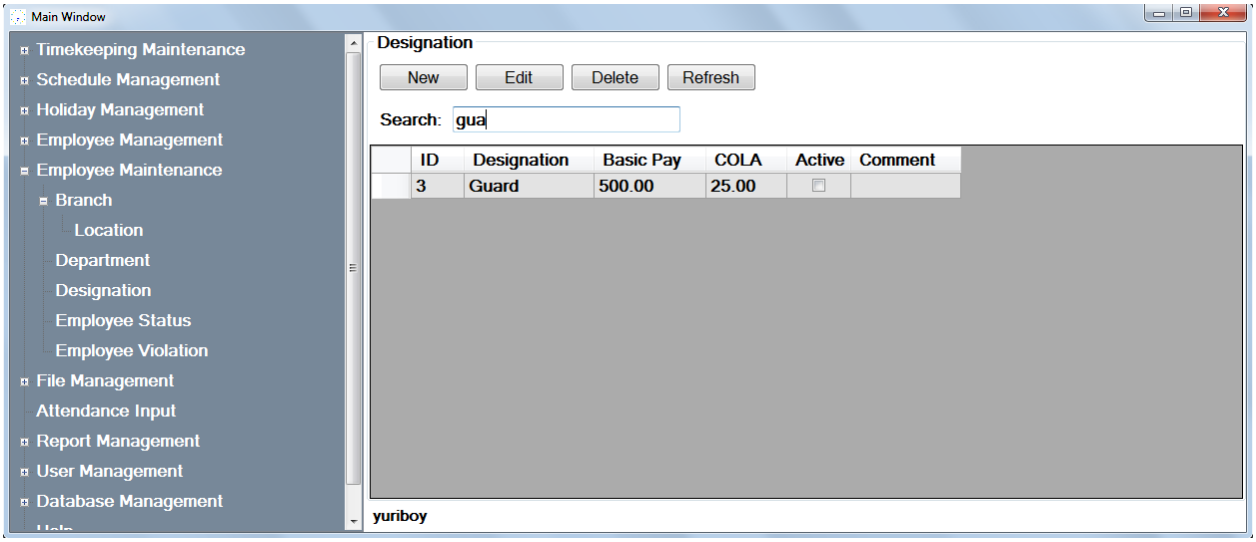
b. Location



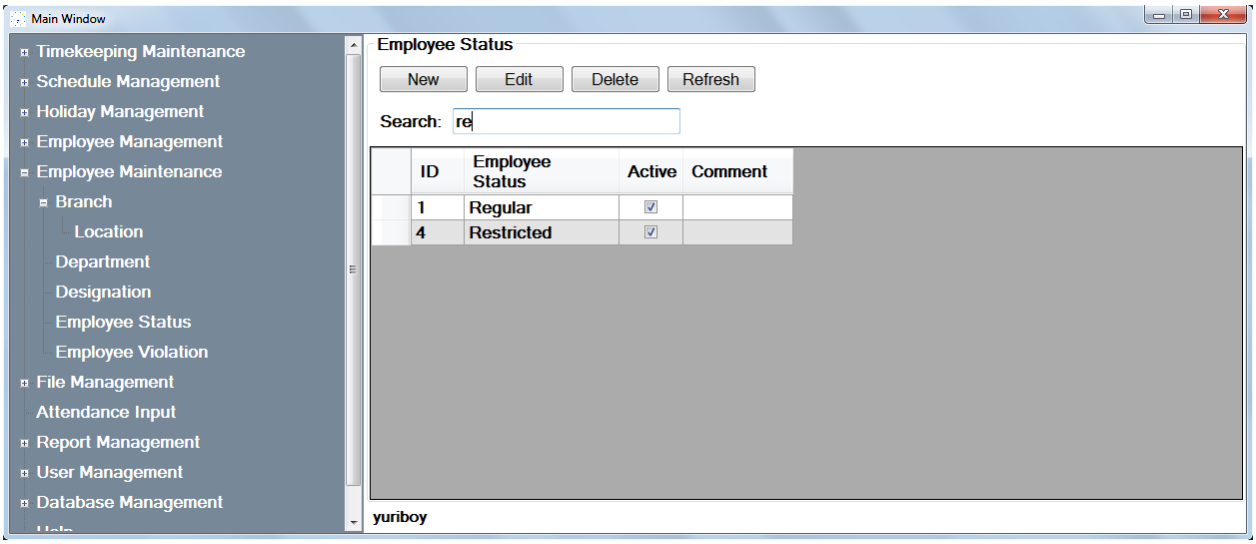
c. Department



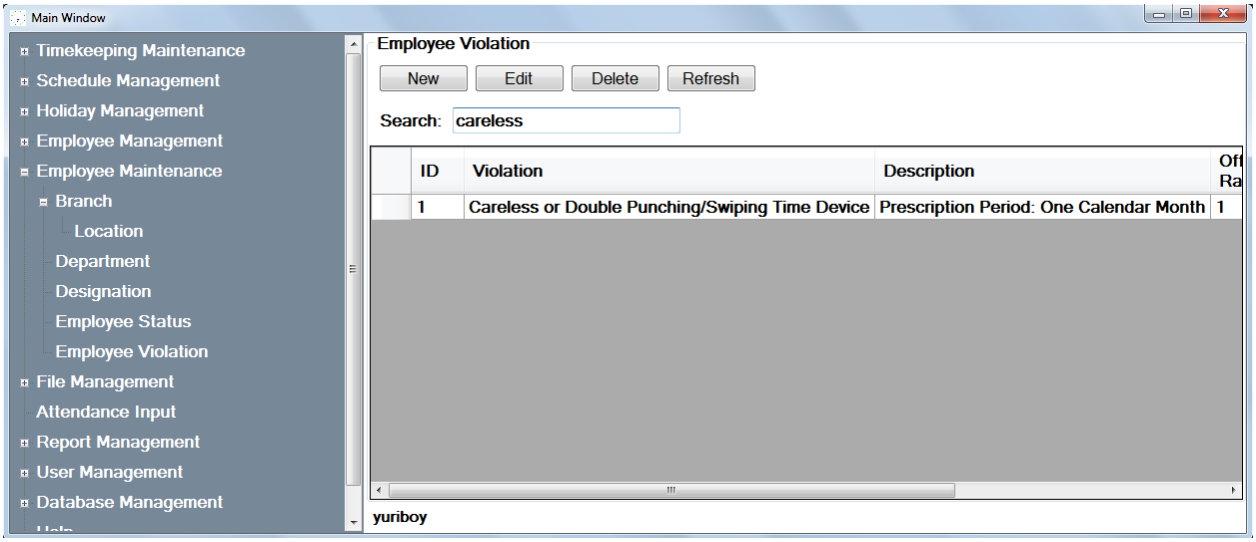
d. Designation



e. Employee Status

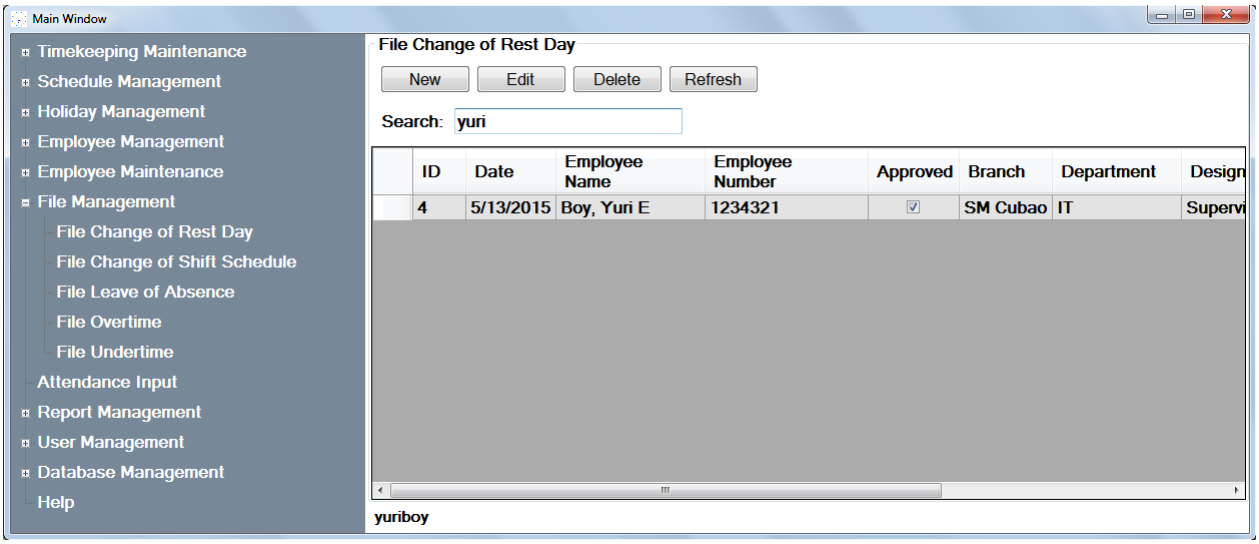


f. Employee Violation

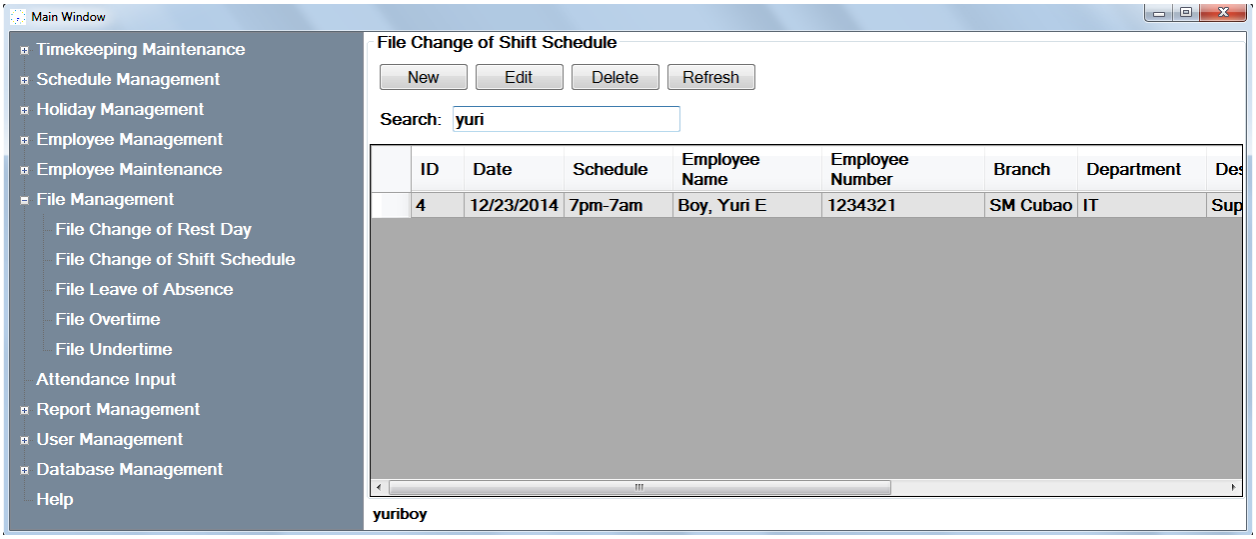


6. File Management

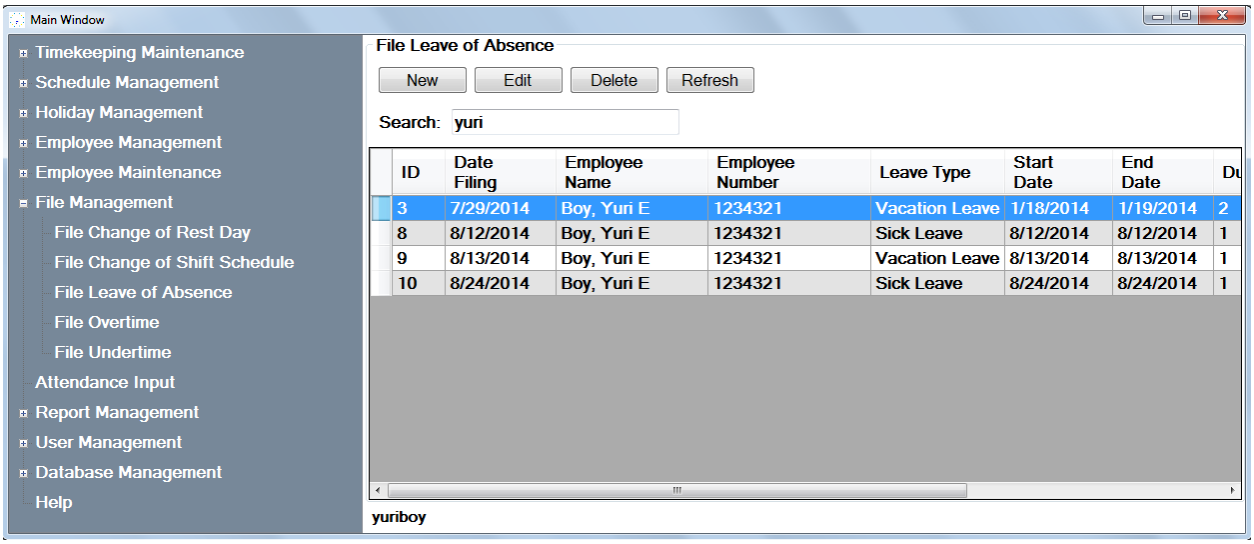
a. File of Change of Rest Day



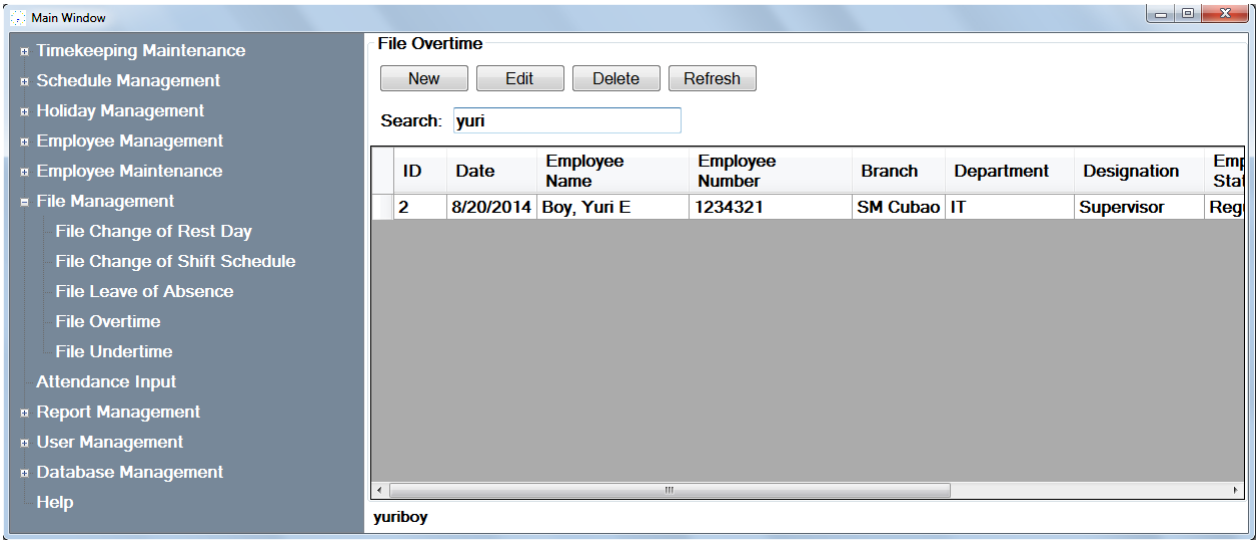
b. File of Change of Shift Schedule



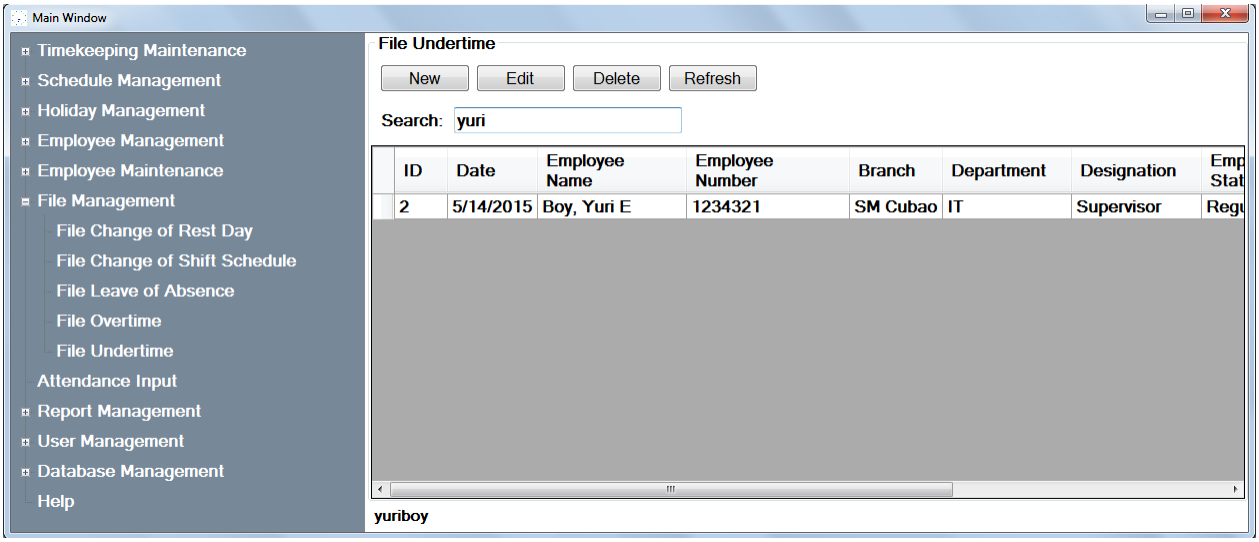
c. File of Leave of Absence



d. File of Overtime Work



e. File of Undertime



D. Add/Edit

The Add function allows the user to add new items that has an effect in the timekeeping system.

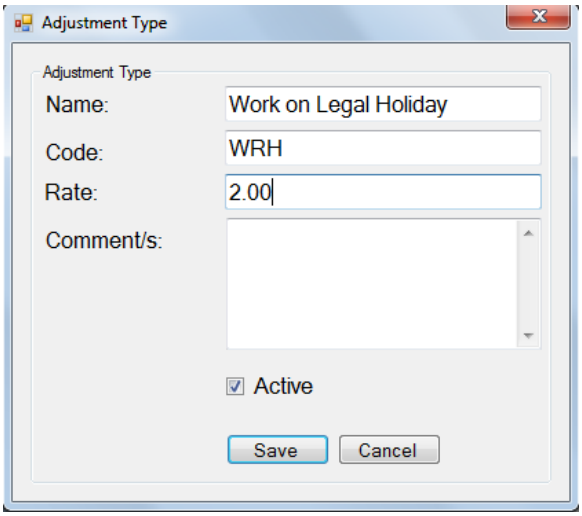
The Edit function allows the user to edit items in the database.

A new window will appear once the Add Button is invoked. Same with the Edit, the same window will appear but with already the details that you may want to edit.

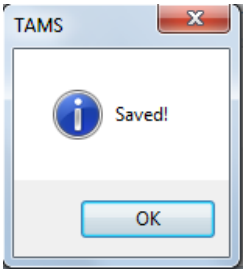
A notification will appear once the changes have been made whether it is successfully saved or not.

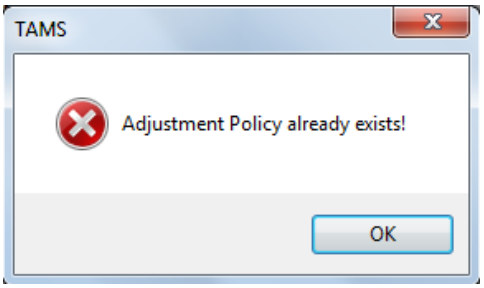
1. Timekeeping Maintenance

a. Adjustment Policy

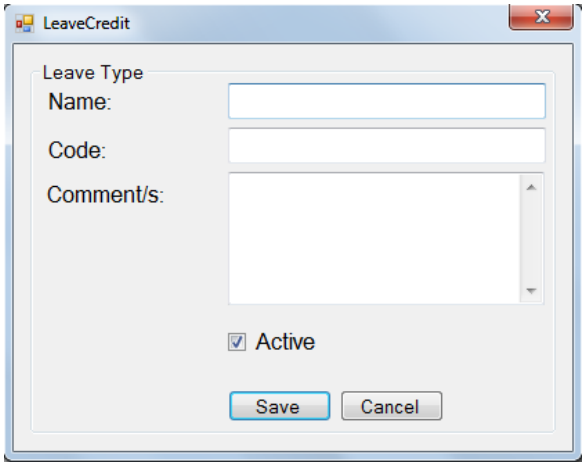


After the details has been set, a notification will appear if the policy has been saved or not.



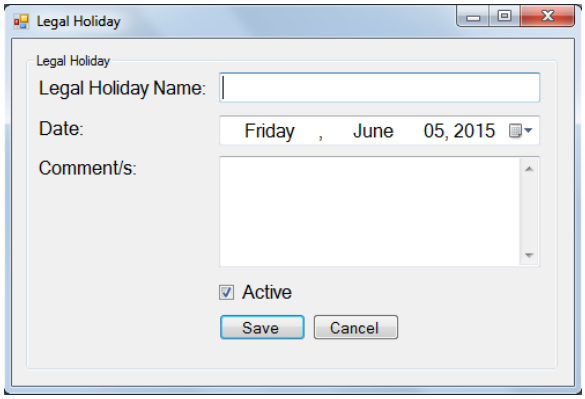


b. Leave Credit

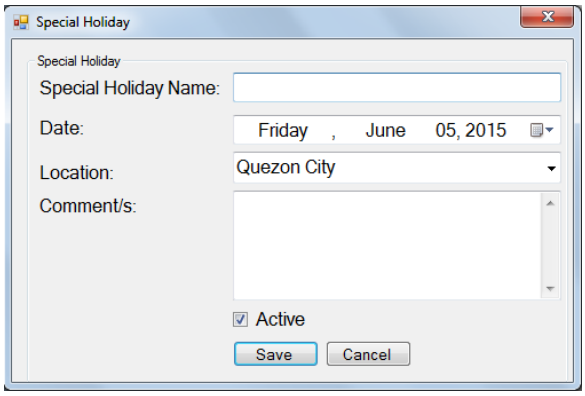


2. Holiday Management

a. Legal Holiday

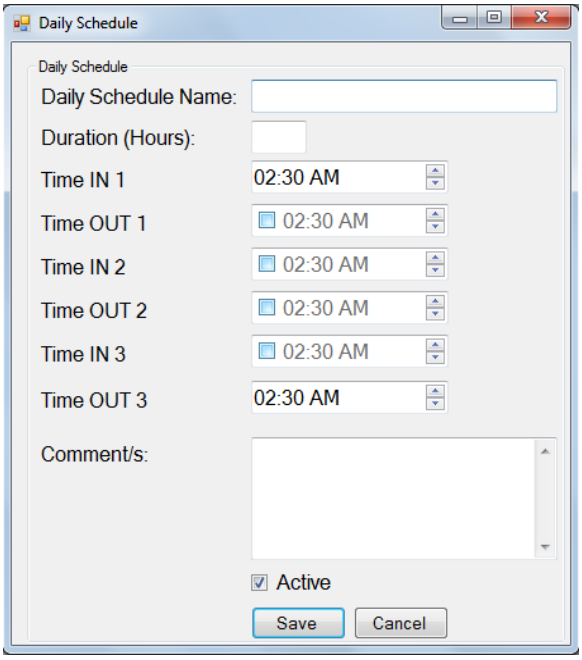


b. Special Holiday

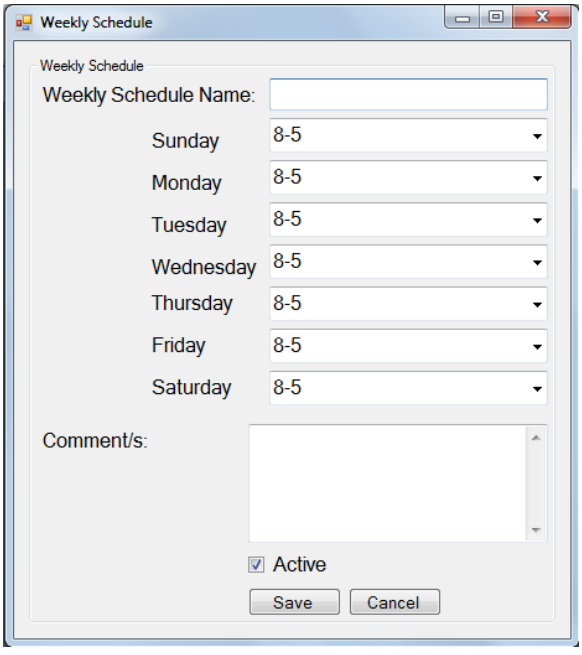


3. Schedule Management

a. Daily Schedule

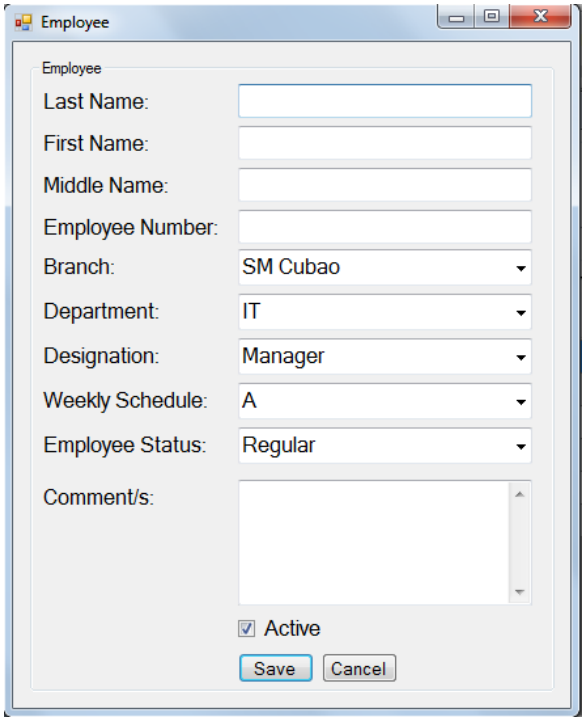


b. Weekly Schedule

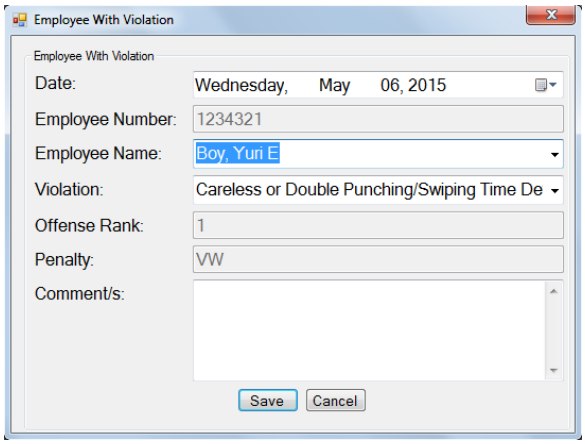


4. Employee Management

a. Employees

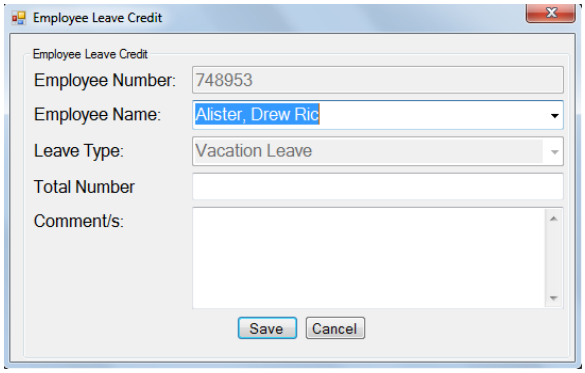


b. Employee with Violation

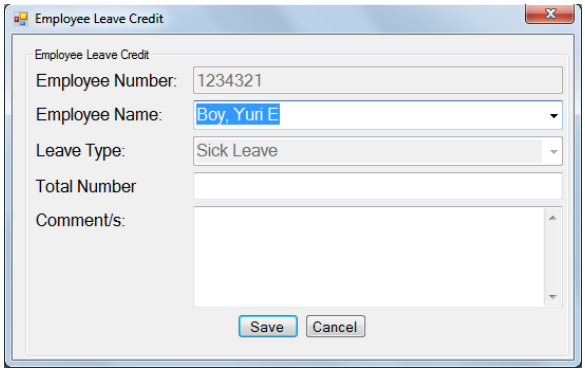


c. Employee Leave Credit

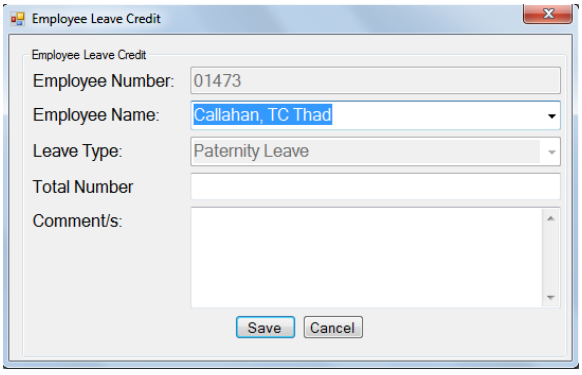
- **Vacation Leave**



- **Sick Leave**

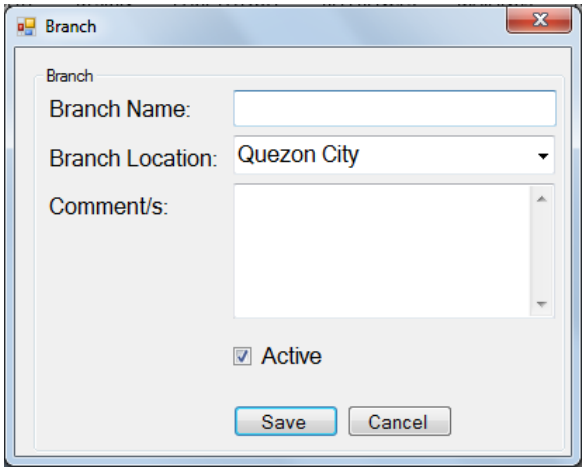


- **Paternity Leave**

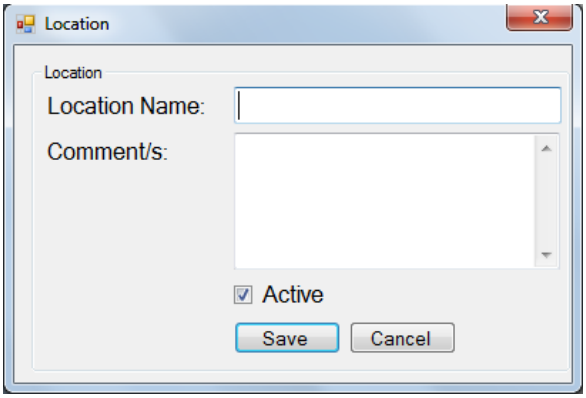


5. Employee Maintenance

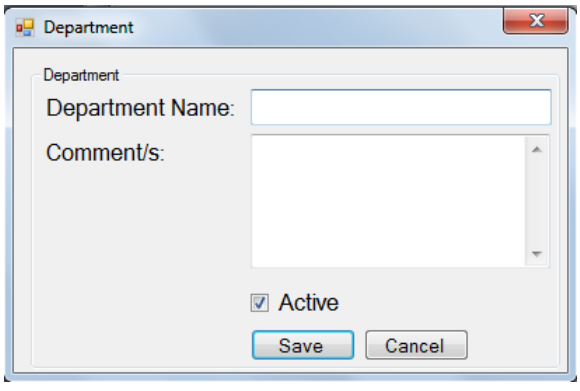
a. Branch



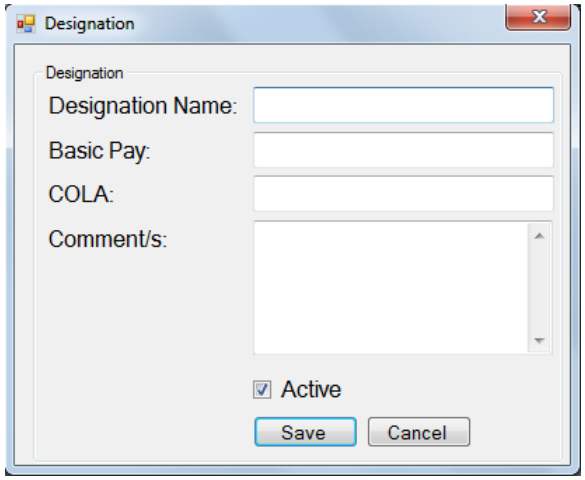
- **Location**



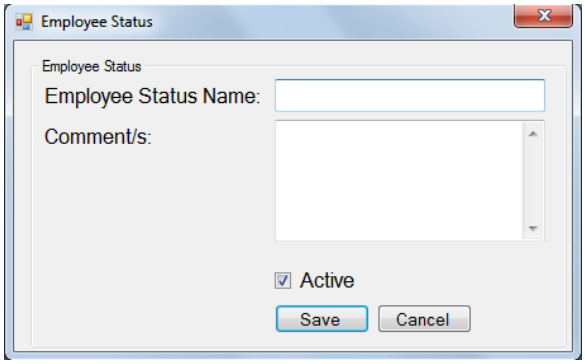
b. Department



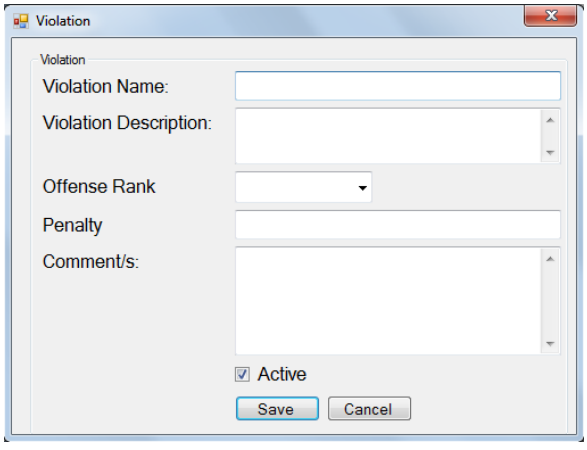
c. Designation



d. Employee Status

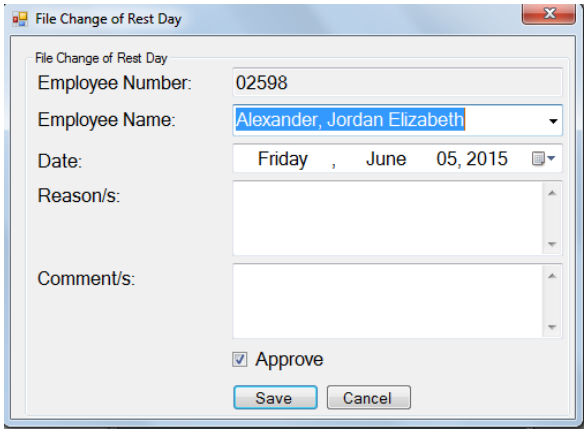


e. Employee Violation

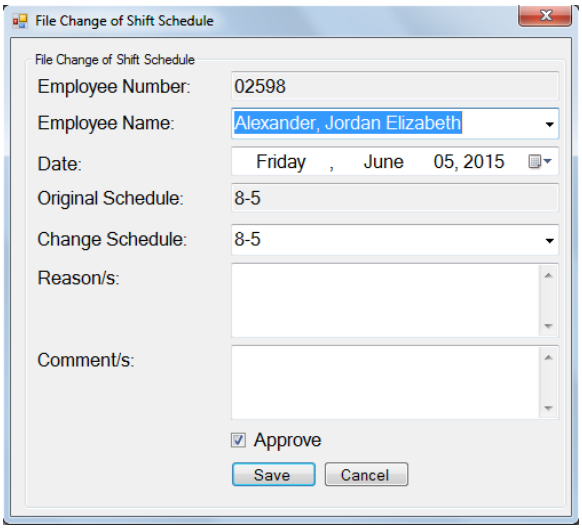


6. File Management

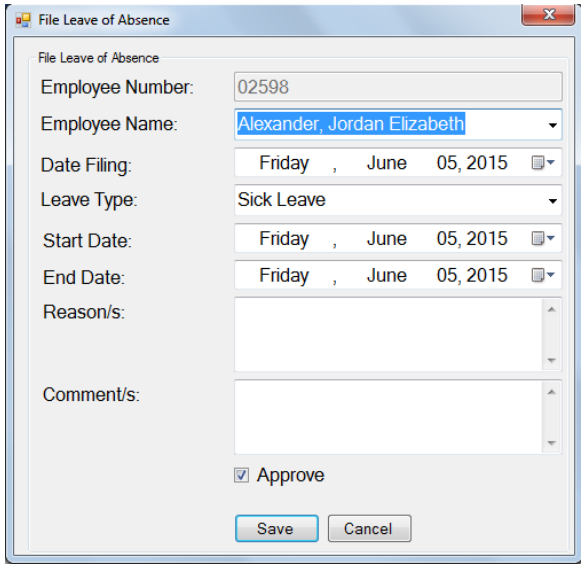
a. File Change of Rest Day



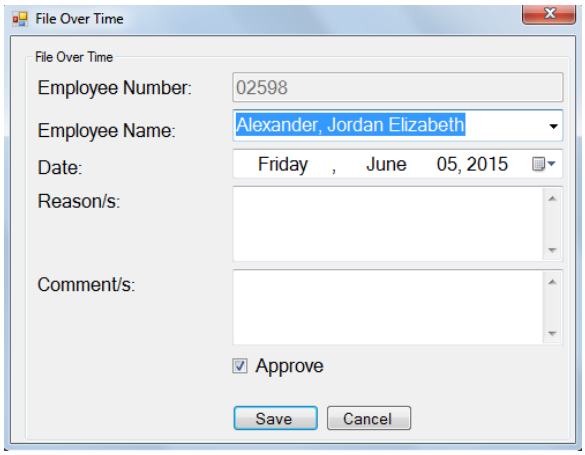
b. File Change of Shift Schedule



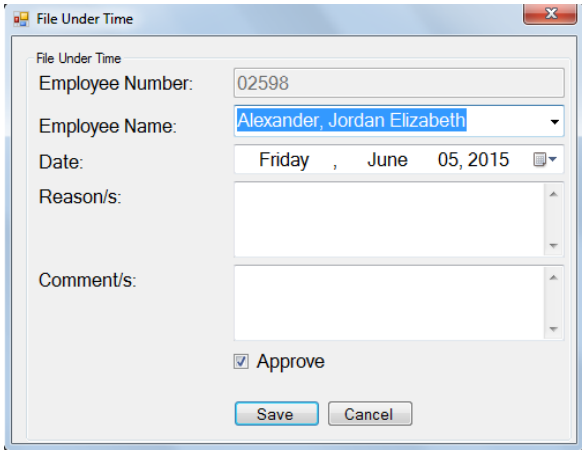
c. File Leave of Absence



d. File Overtime

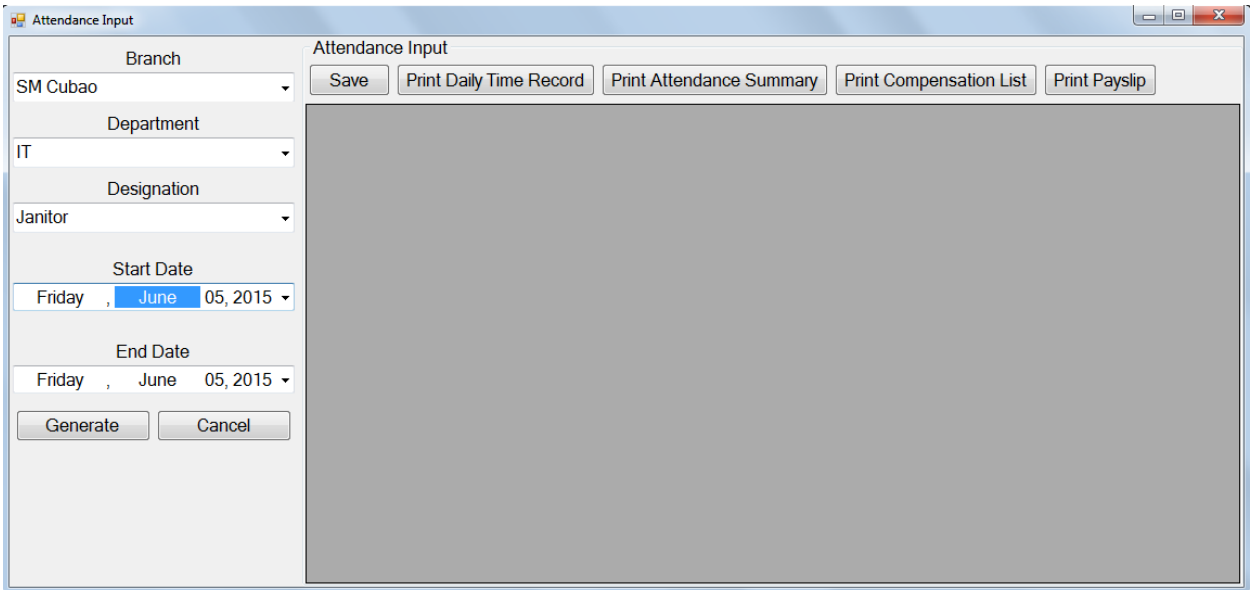


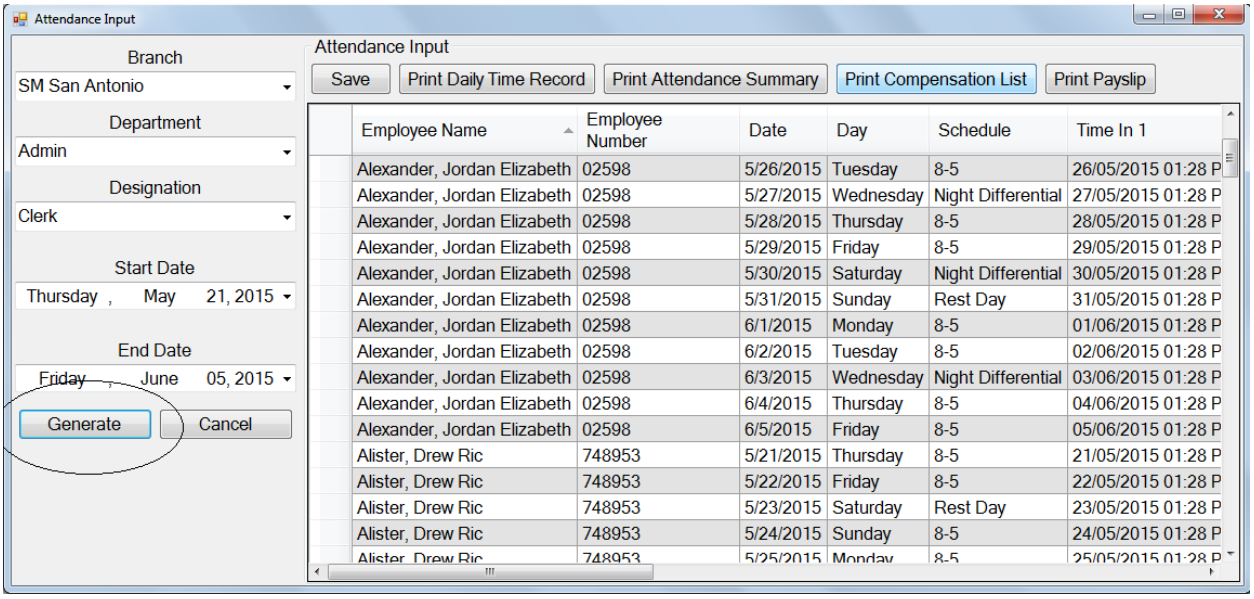
e. File Undertime



E. Attendance Input

Users can input time attendance of the employees here by setting the Branch, Department, Designation as well as the Start and End Dates of cut off.

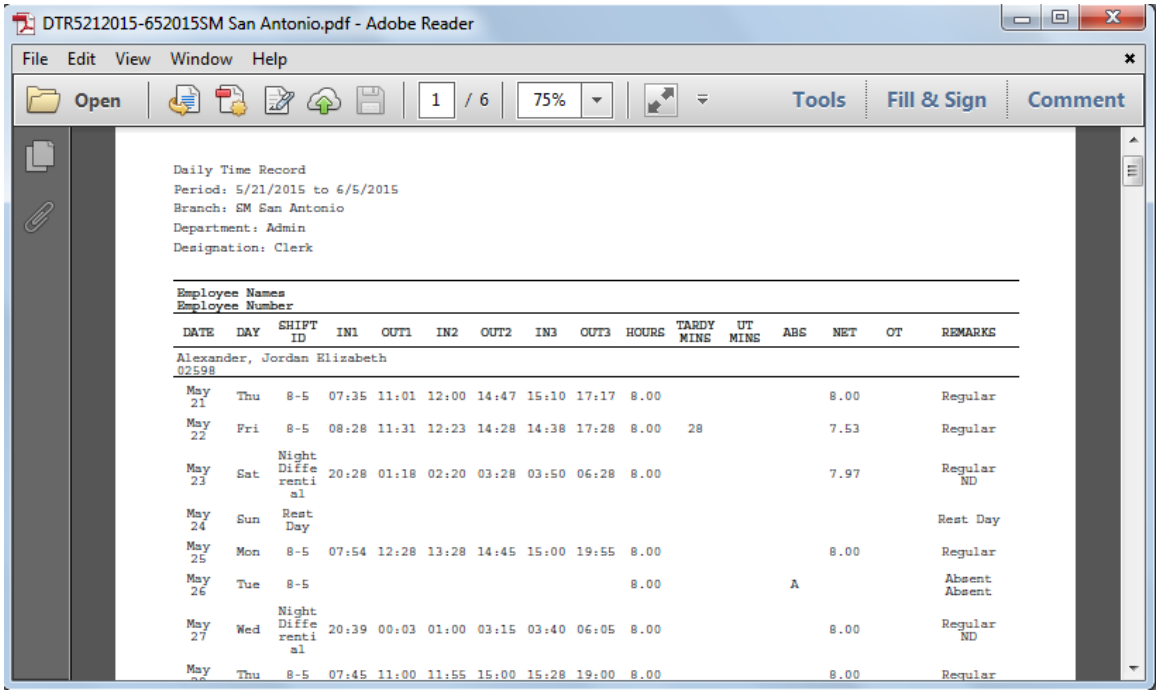




F. Results

1. DTR (Daily Time Record)

The Daily Time Record is the summary of the inputs of all the ins and outs of the attendance of an employee. It also includes the total hours worked, absent, undertime and tardy minutes and overtime of the employee given the range of date.



DTR5212015-652015SM San Antonio.pdf - Adobe Reader

File Edit View Window Help

Open | 1 / 6 | 75% | Tools | Fill & Sign | Comment

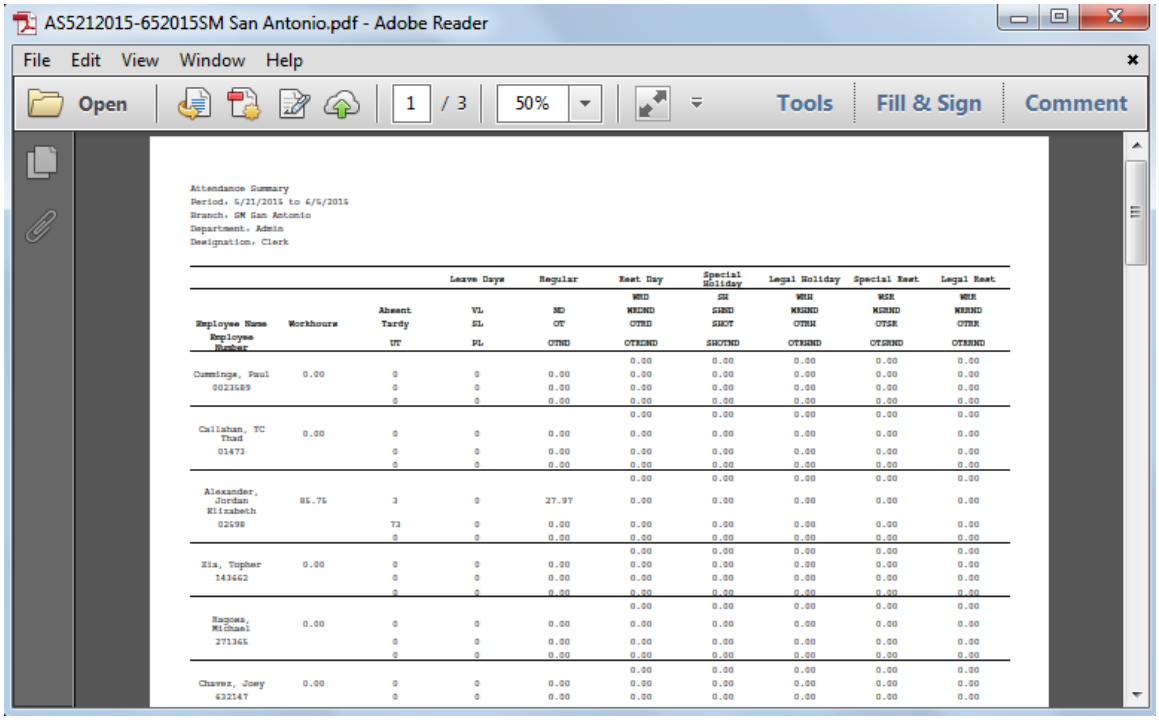
Daily Time Record
Period: 5/21/2015 to 6/5/2015
Branch: SM San Antonio
Department: Admin
Designation: Clerk

Employee Names
Employee Number

DATE	DAY	SHIFT ID	IN1	OUT1	IN2	OUT2	IN3	OUT3	HOURS	TARDY MINS	UT MINS	ABS	NET	OT	REMARKS
Alexander, Jordan Elizabeth 02598															
May 21	Thu	8-5	07:35	11:01	12:00	14:47	15:10	17:17	8.00				8.00		Regular
May 22	Fri	8-5	08:28	11:31	12:23	14:28	14:38	17:28	8.00	28			7.53		Regular
May 23	Sat	Night Differential	20:28	01:18	02:20	03:28	03:50	06:28	8.00				7.97		Regular ND
May 24	Sun	Rest Day													Rest Day
May 25	Mon	8-5	07:54	12:28	13:28	14:45	15:00	19:55	8.00				8.00		Regular
May 26	Tue	8-5							8.00			A			Absent Absent
May 27	Wed	Night Differential	20:39	00:03	01:00	03:15	03:40	06:05	8.00				8.00		Regular ND
May	Thu	8-5	07:45	11:00	11:55	15:00	15:28	19:00	8.00				8.00		Regular

2. Attendance Summary

The Attendance Summary contains the sum of all the computations of the employees hours worked.



AS5212015-652015SM San Antonio.pdf - Adobe Reader

File Edit View Window Help

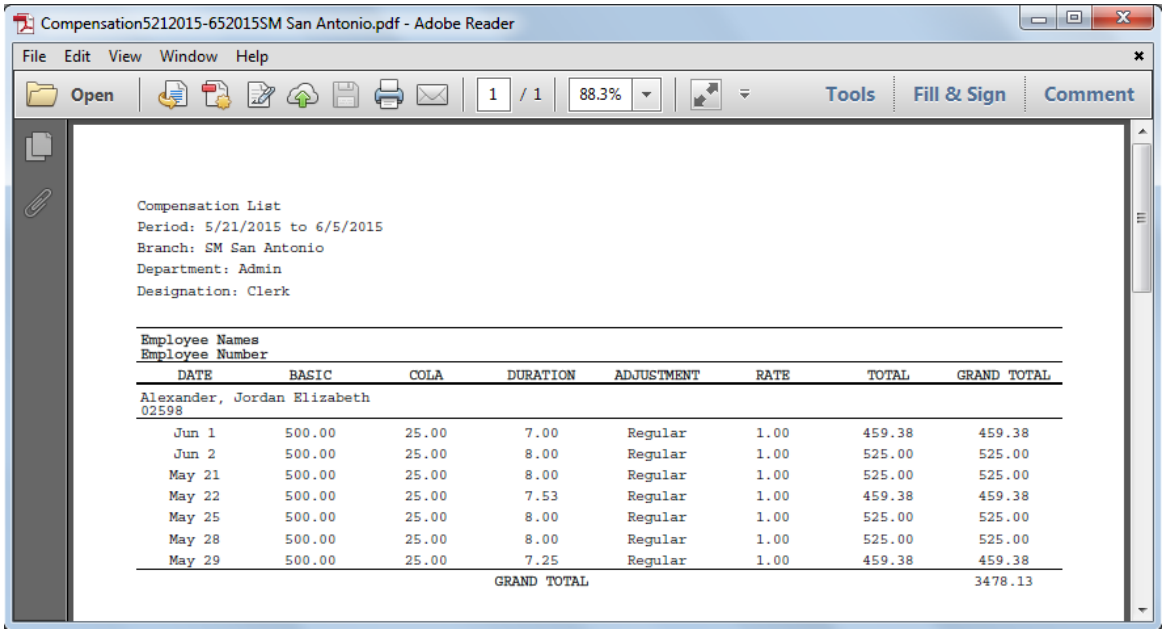
Open | 1 / 3 | 50% | Tools | Fill & Sign | Comment

Attendance Summary
Period: 5/21/2015 to 6/5/2015
Branch: SM San Antonio
Department: Admin
Designation: Clerk

Employee Name / Employee Number	Workhours	Absent / Tardy / UT	Leave Days: VL / SL / PL	Regular: ND / OT / OTND	Rest Day: WRD / RDND / OTRD / OTRDND	Special Holiday: SH / SHND / SHOT / SHOTND	Legal Holiday: WRH / RHND / OTRH / OTRHND	Special Rest: WSR / RSRND / OTSR / OTSRND	Legal Rest: WRR / RRRND / OTRR / OTRRND
					0.00	0.00	0.00	0.00	0.00
Cummings, Paul 0023589	0.00	0	0	0.00	0.00	0.00	0.00	0.00	0.00
		0	0	0.00	0.00	0.00	0.00	0.00	0.00
		0	0	0.00	0.00	0.00	0.00	0.00	0.00
					0.00	0.00	0.00	0.00	0.00
Callahan, TC Thad 01473	0.00	0	0	0.00	0.00	0.00	0.00	0.00	0.00
		0	0	0.00	0.00	0.00	0.00	0.00	0.00
		0	0	0.00	0.00	0.00	0.00	0.00	0.00
					0.00	0.00	0.00	0.00	0.00
Alexander, Jordan Elizabeth 02598	85.75	3	0	27.97	0.00	0.00	0.00	0.00	0.00
		73	0	0.00	0.00	0.00	0.00	0.00	0.00
		0	0	0.00	0.00	0.00	0.00	0.00	0.00
					0.00	0.00	0.00	0.00	0.00
Xia, Topher 143662	0.00	0	0	0.00	0.00	0.00	0.00	0.00	0.00
		0	0	0.00	0.00	0.00	0.00	0.00	0.00
		0	0	0.00	0.00	0.00	0.00	0.00	0.00
					0.00	0.00	0.00	0.00	0.00
Nagoma, Michael 271365	0.00	0	0	0.00	0.00	0.00	0.00	0.00	0.00
		0	0	0.00	0.00	0.00	0.00	0.00	0.00
		0	0	0.00	0.00	0.00	0.00	0.00	0.00
					0.00	0.00	0.00	0.00	0.00
Chavez, Joey 432147	0.00	0	0	0.00	0.00	0.00	0.00	0.00	0.00
		0	0	0.00	0.00	0.00	0.00	0.00	0.00

3. Compensation List

The Compensation List contains the rate, duration as well as its compensation. It also contains the grand total for the range of date.

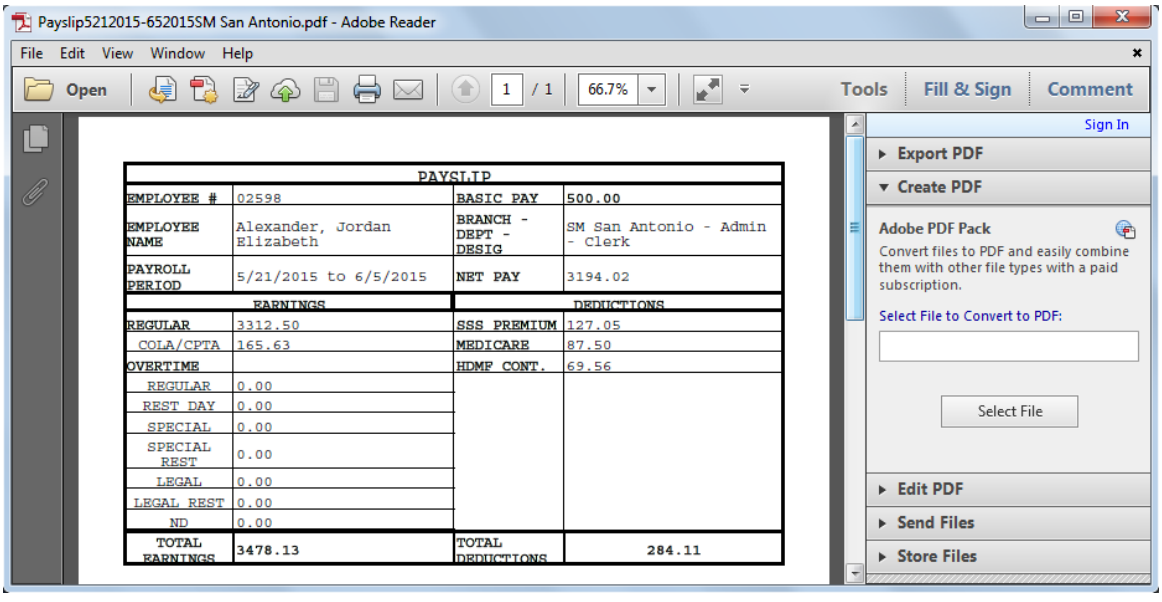


Compensation List
Period: 5/21/2015 to 6/5/2015
Branch: SM San Antonio
Department: Admin
Designation: Clerk

Employee Names
Employee Number

DATE	BASIC	COLA	DURATION	ADJUSTMENT	RATE	TOTAL	GRAND TOTAL
Alexander, Jordan Elizabeth 02598							
Jun 1	500.00	25.00	7.00	Regular	1.00	459.38	459.38
Jun 2	500.00	25.00	8.00	Regular	1.00	525.00	525.00
May 21	500.00	25.00	8.00	Regular	1.00	525.00	525.00
May 22	500.00	25.00	7.53	Regular	1.00	459.38	459.38
May 25	500.00	25.00	8.00	Regular	1.00	525.00	525.00
May 28	500.00	25.00	8.00	Regular	1.00	525.00	525.00
May 29	500.00	25.00	7.25	Regular	1.00	459.38	459.38
			GRAND TOTAL				3478.13

4. Payslip



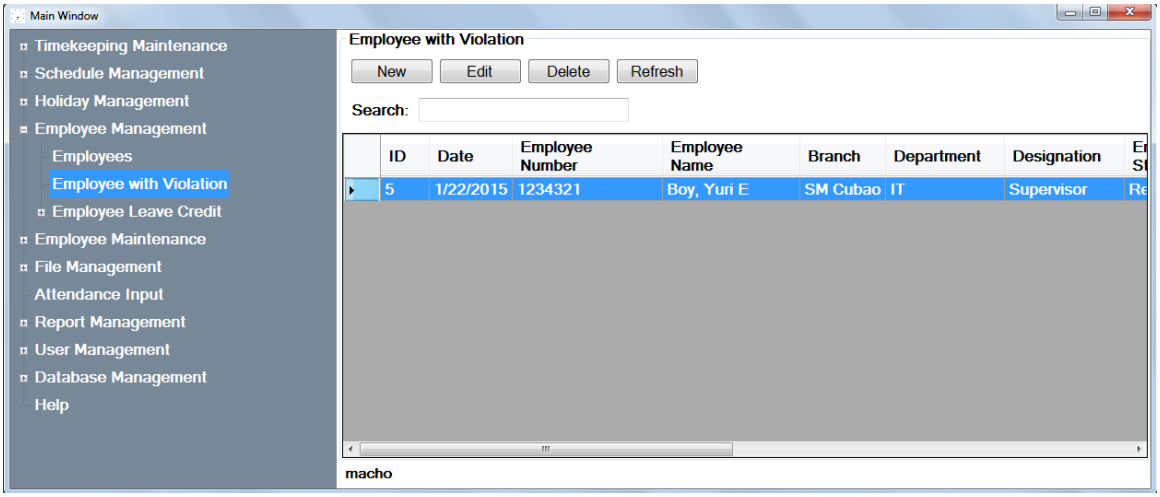
PAYSLIP

EMPLOYEE #	02598	BASIC PAY	500.00
EMPLOYEE NAME	Alexander, Jordan Elizabeth	BRANCH - DEPT - DESIG	SM San Antonio - Admin - Clerk
PAYROLL PERIOD	5/21/2015 to 6/5/2015	NET PAY	3194.02
EARNINGS		DEDUCTIONS	
REGULAR	3312.50	SSS PREMIUM	127.05
COLA/CPTA	165.63	MEDICARE	87.50
OVERTIME		HDMF CONT.	69.56
REGULAR	0.00		
REST DAY	0.00		
SPECIAL	0.00		
SPECIAL REST	0.00		
LEGAL	0.00		
LEGAL REST	0.00		
ND	0.00		
TOTAL EARNINGS	3478.13	TOTAL DEDUCTIONS	284.11

G. Reports

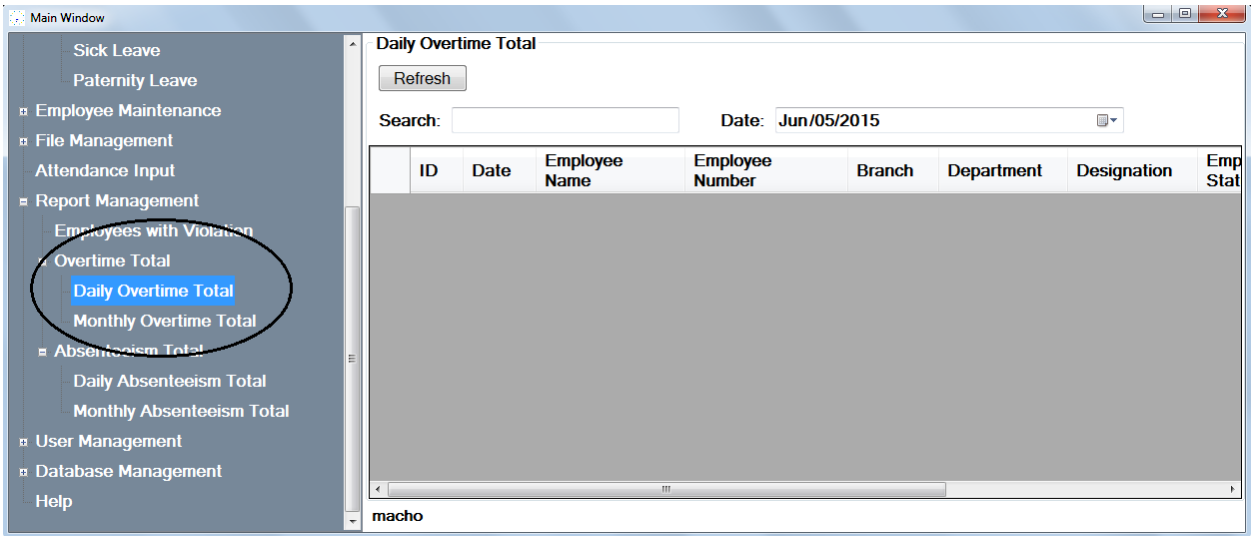
Aside from the results above, other reports include Employees with Violations, Overtime Total and Absenteeism Total.

1. **Employees with Violation**



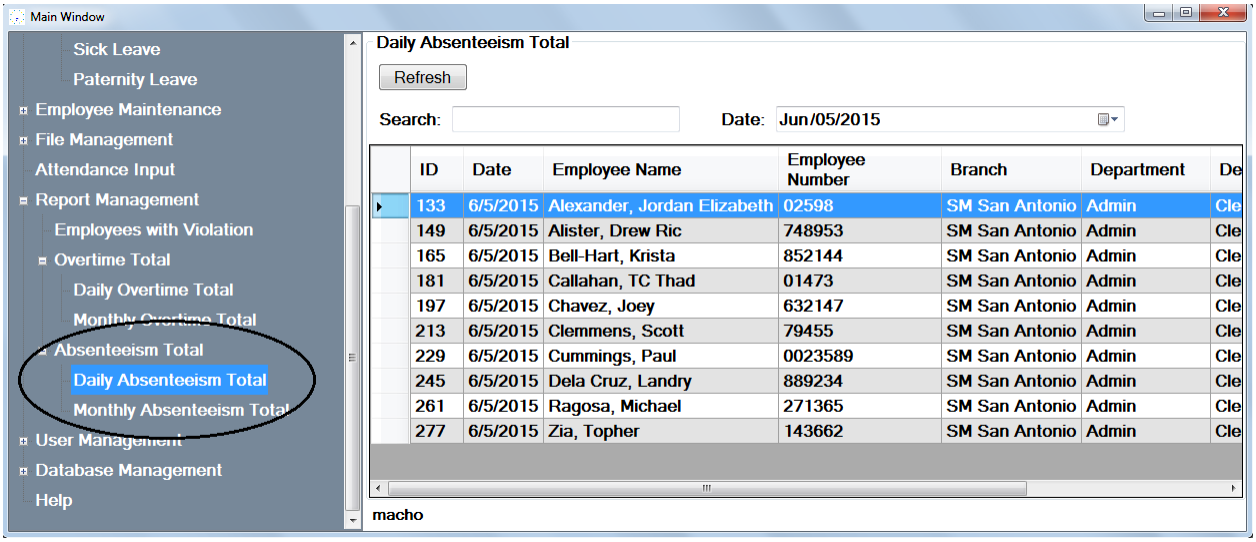
2. **Overtime Total**

Overtime total includes both daily and monthly overtime list of the employees.



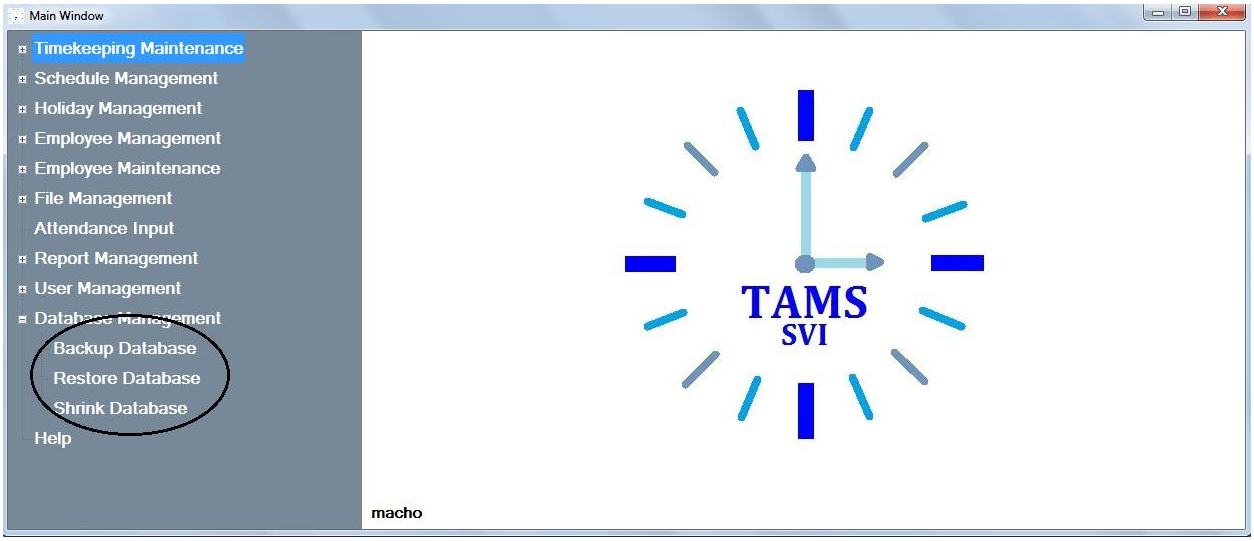
3. **Absenteeism Total**

Absenteeism total includes daily and monthly absenteeism list of the employees.



H. Database Management

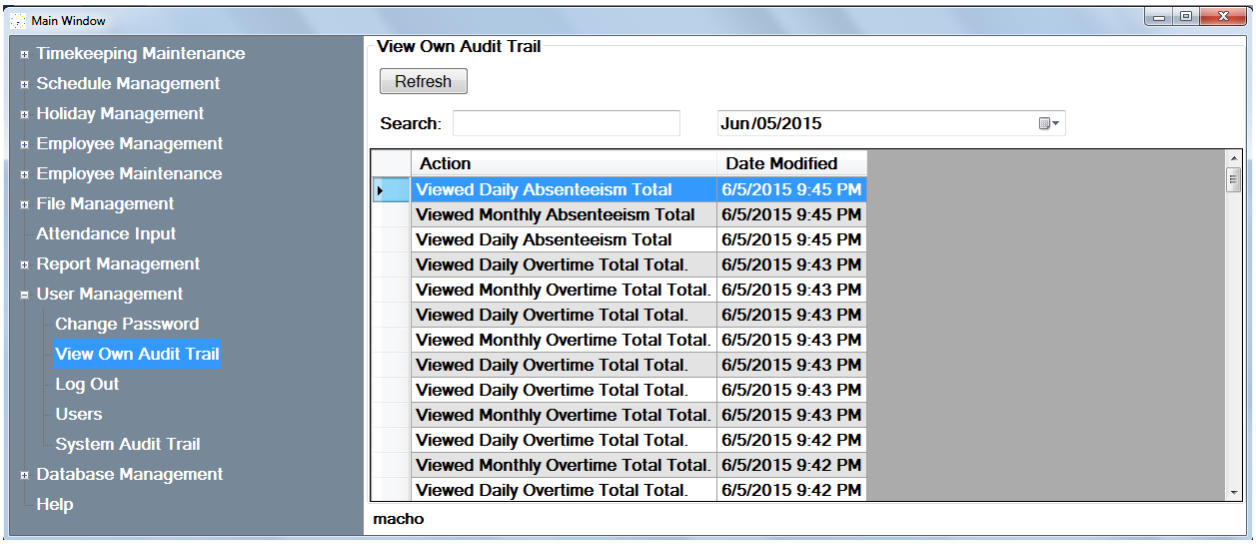
- **Back Up, Restore and Shrink Database**



I. User Management

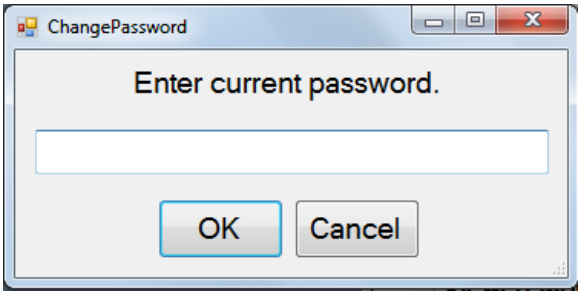
1. Audit Trail

View own audit trail lets the user see all its transactions in the system.

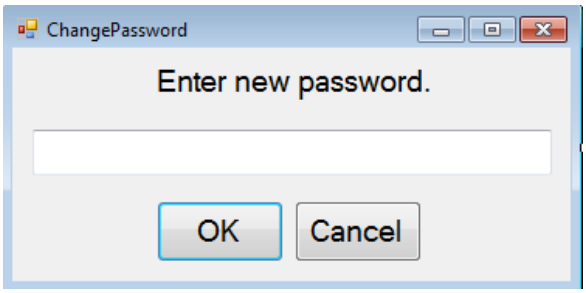


2. Change Password

Change password function lets the user change its password in the system. The system first asks the user to enter the current password.



After the confirmation of the current password, a new password is then prompted from the user followed by its confirmation.



VI. Discussions

The Timekeeping Attendance Monitoring System (TAMS) provides an automated attendance computation for the agency employees of Supervalue, Inc.

TAMS caters to two types of users: the system administrator and the timekeeper. When a user registers an account, the system administrator should approve it first before the user can log on to it. The system administrator is the one that manages all the user accounts for the system as well as the system audit trail.

The two types of users are allowed to input necessary data relevant to the assessment of the employees' attendance. This includes the employee profile (name, employee number, branch, department, designation and schedule), employee clock ins and outs including break times, filed leaves, tardiness and undertime, overtime, change in employee schedule and employee violations.

When the inputs above are entered, necessary outputs and reports are now provided. This includes the Daily Time Record, Attendance Summary, Compensation List, Payslip, Absenteeism Reports, Overtime Reports and Violation Reports

The Daily Time Record per employee shows the total hours worked (regular, overtime, rest day, holidays and night differential), total number of tardiness and undertime, total number of absences, total hours of overtime and total number of sick and vacation leaves.

The Attendance Summary shows the overall total of all the Adjustment Policies in the system for a certain period of time per employee.

The Compensation List shows the adjustment rates of the work the employee has made. It also includes the duration of the work the employee made and the amount of the work. It also includes the total amount of the employee's work during a period of time.

The Payslip includes the total earned payment of the employee as well as the total deductions (SSS premium, Philhealth, and HDMF contribution). The total net pay is shown in the payslip.

The Absenteeism Report shows the list of the employees that were absent for a certain date or month.

The Overtime Report shows the list of the employees that has overtime for a certain date or month.

The Violation Report contains the list of employees that has violation.

By integrating all these functions into a system, it provides an accessible and automated application for the Supervalue, Inc. The system also provides a graphical user interface for a user-friendly experience.

Moreover, the system significantly reduces the inaccuracy and length of computation in the attendance computation.

VII. Conclusion

The TAMS is an automated attendance computation system. The timekeeper and system administrator accounts can input employees together with their clock ins and outs for the automated computation of the employees' attendance. The system administrator may have the overall management of the system, the timekeeper can handle the automated computation just as well. TAMS has improved the accuracy and speed of the employees' attendance computation needed for the employees' compensation.

VIII. Recommendations

TAMS is a young concept which has a lot of improvement potential. A more user friendly approach in the input of attendance by adding color coded schemes may be implemented to avoid confusion in entering clock ins and outs. Also an option in the change of schedule to add a duration to improve calculations in the change of schedule of employees. Checking of combination of special items may be included for further enhancement of the computations. Default input of the employees clock ins and outs may also be implemented for a more user friendly approach.

IX. Bibliography

[1] De Jesus, G. (2008). *I Dream: A Musicale on the Life of Henry Sy, Sr.*

[2] *SM Supermaket About Us*. (n.d.). Retrieved from SM Supermarket : http://www.smsupermarket.com/smsupermarket/about_us.php

[3] *Advantages and Disadvantages of Automation*. (n.d.). Retrieved from Britannica: http://britannica.com/EBchecked/topic/44912/automation/24865/Advantages-and-disadvantages-of-automation

[4] Bradley, G. (2008). *TIMEKEEPING or The Joy of Timekeeping*.

[5] Allan, D., Ashby, N., & Hodge, C. (2010). Science of Timekeeping.

[6] Systems, H. M. (1997). Why Automate Timekeeping.

[7] Sage, H. (2013). Punching in for Improved Workforce Management: The Top Seven Benefits of an Automated Time and Attendance System.

[8] Tayfour, M. F. (2008). Developing Time and Attendance System (TAS).

[9] Norhidayah, A. (2011). *Implementing Payroll and Profits System for Emasjasa Mechanical and Electrical Engineering Consultant, Johor Bahru.*

[10] Patel, B., & Patel, B. (2012). *Payroll Management System.*

[11] Primer, T. (2005). Automated Computerized Process.

[12] Floran, R. (2008). The Computer Edge of the New Employment Opportunities.

[13] Valenzuela, J., & Barry, C. (2011). Timekeeping and Payroll System of the Blanco Family Academy.

[14] Tan, C. (2011). Design and Development of a Database for a Payroll System.

[15] Barsch, P. (2011). Brains and Databases: An Obsession with Timekeeping.

[16] Mayhew, R. (2012). Advantages of Using Computers into Business.

[17] Rao, S. (2011). Benefits of Business Using Computers.

[18] Wong, S. (2006). Employee Attendance Remuneration System (EARS).

[19] Perez, T. (2008). Daily Transaction Monitoring System.

[20] Masturah, B. (2012). ASMMA: Attendance System Using MyKad for Mobile Applications.

[21] Abdallah, K. (2007). PMS: Payroll Management System.

[22] *List of Holidays*. (n.d.). Retrieved from Department of Labor and Employment: http://www.dole.gov.ph/pages/view/9

[23] Magboo, M. (n.d.). CMSC 127 Database Systems.

[24] Woody, B. (2010). *SQL Server Experts*. Retrieved from SQL Server 2008 R2 Launches!: http://blogs.technet.com/b/sqlserverexperts/archive/2010/04/21/sql-server-2008-r2-launches.aspx

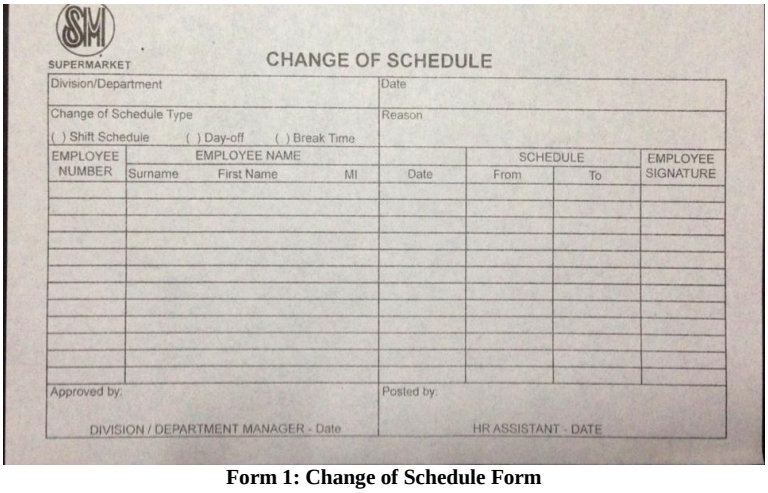
[25] Brooks, J. (2010). *SQL Server 2008 R2 Offers Enhancements, New Management Capabilities*. Retrieved from eWeek: http://www.eweek.com/c/a/IT-Infrastructure/SQL-Server-2008-R2-Offers-Enhancements-New-Management-Capabilities-518969

[26] *port of the book examples to C#*. (n.d.). Retrieved from Kuujinbo.info.: http://kuujinbo.info/iTextInAction2Ed/

[27] *iTextSharp, a .NET PDF library*. (n.d.). Retrieved from Sourceforge.net.: http://sourceforge.net/projects/itextsharp/

X. Appendix

A. Forms

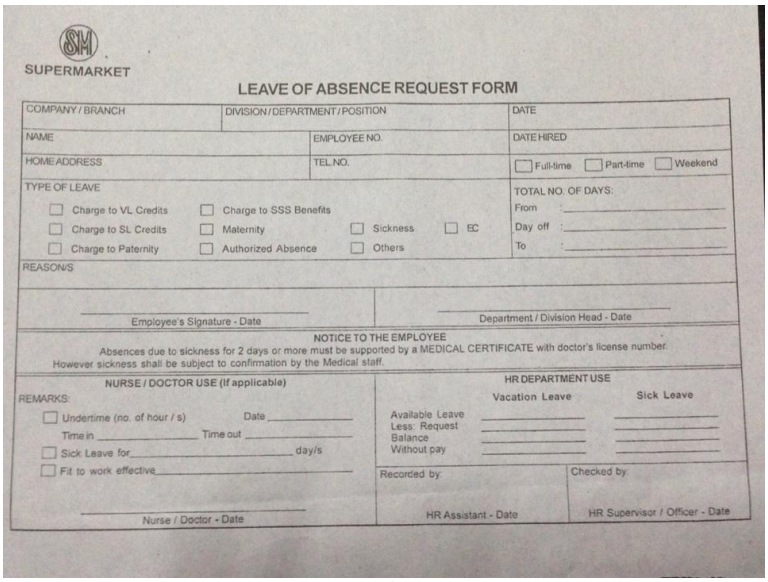


SUPERMARKET

CHANGE OF SCHEDULE

Division/Department				Date			
Change of Schedule Type ( ) Shift Schedule ( ) Day-off ( ) Break Time				Reason			
EMPLOYEE NUMBER	EMPLOYEE NAME				SCHEDULE		EMPLOYEE SIGNATURE
	Surname	First Name	MI	Date	From	To	
Approved by: DIVISION / DEPARTMENT MANAGER - Date				Posted by: HR ASSISTANT - DATE			

Form 1: Change of Schedule Form



SUPERMARKET

LEAVE OF ABSENCE REQUEST FORM

COMPANY / BRANCH	DIVISION / DEPARTMENT / POSITION	DATE
NAME	EMPLOYEE NO.	DATE HIRED
HOME ADDRESS	TEL. NO.	☐ Full-time ☐ Part-time ☐ Weekend

TYPE OF LEAVE

☐ Charge to VL Credits ☐ Charge to SSS Benefits
☐ Charge to SL Credits ☐ Maternity ☐ Sickness ☐ EC
☐ Charge to Paternity ☐ Authorized Absence ☐ Others

TOTAL NO. OF DAYS:
From :
Day off :
To :

REASON/S

Employee's Signature - Date

Department / Division Head - Date

NOTICE TO THE EMPLOYEE
Absences due to sickness for 2 days or more must be supported by a MEDICAL CERTIFICATE with doctor's license number.
However sickness shall be subject to confirmation by the Medical staff.

NURSE / DOCTOR USE (If applicable)

REMARKS:
☐ Undertime (no. of hour / s) Date
Time in Time out
☐ Sick Leave for day/s
☐ Fit to work effective

Nurse / Doctor - Date

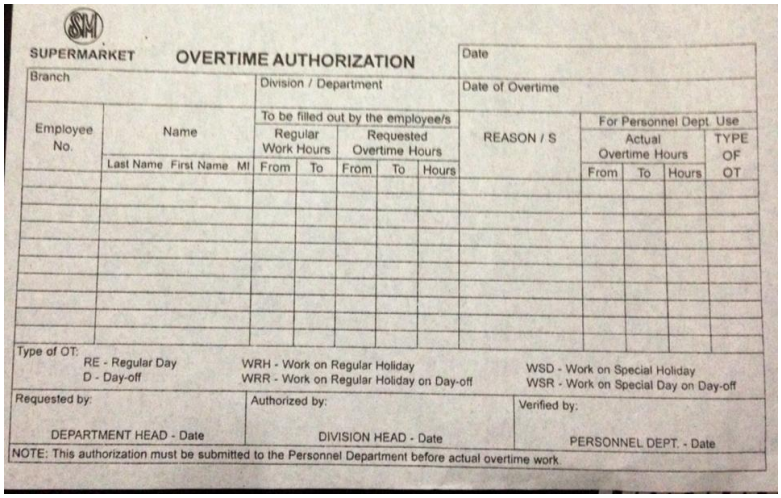
HR DEPARTMENT USE

	Vacation Leave	Sick Leave
Available Leave		
Less: Request		
Balance		
Without pay		

Recorded by:
HR Assistant - Date

Checked by:
HR Supervisor / Officer - Date

Form 2: Leave of Absence Form



SUPERMARKET **OVERTIME AUTHORIZATION**

Date

Branch | Division / Department | Date of Overtime

Employee No.	Name	To be filled out by the employee/s: Regular Work Hours		Requested Overtime Hours			REASON / S	For Personnel Dept. Use: Actual Overtime Hours			TYPE OF OT
	Last Name First Name MI	From	To	From	To	Hours		From	To	Hours	

Type of OT:
RE - Regular Day
D - Day-off
WRH - Work on Regular Holiday
WRR - Work on Regular Holiday on Day-off
WSD - Work on Special Holiday
WSR - Work on Special Day on Day-off

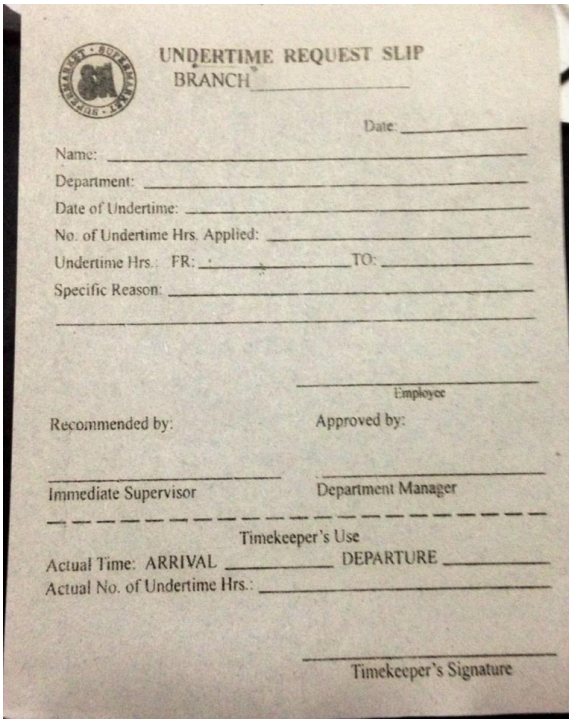
Requested by: DEPARTMENT HEAD - Date

Authorized by: DIVISION HEAD - Date

Verified by: PERSONNEL DEPT. - Date

NOTE: This authorization must be submitted to the Personnel Department before actual overtime work

Form 3: Over Time Authorization Form



UNDERTIME REQUEST SLIP

BRANCH _____

Date: _____

Name: _____

Department: _____

Date of Undertime: _____

No. of Undertime Hrs. Applied: _____

Undertime Hrs.: FR: _____ TO: _____

Specific Reason: _____

_____

_____
Employee

Recommended by:

_____
Immediate Supervisor

Approved by:

_____
Department Manager

Timekeeper's Use

Actual Time: ARRIVAL _____ DEPARTURE _____

Actual No. of Undertime Hrs.: _____

_____
Timekeeper's Signature

Form 4: Under Time Request Slip

B. Source Code

```
Public Class CalendarCell

    Inherits DataGridViewTextBoxCell

    Public Sub New()
        ' Use the short date format.
        Me.Style.Format = "dd/MM/yyyy hh:mm tt"
    End Sub

    Public Overrides Sub
InitializeEditingControl(ByVal rowIndex As
Integer, _
        ByVal initialFormattedValue As Object, _
        ByVal dataGridViewCellStyle As
DataGridViewCellStyle)

        ' Set the value of the editing control to
the current cell value.
        MyBase.InitializeEditingControl(rowIndex,
initialFormattedValue, _
            dataGridViewCellStyle)

        Dim ctl As CalendarEditingControl = _
            CType(DataGridView.EditingControl,
CalendarEditingControl)
        ctl.Value = CType(Me.Value, DateTime)

    End Sub

    Public Overrides ReadOnly Property EditType()
As Type
        Get
            ' Return the type of the editing
contol that CalendarCell uses.
            Return GetType(CalendarEditingControl)
        End Get
    End Property

    Public Overrides ReadOnly Property ValueType()
As Type
        Get
            ' Return the type of the value that
CalendarCell contains.
            Return GetType(DateTime)
        End Get
    End Property

    Public Overrides ReadOnly Property
DefaultNewRowValue() As Object
        Get
            ' Use the current date and time as the
default value.
            Return DateTime.Now
        End Get
    End Property

End Class
Imports System
Imports System.Windows.Forms

Public Class CalendarColumn

    Inherits DataGridViewColumn

    Public Sub New()
        MyBase.New(New CalendarCell())
    End Sub

    Public Overrides Property CellTemplate() As
DataGridViewCell
        Get
            Return MyBase.CellTemplate
        End Get
        Set(ByVal value As DataGridViewCell)

            ' Ensure that the cell used for the
template is a CalendarCell.
            If (value IsNot Nothing) AndAlso _
                Not
value.GetType().IsAssignableFrom(GetType(CalendarC
ell)) _
                Then
                Throw New
InvalidCastException("Must be a CalendarCell")
            End If
            MyBase.CellTemplate = value

        End Set
    End Property

End Class

Public Class Branch

    Private Duplicate As New CheckDuplicates
    Private Trail As New AuditTrail
    Private SQL As New SQLControl
    Private QString As String 'Query String
    Private AString As String 'Query String used
in Audit Trail

    Private Sub CancelBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles CancelBtn.Click

        Me.Close()

    End Sub

    Private Sub Branch_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load

        Try

            SQL.RunQuery("SELECT ID, City FROM
tLocation WHERE isActive='1'")

            BLocationCB.DataSource =
SQL.SQLDS.Tables(0)
            BLocationCB.ValueMember = "ID"
            BLocationCB.DisplayMember = "City"

            If MainWindow.HasRow = True Then

                NBranchTB.Text = MainWindow.NameTB
```

```
                ABranchChB.Checked =
MainWindow.isActive
                CommentTB.Text =
MainWindow.Comment
                BLocationCB.SelectedIndex =
BLocationCB.FindStringExact(MainWindow.locationTB)

            End If

        Catch ex As Exception

            MsgBox(ex.Message,
MsgBoxStyle.Critical)

        End Try

    End Sub

    Private Sub SaveBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles SaveBtn.Click

        Try

            If NBranchTB.Text = "" Then

                MsgBox("Please complete fields.",
MsgBoxStyle.Information)

            Else

                Dim DString As String

                DString = "SELECT * FROM tBranch
WHERE Name = '" & NBranchTB.Text & "' AND
ID_Location = '" & BLocationCB.SelectedValue & "'
AND isActive = '" & ABranchChB.Checked & "'"

                If Duplicate.IsDuplicated(DString)
= True Then

                    MsgBox("Branch already
exists!", MsgBoxStyle.Critical)

                    Exit Sub

                Else

                    If MainWindow.HasRow = False
Then

                        QString = "INSERT INTO
tBranch (Name, ID_Location, isActive, Comment)
VALUES " & _
                                            "(" & _
                                            "'" &
NBranchTB.Text & "', " & _
                                            "'" &
BLocationCB.SelectedValue & "', " & _
                                            "'" &
ABranchChB.Checked & "', " & _
                                            "'" &
CommentTB.Text & "'" & _
                                            ")"

                        AString = "Inserted " &
NBranchTB.Text & " branch."

                    ElseIf MainWindow.HasRow =
True Then

                        QString = "UPDATE tBranch
SET " & _
                                            "Name = '"
& NBranchTB.Text & "', " & _

"ID_Location = '" & BLocationCB.SelectedValue &
"', " & _
                                            "isActive
= '" & ABranchChB.Checked & "', " & _
                                            "Comment =
'" & CommentTB.Text & "' " & _
                                        "WHERE ID = '" &
MainWindow.ID & "'"

                        AString = "Updated " &
NBranchTB.Text & " branch."

                    End If

                    SQL.RunQuery(QString)

                    If
Duplicate.IsDuplicated(DString) = True Then

                        Trail.AddTrail(AString)

                        MsgBox("Saved!",
MsgBoxStyle.Information)

                        Me.Close()

                        MainWindow.LoadDataGrid()

                    Else

                        MsgBox("Branch not
saved!", MsgBoxStyle.Information)

                    End If


                End If

            End If

        Catch ex As Exception

            MsgBox(ex.Message,
MsgBoxStyle.Critical)

        End Try

    End Sub

    Public Sub SelectBranch()

        MainWindow.NewBtn.Visible = True
        MainWindow.EditBtn.Visible = True
        MainWindow.RefreshBtn.Visible = True
        MainWindow.DateTimePicker1.Visible = False
        MainWindow.Label2.Visible = False
        MainWindow.DeleteBtn.Visible = True

        MainWindow.GroupBox.Visible = True
        MainWindow.GroupBox.Text = "Branch"
```

```
        MainWindow.QString = "EXEC selectBranch"

        MainWindow.LoadDataGrid()


MainWindow.DataGrid.Columns("ID").DisplayIndex = 0

MainWindow.DataGrid.Columns("Branch").DisplayIndex
= 1

MainWindow.DataGrid.Columns("Location").DisplayInd
ex = 2

MainWindow.DataGrid.Columns("Active").DisplayIndex
= 3

MainWindow.DataGrid.Columns("Comment").DisplayInde
x = 4

        Trail.AddTrail("Viewed Branch.")

    End Sub

    Public Sub DeleteBranch()

        Dim result1 As DialogResult =
MessageBox.Show("Are you sure to delete " &
MainWindow.NameTB & "?", _
                "Delete Branch", _
                MessageBoxButtons.YesNo)

        If result1 =
Windows.Forms.DialogResult.Yes Then

            MainWindow.QString = "DELETE from
tBranch WHERE ID = '" & MainWindow.ID & "'"

            Trail.AddTrail("Deleted " &
MainWindow.NameTB & " from Branch.")

            MainWindow.LoadDataGrid()

            SelectBranch()

        End If

    End Sub

End Class

Public Class Violation

    Private Duplicate As New CheckDuplicates
    Private Trail As New AuditTrail
    Private SQL As New SQLControl
    Private QString As String 'Query String for
insert or update an employee violation.
    Private AString As String 'Query String for
Audit Trail

    Private Sub CancelBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles CancelBtn.Click

        Me.Close()

    End Sub

    Private Sub SaveBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles SaveBtn.Click

        Try

            If VNameTB.Text = "" Or VDescTB.Text =
"" Or VPenaltyTB.Text = "" Then

                MsgBox("Please complete fields.",
MsgBoxStyle.Information)

            Else

                Dim DString As String

                DString = "SELECT * FROM
tViolation WHERE Name = '" & VNameTB.Text & "' AND
Description = '" & VDescTB.Text & "' AND
OffenseRank = '" & RankCB.SelectedValue & "' AND
Penalty = '" & VPenaltyTB.Text & "' AND isActive =
'" & ActiveChB.Checked & "'"

                If Duplicate.IsDuplicated(DString)
Then

                    MsgBox("Violation already
exists!", MsgBoxStyle.Critical)

                Else

                    If MainWindow.HasRow = False
Then

                        QString = "INSERT INTO
tViolation (Name, Description, OffenseRank,
Penalty, isActive, Comment) VALUES " & _
                                            "(" & _
                                            "'" &
VNameTB.Text & "', " & _
                                            "'" &
VDescTB.Text & "', " & _
                                            "'" &
RankCB.SelectedItem & "', " & _
                                            "'" &
VPenaltyTB.Text & "', " & _
                                            "'" &
ActiveChB.Checked & "', " & _
                                            "'" &
CommentTB.Text & "'" & _
                                            ")"

                        AString = "Inserted " &
VNameTB.Text & " violation."

                    ElseIf MainWindow.HasRow =
True Then

                        QString = "UPDATE
tViolation SET " & _
                                            "Name = '"
& VNameTB.Text & "', " & _

"Description = '" & VDescTB.Text & "', " & _

"OffenseRank = '" & RankCB.SelectedItem & "', " &
_
                                            "Penalty =
'" & VPenaltyTB.Text & "', " & _
```

```
                                        "isActive
= '" & ActiveChB.Checked & "', " & _
                                        "Comment =
'" & CommentTB.Text & "' " & _
                                    "WHERE ID = '" &
MainWindow.ID & "'"

                        AString = "Updated " &
VNameTB.Text & " violation."

                    End If

                    SQL.RunQuery(QString)

                    If
Duplicate.IsDuplicated(DString) Then

                        Trail.AddTrail(AString)

                        MsgBox("Saved!",
MsgBoxStyle.Information)

                        Me.Close()

                        MainWindow.LoadDataGrid()

                    Else

                        MsgBox("Violation not
saved!", MsgBoxStyle.Information)

                    End If

                End If

            End If

        Catch ex As Exception

            MsgBox(ex.Message,
MsgBoxStyle.Critical)

        End Try

    End Sub

    Private Sub Violation_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load

        If MainWindow.HasRow = True Then

            RankCB.SelectedIndex =
RankCB.FindStringExact(MainWindow.rankTB)

            VNameTB.Text = MainWindow.NameTB
            VDescTB.Text = MainWindow.descTB
            VPenaltyTB.Text = MainWindow.penaltyTB
            ActiveChB.Checked =
MainWindow.isActive
            CommentTB.Text = MainWindow.Comment

        End If

    End Sub

    Public Sub SelectViolation()

        MainWindow.NewBtn.Visible = True
        MainWindow.EditBtn.Visible = True
        MainWindow.RefreshBtn.Visible = True
        MainWindow.DeleteBtn.Visible = True
        MainWindow.DateTimePicker1.Visible = False
        MainWindow.DeleteBtn.Visible = True
        MainWindow.Label2.Visible = False

        MainWindow.GroupBox.Visible = True
        MainWindow.GroupBox.Text = "Employee
Violation"

        MainWindow.QString = "EXEC
selectEmployeeViolation"

        MainWindow.LoadDataGrid()


MainWindow.DataGrid.Columns("ID").DisplayIndex = 0

MainWindow.DataGrid.Columns("Violation").DisplayIn
dex = 1

MainWindow.DataGrid.Columns("Description").Display
Index = 2
        MainWindow.DataGrid.Columns("Offense
Rank").DisplayIndex = 3

MainWindow.DataGrid.Columns("Penalty").DisplayInde
x = 4

MainWindow.DataGrid.Columns("Active").DisplayIndex
= 5

MainWindow.DataGrid.Columns("Comment").DisplayInde
x = 6

        Trail.AddTrail("Viewed Employee
Violation.")

    End Sub

    Public Sub DeleteViolation()

        Dim result1 As DialogResult =
MessageBox.Show("Are you sure to delete " &
MainWindow.NameTB & "?", _
                "Delete Employee Violation", _
                MessageBoxButtons.YesNo)

        If result1 =
Windows.Forms.DialogResult.Yes Then

            MainWindow.QString = "DELETE from
tEmployeeViolation WHERE ID = '" & MainWindow.ID &
"'"

            MainWindow.LoadDataGrid()

            Trail.AddTrail("Deleted " &
MainWindow.NameTB & " from Employee Violation.")

            SelectViolation()

        End If

    End Sub

End Class
Public Class Location1
```

```
    Private Duplicate As New CheckDuplicates
    Private Trail As New AuditTrail
    Private SQL As New SQLControl
    Private QString As String 'Query String for
insert or update a location.
    Private AString As String 'Query String for
Audit Trail

    Private Sub CancelBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles CancelBtn.Click

        Me.Close()

    End Sub

    Private Sub SaveBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles SaveBtn.Click

        Try

            If NLocationTB.Text = "" Then

                MsgBox("Enter location name.",
MsgBoxStyle.Information)

            Else

                Dim DString As String

                DString = "SELECT * FROM tLocation
WHERE City = '" & NLocationTB.Text & "' AND
isActive = '" & ALocationChB.Checked & "'"

                If Duplicate.IsDuplicated(DString)
Then

                    MsgBox("Location already
exists!", MsgBoxStyle.Critical)

                Else

                    If MainWindow.HasRow = False
Then

                        QString = "INSERT INTO
tLocation (City, isActive, Comment) VALUES " & _
                                            "(" & _
                                            "'" &
NLocationTB.Text & "', " & _
                                            "'" &
ALocationChB.Checked & "', " & _
                                            "'" &
CommentTB.Text & "'" & _
                                            ")"

                        AString = "Inserted " &
NLocationTB.Text & " location."

                    ElseIf MainWindow.HasRow =
True Then

                        QString = "UPDATE
tLocation SET " & _
                                            "City = '"
& NLocationTB.Text & "', " & _
                                            "isActive
= '" & ALocationChB.Checked & "', " & _
                                            "Comment =
'" & CommentTB.Text & "'" & _
                                        "WHERE ID = '" &
MainWindow.ID & "'"

                        AString = "Updated " &
NLocationTB.Text & " location."

                    End If

                    SQL.RunQuery(QString)

                    If
Duplicate.IsDuplicated(DString) Then

                        Trail.AddTrail(AString)

                        MsgBox("Saved!",
MsgBoxStyle.Information)

                        Me.Close()

                        MainWindow.LoadDataGrid()

                    Else

                        MsgBox("Location not
saved!", MsgBoxStyle.Information)

                    End If

                End If

            End If

        Catch ex As Exception

            MsgBox(ex.Message)

        End Try

    End Sub

    Public Sub SelectLocation()

        MainWindow.NewBtn.Visible = True
        MainWindow.EditBtn.Visible = True
        MainWindow.RefreshBtn.Visible = True
        MainWindow.DateTimePicker1.Visible = False
        MainWindow.DeleteBtn.Visible = True
        MainWindow.Visible = True

        MainWindow.GroupBox.Visible = True
        MainWindow.GroupBox.Text = "Location"

        MainWindow.QString = "EXEC selectLocation"

        MainWindow.LoadDataGrid()


MainWindow.DataGrid.Columns("ID").DisplayIndex = 0

MainWindow.DataGrid.Columns("Location").DisplayInd
ex = 1

MainWindow.DataGrid.Columns("Active").DisplayIndex
= 2
```

```
MainWindow.DataGrid.Columns("Comment").DisplayInde
x = 3

        Trail.AddTrail("Viewed Location.")

    End Sub

    Public Sub DeleteLocation()

        Dim result1 As DialogResult =
MessageBox.Show("Are you sure to delete " &
MainWindow.locationTB & "?", _
                "Delete Location", _
                MessageBoxButtons.YesNo)

        If result1 =
Windows.Forms.DialogResult.Yes Then

            MainWindow.QString = "DELETE from
tLocation WHERE ID = '" & MainWindow.ID & "'"

            MainWindow.LoadDataGrid()

            Trail.AddTrail("Deleted " &
MainWindow.locationTB & " from Location.")

            SelectLocation()

        End If

    End Sub

    Private Sub Location1_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load

        If MainWindow.HasRow = True Then

            NLocationTB.Text =
MainWindow.locationTB
            ALocationChB.Checked =
MainWindow.isActive
            CommentTB.Text = MainWindow.Comment

        End If

    End Sub

End Class
Public Class EmployeeStatus

    Private Duplicate As New CheckDuplicates
    Private Trail As New AuditTrail
    Private SQL As New SQLControl
    Private QString As String 'Query String for
insert or update an employee status.
    Private AString As String 'Query String for
Audit Trail

    Private Sub CancelBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles CancelBtn.Click

        Me.Close()

    End Sub

    Private Sub SaveBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles SaveBtn.Click

        Try

            If NEmployeeStatusTB.Text = "" Then

                MsgBox("Enter employee status
name.", MsgBoxStyle.Information)

            Else

                Dim DString As String

                DString = "SELECT * FROM
tEmployeeStatus WHERE Name = '" &
NEmployeeStatusTB.Text & "' AND isActive = '" &
AEmployeeStatusChB.Checked & "'"

                If Duplicate.IsDuplicated(DString)
Then

                    MsgBox("Employee Status
already exists!", MsgBoxStyle.Critical)

                    Exit Sub

                Else

                    If MainWindow.HasRow = False
Then

                        QString = "INSERT INTO
tEmployeeStatus (Name, isActive, Comment) VALUES "
& _
                                                "(" &
_
                                                "'" &
NEmployeeStatusTB.Text & "', " & _
                                                "'" &
AEmployeeStatusChB.Checked & "', " & _
                                                "'" &
CommentTB.Text & "'" & _
                                                ")"

                        AString = "Inserted " &
NEmployeeStatusTB.Text & " employee status."

                    ElseIf MainWindow.HasRow =
True Then

                        QString = "UPDATE
tEmployeeStatus SET " & _
                                            "Name = '"
& NEmployeeStatusTB.Text & "', " & _
                                            "isActive
= '" & AEmployeeStatusChB.Checked & "', " & _
                                            "Comment =
'" & CommentTB.Text & "' " & _
                                        "WHERE ID = '" &
MainWindow.ID & "'"

                        AString = "Updated " &
NEmployeeStatusTB.Text & " employee status."

                    End If

                    SQL.RunQuery(QString)
```

```
                    If
Duplicate.IsDuplicated(DString) Then

                        Trail.AddTrail(AString)

                        MsgBox("Saved!",
MsgBoxStyle.Information)

                        Me.Close()

                        MainWindow.LoadDataGrid()

                    Else

                        MsgBox("Employee Status
not saved!", MsgBoxStyle.Information)

                    End If

                End If

            End If

        Catch ex As Exception

            MsgBox(ex.Message,
MsgBoxStyle.Critical)

        End Try

    End Sub

    Public Sub SelectEmployeeStatus()

        MainWindow.NewBtn.Visible = True
        MainWindow.EditBtn.Visible = True
        MainWindow.RefreshBtn.Visible = True
        MainWindow.DateTimePicker1.Visible = False
        MainWindow.DeleteBtn.Visible = True
        MainWindow.Label2.Visible = False

        MainWindow.GroupBox.Visible = True
        MainWindow.GroupBox.Text = "Employee
Status"

        MainWindow.QString = "EXEC
selectEmployeeStatus"

        MainWindow.LoadDataGrid()


MainWindow.DataGrid.Columns("ID").DisplayIndex = 0
        MainWindow.DataGrid.Columns("Employee
Status").DisplayIndex = 1

MainWindow.DataGrid.Columns("Active").DisplayIndex
= 2

MainWindow.DataGrid.Columns("Comment").DisplayInde
x = 3

        Trail.AddTrail("Viewed Employee Status.")

    End Sub

    Public Sub DeleteEmployeeStatus()

        Dim result1 As DialogResult =
MessageBox.Show("Are you sure to delete " &
MainWindow.employeestatusTB & "?", _
                "Delete Employee Status", _
                MessageBoxButtons.YesNo)

        If result1 =
Windows.Forms.DialogResult.Yes Then

            MainWindow.QString = "DELETE from
tEmployeeStatus WHERE ID = '" & MainWindow.ID &
"'"

            MainWindow.LoadDataGrid()

            Trail.AddTrail("Deleted " &
MainWindow.employeestatusTB & " from Employee
Status.")

            SelectEmployeeStatus()

        End If

    End Sub

    Private Sub EmployeeStatus_Load(ByVal sender
As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load

        If MainWindow.HasRow = True Then

            NEmployeeStatusTB.Text =
MainWindow.employeestatusTB
            AEmployeeStatusChB.Checked =
MainWindow.isActive
            CommentTB.Text = MainWindow.Comment

        End If

    End Sub

End Class

Public Class Designation

    Private Duplicate As New CheckDuplicates
    Private Trail As New AuditTrail
    Private SQL As New SQLControl
    Private QString As String 'Query String
    Private AString As String 'Query String for
Audit Trail

    Private Sub CancelBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles CancelBtn.Click

        Me.Close()

    End Sub

    Private Sub SaveBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles SaveBtn.Click

        Try

            If NDesignationTB.Text = "" Or
BasicPayTB.Text = "" Or COLATB.Text = "" Then
```

```
                MsgBox("Please complete entries.",
MsgBoxStyle.Information)

            Else

                Dim DString As String

                DString = "SELECT * FROM
tDesignation WHERE Name = '" & NDesignationTB.Text
& "' AND BasicPay = '" & BasicPayTB.Text & "' AND
COLA = '" & COLATB.Text & "' AND isActive = '" &
ADesignationChB.Checked & "'"

                If Duplicate.IsDuplicated(DString)
Then

                    MsgBox("Designation already
exists!", MsgBoxStyle.Critical)

                    Exit Sub

                Else

                    If MainWindow.HasRow = False
Then

                        QString = "INSERT INTO
tDesignation (Name, BasicPay, COLA, isActive,
Comment) VALUES " & _
                                            "(" &
_
                                            "'" &
NDesignationTB.Text & "', " & _
                                            "'" &
BasicPayTB.Text & "', " & _
                                            "'" &
COLATB.Text & "', " & _
                                            "'" &
ADesignationChB.Checked & "', " & _
                                            "'" &
CommentTB.Text & "'" & _
                                            ")"

                        AString = "Inserted " &
NDesignationTB.Text & " designation."

                    ElseIf MainWindow.HasRow =
True Then

                        QString = "UPDATE
tDesignation SET " & _
                                        "Name = '"
& NDesignationTB.Text & "', " & _
                                        "BasicPay
= '" & BasicPayTB.Text & "', " & _
                                        "COLA = '"
& COLATB.Text & "', " & _
                                        "isActive
= '" & ADesignationChB.Checked & "', " & _
                                        "Comment =
'" & CommentTB.Text & "' " & _
                                    "WHERE ID = '" &
MainWindow.ID & "'"

                        AString = "Updated " &
NDesignationTB.Text & " designation."

                    End If

                    SQL.RunQuery(QString)

                    If
Duplicate.IsDuplicated(DString) Then

                        Trail.AddTrail(AString)

                        MsgBox("Saved!",
MsgBoxStyle.Information)

                        Me.Close()

                        MainWindow.LoadDataGrid()

                    Else

                        MsgBox("Designation not
saved!", MsgBoxStyle.Information)

                    End If

                End If

            End If

        Catch ex As Exception

            MsgBox(ex.Message,
MsgBoxStyle.Critical)

        End Try

    End Sub

    Public Sub SelectDesignation()

        MainWindow.NewBtn.Visible = True
        MainWindow.EditBtn.Visible = True
        MainWindow.RefreshBtn.Visible = True
        MainWindow.DateTimePicker1.Visible = False
        MainWindow.DeleteBtn.Visible = True
        MainWindow.Label2.Visible = False

        MainWindow.GroupBox.Visible = True
        MainWindow.GroupBox.Text = "Designation"

        MainWindow.QString = "EXEC
selectDesignation"

        MainWindow.LoadDataGrid()


MainWindow.DataGrid.Columns("ID").DisplayIndex = 0

MainWindow.DataGrid.Columns("Designation").Display
Index = 1
        MainWindow.DataGrid.Columns("Basic
Pay").DisplayIndex = 2

MainWindow.DataGrid.Columns("COLA").DisplayIndex =
3

MainWindow.DataGrid.Columns("Active").DisplayIndex
= 4

MainWindow.DataGrid.Columns("Comment").DisplayInde
x = 5

        Trail.AddTrail("Viewed Designation.")
```

```
    End Sub

    Public Sub DeleteDesignation()

        Dim result1 As DialogResult =
MessageBox.Show("Are you sure to delete " &
MainWindow.designationTB & "?", _
                "Delete Designation", _
                MessageBoxButtons.YesNo)

        If result1 =
Windows.Forms.DialogResult.Yes Then

            MainWindow.QString = "DELETE from
tDesignation WHERE ID = '" & MainWindow.ID & "'"

            MainWindow.LoadDataGrid()

            Trail.AddTrail("Deleted " &
MainWindow.designationTB & " from Designation.")

            SelectDesignation()

        End If

    End Sub

    Private Sub Designation_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load

        If MainWindow.HasRow = True Then

            NDesignationTB.Text =
MainWindow.designationTB
            BasicPayTB.Text =
MainWindow.basicpayTB
            COLATB.Text = MainWindow.COLA
            ADesignationChB.Checked =
MainWindow.isActive
            CommentTB.Text = MainWindow.Comment

        End If

    End Sub

End Class
Public Class Department

    Private Duplicate As New CheckDuplicates
    Private Trail As New AuditTrail
    Private SQL As New SQLControl
    Private QString As String 'Query String
    Private AString As String 'Query String for
Audit Trail

    Private Sub CancelBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles CancelBtn.Click

        Me.Close()

    End Sub

    Private Sub SaveBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles SaveBtn.Click

        Try

            If NDepartmentTB.Text = "" Then

                MsgBox("Enter department name.",
MsgBoxStyle.Information)

            Else

                Dim DString As String

                DString = "SELECT * FROM
tDepartment WHERE Name = '" & NDepartmentTB.Text &
"' AND isActive = '" & ADepartmentChB.Checked &
"'"

                If Duplicate.IsDuplicated(DString)
= True Then

                    MsgBox("Department already
exists!", MsgBoxStyle.Critical)

                    Exit Sub

                Else

                    If MainWindow.HasRow = False
Then

                        QString = "INSERT INTO
tDepartment (Name, isActive, Comment) VALUES " & _
                                                "(" &
_
                                                "'" &
NDepartmentTB.Text & "', " & _
                                                "'" &
ADepartmentChB.Checked & "', " & _
                                                "'" &
CommentTB.Text & "'" & _
                                                ")"

                        AString = "Inserted " &
NDepartmentTB.Text & " department."

                    ElseIf MainWindow.HasRow =
True Then

                        QString = "UPDATE
tDepartment SET " & _
                                            "Name = '"
& NDepartmentTB.Text & "', " & _
                                            "isActive
= '" & ADepartmentChB.Checked & "', " & _
                                            "Comment =
'" & CommentTB.Text & "'" & _
                                        "WHERE ID = '" &
MainWindow.ID & "'"

                        AString = "Updated " &
NDepartmentTB.Text & " department"

                    End If

                    SQL.RunQuery(QString)

                    If
Duplicate.IsDuplicated(DString) = True Then

                        Trail.AddTrail(AString)
```

```
                        MsgBox("Saved!",
MsgBoxStyle.Information)

                    Else

                        MsgBox("Department not
saved!", MsgBoxStyle.Information)

                    End If



                    Me.Close()

                    MainWindow.LoadDataGrid()

                End If

            End If

        Catch ex As Exception

            MsgBox(ex.Message,
MsgBoxStyle.Critical)

        End Try

    End Sub

    Public Sub SelectDepartment()

        MainWindow.NewBtn.Visible = True
        MainWindow.EditBtn.Visible = True
        MainWindow.RefreshBtn.Visible = True
        MainWindow.DateTimePicker1.Visible = False
        MainWindow.DeleteBtn.Visible = True
        MainWindow.Label2.Visible = False

        MainWindow.GroupBox.Visible = True
        MainWindow.GroupBox.Text = "Department"

        MainWindow.QString = "EXEC
selectDepartment"

        MainWindow.LoadDataGrid()


MainWindow.DataGrid.Columns("ID").DisplayIndex = 0

MainWindow.DataGrid.Columns("Department").DisplayI
ndex = 1

MainWindow.DataGrid.Columns("Active").DisplayIndex
= 2

MainWindow.DataGrid.Columns("Comment").DisplayInde
x = 3

        Trail.AddTrail("Viewed Department.")

    End Sub

    Public Sub DeleteDepartment()

        Dim result1 As DialogResult =
MessageBox.Show("Are you sure to delete " &
MainWindow.departmentTB & "?", _
                "Delete Department", _
                    MessageBoxButtons.YesNo)

        If result1 =
Windows.Forms.DialogResult.Yes Then

            MainWindow.QString = "DELETE from
tDepartment WHERE ID = '" & MainWindow.ID & "'"

            MainWindow.LoadDataGrid()

            Trail.AddTrail("Deleted " &
MainWindow.departmentTB & " from Department.")

            SelectDepartment()

        End If

    End Sub

    Private Sub Department_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load

        If MainWindow.HasRow = True Then

            NDepartmentTB.Text =
MainWindow.departmentTB
            ADepartmentChB.Checked =
MainWindow.isActive
            CommentTB.Text = MainWindow.Comment

        End If

    End Sub

End Class

Public Class EmployeeWithViolation

    Private Duplicate As New CheckDuplicates
    Private Trail As New AuditTrail
    Private SQL As New SQLControl
    Private QString As String 'Query String for
insert or update an employee with violation.
    Private AString As String 'Query String for
Audit Trail

    Private Sub CancelBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles CancelBtn.Click

        Me.Close()

    End Sub

    Private Sub EmployeeWithViolation_Load(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        Try

            SQL.RunQuery("SELECT ID,
EmployeeNumber, Name from tEmployee WHERE
isActive='1'")

            ENameCB.DataSource =
SQL.SQLDS.Tables(0)
            ENameCB.ValueMember = "ID"
            ENameCB.DisplayMember = "Name"
```

```
            SQL.RunQuery("SELECT ID, OffenseRank,
Penalty, Name from tViolation WHERE isActive='1'")

            ViolationCB.DataSource =
SQL.SQLDS.Tables(0)
            ViolationCB.ValueMember = "ID"
            ViolationCB.DisplayMember = "Name"

            If MainWindow.HasRow = True Then

                ENameCB.SelectedIndex =
ENameCB.FindStringExact(MainWindow.NameTB)
                ViolationCB.SelectedIndex =
ViolationCB.FindStringExact(MainWindow.violationTB
)
                EWVDatePicker.Value =
MainWindow.dateTB
                CommentTB.Text =
MainWindow.Comment

            End If

        Catch ex As Exception

            MsgBox(ex.Message,
MsgBoxStyle.Critical)

        End Try

    End Sub

    Private Sub ENameCB_SelectedIndexChanged(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
ENameCB.SelectedIndexChanged

        If (Not Me.ENameCB.SelectedItem Is
Nothing) Then

            Dim SItem =
TryCast(ENameCB.SelectedItem, DataRowView)
            Me.ENumberTB.Text =
SItem.Row("EmployeeNumber").ToString()

        End If

    End Sub

    Private Sub
ViolationCB_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles ViolationCB.SelectedIndexChanged

        If (Not Me.ViolationCB.SelectedItem Is
Nothing) Then

            Dim SItem =
TryCast(ViolationCB.SelectedItem, DataRowView)
            Me.OffenseRankTB.Text =
SItem.Row("OffenseRank").ToString()
            Me.PenaltyTB.Text =
SItem.Row("Penalty").ToString()

        End If

    End Sub
```

```
    Private Sub SaveBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles SaveBtn.Click

        Try

            Dim DString As String

            DString = "SELECT * FROM
tEmployeeWithViolation WHERE ID_Employee = '" &
ENameCB.SelectedValue & "' AND ID_Violation = '" &
ViolationCB.SelectedValue & "' AND Date = '" &
EWVDatePicker.Value.Date & "'"

            If Duplicate.IsDuplicated(DString) =
True Then

                MsgBox("Employee with Violation on
the said date already exists!",
MsgBoxStyle.Critical)

                Exit Sub

            Else

                If MainWindow.HasRow = False Then

                    QString = "INSERT into
tEmployeeWithViolation " & _
                                        "(ID_Employee,
ID_Violation, Date, Comment) VALUES " & _
                                        "(" & _
                                        "'" &
ENameCB.SelectedValue & "', " & _
                                        "'" &
ViolationCB.SelectedValue & "', " & _
                                        "'" &
EWVDatePicker.Value.Date & "', " & _
                                        "'" &
CommentTB.Text & "'" & _
                                        ")"

                    AString = "Inserted " &
ENameCB.SelectedValue & " employee with " &
ViolationCB.SelectedValue & " violation."

                ElseIf MainWindow.HasRow = True
Then

                    QString = "UPDATE
tEmployeeWithViolation SET " & _
                                        "ID_Employee =
'" & ENameCB.SelectedValue & "', " & _
                                        "ID_Violation
= '" & ViolationCB.SelectedValue & "', " & _
                                        "Date = '" &
EWVDatePicker.Value.Date & "', " & _
                                        "Comment = '"
& CommentTB.Text & "'"

                    AString = "Updated " &
ENameCB.SelectedValue & " employee with " &
ViolationCB.SelectedValue & " violation."

                End If

                SQL.RunQuery(QString)
```

```
                If Duplicate.IsDuplicated(DString)
= True Then

                    Trail.AddTrail(AString)

                    MsgBox("Saved!",
MsgBoxStyle.Information)

                    Me.Close()

                    MainWindow.LoadDataGrid()

                Else

                    MsgBox("Employee with
Violation not saved!", MsgBoxStyle.Information)

                End If

            End If

        Catch ex As Exception

            MsgBox(ex.Message,
MsgBoxStyle.Critical)

        End Try

    End Sub

    Public Sub SelectEmployeeWithViolation()

        MainWindow.NewBtn.Visible = True
        MainWindow.EditBtn.Visible = True
        MainWindow.RefreshBtn.Visible = True
        MainWindow.DateTimePicker1.Visible = False
        MainWindow.DeleteBtn.Visible = True
        MainWindow.Label2.Visible = False

        MainWindow.GroupBox.Visible = True
        MainWindow.GroupBox.Text = "Employee with
Violation"

        MainWindow.QString = "EXEC
selectEmployeeWithViolation"

        MainWindow.LoadDataGrid()


MainWindow.DataGrid.Columns("ID").DisplayIndex = 0

MainWindow.DataGrid.Columns("Date").DisplayIndex =
1
        MainWindow.DataGrid.Columns("Employee
Number").DisplayIndex = 2
        MainWindow.DataGrid.Columns("Employee
Name").DisplayIndex = 3

MainWindow.DataGrid.Columns("Branch").DisplayIndex
= 4

MainWindow.DataGrid.Columns("Department").DisplayI
ndex = 5

MainWindow.DataGrid.Columns("Designation").Display
Index = 6
        MainWindow.DataGrid.Columns("Employee
Status").DisplayIndex = 7

MainWindow.DataGrid.Columns("Violation").DisplayIn
dex = 8
        MainWindow.DataGrid.Columns("Offense
Rank").DisplayIndex = 9

MainWindow.DataGrid.Columns("Penalty").DisplayInde
x = 10

MainWindow.DataGrid.Columns("Comment").DisplayInde
x = 11

        Trail.AddTrail("Viewed Employees with
Violation.")

    End Sub

    Public Sub DeleteEmployeeWithViolation()

        Dim result1 As DialogResult =
MessageBox.Show("Are you sure to delete " &
MainWindow.NameTB & "'s violation?", _
                "Delete Employee With Violation",
_
                MessageBoxButtons.YesNo)

        If result1 =
Windows.Forms.DialogResult.Yes Then

            MainWindow.QString = "DELETE from
tEmployeeWithViolation WHERE ID = '" &
MainWindow.ID & "'"

            MainWindow.LoadDataGrid()

            Trail.AddTrail("Deleted " &
MainWindow.NameTB & "'s violation from Employee
With Violation.")

            SelectEmployeeWithViolation()

        End If

    End Sub

End Class
Public Class EmployeeLeaveCredit

    Private Duplicate As New CheckDuplicates
    Private Trail As New AuditTrail
    Private SQL As New SQLControl
    Private QString As String 'Query String for
insert or update an employee leave credit.
    Private AString As String 'Query String for
Audit Trail

    Private Sub EmployeeLeaveCredit_Load(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        SQL.RunQuery("SELECT ID, EmployeeNumber,
Name from tEmployee WHERE isActive='1'")

        ENameCB.DataSource = SQL.SQLDS.Tables(0)
        ENameCB.ValueMember = "ID"
        ENameCB.DisplayMember = "Name"

        SQL.RunQuery("SELECT ID, Name from
tLeaveType WHERE isActive='1'")
```

```
        LeaveTypeCB.DataSource =
SQL.SQLDS.Tables(0)
        LeaveTypeCB.ValueMember = "ID"
        LeaveTypeCB.DisplayMember = "Name"

        If MainWindow.HasRow = True Then

            CommentTB.Text = MainWindow.Comment
            TotalNumber.Text =
MainWindow.totalnumberTB

            ENameCB.SelectedIndex =
ENameCB.FindStringExact(MainWindow.NameTB)
            LeaveTypeCB.SelectedIndex =
LeaveTypeCB.FindStringExact(MainWindow.leavetypeTB
)

        Else

            LeaveTypeCB.SelectedIndex =
LeaveTypeCB.FindStringExact(MainWindow.leavetypeTB
)

        End If

    End Sub

    Private Sub CancelBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles CancelBtn.Click

        Me.Close()

    End Sub

    Private Sub SaveBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles SaveBtn.Click

        Try

            If TotalNumber.Text = "" Then

                MsgBox("Enter total number of
leave", MsgBoxStyle.Information)

            Else

                Dim DString As String

                DString = "SELECT * FROM
tLeaveRemaining " & _
                                "WHERE ID_Employee
= '" & ENameCB.SelectedValue & "' AND " & _

"ID_LeaveType = '" & LeaveTypeCB.SelectedValue &
"'"

                If Duplicate.IsDuplicated(DString)
= True Then

                    MsgBox("Leave Credit already
exists for the employee.", MsgBoxStyle.Critical)

                    Exit Sub

                Else

                    If MainWindow.HasRow = False
Then

                        QString = "INSERT into
tLeaveRemaining (ID_Employee, TotalNumber,
ID_LeaveType, Comment, Remaining) VALUES " & _
                                            "(" & _
                                            "'" &
ENameCB.SelectedValue & "', " & _
                                            "'" &
TotalNumber.Text & "', " & _
                                            "'" &
LeaveTypeCB.SelectedValue & "', " & _
                                            "'" &
CommentTB.Text & "', " & _
                                            "'" &
TotalNumber.Text & "'" & _
                                            ")"

                        AString = "Inserted leave
credit for employee " & ENameCB.SelectedValue &
"."

                    ElseIf MainWindow.HasRow =
True Then

                        QString = "UPDATE
tLeaveRemaining SET " & _

"ID_Employee = '" & ENameCB.SelectedValue & "', "
& _

"TotalNumber = '" & TotalNumber.Text & "', " & _

"ID_LeaveType = '" & LeaveTypeCB.SelectedValue &
"', " & _
                                            "Comment =
'" & CommentTB.Text & "', " & _
                                            "Remaining
= '" & TotalNumber.Text & "' " & _
                                              "WHERE ID = '" &
MainWindow.ID & "'"

                        AString = "Updated leave
credit for employee " & ENameCB.SelectedValue &
"."

                    End If


                    SQL.RunQuery(QString)

                    If
Duplicate.IsDuplicated(DString) = True Then

                        Trail.AddTrail(AString)

                        MsgBox("Saved!",
MsgBoxStyle.Information)

                        Me.Close()

                        MainWindow.LoadDataGrid()

                    Else

                        MsgBox("Employee Leave
Credit not saved!", MsgBoxStyle.Information)
```

```
                End If

            End If

        End If

    Catch ex As Exception

        MsgBox(ex.Message)

    End Try

End Sub

Private Sub ENameCB_SelectedIndexChanged(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
ENameCB.SelectedIndexChanged

    If (Not Me.ENameCB.SelectedItem Is
Nothing) Then

        Dim SItem =
TryCast(ENameCB.SelectedItem, DataRowView)
        Me.ENumberTB.Text =
SItem.Row("EmployeeNumber").ToString()

    End If

End Sub

Public Sub SelectEmployeeLeaveCredit(ByVal
Type As String)

    MainWindow.NewBtn.Visible = True
    MainWindow.EditBtn.Visible = True
    MainWindow.RefreshBtn.Visible = True
    MainWindow.DateTimePicker1.Visible = False
    MainWindow.DeleteBtn.Visible = True
    MainWindow.Label2.Visible = False

    MainWindow.GroupBox.Visible = True
    MainWindow.GroupBox.Text = Type

    MainWindow.QString = "EXEC selectLeave @LT
= '" & Type & "'"
    MainWindow.LoadDataGrid()


MainWindow.DataGrid.Columns("ID").DisplayIndex = 0
    MainWindow.DataGrid.Columns("Employee
Name").DisplayIndex = 1
    MainWindow.DataGrid.Columns("Employee
Number").DisplayIndex = 2

MainWindow.DataGrid.Columns("Branch").DisplayIndex
= 3

MainWindow.DataGrid.Columns("Department").DisplayI
ndex = 4

MainWindow.DataGrid.Columns("Designation").Display
Index = 5
    MainWindow.DataGrid.Columns("Employee
Status").DisplayIndex = 6
    MainWindow.DataGrid.Columns("Leave
Type").DisplayIndex = 7
    MainWindow.DataGrid.Columns("Total
Number").DisplayIndex = 8
MainWindow.DataGrid.Columns("Remaining").DisplayIn
dex = 9

MainWindow.DataGrid.Columns("Comment").DisplayInde
x = 10

    Trail.AddTrail("Viewed " & Type & "
Credit.")

End Sub

End Class
Public Class Employee

    Private Duplicate As New CheckDuplicates
    Private Trail As New AuditTrail
    Private SQL As New SQLControl
    Private QString As String 'Query String for
insert or update an employee.
    Private AString As String 'Query String for
Audit Trail

    Private Sub AddEmployee_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load

    SQL.RunQuery("SELECT ID, Name from
tBranch")

    BranchCB.DataSource = SQL.SQLDS.Tables(0)
    BranchCB.ValueMember = "ID"
    BranchCB.DisplayMember = "Name"

    SQL.RunQuery("SELECT ID, Name from
tDepartment")

    DepartmentCB.DataSource =
SQL.SQLDS.Tables(0)
    DepartmentCB.ValueMember = "ID"
    DepartmentCB.DisplayMember = "Name"

    SQL.RunQuery("SELECT ID, Name from
tDesignation")

    DesignationCB.DataSource =
SQL.SQLDS.Tables(0)
    DesignationCB.ValueMember = "ID"
    DesignationCB.DisplayMember = "Name"

    SQL.RunQuery("SELECT ID, Name from
tEmployeeStatus")

    EmployeeStatusCB.DataSource =
SQL.SQLDS.Tables(0)
    EmployeeStatusCB.ValueMember = "ID"
    EmployeeStatusCB.DisplayMember = "Name"

    SQL.RunQuery("SELECT ID, Name from
tWeeklySchedule")

    WeeklyScheduleCB.DataSource =
SQL.SQLDS.Tables(0)
    WeeklyScheduleCB.ValueMember = "ID"
    WeeklyScheduleCB.DisplayMember = "Name"

    If MainWindow.HasRow = True Then

        NLNameTB.Text = MainWindow.lname
```

```
            NFNameTB.Text = MainWindow.fname
            NMNameTB.Text = MainWindow.mname
            AEmployeeChB.Checked =
MainWindow.isActive
            CommentTB.Text = MainWindow.Comment
            ENumberTB.Text = MainWindow.enumber
            ENumberTB.ReadOnly = True

            BranchCB.SelectedIndex =
BranchCB.FindStringExact(MainWindow.branchTB)
            DepartmentCB.SelectedIndex =
DepartmentCB.FindStringExact(MainWindow.department
TB)
            DesignationCB.SelectedIndex =
DesignationCB.FindStringExact(MainWindow.designati
onTB)
            EmployeeStatusCB.SelectedIndex =
EmployeeStatusCB.FindStringExact(MainWindow.employ
eestatusTB)
            WeeklyScheduleCB.SelectedIndex =
WeeklyScheduleCB.FindStringExact(MainWindow.weekly
sched)

        End If

    End Sub


    Private Sub CancelBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles CancelBtn.Click

        Me.Close()

    End Sub

    Private Sub SaveBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles SaveBtn.Click

        Try

            If ENumberTB.Text = "" Or
NLNameTB.Text = "" Or NFNameTB.Text = "" Then

                MsgBox("Please complete fields.",
MsgBoxStyle.Information)

            Else

                Dim DString As String

                DString = "SELECT * FROM tEmployee
" & _
                          "WHERE EmployeeNumber =
'" & ENumberTB.Text & "' AND " & _
                                "LastName = '" &
NLNameTB.Text & "' AND " & _
                                "FirstName = '" &
NFNameTB.Text & "' AND " & _
                                "MiddleName = '" &
NMNameTB.Text & "' AND " & _
                                "ID_Branch = '" &
BranchCB.SelectedValue & "' AND " & _
                                "ID_Department =
'" & DepartmentCB.SelectedValue & "' AND " & _
                                "ID_Designation =
'" & DesignationCB.SelectedValue & "' AND " & _
                                "ID_EmployeeStatus
= '" & EmployeeStatusCB.SelectedValue & "' AND " &
_
                                "ID_WeeklySchedule
= '" & WeeklyScheduleCB.SelectedValue & "' AND " &
_
                                "isActive = '" &
AEmployeeChB.Checked & "'"

                If Duplicate.IsDuplicated(DString)
= True Then

                    MsgBox("Employee already
exists!", MsgBoxStyle.Critical)

                Else

                    If MainWindow.HasRow = False
Then

                        QString = "INSERT INTO
tEmployee " & _

"(LastName, FirstName, MiddleName, EmployeeNumber,
ID_Branch, ID_Department, ID_Designation,
ID_EmployeeStatus, ID_WeeklySchedule, isActive,
Comment) " & _
                                            "VALUES "
& _
                                            "(" & _
                                            "'" &
NLNameTB.Text & "', " & _
                                            "'" &
NFNameTB.Text & "', " & _
                                            "'" &
NMNameTB.Text & "', " & _
                                            "'" &
ENumberTB.Text & "', " & _
                                            "'" &
BranchCB.SelectedValue & "', " & _
                                            "'" &
DepartmentCB.SelectedValue & "', " & _
                                            "'" &
DesignationCB.SelectedValue & "', " & _
                                            "'" &
EmployeeStatusCB.SelectedValue & "', " & _
                                            "'" &
WeeklyScheduleCB.SelectedValue & "', " & _
                                            "'" &
AEmployeeChB.Checked & "', " & _
                                            "'" &
CommentTB.Text & "'" & _
                                            ")"

                        AString = "Inserted " &
NLNameTB.Text & ", " & NFNameTB.Text & " " &
NMNameTB.Text & " employee."

                    ElseIf MainWindow.HasRow =
True Then

                        QString = "UPDATE
tEmployee SET " & _
                                            "LastName
= '" & NLNameTB.Text & "', " & _
                                            "FirstName
= '" & NFNameTB.Text & "', " & _

"MiddleName = '" & NMNameTB.Text & "', " & _
```

```
"EmployeeNumber = '" & ENumberTB.Text & "', " & _
                                        "ID_Branch
= '" & BranchCB.SelectedValue & "', " & _

"ID_Department = '" & DepartmentCB.SelectedValue &
"', " & _

"ID_Designation = '" & DesignationCB.SelectedValue
& "', " & _

"ID_EmployeeStatus = '" &
EmployeeStatusCB.SelectedValue & "', " & _

"ID_WeeklySchedule = '" &
WeeklyScheduleCB.SelectedValue & "', " & _
                                        "isActive
= '" & AEmployeeChB.Checked & "', " & _
                                        "Comment =
'" & CommentTB.Text & "' " & _
                                    "WHERE ID = '" &
MainWindow.ID & "'"


                        AString = "Updated " &
NLNameTB.Text & ", " & NFNameTB.Text & " " &
NMNameTB.Text & " employee."

                        End If

                    SQL.RunQuery(QString)

                    If
Duplicate.IsDuplicated(DString) = True Then

                        Trail.AddTrail(AString)

                        MsgBox("Saved!",
MsgBoxStyle.Information)

                        Me.Close()

                        MainWindow.LoadDataGrid()

                    Else

                        MsgBox("Employee not
saved!", MsgBoxStyle.Information)

                    End If

                End If

            End If

        Catch ex As Exception

            MsgBox(ex.Message,
MsgBoxStyle.Critical)

        End Try

    End Sub

    Public Sub SelectEmployee()

        MainWindow.NewBtn.Visible = True
        MainWindow.EditBtn.Visible = True
        MainWindow.RefreshBtn.Visible = True
        MainWindow.DateTimePicker1.Visible = False
        MainWindow.DeleteBtn.Visible = True
        MainWindow.Label2.Visible = False

        MainWindow.GroupBox.Visible = True
        MainWindow.GroupBox.Text = "Employees"

        MainWindow.QString = "EXEC selectEmployee"

        MainWindow.LoadDataGrid()


MainWindow.DataGrid.Columns("ID").DisplayIndex = 0
        MainWindow.DataGrid.Columns("Last
Name").DisplayIndex = 1
        MainWindow.DataGrid.Columns("First
Name").DisplayIndex = 2
        MainWindow.DataGrid.Columns("Middle
Name").DisplayIndex = 3
        MainWindow.DataGrid.Columns("Employee
Number").DisplayIndex = 4

MainWindow.DataGrid.Columns("Branch").DisplayIndex
= 5

MainWindow.DataGrid.Columns("Department").DisplayI
ndex = 6

MainWindow.DataGrid.Columns("Designation").Display
Index = 7
        MainWindow.DataGrid.Columns("Employee
Status").DisplayIndex = 8
        MainWindow.DataGrid.Columns("Weekly
Schedule").DisplayIndex = 9

MainWindow.DataGrid.Columns("Active").DisplayIndex
= 10

MainWindow.DataGrid.Columns("Comment").DisplayInde
x = 11

        Trail.AddTrail("Viewed Employees.")

    End Sub

    Public Sub DeleteEmployee()

        Dim result1 As DialogResult =
MessageBox.Show("Are you sure to delete " &
MainWindow.lname & ", " & MainWindow.fname & " " &
MainWindow.mname & "?", _
                "Delete Employee", _
                MessageBoxButtons.YesNo)

        If result1 =
Windows.Forms.DialogResult.Yes Then

            MainWindow.QString = "DELETE from
tEmployee WHERE ID = '" & MainWindow.ID & "'"

            MainWindow.LoadDataGrid()

            Trail.AddTrail("Deleted " &
MainWindow.lname & ", " & MainWindow.fname & " " &
MainWindow.mname & "from Employee.")

            SelectEmployee()
```

```
        End If

    End Sub

End Class
Public Class FileUnderTime

    Private Duplicate As New CheckDuplicates
    Private Trail As New AuditTrail
    Private SQL As New SQLControl
    Private QString As String 'Query String for
insert or update an employee.
    Private AString As String 'Query String for
Audit Trail

    Private Sub CancelBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles CancelBtn.Click

        Me.Close()

    End Sub

    Private Sub FileUnderTime_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load

        SQL.RunQuery("SELECT ID, EmployeeNumber,
Name from tEmployee WHERE isActive='1'")

        EmployeeNameCB.DataSource =
SQL.SQLDS.Tables(0)
        EmployeeNameCB.ValueMember = "ID"
        EmployeeNameCB.DisplayMember = "Name"

        If MainWindow.HasRow = True Then

            DateFUTDTP.Value = MainWindow.dateTB
            ReasonTB.Text = MainWindow.reason
            CommentTB.Text = MainWindow.Comment
            ApproveChB.Checked =
MainWindow.approve
            EmployeeNameCB.SelectedIndex =
EmployeeNameCB.FindStringExact(MainWindow.NameTB)

        End If

    End Sub

    Private Sub
EmployeeNameCB_SelectedIndexChanged(ByVal sender
As System.Object, ByVal e As System.EventArgs)
Handles EmployeeNameCB.SelectedIndexChanged

        If (Not Me.EmployeeNameCB.SelectedItem Is
Nothing) Then

            Dim SItem =
TryCast(EmployeeNameCB.SelectedItem, DataRowView)
            Me.ENumberTB.Text =
SItem.Row("EmployeeNumber").ToString()

        End If

    End Sub

    Private Sub SaveBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles SaveBtn.Click

        Try

            Dim DString As String

            DString = "SELECT * FROM
tFileUndertime WHERE " & _
                                "ID_Employee = '" &
EmployeeNameCB.SelectedValue & "' AND " & _
                                "Date = '" &
DateFUTDTP.Value.Date & "' AND " & _
                                "isApproved = '" &
ApproveChB.Checked & "' AND " & _
                                "Reason = '" &
ReasonTB.Text & "'"

            If Duplicate.IsDuplicated(DString) =
True Then

                MsgBox("Filed undertime already
exists for the employee.", MsgBoxStyle.Critical)

                Exit Sub

            Else

                If MainWindow.HasRow = False Then

                    QString = "INSERT INTO
tFileUndertime (ID_Employee, Date, isApproved,
Reason, Comment) VALUES " & _
                                        "(" & _
                                        "'" &
EmployeeNameCB.SelectedValue & "', " & _
                                        "'" &
DateFUTDTP.Value.Date & "', " & _
                                        "'" &
ApproveChB.Checked & "', " & _
                                        "'" &
ReasonTB.Text & "', " & _
                                        "'" &
CommentTB.Text & "'" & _
                                        ")"

                    AString = "Inserted file of
undertime for " & EmployeeNameCB.SelectedValue & "
on " & DateFUTDTP.Value.Date & "."

                ElseIf MainWindow.HasRow = True
Then

                    QString = "UPDATE
tFileUndertime SET " & _
                                        "ID_Employee =
'" & EmployeeNameCB.SelectedValue & "', " & _
                                        "Date = '" &
DateFUTDTP.Value.Date & "', " & _
                                        "isApproved =
'" & ApproveChB.Checked & "', " & _
                                        "Reason = '" &
ReasonTB.Text & "', " & _
                                        "Comment = '"
& CommentTB.Text & "'"

                    AString = "Updated filed
undertime for " & EmployeeNameCB.SelectedValue & "
on " & DateFUTDTP.Value.Date & "."

                End If
```

```
                SQL.RunQuery(QString)

                If Duplicate.IsDuplicated(DString)
= True Then

                    Trail.AddTrail(AString)

                    MsgBox("Saved!",
MsgBoxStyle.Information)

                    Me.Close()

                    MainWindow.LoadDataGrid()

                Else

                    MsgBox("File of undertime not
saved!", MsgBoxStyle.Information)

                End If

            End If

        Catch ex As Exception

            MsgBox(ex.Message,
MsgBoxStyle.Critical)

        End Try

    End Sub

    Public Sub SelectFileUnderTime()

        MainWindow.NewBtn.Visible = True
        MainWindow.EditBtn.Visible = True
        MainWindow.RefreshBtn.Visible = True
        MainWindow.DeleteBtn.Visible = True
        MainWindow.DateTimePicker1.Visible = False
        MainWindow.Label2.Visible = False

        MainWindow.GroupBox.Visible = True
        MainWindow.GroupBox.Text = "File
Undertime"

        MainWindow.QString = "EXEC selectFLUT"

        MainWindow.LoadDataGrid()


MainWindow.DataGrid.Columns("ID").DisplayIndex = 0

MainWindow.DataGrid.Columns("Date").DisplayIndex =
1
        MainWindow.DataGrid.Columns("Employee
Name").DisplayIndex = 2
        MainWindow.DataGrid.Columns("Employee
Number").DisplayIndex = 3

MainWindow.DataGrid.Columns("Branch").DisplayIndex
= 4

MainWindow.DataGrid.Columns("Department").DisplayI
ndex = 5

MainWindow.DataGrid.Columns("Designation").Display
Index = 6

        MainWindow.DataGrid.Columns("Employee
Status").DisplayIndex = 7

MainWindow.DataGrid.Columns("Reason").DisplayIndex
= 8

MainWindow.DataGrid.Columns("Approved").DisplayInd
ex = 9

MainWindow.DataGrid.Columns("Comment").DisplayInde
x = 10

        Trail.AddTrail("Viewed File Undertime.")

    End Sub

    Public Sub DeleteFileUnderTime()

        Dim result1 As DialogResult =
MessageBox.Show("Are you sure to delete " &
MainWindow.NameTB & "'s Filed Undertime?", _
                "Delete File Undertime", _
                MessageBoxButtons.YesNo)

        If result1 =
Windows.Forms.DialogResult.Yes Then

            MainWindow.QString = "DELETE from
tFileUndertime WHERE ID = '" & MainWindow.ID & "'"

            MainWindow.LoadDataGrid()

            Trail.AddTrail("Deleted " &
MainWindow.NameTB & "'s Filed Undertime from File
Undertime.")

            SelectFileUnderTime()

        End If

    End Sub

End Class
Public Class FileOverTime

    Private Duplicate As New CheckDuplicates
    Private Trail As New AuditTrail
    Private SQL As New SQLControl
    Private QString As String 'Query String for
filing an employee overtime.
    Private AString As String 'Query String for
Audit Trail

    Private Sub CancelBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles CancelBtn.Click

        Me.Close()

    End Sub

    Private Sub FileOverTime_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load

        SQL.RunQuery("SELECT ID, EmployeeNumber,
Name from tEmployee WHERE isActive='1'")
```

```
        EmployeeNameCB.DataSource =
SQL.SQLDS.Tables(0)
        EmployeeNameCB.ValueMember = "ID"
        EmployeeNameCB.DisplayMember = "Name"

        If MainWindow.HasRow = True Then

            DateFOTDTP.Value = MainWindow.dateTB
            ReasonTB.Text = MainWindow.reason
            CommentTB.Text = MainWindow.Comment
            ApproveChB.Checked =
MainWindow.approve
            EmployeeNameCB.SelectedIndex =
EmployeeNameCB.FindStringExact(MainWindow.NameTB)

        End If

    End Sub

    Private Sub
EmployeeNameCB_SelectedIndexChanged(ByVal sender
As System.Object, ByVal e As System.EventArgs)
Handles EmployeeNameCB.SelectedIndexChanged

        If (Not Me.EmployeeNameCB.SelectedItem Is
Nothing) Then
            Dim SItem =
TryCast(EmployeeNameCB.SelectedItem, DataRowView)
            Me.ENumberTB.Text =
SItem.Row("EmployeeNumber").ToString()
        End If

    End Sub

    Private Sub SaveBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles SaveBtn.Click

        Try

            Dim DString As String

            DString = "SELECT * FROM tFileOvertime
WHERE " & _
                              "ID_Employee = '" &
EmployeeNameCB.SelectedValue & "' AND " & _
                              "Date = '" &
DateFOTDTP.Value.Date.Date & "' AND " & _
                              "isApproved = '" &
ApproveChB.Checked & "' AND " & _
                              "Reason = '" &
ReasonTB.Text & "'"

            If Duplicate.IsDuplicated(DString) =
True Then

                MsgBox("File of overtime for the
employee already exists!", MsgBoxStyle.Critical)

                Exit Sub

            Else

                If MainWindow.HasRow = False Then

                    QString = "INSERT INTO
tFileOvertime (ID_Employee, Date, isApproved,
Reason, Comment) VALUES " & _
                                      "(" & _
                                      "'" &
EmployeeNameCB.SelectedValue & "', " & _
                                      "'" &
DateFOTDTP.Value.Date & "', " & _
                                      "'" &
ApproveChB.Checked & "', " & _
                                      "'" &
ReasonTB.Text & "', " & _
                                      "'" &
CommentTB.Text & "'" & _
                                      ")"

                    AString = "Inserted a file of
overtime for " & EmployeeNameCB.SelectedValue & "
on " & DateFOTDTP.Value.Date & "."

                ElseIf MainWindow.HasRow = True
Then

                    QString = "UPDATE
tFileOvertime SET " & _
                                      "ID_Employee =
'" & EmployeeNameCB.SelectedValue & "', " & _
                                      "Date = '" &
DateFOTDTP.Value.Date & "', " & _
                                      "isApproved =
'" & ApproveChB.Checked & "', " & _
                                      "Reason = '" &
ReasonTB.Text & "', " & _
                                      "Comment = '"
& CommentTB.Text & "'"

                    AString = "Updated a file of
overtime for " & EmployeeNameCB.SelectedValue & "
on " & DateFOTDTP.Value.Date & "."

                End If

                SQL.RunQuery(QString)

                If Duplicate.IsDuplicated(DString)
= True Then

                    Trail.AddTrail(AString)

                    MsgBox("Saved!",
MsgBoxStyle.Information)

                    Me.Close()

                    MainWindow.LoadDataGrid()

                Else

                    MsgBox("File of overtime not
saved!", MsgBoxStyle.Information)

                End If

            End If

        Catch ex As Exception

            MsgBox(ex.Message,
MsgBoxStyle.Critical)

        End Try

    End Sub
```

```
    Public Sub SelectFileOverTime()

        MainWindow.NewBtn.Visible = True
        MainWindow.EditBtn.Visible = True
        MainWindow.RefreshBtn.Visible = True
        MainWindow.DeleteBtn.Visible = True
        MainWindow.DateTimePicker1.Visible = False
        MainWindow.Label2.Visible = False

        MainWindow.GroupBox.Visible = True
        MainWindow.GroupBox.Text = "File Overtime"

        MainWindow.QString = "EXEC selectFLOT"

        MainWindow.LoadDataGrid()


MainWindow.DataGrid.Columns("ID").DisplayIndex = 0

MainWindow.DataGrid.Columns("Date").DisplayIndex =
1
        MainWindow.DataGrid.Columns("Employee
Name").DisplayIndex = 2
        MainWindow.DataGrid.Columns("Employee
Number").DisplayIndex = 3

MainWindow.DataGrid.Columns("Branch").DisplayIndex
= 4

MainWindow.DataGrid.Columns("Department").DisplayI
ndex = 5

MainWindow.DataGrid.Columns("Designation").Display
Index = 6
        MainWindow.DataGrid.Columns("Employee
Status").DisplayIndex = 7

MainWindow.DataGrid.Columns("Reason").DisplayIndex
= 8

MainWindow.DataGrid.Columns("Approved").DisplayInd
ex = 9

MainWindow.DataGrid.Columns("Comment").DisplayInde
x = 10

        Trail.AddTrail("Viewed File Overtime.")

    End Sub

    Public Sub DeleteFileOverTime()

        Dim result1 As DialogResult =
MessageBox.Show("Are you sure to delete " &
MainWindow.NameTB & "'s Filed Overtime?", _
                "Delete File Overtime", _
                MessageBoxButtons.YesNo)

        If result1 =
Windows.Forms.DialogResult.Yes Then

            MainWindow.LoadDataGrid()

            MainWindow.QString = "DELETE from
tFileOvertime WHERE ID = '" & MainWindow.ID & "'"

            Trail.AddTrail("Deleted " &
MainWindow.NameTB & "'s Filed Overtime from File
Overtime.")

            SelectFileOverTime()

        End If

    End Sub

End Class
Public Class FileLeaveofAbsence

    Private Duplicate As New CheckDuplicates
    Private Trail As New AuditTrail
    Private SQL As New SQLControl
    Private QString As String 'Query String for
insert or update an employee.
    Private AString As String 'Query String for
Audit Trail
    Dim insertok As Boolean

    Private Sub FileLeaveofAbsence_Load(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        Try

            SQL.RunQuery("SELECT ID,
EmployeeNumber, Name from tEmployee WHERE
isActive='1'")

            EmployeeNameCB.DataSource =
SQL.SQLDS.Tables(0)
            EmployeeNameCB.ValueMember = "ID"
            EmployeeNameCB.DisplayMember = "Name"

            SQL.RunQuery("SELECT ID, Name from
tLeaveType WHERE isActive='1'")

            LeaveTypeCB.DataSource =
SQL.SQLDS.Tables(0)
            LeaveTypeCB.ValueMember = "ID"
            LeaveTypeCB.DisplayMember = "Name"

            If MainWindow.HasRow = True Then

                FilingDateDTP.Value =
MainWindow.dateTB
                StartDateDTP.Value =
MainWindow.startdateTB
                EndDateDTP.Value =
MainWindow.enddateTB
                ReasonTB.Text = MainWindow.reason
                CommentTB.Text =
MainWindow.Comment
                ApproveChB.Checked =
MainWindow.approve
                EmployeeNameCB.SelectedIndex =
EmployeeNameCB.FindStringExact(MainWindow.NameTB)
                LeaveTypeCB.SelectedIndex =
LeaveTypeCB.FindStringExact(MainWindow.leavetypeTB
)

            End If

        Catch ex As Exception

            MsgBox(ex.Message)
```

```
        End Try

    End Sub

    Private Sub CancelBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles CancelBtn.Click

        Me.Close()

    End Sub

    Private Sub
EmployeeNameCB_SelectedIndexChanged(ByVal sender
As System.Object, ByVal e As System.EventArgs)
Handles EmployeeNameCB.SelectedIndexChanged

        Try

            If (Not Me.EmployeeNameCB.SelectedItem
Is Nothing) Then

                Dim SItem =
TryCast(EmployeeNameCB.SelectedItem, DataRowView)
                Me.ENumberTB.Text =
SItem.Row("EmployeeNumber").ToString()

            End If

        Catch ex As Exception

            MsgBox(ex.Message)

        End Try

    End Sub

    Private Function CheckEmployeeLeaveCredit()

        SQL.RunQuery("SELECT * FROM
tLeaveRemaining WHERE ID_Employee = '" &
EmployeeNameCB.SelectedValue & "' AND ID_LeaveType
= '" & LeaveTypeCB.SelectedValue & "'")

        If SQL.SQLDS.Tables(0).Rows.Count > 0 Then

            Return True

        Else : Return False

        End If

    End Function

    Private Sub UpdateEmployeeLeaveCredit()

        Try

            Dim Duration, Remaining As Integer

            insertok = False

            Duration =
EndDateDTP.Value.Date.Subtract(StartDateDTP.Value.
Date).Days + 1

            'Get Remaining from the Employee Leave
Credit
            SQL.RunQuery("SELECT Remaining " & _
                         "FROM tLeaveRemaining " &
_
                         "WHERE " & _
                                "ID_Employee = '"
& EmployeeNameCB.SelectedValue & "' " & _
                                "AND ID_LeaveType
= '" & LeaveTypeCB.SelectedValue & "'")

            Remaining =
Convert.ToInt32(SQL.SQLDS.Tables(0).Rows(0).Item("
Remaining").ToString)

            'Update Remaining Leave of the
Employee
            SQL.RunQuery("UPDATE tLeaveRemaining "
& _
                         "SET " & _
                                "Remaining = '" &
(Remaining - Duration).ToString & "' " & _
                         "WHERE ID_Employee = '" &
EmployeeNameCB.SelectedValue & "' " & _
                         "AND ID_LeaveType = '" &
LeaveTypeCB.SelectedValue & "'")

            If (Remaining - Duration) > 0 Then

                insertok = True

            Else

                insertok = False

            End If

        Catch ex As Exception

            MsgBox(ex.Message,
MsgBoxStyle.Critical)

        End Try

    End Sub

    Private Sub SaveBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles SaveBtn.Click

        Try

            Dim DString As String

            DString = "SELECT * FROM
tFileLeaveOfAbsence WHERE " & _
                                "ID_Employee = '" &
EmployeeNameCB.SelectedValue & "' AND " & _
                                "DateFiling = '" &
FilingDateDTP.Value.Date & "' AND " & _
                                "StartDate = '" &
StartDateDTP.Value.Date & "' AND " & _
                                "EndDate = '" &
EndDateDTP.Value.Date & "' AND " & _
                                "ID_LeaveType = '" &
LeaveTypeCB.SelectedValue & "' AND " & _
                                "isApproved = '" &
ApproveChB.Checked & "' AND " & _
                                "Reason = '" &
ReasonTB.Text & "'"
```

```
            If Duplicate.IsDuplicated(DString) =
True Then

                MsgBox("Filed Leave of Absence for
the employee already exists!",
MsgBoxStyle.Critical)

                Exit Sub

            Else

                If MainWindow.HasRow = False Then

                    If CheckEmployeeLeaveCredit()
= True Then

                        QString = "INSERT INTO
tFileLeaveOfAbsence (ID_Employee, ID_LeaveType,
DateFiling, StartDate, EndDate, isApproved,
Reason, Comment) VALUES " & _
                                    "(" & _
                                    "'" &
EmployeeNameCB.SelectedValue & "', " & _
                                    "'" &
LeaveTypeCB.SelectedValue & "', " & _
                                    "'" &
FilingDateDTP.Value.Date & "', " & _
                                    "'" &
StartDateDTP.Value.Date & "', " & _
                                    "'" &
EndDateDTP.Value.Date & "', " & _
                                    "'" &
ApproveChB.Checked.ToString & "', " & _
                                    "'" &
ReasonTB.Text & "', " & _
                                    "'" &
CommentTB.Text & "'" & _
                                    ")"

                        AString = "Inserted a file
of leave of absence for " &
EmployeeNameCB.SelectedValue & " on " &
StartDateDTP.Value.Date & " until " &
EndDateDTP.Value.Date & "."



                    Else

                        MsgBox("Set Employee Leave
Credit first.", MsgBoxStyle.Critical)

                    End If


                ElseIf MainWindow.HasRow = True
Then

                    QString = "UPDATE
tFileLeaveOfAbsence SET " & _
                                    "ID_Employee =
'" & EmployeeNameCB.SelectedValue & "', " & _
                                    "ID_LeaveType
= '" & LeaveTypeCB.SelectedValue & "', " & _
                                    "DateFiling =
'" & FilingDateDTP.Value.Date & "', " & _
                                    "StartDate =
'" & StartDateDTP.Value.Date & "', " & _
                                    "EndDate = '"
& EndDateDTP.Value.Date & "', " & _
                                    "isApproved =
'" & ApproveChB.Checked.ToString & "', " & _
                                    "Reason = '" &
ReasonTB.Text & "', " & _
                                    "Comment = '"
& CommentTB.Text & "'"

                    AString = "Updated a file of
leave of absence for " &
EmployeeNameCB.SelectedValue & " on " &
StartDateDTP.Value.Date & " until " &
EndDateDTP.Value.Date & "."

                End If

                UpdateEmployeeLeaveCredit()

                If insertok Then

                    SQL.RunQuery(QString)

                    If
Duplicate.IsDuplicated(DString) = True Then

                        Trail.AddTrail(AString)

                        MsgBox("Saved!",
MsgBoxStyle.Information)

                    Else

                        MsgBox("Leave of Absence
not saved!", MsgBoxStyle.Information)

                    End If

                    Me.Close()

                    MainWindow.LoadDataGrid()

                Else

                    MsgBox("Leave Credit not
enough.", MsgBoxStyle.Information)

                    Exit Sub

                End If



            End If

        Catch ex As Exception

            MsgBox(ex.Message,
MsgBoxStyle.Critical)

        End Try

    End Sub

    Public Sub SelectFileLeaveOfAbsence()

        MainWindow.NewBtn.Visible = True
        MainWindow.EditBtn.Visible = True
        MainWindow.RefreshBtn.Visible = True
```

```
        MainWindow.DeleteBtn.Visible = True
        MainWindow.DateTimePicker1.Visible = False
        MainWindow.Label2.Visible = False

        MainWindow.GroupBox.Visible = True
        MainWindow.GroupBox.Text = "File Leave of
Absence"

        MainWindow.QString = "EXEC selectFLOA"

        MainWindow.LoadDataGrid()


MainWindow.DataGrid.Columns("ID").DisplayIndex = 0
        MainWindow.DataGrid.Columns("Date
Filing").DisplayIndex = 1
        MainWindow.DataGrid.Columns("Employee
Name").DisplayIndex = 2
        MainWindow.DataGrid.Columns("Employee
Number").DisplayIndex = 3
        MainWindow.DataGrid.Columns("Leave
Type").DisplayIndex = 4
        MainWindow.DataGrid.Columns("Start
Date").DisplayIndex = 5
        MainWindow.DataGrid.Columns("End
Date").DisplayIndex = 6

MainWindow.DataGrid.Columns("Duration").DisplayInd
ex = 7

MainWindow.DataGrid.Columns("Reason").DisplayIndex
= 8

MainWindow.DataGrid.Columns("Approved").DisplayInd
ex = 9

MainWindow.DataGrid.Columns("Comment").DisplayInde
x = 10

        Trail.AddTrail("Viewed File Leave of
Absence.")

    End Sub

    Public Sub DeleteFileLeaveOfAbsence()

        Dim result1 As DialogResult =
MessageBox.Show("Are you sure to delete " &
MainWindow.NameTB & "'s Filed Leave of Absence?",
_
                "Delete File Leave of Absence", _
                MessageBoxButtons.YesNo)

        If result1 =
Windows.Forms.DialogResult.Yes Then

            MainWindow.QString = "DELETE from
tFileLeaveOfAbsence WHERE ID = '" & MainWindow.ID
& "'"

            MainWindow.LoadDataGrid()

            Trail.AddTrail("Deleted " &
MainWindow.NameTB & "'s Filed Leave of Absence
from File Leave of Absence.")

            SelectFileLeaveOfAbsence()

        End If

    End Sub

End Class
Public Class FileChangeShiftSchedule

    Private Duplicate As New CheckDuplicates
    Private Trail As New AuditTrail
    Private SQL As New SQLControl
    Private QString As String 'Query String for
filing a change of shift schedule.
    Private AString As String 'Query String for
Audit Trail

    Private Sub DateSSDTP_ValueChanged(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles DateSSDTP.ValueChanged

        Dim day, qstring As String

        day = DateSSDTP.Value.DayOfWeek.ToString

        qstring = "SELECT " & _
                        "sun.Name AS Sunday, " & _
                        "mon.Name AS Monday, " & _
                        "tue.Name AS Tuesday, " &
_
                        "wed.Name AS Wednesday, "
& _
                        "thu.Name AS Thursday, " &
_
                        "fri.Name AS Friday, " & _
                        "sat.Name AS Saturday " &
_
                "FROM tEmployee " & _
                        "INNER JOIN
tWeeklySchedule AS week ON week.ID =
tEmployee.ID_WeeklySchedule " & _
                        "INNER JOIN tDailySchedule
AS sun ON sun.ID = week.ID_SundayDS " & _
                        "INNER JOIN tDailySchedule
AS mon ON mon.ID = week.ID_MondayDS " & _
                        "INNER JOIN tDailySchedule
AS tue ON tue.ID = week.ID_TuesdayDS " & _
                        "INNER JOIN tDailySchedule
AS wed ON wed.ID = week.ID_WednesdayDS " & _
                        "INNER JOIN tDailySchedule
AS thu ON thu.ID = week.ID_ThursdayDS " & _
                        "INNER JOIN tDailySchedule
AS fri ON fri.ID = week.ID_FridayDS " & _
                        "INNER JOIN tDailySchedule
AS sat ON sat.ID = week.ID_SaturdayDS " & _
                 "WHERE tEmployee.ID='" &
EmployeeNameCB.SelectedValue & "'"

        SQL.RunQuery(qstring)

        Me.OSchedTB.Text =
SQL.SQLDS.Tables(0).Rows(0).Item(day).ToString

    End Sub

    Private Sub FileChangeShiftSchedule_Load(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        SQL.RunQuery("SELECT ID, Name,
EmployeeNumber from tEmployee WHERE isActive='1'")
```

```
        EmployeeNameCB.DataSource =
SQL.SQLDS.Tables(0)
        EmployeeNameCB.ValueMember = "ID"
        EmployeeNameCB.DisplayMember = "Name"

        SQL.RunQuery("SELECT ID, Name from
tDailySchedule WHERE isActive='1'")

        ScheduleCB.DataSource =
SQL.SQLDS.Tables(0)
        ScheduleCB.ValueMember = "ID"
        ScheduleCB.DisplayMember = "Name"

        If MainWindow.HasRow = True Then

            DateSSDTP.Value = MainWindow.dateTB
            ReasonTB.Text = MainWindow.reason
            CommentTB.Text = MainWindow.Comment
            ApproveChB.Checked =
MainWindow.approve
            EmployeeNameCB.SelectedIndex =
EmployeeNameCB.FindStringExact(MainWindow.NameTB)
            ScheduleCB.SelectedIndex =
ScheduleCB.FindStringExact(MainWindow.newschedTB)

        End If

        Dim day, qstring As String

        day = DateSSDTP.Value.DayOfWeek.ToString

        qstring = "SELECT " & _
                        "sun.Name AS Sunday, " & _
                        "mon.Name AS Monday, " & _
                        "tue.Name AS Tuesday, " &
_
                        "wed.Name AS Wednesday, "
& _
                        "thu.Name AS Thursday, " &
_
                        "fri.Name AS Friday, " & _
                        "sat.Name AS Saturday " &
_
                  "FROM tEmployee " & _
                        "INNER JOIN
tWeeklySchedule AS week ON week.ID =
tEmployee.ID_WeeklySchedule " & _
                        "INNER JOIN tDailySchedule
AS sun ON sun.ID = week.ID_SundayDS " & _
                        "INNER JOIN tDailySchedule
AS mon ON mon.ID = week.ID_MondayDS " & _
                        "INNER JOIN tDailySchedule
AS tue ON tue.ID = week.ID_TuesdayDS " & _
                        "INNER JOIN tDailySchedule
AS wed ON wed.ID = week.ID_WednesdayDS " & _
                        "INNER JOIN tDailySchedule
AS thu ON thu.ID = week.ID_ThursdayDS " & _
                        "INNER JOIN tDailySchedule
AS fri ON fri.ID = week.ID_FridayDS " & _
                        "INNER JOIN tDailySchedule
AS sat ON sat.ID = week.ID_SaturdayDS " & _
                   "WHERE tEmployee.ID='" &
EmployeeNameCB.SelectedValue & "'"

        SQL.RunQuery(qstring)

        Me.OSchedTB.Text =
SQL.SQLDS.Tables(0).Rows(0).Item(day).ToString

    End Sub

    Private Sub
EmployeeNameCB_SelectedIndexChanged(ByVal sender
As System.Object, ByVal e As System.EventArgs)
Handles EmployeeNameCB.SelectedIndexChanged

        If (Not Me.EmployeeNameCB.SelectedItem Is
Nothing) Then

            Dim SItem =
TryCast(EmployeeNameCB.SelectedItem, DataRowView)
            Me.ENumberTB.Text =
SItem.Row("EmployeeNumber").ToString()

        End If

    End Sub

    Private Sub CancelBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles CancelBtn.Click

        Me.Close()

    End Sub

    Private Sub SaveBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles SaveBtn.Click

        Try

            Dim DString As String

            DString = "SELECT * FROM
tFileChangeShiftSchedule WHERE " & _
                              "ID_Employee = '" &
EmployeeNameCB.SelectedValue & "' AND " & _
                              "ID_DS = '" &
ScheduleCB.SelectedValue & "' AND " & _
                              "Date = '" &
DateSSDTP.Value.Date & "' AND " & _
                              "isApproved = '" &
ApproveChB.Checked & "' AND " & _
                              "Reason = '" &
ReasonTB.Text & "'"

            If Duplicate.IsDuplicated(DString) =
True Then

                MsgBox("Filed change of shift
schedule already exists for the employee.",
MsgBoxStyle.Critical)

                Exit Sub

            Else

                If MainWindow.HasRow = False Then

                    QString = "INSERT INTO
tFileChangeShiftSchedule (ID_Employee, ID_DS,
Date, isApproved, Reason, Comment) VALUES " & _
                                        "(" & _
                                        "'" &
EmployeeNameCB.SelectedValue & "', " & _
```

```
                                    "'" &
ScheduleCB.SelectedValue & "', " & _
                                    "'" &
DateSSDTP.Value.Date & "', " & _
                                    "'" &
ApproveChB.Checked & "', " & _
                                    "'" &
ReasonTB.Text & "', " & _
                                    "'" &
CommentTB.Text & "'" & _
                                    ")"

                    AString = "Inserted a file of
change of shift schedule for " &
EmployeeNameCB.SelectedValue & " on " &
DateSSDTP.Value.Date & "."

                ElseIf MainWindow.HasRow = True
Then

                    QString = "UPDATE
tFileChangeShiftSchedule SET " & _
                                    "ID_Employee =
'" & EmployeeNameCB.SelectedValue & "', " & _
                                    "ID_DS = '" &
ScheduleCB.SelectedValue & "', " & _
                                    "Date = '" &
DateSSDTP.Value.Date & "', " & _
                                    "isApproved =
'" & ApproveChB.Checked & "', " & _
                                    "Reason = '" &
ReasonTB.Text & "', " & _
                                    "Comment = '"
& CommentTB.Text & "' " & _
                               "WHERE ID = '" &
MainWindow.ID & "'"

                    AString = "Updated a file of
change of shift schedule for " &
EmployeeNameCB.SelectedValue & " on " &
DateSSDTP.Value.Date & "."

                End If

                SQL.RunQuery(QString)

                If Duplicate.IsDuplicated(DString)
= True Then

                    Trail.AddTrail(AString)

                    MsgBox("Saved!",
MsgBoxStyle.Information)

                    Me.Close()

                    MainWindow.LoadDataGrid()

                Else

                    MsgBox("File Change Schedule
not saved!", MsgBoxStyle.Information)

                End If

            End If

        Catch ex As Exception

            MsgBox(ex.Message,
MsgBoxStyle.Critical)

        End Try

    End Sub

    Public Sub SelectFileChangeShiftSchedule()

        MainWindow.NewBtn.Visible = True
        MainWindow.EditBtn.Visible = True
        MainWindow.RefreshBtn.Visible = True
        MainWindow.DeleteBtn.Visible = True
        MainWindow.DateTimePicker1.Visible = False
        MainWindow.Label2.Visible = False

        MainWindow.GroupBox.Visible = True
        MainWindow.GroupBox.Text = "File Change of
Shift Schedule"

        MainWindow.QString = "EXEC selectFCSS"

        MainWindow.LoadDataGrid()


MainWindow.DataGrid.Columns("ID").DisplayIndex = 0

MainWindow.DataGrid.Columns("Date").DisplayIndex =
1

MainWindow.DataGrid.Columns("Schedule").DisplayInd
ex = 2
        MainWindow.DataGrid.Columns("Employee
Name").DisplayIndex = 3
        MainWindow.DataGrid.Columns("Employee
Number").DisplayIndex = 4

MainWindow.DataGrid.Columns("Branch").DisplayIndex
= 5

MainWindow.DataGrid.Columns("Department").DisplayI
ndex = 6

MainWindow.DataGrid.Columns("Designation").Display
Index = 7
        MainWindow.DataGrid.Columns("Employee
Status").DisplayIndex = 8

MainWindow.DataGrid.Columns("Reason").DisplayIndex
= 9

MainWindow.DataGrid.Columns("Approved").DisplayInd
ex = 10

MainWindow.DataGrid.Columns("Comment").DisplayInde
x = 11

        Trail.AddTrail("Viewed File Change of
Shift Schedule.")

    End Sub

    Public Sub DeleteFileChangeShiftSchedule()

        Dim result1 As DialogResult =
MessageBox.Show("Are you sure to delete " &
MainWindow.NameTB & "'s Filed Change of Shift
Schedule?", _
```

```
                "Delete File Change of Shift
Schedule", _
                MessageBoxButtons.YesNo)

        If result1 =
Windows.Forms.DialogResult.Yes Then

            MainWindow.QString = "DELETE from
tFileChangeShiftSchedule WHERE ID = '" &
MainWindow.ID & "'"

            Trail.AddTrail("Deleted " &
MainWindow.NameTB & "'s Filed Change of Shift
Schedule from File Change of Shift Schedule.")

            MainWindow.LoadDataGrid()

            SelectFileChangeShiftSchedule()

        End If

    End Sub

End Class
Public Class FileChangeRestDay

    Private Duplicate As New CheckDuplicates
    Private Trail As New AuditTrail
    Private SQL As New SQLControl
    Private QString As String 'Query String for
filing a change in rest day.
    Private AString As String 'Query String for
Audit Trail

    Private Sub CancelBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles CancelBtn.Click

        Me.Close()

    End Sub

    Private Sub SaveBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles SaveBtn.Click

        Try

            Dim DString As String

            DString = "SELECT * FROM
tFileChangeRestDay WHERE " & _
                          "ID_Employee = '" &
EmployeeNameCB.SelectedValue & "' AND " & _
                          "Date = '" &
DateRDDTP.Value.Date & "' AND " & _
                          "isApproved = '" &
ApproveChB.Checked & "' AND " & _
                          "Reason = '" &
ReasonTB.Text & "'"

            If Duplicate.IsDuplicated(DString) =
True Then

                MsgBox("Filed Change of Rest Day
already exists for the employee.",
MsgBoxStyle.Critical)

                Exit Sub

            Else

                If MainWindow.HasRow = False Then

                    QString = "INSERT INTO
tFileChangeRestDay (ID_Employee, Date, isApproved,
Reason, Comment) VALUES " & _
                                        "(" & _
                                        "'" &
EmployeeNameCB.SelectedValue & "', " & _
                                        "'" &
DateRDDTP.Value.Date & "', " & _
                                        "'" &
ApproveChB.Checked & "', " & _
                                        "'" &
ReasonTB.Text & "', " & _
                                        "'" &
CommentTB.Text & "'" & _
                                        ")"

                    AString = "Inserted a file of
change of rest day for " &
EmployeeNameCB.SelectedValue & " on " &
DateRDDTP.Value.Date & "."

                ElseIf MainWindow.HasRow = True
Then

                    QString = "UPDATE
tFileChangeRestDay SET " & _
                                        "ID_Employee =
'" & EmployeeNameCB.SelectedValue & "', " & _
                                        "Date = '" &
DateRDDTP.Value.Date & "', " & _
                                        "isApproved =
'" & ApproveChB.Checked & "', " & _
                                        "Reason = '" &
ReasonTB.Text & "', " & _
                                        "Comment = '"
& CommentTB.Text & "' " & _
                                   "WHERE ID = '" &
MainWindow.ID & "'"

                    AString = "Updated a file of
change of rest day."

                End If

                SQL.RunQuery(QString)

                If Duplicate.IsDuplicated(DString)
= True Then

                    Trail.AddTrail(AString)

                    MsgBox("Saved!",
MsgBoxStyle.Information)

                    Me.Close()

                    MainWindow.LoadDataGrid()

                Else

                    MsgBox("File of Rest Day not
saved!", MsgBoxStyle.Information)

                End If
```

```
            End If

        Catch ex As Exception

            MsgBox(ex.Message,
MsgBoxStyle.Critical)

        End Try

    End Sub

    Private Sub FileChangeRestDay_Load(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        SQL.RunQuery("SELECT ID, Name,
EmployeeNumber FROM tEmployee WHERE isActive =
'1'")

        EmployeeNameCB.DataSource =
SQL.SQLDS.Tables(0)
        EmployeeNameCB.ValueMember = "ID"
        EmployeeNameCB.DisplayMember = "Name"

        If MainWindow.HasRow = True Then

            DateRDDTP.Value = MainWindow.dateTB
            ReasonTB.Text = MainWindow.reason
            CommentTB.Text = MainWindow.Comment
            ApproveChB.Checked =
MainWindow.approve
            EmployeeNameCB.SelectedIndex =
EmployeeNameCB.FindStringExact(MainWindow.NameTB)

        End If

    End Sub

    Private Sub
EmployeeNameCB_SelectedIndexChanged(ByVal sender
As System.Object, ByVal e As System.EventArgs)
Handles EmployeeNameCB.SelectedIndexChanged

        If (Not Me.EmployeeNameCB.SelectedItem Is
Nothing) Then

            Dim SItem =
TryCast(EmployeeNameCB.SelectedItem, DataRowView)
            Me.ENumberTB.Text =
SItem.Row("EmployeeNumber").ToString()

        End If

    End Sub

    Public Sub SelectFileChangeRestDay()

        MainWindow.NewBtn.Visible = True
        MainWindow.EditBtn.Visible = True
        MainWindow.RefreshBtn.Visible = True
        MainWindow.DeleteBtn.Visible = True
        MainWindow.DateTimePicker1.Visible = False
        MainWindow.Label2.Visible = False

        MainWindow.GroupBox.Visible = True
        MainWindow.GroupBox.Text = "File Change of
Rest Day"

        MainWindow.QString = "EXEC selectFCRD"

        MainWindow.LoadDataGrid()


MainWindow.DataGrid.Columns("ID").DisplayIndex = 0

MainWindow.DataGrid.Columns("Date").DisplayIndex =
1
        MainWindow.DataGrid.Columns("Employee
Name").DisplayIndex = 2
        MainWindow.DataGrid.Columns("Employee
Number").DisplayIndex = 3

MainWindow.DataGrid.Columns("Branch").DisplayIndex
= 4

MainWindow.DataGrid.Columns("Department").DisplayI
ndex = 5

MainWindow.DataGrid.Columns("Designation").Display
Index = 6
        MainWindow.DataGrid.Columns("Employee
Status").DisplayIndex = 7

MainWindow.DataGrid.Columns("Reason").DisplayIndex
= 8

MainWindow.DataGrid.Columns("Approved").DisplayInd
ex = 9

MainWindow.DataGrid.Columns("Comment").DisplayInde
x = 10

        Trail.AddTrail("Viewed File Change of Rest
Day.")

    End Sub

    Public Sub DeleteFileChangeRestDay()

        Dim result1 As DialogResult =
MessageBox.Show("Are you sure to delete " &
MainWindow.NameTB & "'s Filed Change of Rest
Day?", _
               "Delete File Change of Rest Day", _
               MessageBoxButtons.YesNo)

        If result1 =
Windows.Forms.DialogResult.Yes Then

            MainWindow.QString = "DELETE from
tFileChangeRestDay WHERE ID = '" & MainWindow.ID &
"'"

            MainWindow.LoadDataGrid()

            Trail.AddTrail("Deleted " &
MainWindow.NameTB & "'s Filed Change of Rest Day
from File Change of Rest Day.")

            SelectFileChangeRestDay()

        End If

    End Sub

End Class
```

```
Imports iTextSharp.text
Imports iTextSharp.text.pdf
Imports System.IO

Public Class PrintPDFPayslip

    Dim Payslip As New Document
    Private SQL As New SQLControl
    Private qstring As String
    Private dTable, dTable2 As New DataTable
    Private printString As String
    Dim result1, result2 As Integer
    Dim bfTimes As BaseFont
    Dim times, times2, times3 As Font
    Dim pdfWrite As PdfWriter
    Dim sDate, eDate, s, e, nameD2, text, path As
String
    Dim n, x As Integer

    Public Sub PDFPayslip()

        Try

            dTable.Clear()
            dTable2.Clear()

            SetDataTable()

            SetDataTable2()

            CreatePDF()

            SetFont()

            Payslip.Open()

            '-----Create PDF Table-----


            '-----Size and Position of PDF Table--
---



            'MsgBox(dTable.Rows.Count)


            '-----Table Contents-----
            For i As Integer = 0 To
dTable2.Rows.Count - 1

                Dim pTable1 As PdfPTable
                pTable1 = New PdfPTable(6)
                pTable1.TotalWidth = 100%
                pTable1.WidthPercentage = 100
                pTable1.TotalWidth =
Payslip.PageSize.Width - Payslip.LeftMargin -
Payslip.RightMargin

                pTable1.WriteSelectedRows(0, -1,
Payslip.LeftMargin, Payslip.RightMargin,
pdfWrite.DirectContent)

                Dim cell As PdfPCell
                Dim text As String

                cell = New PdfPCell(New
Phrase("PAYSLIP", times3))
                cell.Colspan = 6
                cell.BorderWidthBottom = 2
                cell.BorderWidthLeft = 2
                cell.BorderWidthRight = 2
                cell.BorderWidthTop = 2
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                pTable1.AddCell(cell)

                cell = New PdfPCell(New
Phrase("EMPLOYEE #", times2))
                cell.Colspan = 1
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 2
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                text =
dTable2.Rows(i).Item("ENUMBER")

                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 2
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                cell = New PdfPCell(New
Phrase("BASIC PAY", times2))
                cell.Colspan = 1
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                text =
dTable2.Rows(i).Item("BASIC")
                cell = New PdfPCell(New
Phrase(text, times2))
                cell.Colspan = 2
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 2
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                text = "EMPLOYEE NAME"
                cell = New PdfPCell(New
Phrase(text, times2))
```

```
                cell.Colspan = 1
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 2
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                text =
dTable2.Rows(i).Item("NAME")
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 2
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                text = "BRANCH - DEPT - DESIG"
                cell = New PdfPCell(New
Phrase(text, times2))
                cell.Colspan = 1
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                text =
AttendanceInput.BranchCB.SelectedItem("Name") & "
- " & AttendanceInput.DeptCB.SelectedItem("Name")
& " - " &
AttendanceInput.DesigCB.SelectedItem("Name")
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 2
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 2
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                text = "PAYROLL PERIOD"
                cell = New PdfPCell(New
Phrase(text, times2))
                cell.Colspan = 1
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 2
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                text =
AttendanceInput.StartDTP.Value.Date & " to " &
AttendanceInput.EndDTP.Value.Date
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 2
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                text = "NET PAY"
                cell = New PdfPCell(New
Phrase(text, times2))
                cell.Colspan = 1
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                Dim NETPAY As Object
                NETPAY =
dTable.Compute("Sum(NETPAY)", "NAME = '" &
dTable2.Rows(i).Item("NAME") & "'")

                cell = New PdfPCell(New
Phrase(NETPAY, times))
                cell.Colspan = 2
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 2
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                text = "EARNINGS"
                cell = New PdfPCell(New
Phrase(text, times2))
                cell.Colspan = 3
                cell.BorderWidthBottom = 2
                cell.BorderWidthLeft = 2
                cell.BorderWidthRight = 2
                cell.BorderWidthTop = 2
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                pTable1.AddCell(cell)

                text = "DEDUCTIONS"
                cell = New PdfPCell(New
Phrase(text, times2))
```

```
                cell.Colspan = 3
                cell.BorderWidthBottom = 2
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 2
                cell.BorderWidthTop = 2
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                pTable1.AddCell(cell)

                text = "REGULAR"
                cell = New PdfPCell(New
Phrase(text, times2))
                cell.Colspan = 1
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 2
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                Dim REGULAR As Object
                REGULAR =
dTable.Compute("Sum(REGULAR)", "NAME = '" &
dTable2.Rows(i).Item("NAME") & "'")
                cell = New PdfPCell(New
Phrase(REGULAR, times))
                cell.Colspan = 2
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                text = "SSS PREMIUM"
                cell = New PdfPCell(New
Phrase(text, times2))
                cell.Colspan = 1
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                Dim SSS As Object
                SSS = dTable.Compute("Sum(SSS)",
"NAME = '" & dTable2.Rows(i).Item("NAME") & "'")

                cell = New PdfPCell(New
Phrase(SSS, times))
                cell.Colspan = 2
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 2
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                text = "COLA/CPTA"
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 2
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                pTable1.AddCell(cell)


                Dim CPTA As Object
                CPTA = dTable.Compute("Sum(CPTA)",
"NAME = '" & dTable2.Rows(i).Item("NAME") & "'")

                cell = New PdfPCell(New
Phrase(CPTA, times))
                cell.Colspan = 2
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                text = "MEDICARE"
                cell = New PdfPCell(New
Phrase(text, times2))
                cell.Colspan = 1
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)


                Dim MEDICARE As Object
                MEDICARE =
dTable.Compute("Sum(MEDICARE)", "NAME = '" &
dTable2.Rows(i).Item("NAME") & "'")

                cell = New PdfPCell(New
Phrase(MEDICARE, times))
                cell.Colspan = 2
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 2
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)
```

```
                text = "OVERTIME"
                cell = New PdfPCell(New
Phrase(text, times2))
                cell.Colspan = 1
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 2
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                text = " "
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 2
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                text = "HDMF CONT."
                cell = New PdfPCell(New
Phrase(text, times2))
                cell.Colspan = 1
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                Dim HDMF As Object
                HDMF = dTable.Compute("Sum(HDMF)",
"NAME = '" & dTable2.Rows(i).Item("NAME") & "'")

                cell = New PdfPCell(New
Phrase(HDMF, times))
                cell.Colspan = 2
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 2
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                text = "REGULAR"
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 2
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                pTable1.AddCell(cell)

                Dim OT As Object
                OT = dTable.Compute("Sum(OT)",
"NAME = '" & dTable2.Rows(i).Item("NAME") & "'")

                cell = New PdfPCell(New Phrase(OT,
times))
                cell.Colspan = 2
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                text = " "
                cell = New PdfPCell(New
Phrase(text, times2))
                cell.Colspan = 1
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                text = " "
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 2
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 2
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                text = "REST DAY"
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 2
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                pTable1.AddCell(cell)

                Dim WRD As Object
                WRD = dTable.Compute("Sum(WRD)",
"NAME = '" & dTable2.Rows(i).Item("NAME") & "'")

                cell = New PdfPCell(New
Phrase(WRD, times))
```

```
                cell.Colspan = 2
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                text = " "
                cell = New PdfPCell(New
Phrase(text, times2))
                cell.Colspan = 1
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                text = " "
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 2
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 2
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                text = "SPECIAL"
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 2
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                pTable1.AddCell(cell)

                Dim SH As Object
                SH = dTable.Compute("Sum(SH)",
"NAME = '" & dTable2.Rows(i).Item("NAME") & "'")

                cell = New PdfPCell(New Phrase(SH,
times))
                cell.Colspan = 2
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                text = " "
                cell = New PdfPCell(New
Phrase(text, times2))
                cell.Colspan = 1
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                text = " "
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 2
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 2
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                text = "SPECIAL REST"
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 2
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                pTable1.AddCell(cell)

                Dim WSR As Object
                WSR = dTable.Compute("Sum(WSR)",
"NAME = '" & dTable2.Rows(i).Item("NAME") & "'")

                cell = New PdfPCell(New
Phrase(WSR, times))
                cell.Colspan = 2
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                text = " "
                cell = New PdfPCell(New
Phrase(text, times2))
                cell.Colspan = 1
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
```

```
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                text = " "
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 2
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 2
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                text = "LEGAL"
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 2
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                pTable1.AddCell(cell)

                Dim WRH As Object
                WRH = dTable.Compute("Sum(WRH)",
"NAME = '" & dTable2.Rows(i).Item("NAME") & "'")

                cell = New PdfPCell(New
Phrase(WRH, times))
                cell.Colspan = 2
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                text = " "
                cell = New PdfPCell(New
Phrase(text, times2))
                cell.Colspan = 1
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                text = " "
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 2
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 2
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                text = "LEGAL REST"
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 2
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                pTable1.AddCell(cell)

                Dim WRR As Object
                WRR = dTable.Compute("Sum(WRR)",
"NAME = '" & dTable2.Rows(i).Item("NAME") & "'")

                cell = New PdfPCell(New
Phrase(WRR, times))
                cell.Colspan = 2
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                text = " "
                cell = New PdfPCell(New
Phrase(text, times2))
                cell.Colspan = 1
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                text = " "
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 2
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 2
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)
```

```
                text = "ND"
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 2
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                pTable1.AddCell(cell)

                Dim ND As Object
                ND = dTable.Compute("Sum(ND)",
"NAME = '" & dTable2.Rows(i).Item("NAME") & "'")

                cell = New PdfPCell(New Phrase(ND,
times))
                cell.Colspan = 2
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                text = " "
                cell = New PdfPCell(New
Phrase(text, times2))
                cell.Colspan = 1
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                text = " "
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 2
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 2
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                text = "TOTAL EARNINGS"
                cell = New PdfPCell(New
Phrase(text, times2))
                cell.Colspan = 1
                cell.BorderWidthBottom = 2
                cell.BorderWidthLeft = 2
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 2
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                pTable1.AddCell(cell)

                Dim TOTAL As Object
                TOTAL =
dTable.Compute("Sum(TOTAL)", "NAME = '" &
dTable2.Rows(i).Item("NAME") & "'")

                cell = New PdfPCell(New
Phrase(TOTAL, times2))
                cell.Colspan = 2
                cell.BorderWidthBottom = 2
                cell.BorderWidthLeft = 1
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 2
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                text = "TOTAL DEDUCTIONS"
                cell = New PdfPCell(New
Phrase(text, times2))
                cell.Colspan = 1
                cell.BorderWidthBottom = 2
                cell.BorderWidthLeft = 1
                cell.BorderWidthRight = 1
                cell.BorderWidthTop = 2
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                pTable1.AddCell(cell)

                Dim TOTALDEDUCTIONS As Object
                TOTALDEDUCTIONS =
dTable.Compute("Sum(TOTALDEDUCTIONS)", "NAME = '"
& dTable2.Rows(i).Item("NAME") & "'")

                cell = New PdfPCell(New
Phrase(TOTALDEDUCTIONS, times2))
                cell.Colspan = 2
                cell.BorderWidthBottom = 2
                cell.BorderWidthLeft = 1
                cell.BorderWidthRight = 2
                cell.BorderWidthTop = 2
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                pTable1.AddCell(cell)

                text = " "
                cell = New PdfPCell(New
Phrase(text, times2))
                cell.Colspan = 6
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                pTable1.AddCell(cell)

                text = " "
```

```
                cell = New PdfPCell(New
Phrase(text, times2))
                cell.Colspan = 6
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                cell.BorderWidthTop = 0
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                pTable1.AddCell(cell)


                Payslip.Add(pTable1)
                Payslip.NewPage()

            Next



            Payslip.Close()

            '-----Opens the PDF-----
            System.Diagnostics.Process.Start(path)

        Catch ex As Exception

            MsgBox(ex.Message)

        End Try

    End Sub

    Private Sub SetDataTable()

        If SQL.SQLDS IsNot Nothing Then

            SQL.SQLDS.Clear()

        End If

        qstring = "EXEC PrintPayslip @Branch = '"
& AttendanceInput.BranchCB.SelectedValue & "',
@Dept = '" & AttendanceInput.DeptCB.SelectedValue
& "', @Desig = '" &
AttendanceInput.DesigCB.SelectedValue & "', @SDate
= '" & AttendanceInput.StartDTP.Value.Date & "',
@EDate = '" & AttendanceInput.EndDTP.Value.Date &
"'"

        SQL.RunQuery(qstring)

        SQL.SQLDA.Fill(dTable)

    End Sub

    Private Sub SetDataTable2()

        If SQL.SQLDS IsNot Nothing Then

            SQL.SQLDS.Clear()

        End If

        qstring = "SELECT " & _
                        "EMP.NAME AS NAME, " & _
                        "EMP.EMPLOYEENUMBER AS
ENUMBER, " & _
                        "TC.BASICPAY AS BASIC, " &
_
                        "TC.COLA AS COLA, " & _
                        "COUNT(*) AS COUNTROW " &
_
                    "FROM TCOMPENSATIONLIST AS TC "
& _
                        "INNER JOIN TEMPLOYEE AS
EMP ON EMP.ID = TC.ID_EMPLOYEE " & _
                        "INNER JOIN
TADJUSTMENTPOLICY AS ADJ ON ADJ.ID =
TC.ID_ADJPOLICY " & _
                    "WHERE " & _
                        "EMP.ID_BRANCH = '" &
AttendanceInput.BranchCB.SelectedValue & "' AND "
& _
                        "EMP.ID_DEPARTMENT = '" &
AttendanceInput.DeptCB.SelectedValue & "' AND " &
_
                        "EMP.ID_DESIGNATION = '" &
AttendanceInput.DesigCB.SelectedValue & "' AND " &
_
                        "TC.DATE BETWEEN '" &
AttendanceInput.StartDTP.Value.Date & "' AND '" &
AttendanceInput.EndDTP.Value.Date & "' " & _
                    "GROUP BY EMP.NAME,
EMP.EMPLOYEENUMBER, TC.BASICPAY, TC.COLA"

        SQL.RunQuery(qstring)

        SQL.SQLDA.Fill(dTable2)

    End Sub

    Private Sub CreatePDF()

        '-----Sets Dates for Filename-----
        sDate =
AttendanceInput.StartDTP.Value.Date
        s = sDate.Replace("/", "")

        eDate = AttendanceInput.EndDTP.Value.Date
        e = eDate.Replace("/", "")

        '-----Creating Pdf-----
        Dim filePath As Object
        Dim defaultPath As String
        defaultPath =
Application.StartupPath.ToString & "\Payslip" & s
& "-" & e &
AttendanceInput.BranchCB.SelectedItem("Name") &
".pdf"
        filePath = InputBox("Specify file path.",
"Print Payslip", defaultPath)
        path = filePath

        pdfWrite = PdfWriter.GetInstance(Payslip,
New FileStream(path, FileMode.Create))

    End Sub

    Private Sub SetFont()

        '-----Set FONT-----
        bfTimes =
BaseFont.CreateFont(BaseFont.COURIER,
BaseFont.CP1252, False)
        times = New Font(bfTimes, 12)
        times2 = New Font(bfTimes, 12, Font.BOLD)
```

```
        times3 = New Font(bfTimes, 14, Font.BOLD)

    End Sub

End Class

Imports iTextSharp.text
Imports iTextSharp.text.pdf
Imports System.IO

Public Class PrintMonthlyOvertime

    Private SQL As New SQLControl
    Private dTable As New DataTable
    Dim qstring As String
    Dim datey As DateTime



    Private Sub SetTable()

        If SQL.SQLDS IsNot Nothing Then

            SQL.SQLDS.Clear()

        End If

        qstring = "SELECT " & _
                        "tFileOvertime.ID AS ID, "
& _
                        "tFileOvertime.Date, " & _
                        "Employee.Name AS
[Employee Name], " & _
                        "ENumber.EmployeeNumber AS
[Employee Number], " & _
                        "Branch.Name AS Branch, "
& _
                        "Dept.Name AS Department,
" & _
                        "Desig.Name AS
Designation, " & _
                        "tFileOvertime.Reason as
Reason " & _
                  "FROM tFileOvertime " & _
                        "INNER JOIN tEmployee AS
Employee ON Employee.ID =
tFileOvertime.ID_Employee " & _
                        "INNER JOIN tEmployee AS
ENumber ON ENumber.ID = tFileOvertime.ID_Employee
" & _
                        "INNER JOIN tBranch AS
Branch ON Branch.ID = Employee.ID_Branch " & _
                        "INNER JOIN tDepartment AS
Dept ON Dept.ID = Employee.ID_Department " & _
                        "INNER JOIN tDesignation
AS Desig ON Desig.ID = Employee.ID_Designation " &
_
                        "INNER JOIN
tEmployeeStatus AS EStatus ON EStatus.ID =
Employee.ID_EmployeeStatus " & _
                  "WHERE " & _
                        "tFileOverTime.Date
BETWEEN '" & datey & "' AND '" &
datey.Date.AddMonths(1).AddDays(-1) & "'" & _
                  "ORDER BY Branch.Name,
Dept.Name, Desig.Name"

        SQL.RunQuery(qstring)

        SQL.SQLDA.Fill(dTable)

    End Sub


    Public Sub PrintMOvertime()

        Try

            datey =
DateTime.Parse(MainWindow.DateTimePicker1.Value.Da
te.Month & " 1, " &
MainWindow.DateTimePicker1.Value.Date.Year)

            Dim sDate, s As String

            sDate =
MainWindow.DateTimePicker1.Value.Date
            s = sDate.Replace("/", "")

            dTable.Clear()

            SetTable()

            '-----Create PDF-----
            Dim pdfDAbsent As New Document
            Dim filePath As Object
            Dim defaultPath As String
            defaultPath =
Application.StartupPath.ToString &
"\MonthlyOvertime" & s & ".pdf"
            filePath = InputBox("Specify file
path.", "Print Monthly Overtime", defaultPath)
            Dim path As String
            path = filePath
            Dim pdfWrite As PdfWriter =
PdfWriter.GetInstance(pdfDAbsent, New
FileStream(path, FileMode.Create))

            '-----Open PDF-----
            pdfDAbsent.Open()

            '-----Set FONT-----
            Dim bfTimes As BaseFont
            bfTimes =
BaseFont.CreateFont(BaseFont.COURIER,
BaseFont.CP1252, False)
            Dim times, times2 As Font
            times = New Font(bfTimes, 8)
            times2 = New Font(bfTimes, 8,
Font.BOLD)

            '-----Report Header-----
            pdfDAbsent.Add(New Paragraph("Monthly
Overtime", times))
            pdfDAbsent.Add(New Paragraph("Date: "
& datey & " to " &
datey.Date.AddMonths(1).AddDays(-1), times))
            pdfDAbsent.Add(New Paragraph(" "))

            '-----Create PDF Table-----
            Dim pTable1 As PdfPTable

            '-----Size and Position of PDF Table--
---
            pTable1 = New PdfPTable(14)
            pTable1.TotalWidth = 100%
            pTable1.WidthPercentage = 100
```

```
            pTable1.TotalWidth =
pdfDAbsent.PageSize.Width - pdfDAbsent.LeftMargin
- pdfDAbsent.RightMargin

            pTable1.WriteSelectedRows(0, -1,
pdfDAbsent.LeftMargin, pdfDAbsent.RightMargin,
pdfWrite.DirectContent)

            Dim cell, cell2, head2 As PdfPCell

            '-----TABLE HEADER------
            cell = New PdfPCell(New Phrase("ID",
times2))
            cell.Colspan = 1
            cell.HorizontalAlignment = 1
            cell.VerticalAlignment =
Element.ALIGN_MIDDLE
            cell.BorderWidthLeft = 0
            cell.BorderWidthRight = 0
            cell.BorderWidthBottom = 1
            cell.BorderWidthTop = 1
            pTable1.AddCell(cell)

            cell = New PdfPCell(New Phrase("DATE",
times2))
            cell.Colspan = 1
            cell.HorizontalAlignment = 1
            cell.VerticalAlignment =
Element.ALIGN_MIDDLE
            cell.BorderWidthLeft = 0
            cell.BorderWidthRight = 0
            cell.BorderWidthBottom = 1
            cell.BorderWidthTop = 1
            pTable1.AddCell(cell)

            head2 = New PdfPCell(New
Phrase("EMPLOYEE NAME", times2))
            head2.Colspan = 2
            head2.HorizontalAlignment = 1
            head2.VerticalAlignment =
Element.ALIGN_MIDDLE
            head2.BorderWidthRight = 0
            head2.BorderWidthLeft = 0
            head2.BorderWidthTop = 1
            head2.BorderWidthBottom = 1
            pTable1.AddCell(head2)

            head2 = New PdfPCell(New
Phrase("EMPLOYEE NUMBER", times2))
            head2.Colspan = 2
            head2.HorizontalAlignment = 1
            head2.VerticalAlignment =
Element.ALIGN_MIDDLE
            head2.BorderWidthRight = 0
            head2.BorderWidthLeft = 0
            head2.BorderWidthTop = 1
            head2.BorderWidthBottom = 1
            pTable1.AddCell(head2)

            head2 = New PdfPCell(New
Phrase("BRANCH", times2))
            head2.Colspan = 2
            head2.HorizontalAlignment = 1
            head2.VerticalAlignment =
Element.ALIGN_MIDDLE
            head2.BorderWidthRight = 0
            head2.BorderWidthLeft = 0
            head2.BorderWidthTop = 1
            head2.BorderWidthBottom = 1
            pTable1.AddCell(head2)

            head2 = New PdfPCell(New
Phrase("DEPARTMENT", times2))
            head2.Colspan = 2
            head2.HorizontalAlignment = 1
            head2.VerticalAlignment =
Element.ALIGN_MIDDLE
            head2.BorderWidthRight = 0
            head2.BorderWidthLeft = 0
            head2.BorderWidthTop = 1
            head2.BorderWidthBottom = 1
            pTable1.AddCell(head2)

            head2 = New PdfPCell(New
Phrase("DESIGNATION", times2))
            head2.Colspan = 2
            head2.HorizontalAlignment = 1
            head2.VerticalAlignment =
Element.ALIGN_MIDDLE
            head2.BorderWidthRight = 0
            head2.BorderWidthLeft = 0
            head2.BorderWidthTop = 1
            head2.BorderWidthBottom = 1
            pTable1.AddCell(head2)

            head2 = New PdfPCell(New
Phrase("REASON", times2))
            head2.Colspan = 2
            head2.HorizontalAlignment = 1
            head2.VerticalAlignment =
Element.ALIGN_MIDDLE
            head2.BorderWidthRight = 0
            head2.BorderWidthLeft = 0
            head2.BorderWidthTop = 1
            head2.BorderWidthBottom = 1
            pTable1.AddCell(head2)

            For i As Integer = 0 To
dTable.Rows.Count - 1

                Dim text As String
                Dim datetext As DateTime

                text = dTable.Rows(i).Item("ID")
                cell2 = New PdfPCell(New
Phrase(text, times))
                cell2.Colspan = 1
                cell2.HorizontalAlignment = 1
                cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell2.BorderWidthLeft = 0
                cell2.BorderWidthRight = 0
                cell2.BorderWidthBottom = 0.5
                cell2.BorderWidthTop = 0
                pTable1.AddCell(cell2)

                text = dTable.Rows(i).Item("Date")
                datetext = DateTime.Parse(text)
                text = datetext.ToString("MMM dd")
                cell2 = New PdfPCell(New
Phrase(text, times))
                cell2.Colspan = 1
                cell2.HorizontalAlignment = 1
                cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell2.BorderWidthLeft = 0
                cell2.BorderWidthRight = 0
                cell2.BorderWidthBottom = 0.5
```

```
                cell2.BorderWidthTop = 0
                pTable1.AddCell(cell2)

                text =
dTable.Rows(i).Item("Employee Name")
                cell2 = New PdfPCell(New
Phrase(text, times))
                cell2.Colspan = 2
                cell2.HorizontalAlignment = 1
                cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell2.BorderWidthLeft = 0
                cell2.BorderWidthRight = 0
                cell2.BorderWidthBottom = 0.5
                cell2.BorderWidthTop = 0
                pTable1.AddCell(cell2)

                text =
dTable.Rows(i).Item("Employee Number")
                cell2 = New PdfPCell(New
Phrase(text, times))
                cell2.Colspan = 2
                cell2.HorizontalAlignment = 1
                cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell2.BorderWidthLeft = 0
                cell2.BorderWidthRight = 0
                cell2.BorderWidthBottom = 0.5
                cell2.BorderWidthTop = 0
                pTable1.AddCell(cell2)

                text =
dTable.Rows(i).Item("Branch")
                cell2 = New PdfPCell(New
Phrase(text, times))
                cell2.Colspan = 2
                cell2.HorizontalAlignment = 1
                cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell2.BorderWidthLeft = 0
                cell2.BorderWidthRight = 0
                cell2.BorderWidthBottom = 0.5
                cell2.BorderWidthTop = 0
                pTable1.AddCell(cell2)

                text =
dTable.Rows(i).Item("Department")
                cell2 = New PdfPCell(New
Phrase(text, times))
                cell2.Colspan = 2
                cell2.HorizontalAlignment = 1
                cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell2.BorderWidthLeft = 0
                cell2.BorderWidthRight = 0
                cell2.BorderWidthBottom = 0.5
                cell2.BorderWidthTop = 0
                pTable1.AddCell(cell2)

                text =
dTable.Rows(i).Item("Designation")
                cell2 = New PdfPCell(New
Phrase(text, times))
                cell2.Colspan = 2
                cell2.HorizontalAlignment = 1
                cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell2.BorderWidthLeft = 0
                cell2.BorderWidthRight = 0
                cell2.BorderWidthBottom = 0.5
                cell2.BorderWidthTop = 0
                pTable1.AddCell(cell2)

                text =
dTable.Rows(i).Item("Reason")
                cell2 = New PdfPCell(New
Phrase(text, times))
                cell2.Colspan = 2
                cell2.HorizontalAlignment = 1
                cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell2.BorderWidthLeft = 0
                cell2.BorderWidthRight = 0
                cell2.BorderWidthBottom = 0.5
                cell2.BorderWidthTop = 0
                pTable1.AddCell(cell2)

            Next

            pdfDAbsent.Add(pTable1)

            pdfDAbsent.NewPage()
            pdfDAbsent.Close()

            'opens the file
            System.Diagnostics.Process.Start(path)

        Catch ex As Exception

            MsgBox(ex.Message,
MsgBoxStyle.Information)

        End Try

    End Sub

End Class

Imports iTextSharp.text
Imports iTextSharp.text.pdf
Imports System.IO

Public Class PrintMonthlyAbsenteeism

    Private SQL As New SQLControl
    Private dTable As New DataTable
    Dim qstring As String
    Dim datey As DateTime



    Private Sub SetTable()

        If SQL.SQLDS IsNot Nothing Then

            SQL.SQLDS.Clear()

        End If

        qstring = "SELECT " & _
                        "tAttendanceComputation.ID
AS ID, " & _

"tAttendanceComputation.Date AS Date, " & _
                        "Employee.Name AS
[Employee Name], " & _
                        "ENumber.EmployeeNumber AS
[Employee Number], " & _
```

```
                        "Branch.Name AS Branch, "
& _
                        "Dept.Name AS Department,
" & _
                        "Desig.Name AS Designation
" & _
                    "FROM tAttendanceComputation " &
_
                        "INNER JOIN tEmployee AS
Employee ON Employee.ID =
tAttendanceComputation.ID_Employee " & _
                        "INNER JOIN tEmployee AS
ENumber ON ENumber.ID =
tAttendanceComputation.ID_Employee " & _
                        "INNER JOIN tBranch AS
Branch ON Branch.ID = Employee.ID_Branch " & _
                        "INNER JOIN tDepartment AS
Dept ON Dept.ID = Employee.ID_Department " & _
                        "INNER JOIN tDesignation
AS Desig ON Desig.ID = Employee.ID_Designation " &
_
                        "INNER JOIN
tEmployeeStatus AS EStatus ON EStatus.ID =
Employee.ID_EmployeeStatus " & _
                    "WHERE " & _

"tAttendanceComputation.Absent = 1 AND " & _

"tAttendanceComputation.Date BETWEEN '" & datey &
"' AND '" & datey.Date.AddMonths(1).AddDays(-1) &
"'" & _
                    "ORDER BY Branch.Name,
Dept.Name, Desig.Name"

        SQL.RunQuery(qstring)

        SQL.SQLDA.Fill(dTable)

    End Sub


    Public Sub PrintMAbsenteeism()

        Try

            datey =
DateTime.Parse(MainWindow.DateTimePicker1.Value.Da
te.Month & " 1, " &
MainWindow.DateTimePicker1.Value.Date.Year)

            Dim sDate, s As String

            sDate =
MainWindow.DateTimePicker1.Value.Date
            s = sDate.Replace("/", "")

            dTable.Clear()

            SetTable()

            '-----Create PDF-----
            Dim pdfDAbsent As New Document
            Dim filePath As Object
            Dim defaultPath As String
            defaultPath =
Application.StartupPath.ToString &
"\MonthlyAbsenteeism" & s & ".pdf"
            filePath = InputBox("Specify file
path.", "Print Monthly Absenteeism", defaultPath)
            Dim path As String
            path = filePath
            Dim pdfWrite As PdfWriter =
PdfWriter.GetInstance(pdfDAbsent, New
FileStream(path, FileMode.Create))

            '-----Open PDF-----
            pdfDAbsent.Open()

            '-----Set FONT-----
            Dim bfTimes As BaseFont
            bfTimes =
BaseFont.CreateFont(BaseFont.COURIER,
BaseFont.CP1252, False)
            Dim times, times2 As Font
            times = New Font(bfTimes, 8)
            times2 = New Font(bfTimes, 8,
Font.BOLD)

            '-----Report Header-----
            pdfDAbsent.Add(New Paragraph("Monthly
Absenteeism", times))
            pdfDAbsent.Add(New Paragraph("Date: "
& datey & " to " &
datey.Date.AddMonths(1).AddDays(-1), times))
            pdfDAbsent.Add(New Paragraph(" "))

            '-----Create PDF Table-----
            Dim pTable1 As PdfPTable

            '-----Size and Position of PDF Table--
---
            pTable1 = New PdfPTable(12)
            pTable1.TotalWidth = 100%
            pTable1.WidthPercentage = 100
            pTable1.TotalWidth =
pdfDAbsent.PageSize.Width - pdfDAbsent.LeftMargin
- pdfDAbsent.RightMargin

            pTable1.WriteSelectedRows(0, -1,
pdfDAbsent.LeftMargin, pdfDAbsent.RightMargin,
pdfWrite.DirectContent)

            Dim cell, cell2, head2 As PdfPCell

            '-----TABLE HEADER------
            cell = New PdfPCell(New Phrase("ID",
times2))
            cell.Colspan = 1
            cell.HorizontalAlignment = 1
            cell.VerticalAlignment =
Element.ALIGN_MIDDLE
            cell.BorderWidthLeft = 0
            cell.BorderWidthRight = 0
            cell.BorderWidthBottom = 1
            cell.BorderWidthTop = 1
            pTable1.AddCell(cell)

            cell = New PdfPCell(New Phrase("DATE",
times2))
            cell.Colspan = 1
            cell.HorizontalAlignment = 1
            cell.VerticalAlignment =
Element.ALIGN_MIDDLE
            cell.BorderWidthLeft = 0
            cell.BorderWidthRight = 0
            cell.BorderWidthBottom = 1
            cell.BorderWidthTop = 1
            pTable1.AddCell(cell)
```

```
            head2 = New PdfPCell(New
Phrase("EMPLOYEE NAME", times2))
            head2.Colspan = 2
            head2.HorizontalAlignment = 1
            head2.VerticalAlignment =
Element.ALIGN_MIDDLE
            head2.BorderWidthRight = 0
            head2.BorderWidthLeft = 0
            head2.BorderWidthTop = 1
            head2.BorderWidthBottom = 1
            pTable1.AddCell(head2)

            head2 = New PdfPCell(New
Phrase("EMPLOYEE NUMBER", times2))
            head2.Colspan = 2
            head2.HorizontalAlignment = 1
            head2.VerticalAlignment =
Element.ALIGN_MIDDLE
            head2.BorderWidthRight = 0
            head2.BorderWidthLeft = 0
            head2.BorderWidthTop = 1
            head2.BorderWidthBottom = 1
            pTable1.AddCell(head2)

            head2 = New PdfPCell(New
Phrase("BRANCH", times2))
            head2.Colspan = 2
            head2.HorizontalAlignment = 1
            head2.VerticalAlignment =
Element.ALIGN_MIDDLE
            head2.BorderWidthRight = 0
            head2.BorderWidthLeft = 0
            head2.BorderWidthTop = 1
            head2.BorderWidthBottom = 1
            pTable1.AddCell(head2)

            head2 = New PdfPCell(New
Phrase("DEPARTMENT", times2))
            head2.Colspan = 2
            head2.HorizontalAlignment = 1
            head2.VerticalAlignment =
Element.ALIGN_MIDDLE
            head2.BorderWidthRight = 0
            head2.BorderWidthLeft = 0
            head2.BorderWidthTop = 1
            head2.BorderWidthBottom = 1
            pTable1.AddCell(head2)

            head2 = New PdfPCell(New
Phrase("DESIGNATION", times2))
            head2.Colspan = 2
            head2.HorizontalAlignment = 1
            head2.VerticalAlignment =
Element.ALIGN_MIDDLE
            head2.BorderWidthRight = 0
            head2.BorderWidthLeft = 0
            head2.BorderWidthTop = 1
            head2.BorderWidthBottom = 1
            pTable1.AddCell(head2)

            For i As Integer = 0 To
dTable.Rows.Count - 1

                Dim text As String
                Dim datetext As DateTime

                text = dTable.Rows(i).Item("ID")
                cell2 = New PdfPCell(New
Phrase(text, times))
                cell2.Colspan = 1
                cell2.HorizontalAlignment = 1
                cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell2.BorderWidthLeft = 0
                cell2.BorderWidthRight = 0
                cell2.BorderWidthBottom = 0.5
                cell2.BorderWidthTop = 0
                pTable1.AddCell(cell2)

                text = dTable.Rows(i).Item("Date")
                datetext = DateTime.Parse(text)
                text = datetext.ToString("MMM dd")
                cell2 = New PdfPCell(New
Phrase(text, times))
                cell2.Colspan = 1
                cell2.HorizontalAlignment = 1
                cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell2.BorderWidthLeft = 0
                cell2.BorderWidthRight = 0
                cell2.BorderWidthBottom = 0.5
                cell2.BorderWidthTop = 0
                pTable1.AddCell(cell2)

                text =
dTable.Rows(i).Item("Employee Name")
                cell2 = New PdfPCell(New
Phrase(text, times))
                cell2.Colspan = 2
                cell2.HorizontalAlignment = 1
                cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell2.BorderWidthLeft = 0
                cell2.BorderWidthRight = 0
                cell2.BorderWidthBottom = 0.5
                cell2.BorderWidthTop = 0
                pTable1.AddCell(cell2)

                text =
dTable.Rows(i).Item("Employee Number")
                cell2 = New PdfPCell(New
Phrase(text, times))
                cell2.Colspan = 2
                cell2.HorizontalAlignment = 1
                cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell2.BorderWidthLeft = 0
                cell2.BorderWidthRight = 0
                cell2.BorderWidthBottom = 0.5
                cell2.BorderWidthTop = 0
                pTable1.AddCell(cell2)

                text =
dTable.Rows(i).Item("Branch")
                cell2 = New PdfPCell(New
Phrase(text, times))
                cell2.Colspan = 2
                cell2.HorizontalAlignment = 1
                cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell2.BorderWidthLeft = 0
                cell2.BorderWidthRight = 0
                cell2.BorderWidthBottom = 0.5
                cell2.BorderWidthTop = 0
                pTable1.AddCell(cell2)
```

```
                text =
dTable.Rows(i).Item("Department")
                cell2 = New PdfPCell(New
Phrase(text, times))
                cell2.Colspan = 2
                cell2.HorizontalAlignment = 1
                cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell2.BorderWidthLeft = 0
                cell2.BorderWidthRight = 0
                cell2.BorderWidthBottom = 0.5
                cell2.BorderWidthTop = 0
                pTable1.AddCell(cell2)

                text =
dTable.Rows(i).Item("Designation")
                cell2 = New PdfPCell(New
Phrase(text, times))
                cell2.Colspan = 2
                cell2.HorizontalAlignment = 1
                cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell2.BorderWidthLeft = 0
                cell2.BorderWidthRight = 0
                cell2.BorderWidthBottom = 0.5
                cell2.BorderWidthTop = 0
                pTable1.AddCell(cell2)

            Next

            pdfDAbsent.Add(pTable1)

            pdfDAbsent.NewPage()
            pdfDAbsent.Close()

            'opens the file
            System.Diagnostics.Process.Start(path)

        Catch ex As Exception

            MsgBox(ex.Message,
MsgBoxStyle.Information)

        End Try

    End Sub

End Class

Imports iTextSharp.text
Imports iTextSharp.text.pdf
Imports System.IO

Public Class PrintDTR

    Private SQL As New SQLControl
    Private HRow As Boolean
    Private qstring As String
    Private dTable, dTable2, dTable3 As New
DataTable
    Private printString As String
    Dim result1, result2 As Integer

    Private Sub SetDataTable1()

        If SQL.SQLDS IsNot Nothing Then

            SQL.SQLDS.Clear()

        End If

        qstring = "EXEC PrintDTR @Branch = '" &
AttendanceInput.BranchCB.SelectedValue & "',
@Department = '" &
AttendanceInput.DeptCB.SelectedValue & "',
@Designation = '" &
AttendanceInput.DesigCB.SelectedValue & "', @SDate
= '" & AttendanceInput.StartDTP.Value.Date & "',
@EDate = '" & AttendanceInput.EndDTP.Value.Date &
"'"

        SQL.RunQuery(qstring)

        SQL.SQLDA.Fill(dTable)

    End Sub

    Private Sub GetSum()

        If SQL.SQLDS IsNot Nothing Then

            SQL.SQLDS.Clear()

        End If

        '-----Get Number of Rows per Employee-----
        qstring = "SELECT " & _
                        "Emp.Name AS Name, " & _
                        "COUNT(*) AS NCount " & _
                  "FROM tAttendanceComputation AS
tA " & _
                        "INNER JOIN tEmployee AS
Emp ON Emp.ID = tA.ID_Employee " & _
                        "INNER JOIN tBranch AS
Branch ON Branch.ID = Emp.ID_Branch " & _
                        "INNER JOIN tDepartment AS
Dept ON Dept.ID = Emp.ID_Department " & _
                        "INNER JOIN tDesignation
AS Desig ON Desig.ID = Emp.ID_Designation " & _
                  "WHERE " & _
                        "Branch.ID = '" &
AttendanceInput.BranchCB.SelectedValue & "' AND "
& _
                        "Dept.ID = '" &
AttendanceInput.DeptCB.SelectedValue & "' AND " &
_
                        "Desig.ID = '" &
AttendanceInput.DesigCB.SelectedValue & "' AND " &
_
                        "tA.Date BETWEEN '" &
AttendanceInput.StartDTP.Value.Date & "' AND '" &
AttendanceInput.EndDTP.Value.Date & "' " & _
                  "GROUP BY Emp.Name "

        SQL.RunQuery(qstring)

        SQL.SQLDA.Fill(dTable2)

        '-----Get Sum of Hours Worked, Absent, UT,
Late and OT-----
        For i As Integer = 0 To dTable2.Rows.Count
- 1

            qstring = "SELECT " & _
                            "Emp.Name AS Name, " &
_
```

```
                            "SUM(DS.Hour) AS
[Total Hours], " & _
                            "SUM(tA.UT) AS [Total
UT], " & _
                            "SUM(tA.Late) AS
[Total Late], " & _
                            "ISNULL(SUM(CASE
tA.Absent WHEN 1 THEN 1 END), 0) AS [Total
Absent], " & _

"CONVERT(DECIMAL(10,2),
(SUM(tA.RegularHours)/60.0)) AS [Total Reg Hours],
" & _

"CONVERT(DECIMAL(10,2), (SUM(tA.WRD) + SUM(tA.SH)
+ SUM(tA.WRH) + SUM(tA.WSR) + SUM(tA.WRR) +
SUM(tA.OT))/60.00) AS [OT Sum] " & _
                        "FROM tAttendanceComputation
AS tA " & _
                            "INNER JOIN
tDailySchedule AS DS ON DS.ID = tA.ID_DS " & _
                            "INNER JOIN tEmployee
AS Emp ON Emp.ID = tA.ID_Employee " & _
                        "WHERE " & _
                            "Emp.Name = '" &
dTable.AsDataView.ToTable(True, "Employee
Name").Rows(i).Item(0) & "' AND " & _
                            "tA.Date BETWEEN '" &
AttendanceInput.StartDTP.Value.Date & "' AND '" &
AttendanceInput.EndDTP.Value.Date & "'" & _
                        "GROUP BY Emp.Name"

            SQL.RunQuery(qstring)

            SQL.SQLDA.Fill(dTable3)

        Next

    End Sub

    Public Sub PrintPDFDTR()

        Try

            dTable.Clear()
            dTable2.Clear()
            dTable3.Clear()

            '-----Set Dates For File Name-----
            Dim sDate, eDate, s, e As String

            sDate =
AttendanceInput.StartDTP.Value.Date
            s = sDate.Replace("/", "")

            eDate =
AttendanceInput.EndDTP.Value.Date
            e = eDate.Replace("/", "")

            '-----Set Data Table from Database----
-
            SetDataTable1()

            '-----Get Sum for Footer-----
            GetSum()

            '-----Create PDF-----
            Dim pdfDTR As New Document
            Dim filePath As Object
            Dim defaultPath As String
            defaultPath =
Application.StartupPath.ToString & "\DTR" & s & "-
" & e &
AttendanceInput.BranchCB.SelectedItem("Name") &
".pdf"
            filePath = InputBox("Specify file
path.", "Print DTR", defaultPath)
            Dim path As String
            path = filePath
            Dim pdfWrite As PdfWriter =
PdfWriter.GetInstance(pdfDTR, New FileStream(path,
FileMode.Create))

            '-----Open PDF-----
            pdfDTR.Open()

            '-----Set FONT-----
            Dim bfTimes As BaseFont
            bfTimes =
BaseFont.CreateFont(BaseFont.COURIER,
BaseFont.CP1252, False)
            Dim times, times2 As Font
            times = New Font(bfTimes, 7)
            times2 = New Font(bfTimes, 8,
Font.BOLD)

            '-----Report Header-----
            pdfDTR.Add(New Paragraph("Daily Time
Record", times))
            pdfDTR.Add(New Paragraph("Period: " &
AttendanceInput.StartDTP.Value.Date & " to " &
AttendanceInput.EndDTP.Value.Date, times))
            pdfDTR.Add(New Paragraph("Branch: " &
AttendanceInput.BranchCB.SelectedItem("Name").ToSt
ring, times))
            pdfDTR.Add(New Paragraph("Department:
" &
AttendanceInput.DeptCB.SelectedItem("Name").ToStri
ng, times))
            pdfDTR.Add(New Paragraph("Designation:
" &
AttendanceInput.DesigCB.SelectedItem("Name").ToStr
ing, times))
            pdfDTR.Add(New Paragraph(" "))

            '-----Create PDF Table-----
            Dim pTable1 As PdfPTable

            '-----Size and Position of PDF Table--
---


            '--------Contents----------
            Dim nameD2, text As String

            Dim n As Integer

            Dim x As Integer

            x = 0

            For i As Integer = 0 To
dTable2.Rows.Count - 1

                pTable1 = New PdfPTable(17)
                pTable1.TotalWidth = 100%
                pTable1.WidthPercentage = 100
```

```
                pTable1.TotalWidth =
pdfDTR.PageSize.Width - pdfDTR.LeftMargin -
pdfDTR.RightMargin

                pTable1.WriteSelectedRows(0, -1,
pdfDTR.LeftMargin, pdfDTR.RightMargin,
pdfWrite.DirectContent)


                Dim cell, cell2, head2, head3 As
PdfPCell

                '-----TABLE HEADER------
                cell = New PdfPCell(New
Phrase("Employee Names" & Environment.NewLine &
"Employee Number", times2))
                cell.Colspan = 17
                cell.HorizontalAlignment = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                pTable1.AddCell(cell)

                head2 = New PdfPCell(New
Phrase("DATE", times2))
                head2.Colspan = 1
                head2.HorizontalAlignment = 1
                head2.VerticalAlignment =
Element.ALIGN_MIDDLE
                head2.BorderWidthLeft = 0
                head2.BorderWidthRight = 0
                head2.BorderWidthTop = 0
                head2.BorderWidthBottom = 0
                pTable1.AddCell(head2)

                head2 = New PdfPCell(New
Phrase("DAY", times2))
                head2.Colspan = 1
                head2.HorizontalAlignment = 1
                head2.VerticalAlignment =
Element.ALIGN_MIDDLE
                head2.BorderWidthRight = 0
                head2.BorderWidthLeft = 0
                head2.BorderWidthTop = 0
                head2.BorderWidthBottom = 0
                pTable1.AddCell(head2)

                head2 = New PdfPCell(New
Phrase("SHIFT ID", times2))
                head2.Colspan = 1
                head2.HorizontalAlignment = 1
                head2.VerticalAlignment =
Element.ALIGN_MIDDLE
                head2.BorderWidthLeft = 0
                head2.BorderWidthRight = 0
                head2.BorderWidthTop = 0
                head2.BorderWidthBottom = 0
                pTable1.AddCell(head2)

                head2 = New PdfPCell(New
Phrase("IN1", times2))
                head2.Colspan = 1
                head2.HorizontalAlignment = 1
                head2.VerticalAlignment =
Element.ALIGN_MIDDLE
                head2.BorderWidthLeft = 0
                head2.BorderWidthRight = 0
                head2.BorderWidthTop = 0
                head2.BorderWidthBottom = 0
                pTable1.AddCell(head2)

                head3 = New PdfPCell(New
Phrase("OUT1", times2))
                head3.Colspan = 1
                head3.HorizontalAlignment = 1
                head3.VerticalAlignment =
Element.ALIGN_MIDDLE
                head3.BorderWidthLeft = 0
                head3.BorderWidthRight = 0
                head3.BorderWidthTop = 0
                head3.BorderWidthBottom = 0
                pTable1.AddCell(head3)

                head2 = New PdfPCell(New
Phrase("IN2", times2))
                head2.Colspan = 1
                head2.HorizontalAlignment = 1
                head2.VerticalAlignment =
Element.ALIGN_MIDDLE
                head2.BorderWidthLeft = 0
                head2.BorderWidthRight = 0
                head2.BorderWidthTop = 0
                head2.BorderWidthBottom = 0
                pTable1.AddCell(head2)

                head3 = New PdfPCell(New
Phrase("OUT2", times2))
                head3.Colspan = 1
                head3.HorizontalAlignment = 1
                head3.VerticalAlignment =
Element.ALIGN_MIDDLE
                head3.BorderWidthLeft = 0
                head3.BorderWidthRight = 0
                head3.BorderWidthTop = 0
                head3.BorderWidthBottom = 0
                pTable1.AddCell(head3)

                head2 = New PdfPCell(New
Phrase("IN3", times2))
                head2.Colspan = 1
                head2.HorizontalAlignment = 1
                head2.VerticalAlignment =
Element.ALIGN_MIDDLE
                head2.BorderWidthLeft = 0
                head2.BorderWidthRight = 0
                head2.BorderWidthTop = 0
                head2.BorderWidthBottom = 0
                pTable1.AddCell(head2)

                head3 = New PdfPCell(New
Phrase("OUT3", times2))
                head3.Colspan = 1
                head3.HorizontalAlignment = 1
                head3.VerticalAlignment =
Element.ALIGN_MIDDLE
                head3.BorderWidthLeft = 0
                head3.BorderWidthRight = 0
                head3.BorderWidthTop = 0
                head3.BorderWidthBottom = 0
                pTable1.AddCell(head3)

                head3 = New PdfPCell(New
Phrase("HOURS", times2))
                head3.Colspan = 1
                head3.HorizontalAlignment = 1
                head3.VerticalAlignment =
Element.ALIGN_MIDDLE
                head3.BorderWidthLeft = 0
                head3.BorderWidthRight = 0
```

```
                head3.BorderWidthTop = 0
                head3.BorderWidthBottom = 0
                pTable1.AddCell(head3)

                head3 = New PdfPCell(New
Phrase("TARDY MINS", times2))
                head3.Colspan = 1
                head3.HorizontalAlignment = 1
                head3.VerticalAlignment =
Element.ALIGN_MIDDLE
                head3.BorderWidthLeft = 0
                head3.BorderWidthRight = 0
                head3.BorderWidthTop = 0
                head3.BorderWidthBottom = 0
                pTable1.AddCell(head3)

                head3 = New PdfPCell(New
Phrase("UT MINS", times2))
                head3.Colspan = 1
                head3.HorizontalAlignment = 1
                head3.VerticalAlignment =
Element.ALIGN_MIDDLE
                head3.BorderWidthLeft = 0
                head3.BorderWidthRight = 0
                head3.BorderWidthTop = 0
                head3.BorderWidthBottom = 0
                pTable1.AddCell(head3)

                head3 = New PdfPCell(New
Phrase("ABS", times2))
                head3.Colspan = 1
                head3.HorizontalAlignment = 1
                head3.VerticalAlignment =
Element.ALIGN_MIDDLE
                head3.BorderWidthLeft = 0
                head3.BorderWidthRight = 0
                head3.BorderWidthTop = 0
                head3.BorderWidthBottom = 0
                pTable1.AddCell(head3)

                head3 = New PdfPCell(New
Phrase("NET", times2))
                head3.Colspan = 1
                head3.HorizontalAlignment = 1
                head3.VerticalAlignment =
Element.ALIGN_MIDDLE
                head3.BorderWidthLeft = 0
                head3.BorderWidthRight = 0
                head3.BorderWidthTop = 0
                head3.BorderWidthBottom = 0
                pTable1.AddCell(head3)

                head3 = New PdfPCell(New
Phrase("OT", times2))
                head3.Colspan = 1
                head3.HorizontalAlignment = 1
                head3.VerticalAlignment =
Element.ALIGN_MIDDLE
                head3.BorderWidthLeft = 0
                head3.BorderWidthRight = 0
                head3.BorderWidthTop = 0
                head3.BorderWidthBottom = 0
                pTable1.AddCell(head3)

                head3 = New PdfPCell(New
Phrase("REMARKS", times2))
                head3.Colspan = 2
                head3.HorizontalAlignment = 1
                head3.VerticalAlignment =
Element.ALIGN_MIDDLE
                head3.BorderWidthLeft = 0
                head3.BorderWidthRight = 0
                head3.BorderWidthTop = 0
                head3.BorderWidthBottom = 0
                pTable1.AddCell(head3)
                '--------END TABLE HEADER--------

                '-----Employee Name and Number----
                nameD2 =
dTable2.Rows(i).Item("Name")
                nameD2 += Environment.NewLine
                nameD2 +=
dTable.Rows(x).Item("Employee Number")

                cell2 = New PdfPCell(New
Phrase(nameD2, times))
                cell2.Colspan = 17
                cell2.HorizontalAlignment = 0
                cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell2.BorderWidthLeft = 0
                cell2.BorderWidthRight = 0
                cell2.BorderWidthTop = 1
                cell2.BorderWidthBottom = 1
                pTable1.AddCell(cell2)
                '------End Employee Name and
Number------

                n =
Integer.Parse(dTable2.Rows(i).Item("NCount"))

                For j As Integer = 0 To n - 1

                    x += 1

                    '--------Date--------
                    text = dTable.Rows(x -
1).Item("Date")

                    cell2 = New PdfPCell(New
Phrase(text, times))
                    cell2.Colspan = 1
                    cell2.HorizontalAlignment = 1
                    cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                    cell2.BorderWidthLeft = 0
                    cell2.BorderWidthRight = 0
                    cell2.BorderWidthTop = 0
                    cell2.BorderWidthBottom = 0.5
                    pTable1.AddCell(cell2)
                    '-------End Date-------

                    '-------Day-------
                    text = dTable.Rows(x -
1).Item("Day")

                    cell2 = New PdfPCell(New
Phrase(text, times))
                    cell2.Colspan = 1
                    cell2.HorizontalAlignment = 1
                    cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                    cell2.BorderWidthLeft = 0
                    cell2.BorderWidthRight = 0
                    cell2.BorderWidthTop = 0
                    cell2.BorderWidthBottom = 0.5
                    pTable1.AddCell(cell2)
```

```
                    '-------End Day---

                    text = dTable.Rows(x -
1).Item("Sched")

                    cell2 = New PdfPCell(New
Phrase(text, times))
                    cell2.Colspan = 1
                    cell2.HorizontalAlignment = 1
                    cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                    cell2.BorderWidthLeft = 0
                    cell2.BorderWidthRight = 0
                    cell2.BorderWidthTop = 0
                    cell2.BorderWidthBottom = 0.5
                    pTable1.AddCell(cell2)

                    text = dTable.Rows(x -
1).Item("In 1")

                    cell2 = New PdfPCell(New
Phrase(text, times))
                    cell2.Colspan = 1
                    cell2.HorizontalAlignment = 1
                    cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                    cell2.BorderWidthLeft = 0
                    cell2.BorderWidthRight = 0
                    cell2.BorderWidthTop = 0
                    cell2.BorderWidthBottom = 0.5
                    pTable1.AddCell(cell2)

                    text = dTable.Rows(x -
1).Item("Out 1")

                    cell2 = New PdfPCell(New
Phrase(text, times))
                    cell2.Colspan = 1
                    cell2.HorizontalAlignment = 1
                    cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                    cell2.BorderWidthLeft = 0
                    cell2.BorderWidthRight = 0
                    cell2.BorderWidthTop = 0
                    cell2.BorderWidthBottom = 0.5
                    pTable1.AddCell(cell2)

                    text = dTable.Rows(x -
1).Item("In 2")

                    cell2 = New PdfPCell(New
Phrase(text, times))
                    cell2.Colspan = 1
                    cell2.HorizontalAlignment = 1
                    cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                    cell2.BorderWidthLeft = 0
                    cell2.BorderWidthRight = 0
                    cell2.BorderWidthTop = 0
                    cell2.BorderWidthBottom = 0.5
                    pTable1.AddCell(cell2)

                    text = dTable.Rows(x -
1).Item("Out 2")

                    cell2 = New PdfPCell(New
Phrase(text, times))
                    cell2.Colspan = 1
                    cell2.HorizontalAlignment = 1
                    cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                    cell2.BorderWidthLeft = 0
                    cell2.BorderWidthRight = 0
                    cell2.BorderWidthTop = 0
                    cell2.BorderWidthBottom = 0.5
                    pTable1.AddCell(cell2)

                    text = dTable.Rows(x -
1).Item("In 3")

                    cell2 = New PdfPCell(New
Phrase(text, times))
                    cell2.Colspan = 1
                    cell2.HorizontalAlignment = 1
                    cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                    cell2.BorderWidthLeft = 0
                    cell2.BorderWidthRight = 0
                    cell2.BorderWidthTop = 0
                    cell2.BorderWidthBottom = 0.5
                    pTable1.AddCell(cell2)

                    text = dTable.Rows(x -
1).Item("Out 3")

                    cell2 = New PdfPCell(New
Phrase(text, times))
                    cell2.Colspan = 1
                    cell2.HorizontalAlignment = 1
                    cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                    cell2.BorderWidthLeft = 0
                    cell2.BorderWidthRight = 0
                    cell2.BorderWidthTop = 0
                    cell2.BorderWidthBottom = 0.5
                    pTable1.AddCell(cell2)

                    text = dTable.Rows(x -
1).Item("Hours")

                    If text = "0.00" Then

                        text = " "

                    End If

                    cell2 = New PdfPCell(New
Phrase(text, times))
                    cell2.Colspan = 1
                    cell2.HorizontalAlignment = 1
                    cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                    cell2.BorderWidthLeft = 0
                    cell2.BorderWidthRight = 0
                    cell2.BorderWidthTop = 0
                    cell2.BorderWidthBottom = 0.5
                    pTable1.AddCell(cell2)

                    text = dTable.Rows(x -
1).Item("Late")

                    If text = "0" Then

                        text = " "

                    End If
```

```
                    cell2 = New PdfPCell(New
Phrase(text, times))
                    cell2.Colspan = 1
                    cell2.HorizontalAlignment = 1
                    cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                    cell2.BorderWidthLeft = 0
                    cell2.BorderWidthRight = 0
                    cell2.BorderWidthTop = 0
                    cell2.BorderWidthBottom = 0.5
                    pTable1.AddCell(cell2)

                    text = dTable.Rows(x -
1).Item("UT")

                    If text = "0" Then

                        text = " "

                    End If

                    cell2 = New PdfPCell(New
Phrase(text, times))
                    cell2.Colspan = 1
                    cell2.HorizontalAlignment = 1
                    cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                    cell2.BorderWidthLeft = 0
                    cell2.BorderWidthRight = 0
                    cell2.BorderWidthTop = 0
                    cell2.BorderWidthBottom = 0.5
                    pTable1.AddCell(cell2)

                    text = dTable.Rows(x -
1).Item("Absent")

                    cell2 = New PdfPCell(New
Phrase(text, times))
                    cell2.Colspan = 1
                    cell2.HorizontalAlignment = 1
                    cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                    cell2.BorderWidthLeft = 0
                    cell2.BorderWidthRight = 0
                    cell2.BorderWidthTop = 0
                    cell2.BorderWidthBottom = 0.5
                    pTable1.AddCell(cell2)

                    text = dTable.Rows(x -
1).Item("RegularHours")

                    If text = "0.00" Then

                        text = " "

                    End If

                    cell2 = New PdfPCell(New
Phrase(text, times))
                    cell2.Colspan = 1
                    cell2.HorizontalAlignment = 1
                    cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                    cell2.BorderWidthLeft = 0
                    cell2.BorderWidthRight = 0
                    cell2.BorderWidthTop = 0
                    cell2.BorderWidthBottom = 0.5
                    pTable1.AddCell(cell2)

                    text = dTable.Rows(x -
1).Item("OT SUM")

                    If text = "0.00" Then

                        text = " "

                    End If

                    cell2 = New PdfPCell(New
Phrase(text, times))
                    cell2.Colspan = 1
                    cell2.HorizontalAlignment = 1
                    cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                    cell2.BorderWidthLeft = 0
                    cell2.BorderWidthRight = 0
                    cell2.BorderWidthTop = 0
                    cell2.BorderWidthBottom = 0.5
                    pTable1.AddCell(cell2)

                    text = dTable.Rows(x -
1).Item("Remarks")

                    cell2 = New PdfPCell(New
Phrase(text, times))
                    cell2.Colspan = 2
                    cell2.HorizontalAlignment = 1
                    cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                    cell2.BorderWidthLeft = 0
                    cell2.BorderWidthRight = 0
                    cell2.BorderWidthTop = 0
                    cell2.BorderWidthBottom = 0.5
                    pTable1.AddCell(cell2)

                Next

                '----------TOTAL----------
                text = "TOTAL"

                cell2 = New PdfPCell(New
Phrase(text, times2))
                cell2.Colspan = 9
                cell2.HorizontalAlignment = 2
                cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell2.BorderWidthLeft = 0
                cell2.BorderWidthRight = 0
                cell2.BorderWidthTop = 1
                cell2.BorderWidthBottom = 1
                pTable1.AddCell(cell2)

                text = dTable3.Rows(i).Item("Total
Hours")

                cell2 = New PdfPCell(New
Phrase(text, times))
                cell2.Colspan = 1
                cell2.HorizontalAlignment = 1
                cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell2.BorderWidthLeft = 0
                cell2.BorderWidthRight = 0
                cell2.BorderWidthTop = 1
                cell2.BorderWidthBottom = 1
                pTable1.AddCell(cell2)
```

```
                text = dTable3.Rows(i).Item("Total
Late")

                cell2 = New PdfPCell(New
Phrase(text, times))
                cell2.Colspan = 1
                cell2.HorizontalAlignment = 1
                cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell2.BorderWidthLeft = 0
                cell2.BorderWidthRight = 0
                cell2.BorderWidthTop = 1
                cell2.BorderWidthBottom = 1
                pTable1.AddCell(cell2)

                text = dTable3.Rows(i).Item("Total
UT")

                cell2 = New PdfPCell(New
Phrase(text, times))
                cell2.Colspan = 1
                cell2.HorizontalAlignment = 1
                cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell2.BorderWidthLeft = 0
                cell2.BorderWidthRight = 0
                cell2.BorderWidthTop = 1
                cell2.BorderWidthBottom = 1
                pTable1.AddCell(cell2)

                text = dTable3.Rows(i).Item("Total
Absent")

                cell2 = New PdfPCell(New
Phrase(text, times))
                cell2.Colspan = 1
                cell2.HorizontalAlignment = 1
                cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell2.BorderWidthLeft = 0
                cell2.BorderWidthRight = 0
                cell2.BorderWidthTop = 1
                cell2.BorderWidthBottom = 1
                pTable1.AddCell(cell2)

                text = dTable3.Rows(i).Item("Total
Reg Hours")

                cell2 = New PdfPCell(New
Phrase(text, times))
                cell2.Colspan = 1
                cell2.HorizontalAlignment = 1
                cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell2.BorderWidthLeft = 0
                cell2.BorderWidthRight = 0
                cell2.BorderWidthTop = 1
                cell2.BorderWidthBottom = 1
                pTable1.AddCell(cell2)

                text = dTable3.Rows(i).Item("OT
Sum")

                cell2 = New PdfPCell(New
Phrase(text, times))
                cell2.Colspan = 1
                cell2.HorizontalAlignment = 1
                cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell2.BorderWidthLeft = 0
                cell2.BorderWidthRight = 0
                cell2.BorderWidthTop = 1
                cell2.BorderWidthBottom = 1
                pTable1.AddCell(cell2)

                text = " "

                cell2 = New PdfPCell(New
Phrase(text, times))
                cell2.Colspan = 2
                cell2.HorizontalAlignment = 1
                cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell2.BorderWidthLeft = 0
                cell2.BorderWidthRight = 0
                cell2.BorderWidthTop = 1
                cell2.BorderWidthBottom = 1
                pTable1.AddCell(cell2)

                text = " "

                cell2 = New PdfPCell(New
Phrase(text, times))
                cell2.Colspan = 17
                cell2.HorizontalAlignment = 1
                cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell2.BorderWidthLeft = 0
                cell2.BorderWidthRight = 0
                cell2.BorderWidthTop = 0
                cell2.BorderWidthBottom = 0
                pTable1.AddCell(cell2)

                pdfDTR.Add(pTable1)

                pdfDTR.NewPage()

            Next



            pdfDTR.Close()

            'opens the file
            System.Diagnostics.Process.Start(path)

        Catch ex As Exception

            MsgBox(ex.Message)

        End Try

    End Sub

End Class

Imports iTextSharp.text
Imports iTextSharp.text.pdf
Imports System.IO

Public Class PrintDailyOvertime

    Private SQL As New SQLControl
    Private dTable As New DataTable
    Dim qstring As String

    Private Sub SetTable()
```

```
        If SQL.SQLDS IsNot Nothing Then

            SQL.SQLDS.Clear()

        End If

        qstring = "SELECT " & _
                          "tFileOvertime.ID AS ID, "
& _
                          "Employee.Name AS
[Employee Name], " & _
                          "ENumber.EmployeeNumber AS
[Employee Number], " & _
                          "Branch.Name AS Branch, "
& _
                          "Dept.Name AS Department,
" & _
                          "Desig.Name AS
Designation, " & _
                          "tFileOvertime.Reason as
Reason " & _
                  "FROM tFileOvertime " & _
                          "INNER JOIN tEmployee AS
Employee ON Employee.ID =
tFileOvertime.ID_Employee " & _
                          "INNER JOIN tEmployee AS
ENumber ON ENumber.ID = tFileOvertime.ID_Employee
" & _
                          "INNER JOIN tBranch AS
Branch ON Branch.ID = Employee.ID_Branch " & _
                          "INNER JOIN tDepartment AS
Dept ON Dept.ID = Employee.ID_Department " & _
                          "INNER JOIN tDesignation
AS Desig ON Desig.ID = Employee.ID_Designation " &
_
                          "INNER JOIN
tEmployeeStatus AS EStatus ON EStatus.ID =
Employee.ID_EmployeeStatus " & _
                  "WHERE tFileOverTime.Date = '" &
MainWindow.DateTimePicker1.Value.Date & "'" & _
                  "ORDER BY Branch.Name,
Dept.Name, Desig.Name"

        SQL.RunQuery(qstring)

        SQL.SQLDA.Fill(dTable)

    End Sub


    Public Sub PrintDOvertime()

        Try

            Dim sDate, s As String

            sDate =
MainWindow.DateTimePicker1.Value.Date
            s = sDate.Replace("/", "")

            dTable.Clear()

            SetTable()

            '-----Create PDF-----
            Dim pdfDAbsent As New Document
            Dim filePath As Object
            Dim defaultPath As String

            defaultPath =
Application.StartupPath.ToString &
"\DailyOvertime" & s & ".pdf"
            filePath = InputBox("Specify file
path.", "Print Daily Overtime", defaultPath)
            Dim path As String
            path = filePath
            Dim pdfWrite As PdfWriter =
PdfWriter.GetInstance(pdfDAbsent, New
FileStream(path, FileMode.Create))

            '-----Open PDF-----
            pdfDAbsent.Open()

            '-----Set FONT-----
            Dim bfTimes As BaseFont
            bfTimes =
BaseFont.CreateFont(BaseFont.COURIER,
BaseFont.CP1252, False)
            Dim times, times2 As Font
            times = New Font(bfTimes, 8)
            times2 = New Font(bfTimes, 8,
Font.BOLD)

            '-----Report Header-----
            pdfDAbsent.Add(New Paragraph("Daily
Overtime", times))
            pdfDAbsent.Add(New Paragraph("Date: "
& MainWindow.DateTimePicker1.Value.Date, times))
            pdfDAbsent.Add(New Paragraph(" "))

            '-----Create PDF Table-----
            Dim pTable1 As PdfPTable

            '-----Size and Position of PDF Table--
---
            pTable1 = New PdfPTable(13)
            pTable1.TotalWidth = 100%
            pTable1.WidthPercentage = 100
            pTable1.TotalWidth =
pdfDAbsent.PageSize.Width - pdfDAbsent.LeftMargin
- pdfDAbsent.RightMargin

            pTable1.WriteSelectedRows(0, -1,
pdfDAbsent.LeftMargin, pdfDAbsent.RightMargin,
pdfWrite.DirectContent)

            Dim cell, cell2, head2 As PdfPCell

            '-----TABLE HEADER------
            cell = New PdfPCell(New Phrase("ID",
times2))
            cell.Colspan = 1
            cell.HorizontalAlignment = 1
            cell.VerticalAlignment =
Element.ALIGN_MIDDLE
            cell.BorderWidthLeft = 0
            cell.BorderWidthRight = 0
            cell.BorderWidthBottom = 1
            cell.BorderWidthTop = 1
            pTable1.AddCell(cell)

            head2 = New PdfPCell(New
Phrase("EMPLOYEE NAME", times2))
            head2.Colspan = 2
            head2.HorizontalAlignment = 1
            head2.VerticalAlignment =
Element.ALIGN_MIDDLE
            head2.BorderWidthRight = 0
```

```
            head2.BorderWidthLeft = 0
            head2.BorderWidthTop = 1
            head2.BorderWidthBottom = 1
            pTable1.AddCell(head2)

            head2 = New PdfPCell(New
Phrase("EMPLOYEE NUMBER", times2))
            head2.Colspan = 2
            head2.HorizontalAlignment = 1
            head2.VerticalAlignment =
Element.ALIGN_MIDDLE
            head2.BorderWidthRight = 0
            head2.BorderWidthLeft = 0
            head2.BorderWidthTop = 1
            head2.BorderWidthBottom = 1
            pTable1.AddCell(head2)

            head2 = New PdfPCell(New
Phrase("BRANCH", times2))
            head2.Colspan = 2
            head2.HorizontalAlignment = 1
            head2.VerticalAlignment =
Element.ALIGN_MIDDLE
            head2.BorderWidthRight = 0
            head2.BorderWidthLeft = 0
            head2.BorderWidthTop = 1
            head2.BorderWidthBottom = 1
            pTable1.AddCell(head2)

            head2 = New PdfPCell(New
Phrase("DEPARTMENT", times2))
            head2.Colspan = 2
            head2.HorizontalAlignment = 1
            head2.VerticalAlignment =
Element.ALIGN_MIDDLE
            head2.BorderWidthRight = 0
            head2.BorderWidthLeft = 0
            head2.BorderWidthTop = 1
            head2.BorderWidthBottom = 1
            pTable1.AddCell(head2)

            head2 = New PdfPCell(New
Phrase("DESIGNATION", times2))
            head2.Colspan = 2
            head2.HorizontalAlignment = 1
            head2.VerticalAlignment =
Element.ALIGN_MIDDLE
            head2.BorderWidthRight = 0
            head2.BorderWidthLeft = 0
            head2.BorderWidthTop = 1
            head2.BorderWidthBottom = 1
            pTable1.AddCell(head2)

            head2 = New PdfPCell(New
Phrase("REASON", times2))
            head2.Colspan = 2
            head2.HorizontalAlignment = 1
            head2.VerticalAlignment =
Element.ALIGN_MIDDLE
            head2.BorderWidthRight = 0
            head2.BorderWidthLeft = 0
            head2.BorderWidthTop = 1
            head2.BorderWidthBottom = 1
            pTable1.AddCell(head2)

            For i As Integer = 0 To
dTable.Rows.Count - 1

                Dim text As String
                text = dTable.Rows(i).Item("ID")
                cell2 = New PdfPCell(New
Phrase(text, times))
                cell2.Colspan = 1
                cell2.HorizontalAlignment = 1
                cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell2.BorderWidthLeft = 0
                cell2.BorderWidthRight = 0
                cell2.BorderWidthBottom = 0.5
                cell2.BorderWidthTop = 0
                pTable1.AddCell(cell2)

                text =
dTable.Rows(i).Item("Employee Name")
                cell2 = New PdfPCell(New
Phrase(text, times))
                cell2.Colspan = 2
                cell2.HorizontalAlignment = 1
                cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell2.BorderWidthLeft = 0
                cell2.BorderWidthRight = 0
                cell2.BorderWidthBottom = 0.5
                cell2.BorderWidthTop = 0
                pTable1.AddCell(cell2)

                text =
dTable.Rows(i).Item("Employee Number")
                cell2 = New PdfPCell(New
Phrase(text, times))
                cell2.Colspan = 2
                cell2.HorizontalAlignment = 1
                cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell2.BorderWidthLeft = 0
                cell2.BorderWidthRight = 0
                cell2.BorderWidthBottom = 0.5
                cell2.BorderWidthTop = 0
                pTable1.AddCell(cell2)

                text =
dTable.Rows(i).Item("Branch")
                cell2 = New PdfPCell(New
Phrase(text, times))
                cell2.Colspan = 2
                cell2.HorizontalAlignment = 1
                cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell2.BorderWidthLeft = 0
                cell2.BorderWidthRight = 0
                cell2.BorderWidthBottom = 0.5
                cell2.BorderWidthTop = 0
                pTable1.AddCell(cell2)

                text =
dTable.Rows(i).Item("Department")
                cell2 = New PdfPCell(New
Phrase(text, times))
                cell2.Colspan = 2
                cell2.HorizontalAlignment = 1
                cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell2.BorderWidthLeft = 0
                cell2.BorderWidthRight = 0
                cell2.BorderWidthBottom = 0.5
                cell2.BorderWidthTop = 0
                pTable1.AddCell(cell2)
```

```
                text =
dTable.Rows(i).Item("Designation")
                cell2 = New PdfPCell(New
Phrase(text, times))
                cell2.Colspan = 2
                cell2.HorizontalAlignment = 1
                cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell2.BorderWidthLeft = 0
                cell2.BorderWidthRight = 0
                cell2.BorderWidthBottom = 0.5
                cell2.BorderWidthTop = 0
                pTable1.AddCell(cell2)

                text =
dTable.Rows(i).Item("Reason")
                cell2 = New PdfPCell(New
Phrase(text, times))
                cell2.Colspan = 2
                cell2.HorizontalAlignment = 1
                cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell2.BorderWidthLeft = 0
                cell2.BorderWidthRight = 0
                cell2.BorderWidthBottom = 0.5
                cell2.BorderWidthTop = 0
                pTable1.AddCell(cell2)

            Next

            pdfDAbsent.Add(pTable1)

            pdfDAbsent.NewPage()
            pdfDAbsent.Close()

            'opens the file
            System.Diagnostics.Process.Start(path)

        Catch ex As Exception

            MsgBox(ex.Message,
MsgBoxStyle.Information)

        End Try

    End Sub

End Class

Imports iTextSharp.text
Imports iTextSharp.text.pdf
Imports System.IO

Public Class PrintDailyAbsenteeism

    Private SQL As New SQLControl
    Private dTable As New DataTable
    Dim qstring As String

    Private Sub SetTable()

        If SQL.SQLDS IsNot Nothing Then

            SQL.SQLDS.Clear()

        End If

        qstring = "SELECT " & _
                            "tAttendanceComputation.ID
AS ID, " & _
                            "Employee.Name AS
[Employee Name], " & _
                            "ENumber.EmployeeNumber AS
[Employee Number], " & _
                            "Branch.Name AS Branch, "
& _
                            "Dept.Name AS Department,
" & _
                            "Desig.Name AS Designation
" & _
                  "FROM tAttendanceComputation " &
_
                            "INNER JOIN tEmployee AS
Employee ON Employee.ID =
tAttendanceComputation.ID_Employee " & _
                            "INNER JOIN tEmployee AS
ENumber ON ENumber.ID =
tAttendanceComputation.ID_Employee " & _
                            "INNER JOIN tBranch AS
Branch ON Branch.ID = Employee.ID_Branch " & _
                            "INNER JOIN tDepartment AS
Dept ON Dept.ID = Employee.ID_Department " & _
                            "INNER JOIN tDesignation
AS Desig ON Desig.ID = Employee.ID_Designation " &
_
                            "INNER JOIN
tEmployeeStatus AS EStatus ON EStatus.ID =
Employee.ID_EmployeeStatus " & _
                  "WHERE " & _

"tAttendanceComputation.Absent = 1 AND " & _

"tAttendanceComputation.Date = '" &
MainWindow.DateTimePicker1.Value.Date & "' " & _
                  "ORDER BY Branch.Name,
Dept.Name, Desig.Name"

        SQL.RunQuery(qstring)

        SQL.SQLDA.Fill(dTable)

    End Sub


    Public Sub PrintDAbsenteeism()

        Try

            Dim sDate, s As String

            sDate =
MainWindow.DateTimePicker1.Value.Date
            s = sDate.Replace("/", "")

            dTable.Clear()

            SetTable()

            '-----Create PDF-----
            Dim pdfDAbsent As New Document
            Dim filePath As Object
            Dim defaultPath As String
            defaultPath =
Application.StartupPath.ToString &
"\DailyAbsenteeism" & s & ".pdf"
            filePath = InputBox("Specify file
path.", "Print Daily Absenteeism", defaultPath)
```

```
            Dim path As String
            path = filePath
            Dim pdfWrite As PdfWriter =
PdfWriter.GetInstance(pdfDAbsent, New
FileStream(path, FileMode.Create))

            '-----Open PDF-----
            pdfDAbsent.Open()

            '-----Set FONT-----
            Dim bfTimes As BaseFont
            bfTimes =
BaseFont.CreateFont(BaseFont.COURIER,
BaseFont.CP1252, False)
            Dim times, times2 As Font
            times = New Font(bfTimes, 8)
            times2 = New Font(bfTimes, 8,
Font.BOLD)

            '-----Report Header-----
            pdfDAbsent.Add(New Paragraph("Daily
Absenteeism", times))
            pdfDAbsent.Add(New Paragraph("Date: "
& MainWindow.DateTimePicker1.Value.Date, times))
            pdfDAbsent.Add(New Paragraph(" "))

            '-----Create PDF Table-----
            Dim pTable1 As PdfPTable

            '-----Size and Position of PDF Table--
---
            pTable1 = New PdfPTable(11)
            pTable1.TotalWidth = 100%
            pTable1.WidthPercentage = 100
            pTable1.TotalWidth =
pdfDAbsent.PageSize.Width - pdfDAbsent.LeftMargin
- pdfDAbsent.RightMargin

            pTable1.WriteSelectedRows(0, -1,
pdfDAbsent.LeftMargin, pdfDAbsent.RightMargin,
pdfWrite.DirectContent)

            Dim cell, cell2, head2 As PdfPCell

            '-----TABLE HEADER------
            cell = New PdfPCell(New Phrase("ID",
times2))
            cell.Colspan = 1
            cell.HorizontalAlignment = 1
            cell.VerticalAlignment =
Element.ALIGN_MIDDLE
            cell.BorderWidthLeft = 0
            cell.BorderWidthRight = 0
            cell.BorderWidthBottom = 1
            cell.BorderWidthTop = 1
            pTable1.AddCell(cell)

            head2 = New PdfPCell(New
Phrase("EMPLOYEE NAME", times2))
            head2.Colspan = 2
            head2.HorizontalAlignment = 1
            head2.VerticalAlignment =
Element.ALIGN_MIDDLE
            head2.BorderWidthRight = 0
            head2.BorderWidthLeft = 0
            head2.BorderWidthTop = 1
            head2.BorderWidthBottom = 1
            pTable1.AddCell(head2)

            head2 = New PdfPCell(New
Phrase("EMPLOYEE NUMBER", times2))
            head2.Colspan = 2
            head2.HorizontalAlignment = 1
            head2.VerticalAlignment =
Element.ALIGN_MIDDLE
            head2.BorderWidthRight = 0
            head2.BorderWidthLeft = 0
            head2.BorderWidthTop = 1
            head2.BorderWidthBottom = 1
            pTable1.AddCell(head2)

            head2 = New PdfPCell(New
Phrase("BRANCH", times2))
            head2.Colspan = 2
            head2.HorizontalAlignment = 1
            head2.VerticalAlignment =
Element.ALIGN_MIDDLE
            head2.BorderWidthRight = 0
            head2.BorderWidthLeft = 0
            head2.BorderWidthTop = 1
            head2.BorderWidthBottom = 1
            pTable1.AddCell(head2)

            head2 = New PdfPCell(New
Phrase("DEPARTMENT", times2))
            head2.Colspan = 2
            head2.HorizontalAlignment = 1
            head2.VerticalAlignment =
Element.ALIGN_MIDDLE
            head2.BorderWidthRight = 0
            head2.BorderWidthLeft = 0
            head2.BorderWidthTop = 1
            head2.BorderWidthBottom = 1
            pTable1.AddCell(head2)

            head2 = New PdfPCell(New
Phrase("DESIGNATION", times2))
            head2.Colspan = 2
            head2.HorizontalAlignment = 1
            head2.VerticalAlignment =
Element.ALIGN_MIDDLE
            head2.BorderWidthRight = 0
            head2.BorderWidthLeft = 0
            head2.BorderWidthTop = 1
            head2.BorderWidthBottom = 1
            pTable1.AddCell(head2)

            For i As Integer = 0 To
dTable.Rows.Count - 1

                Dim text As String

                text = dTable.Rows(i).Item("ID")
                cell2 = New PdfPCell(New
Phrase(text, times))
                cell2.Colspan = 1
                cell2.HorizontalAlignment = 1
                cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell2.BorderWidthLeft = 0
                cell2.BorderWidthRight = 0
                cell2.BorderWidthBottom = 0.5
                cell2.BorderWidthTop = 0
                pTable1.AddCell(cell2)

                text =
dTable.Rows(i).Item("Employee Name")
```

```
                cell2 = New PdfPCell(New
Phrase(text, times))
                cell2.Colspan = 2
                cell2.HorizontalAlignment = 1
                cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell2.BorderWidthLeft = 0
                cell2.BorderWidthRight = 0
                cell2.BorderWidthBottom = 0.5
                cell2.BorderWidthTop = 0
                pTable1.AddCell(cell2)

                text =
dTable.Rows(i).Item("Employee Number")
                cell2 = New PdfPCell(New
Phrase(text, times))
                cell2.Colspan = 2
                cell2.HorizontalAlignment = 1
                cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell2.BorderWidthLeft = 0
                cell2.BorderWidthRight = 0
                cell2.BorderWidthBottom = 0.5
                cell2.BorderWidthTop = 0
                pTable1.AddCell(cell2)

                text =
dTable.Rows(i).Item("Branch")
                cell2 = New PdfPCell(New
Phrase(text, times))
                cell2.Colspan = 2
                cell2.HorizontalAlignment = 1
                cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell2.BorderWidthLeft = 0
                cell2.BorderWidthRight = 0
                cell2.BorderWidthBottom = 0.5
                cell2.BorderWidthTop = 0
                pTable1.AddCell(cell2)

                text =
dTable.Rows(i).Item("Department")
                cell2 = New PdfPCell(New
Phrase(text, times))
                cell2.Colspan = 2
                cell2.HorizontalAlignment = 1
                cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell2.BorderWidthLeft = 0
                cell2.BorderWidthRight = 0
                cell2.BorderWidthBottom = 0.5
                cell2.BorderWidthTop = 0
                pTable1.AddCell(cell2)

                text =
dTable.Rows(i).Item("Designation")
                cell2 = New PdfPCell(New
Phrase(text, times))
                cell2.Colspan = 2
                cell2.HorizontalAlignment = 1
                cell2.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell2.BorderWidthLeft = 0
                cell2.BorderWidthRight = 0
                cell2.BorderWidthBottom = 0.5
                cell2.BorderWidthTop = 0
                pTable1.AddCell(cell2)

            Next

                pdfDAbsent.Add(pTable1)

                pdfDAbsent.NewPage()
                pdfDAbsent.Close()

                'opens the file
                System.Diagnostics.Process.Start(path)

            Catch ex As Exception

                MsgBox(ex.Message,
MsgBoxStyle.Information)

            End Try

        End Sub

End Class

Imports iTextSharp.text
Imports iTextSharp.text.pdf
Imports System.IO

Public Class PrintCompensation

    Private SQL As New SQLControl
    Private qstring As String
    Private dTable, dTable2, dTable3, dTable4 As
New DataTable
    Private printString As String
    Dim result1, result2 As Integer
    Dim pdfCompensation As New Document
    Dim bfTimes As BaseFont
    Dim times, times2 As Font
    Dim pdfWrite As PdfWriter
    Dim sDate, eDate, s, e, nameD2, text, path As
String
    Dim n, x As Integer

    Public Sub PrintPDFCompensation()

        Try

            dTable.Clear()
            dTable2.Clear()
            dTable3.Clear()
            dTable4.Clear()

            SetDataTable()

            SetGroups()

            GetTotalForDate()

            GetGrandTotal()

            CreatePDF()

            '-----Open PDF-----
            pdfCompensation.Open()

            SetFont()

            '-----Report Header-----
            pdfCompensation.Add(New
Paragraph("Compensation List", times))
            pdfCompensation.Add(New
Paragraph("Period: " &
```

```
AttendanceInput.StartDTP.Value.Date & " to " &
AttendanceInput.EndDTP.Value.Date, times))
            pdfCompensation.Add(New
Paragraph("Branch: " &
AttendanceInput.BranchCB.SelectedItem("Name").ToSt
ring, times))
            pdfCompensation.Add(New
Paragraph("Department: " &
AttendanceInput.DeptCB.SelectedItem("Name").ToStri
ng, times))
            pdfCompensation.Add(New
Paragraph("Designation: " &
AttendanceInput.DesigCB.SelectedItem("Name").ToStr
ing, times))
            pdfCompensation.Add(New Paragraph("
"))

            '-----Create PDF Table-----


            '-----Contents-----
            Dim cell As PdfPCell

            x = 0

            For i As Integer = 0 To
dTable2.Rows.Count - 1

                Dim pTable1 As PdfPTable

                '-----Size and Position of PDF
Table-----
                pTable1 = New PdfPTable(8)
                pTable1.TotalWidth = 100%
                pTable1.WidthPercentage = 100
                pTable1.TotalWidth =
pdfCompensation.PageSize.Width -
pdfCompensation.LeftMargin -
pdfCompensation.RightMargin

                pTable1.WriteSelectedRows(0, -1,
pdfCompensation.LeftMargin,
pdfCompensation.RightMargin,
pdfWrite.DirectContent)

                '-----Table Header-----
                Dim head As PdfPCell

                head = New PdfPCell(New
Phrase("Employee Names" & Environment.NewLine &
"Employee Number", times2))
                head.Colspan = 8
                head.HorizontalAlignment = 0
                head.BorderWidthLeft = 0
                head.BorderWidthRight = 0
                head.BorderWidthTop = 1
                head.BorderWidthBottom = 1
                pTable1.AddCell(head)

                head = New PdfPCell(New
Phrase("DATE", times2))
                head.Colspan = 1
                head.HorizontalAlignment =
Element.ALIGN_CENTER
                head.VerticalAlignment =
Element.ALIGN_MIDDLE
                head.BorderWidthLeft = 0
                head.BorderWidthRight = 0
                head.BorderWidthTop = 0
                head.BorderWidthBottom = 1
                pTable1.AddCell(head)

                head = New PdfPCell(New
Phrase("BASIC", times2))
                head.Colspan = 1
                head.HorizontalAlignment =
Element.ALIGN_CENTER
                head.VerticalAlignment =
Element.ALIGN_MIDDLE
                head.BorderWidthLeft = 0
                head.BorderWidthRight = 0
                head.BorderWidthTop = 0
                head.BorderWidthBottom = 1
                pTable1.AddCell(head)

                head = New PdfPCell(New
Phrase("COLA", times2))
                head.Colspan = 1
                head.HorizontalAlignment =
Element.ALIGN_CENTER
                head.VerticalAlignment =
Element.ALIGN_MIDDLE
                head.BorderWidthLeft = 0
                head.BorderWidthRight = 0
                head.BorderWidthTop = 0
                head.BorderWidthBottom = 1
                pTable1.AddCell(head)

                head = New PdfPCell(New
Phrase("DURATION", times2))
                head.Colspan = 1
                head.HorizontalAlignment =
Element.ALIGN_CENTER
                head.VerticalAlignment =
Element.ALIGN_MIDDLE
                head.BorderWidthLeft = 0
                head.BorderWidthRight = 0
                head.BorderWidthTop = 0
                head.BorderWidthBottom = 1
                pTable1.AddCell(head)

                head = New PdfPCell(New
Phrase("ADJUSTMENT", times2))
                head.Colspan = 1
                head.HorizontalAlignment =
Element.ALIGN_CENTER
                head.VerticalAlignment =
Element.ALIGN_MIDDLE
                head.BorderWidthLeft = 0
                head.BorderWidthRight = 0
                head.BorderWidthTop = 0
                head.BorderWidthBottom = 1
                pTable1.AddCell(head)

                head = New PdfPCell(New
Phrase("RATE", times2))
                head.Colspan = 1
                head.HorizontalAlignment =
Element.ALIGN_CENTER
                head.VerticalAlignment =
Element.ALIGN_MIDDLE
                head.BorderWidthLeft = 0
                head.BorderWidthRight = 0
                head.BorderWidthTop = 0
                head.BorderWidthBottom = 1
                pTable1.AddCell(head)
```

head.BorderWidthBottom = 1

```
                head = New PdfPCell(New
Phrase("TOTAL", times2))
                head.Colspan = 1
                head.HorizontalAlignment =
Element.ALIGN_CENTER
                head.VerticalAlignment =
Element.ALIGN_MIDDLE
                head.BorderWidthLeft = 0
                head.BorderWidthRight = 0
                head.BorderWidthTop = 0
                head.BorderWidthBottom = 1
                pTable1.AddCell(head)

                head = New PdfPCell(New
Phrase("GRAND TOTAL", times2))
                head.Colspan = 1
                head.HorizontalAlignment =
Element.ALIGN_CENTER
                head.VerticalAlignment =
Element.ALIGN_MIDDLE
                head.BorderWidthLeft = 0
                head.BorderWidthRight = 0
                head.BorderWidthTop = 0
                head.BorderWidthBottom = 1
                pTable1.AddCell(head)
                '-----End Table Header-----

                nameD2 =
dTable2.Rows(i).Item("Name")
                nameD2 += Environment.NewLine
                nameD2 +=
dTable.Rows(x).Item("Employee Number")

                cell = New PdfPCell(New
Phrase(nameD2, times))
                cell.Colspan = 8
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_LEFT
                cell.BorderWidthTop = 1
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                pTable1.AddCell(cell)

                n =
Integer.Parse(dTable2.Rows(i).Item("NCount"))

                For j As Integer = 0 To n - 1

                    x += 1

                    '-----Date-----

                    If x = 1 Then

                        text = dTable.Rows(x -
1).Item("DATE")

                    Else

                        If dTable.Rows(x -
1).Item("DATE").Equals(dTable.Rows(x -
2).Item("DATE")) Then

                            text = " "

                        Else

                            text = dTable.Rows(x -
1).Item("DATE")

                        End If

                    End If

                    cell = New PdfPCell(New
Phrase(text, times))
                    cell.Colspan = 1
                    cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                    cell.HorizontalAlignment =
Element.ALIGN_CENTER
                    cell.BorderWidthTop = 0
                    cell.BorderWidthBottom = 0
                    cell.BorderWidthLeft = 0
                    cell.BorderWidthRight = 0
                    pTable1.AddCell(cell)

                    '-----Basic-----
                    If x = 1 Then

                        text = dTable.Rows(x -
1).Item("BASIC")

                    Else

                        If dTable.Rows(x -
1).Item("BASIC").Equals(dTable.Rows(x -
2).Item("BASIC")) And dTable.Rows(x -
1).Item("DATE").Equals(dTable.Rows(x -
2).Item("DATE")) Then

                            text = " "

                        Else

                            text = dTable.Rows(x -
1).Item("BASIC")

                        End If

                    End If


                    cell = New PdfPCell(New
Phrase(text, times))
                    cell.Colspan = 1
                    cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                    cell.HorizontalAlignment =
Element.ALIGN_CENTER
                    cell.BorderWidthTop = 0
                    cell.BorderWidthBottom = 0
                    cell.BorderWidthLeft = 0
                    cell.BorderWidthRight = 0
                    pTable1.AddCell(cell)

                    '-----COLA-----
                    If x = 1 Then

                        text = dTable.Rows(x -
1).Item("COLA")

                    Else
```

```
                        If dTable.Rows(x -
1).Item("COLA").Equals(dTable.Rows(x -
2).Item("COLA")) And dTable.Rows(x -
1).Item("DATE").Equals(dTable.Rows(x -
2).Item("DATE")) Then

                            text = " "

                        Else

                            text = dTable.Rows(x -
1).Item("COLA")

                        End If

                    End If



                    cell = New PdfPCell(New
Phrase(text, times))
                    cell.Colspan = 1
                    cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                    cell.HorizontalAlignment =
Element.ALIGN_CENTER
                    cell.BorderWidthTop = 0
                    cell.BorderWidthBottom = 0
                    cell.BorderWidthLeft = 0
                    cell.BorderWidthRight = 0
                    pTable1.AddCell(cell)

                    '-----DURATION-----
                    text = dTable.Rows(x -
1).Item("DURATION")

                    cell = New PdfPCell(New
Phrase(text, times))
                    cell.Colspan = 1
                    cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                    cell.HorizontalAlignment =
Element.ALIGN_CENTER
                    cell.BorderWidthTop = 0
                    cell.BorderWidthBottom = 0
                    cell.BorderWidthLeft = 0
                    cell.BorderWidthRight = 0
                    pTable1.AddCell(cell)

                    '-----ADJUSTMENT POLICY-----
                    text = dTable.Rows(x -
1).Item("ADJUSTMENT POLICY")

                    cell = New PdfPCell(New
Phrase(text, times))
                    cell.Colspan = 1
                    cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                    cell.HorizontalAlignment =
Element.ALIGN_CENTER
                    cell.BorderWidthTop = 0
                    cell.BorderWidthBottom = 0
                    cell.BorderWidthLeft = 0
                    cell.BorderWidthRight = 0
                    pTable1.AddCell(cell)

                    '-----RATE-----
                    text = dTable.Rows(x -
1).Item("RATE")

                    cell = New PdfPCell(New
Phrase(text, times))
                    cell.Colspan = 1
                    cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                    cell.HorizontalAlignment =
Element.ALIGN_CENTER
                    cell.BorderWidthTop = 0
                    cell.BorderWidthBottom = 0
                    cell.BorderWidthLeft = 0
                    cell.BorderWidthRight = 0
                    pTable1.AddCell(cell)

                    '-----TOTAL-----
                    text = dTable.Rows(x -
1).Item("TOTAL")

                    cell = New PdfPCell(New
Phrase(text, times))
                    cell.Colspan = 1
                    cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                    cell.HorizontalAlignment =
Element.ALIGN_CENTER
                    cell.BorderWidthTop = 0
                    cell.BorderWidthBottom = 0
                    cell.BorderWidthLeft = 0
                    cell.BorderWidthRight = 0
                    pTable1.AddCell(cell)

                    '-----GRAND TOTAL-----
                    Dim result() As DataRow
                    result = dTable3.Select("Name
= '" & dTable.Rows(x - 1).Item("Name") & "' AND
Date = '" & dTable.Rows(x - 1).Item("ODate") &
"'")

                    text = result(0)(2)

                    If x = 1 Then

                        text = result(0)(2)

                    Else

                        If dTable.Rows(x -
1).Item("DATE").Equals(dTable.Rows(x -
2).Item("DATE")) Then

                            text = " "

                        Else

                            text = result(0)(2)

                        End If

                    End If

                    cell = New PdfPCell(New
Phrase(text, times))
                    cell.Colspan = 1
                    cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                    cell.HorizontalAlignment =
Element.ALIGN_CENTER
                    cell.BorderWidthTop = 0
                    cell.BorderWidthBottom = 0
```

```
                    cell.BorderWidthLeft = 0
                    cell.BorderWidthRight = 0
                    pTable1.AddCell(cell)

                Next

                text = "GRAND TOTAL"

                cell = New PdfPCell(New
Phrase(text, times2))
                cell.Colspan = 7
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                cell.BorderWidthTop = 1
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                pTable1.AddCell(cell)

                text =
dTable4.Rows(i).Item("TOTAL")

                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                cell.BorderWidthTop = 1
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                pTable1.AddCell(cell)

                text = " "

                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 8
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                cell.BorderWidthTop = 0
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                pTable1.AddCell(cell)

                pdfCompensation.Add(pTable1)

                pdfCompensation.NewPage()

            Next

            '-----End of Contents-----


            pdfCompensation.Close()

            '-----Opens the PDF-----
            System.Diagnostics.Process.Start(path)


        Catch ex As Exception
            MsgBox(ex.Message)

        End Try

    End Sub

    Private Sub SetDataTable()

        If SQL.SQLDS IsNot Nothing Then

            SQL.SQLDS.Clear()

        End If

        qstring = "SELECT " & _
                        "EMP.Name AS [NAME], " & _
                        "EMP.EmployeeNumber AS
[Employee Number], " & _
                        "CONVERT(VARCHAR(6),
COMP.DATE, 107) AS [DATE], " & _

"CONVERT(VARCHAR(10),COMP.Date,120) AS [ODate], "
& _
                        "COMP.BasicPay AS [BASIC],
" & _
                        "COMP.COLA AS [COLA], " &
_
                        "CONVERT(DECIMAL(10,2),
(COMP.Duration/60.0)) AS [DURATION], " & _
                        "ADJ.Code AS [ADJUSTMENT
POLICY], " & _
                        "COMP.Rate AS [RATE], " &
_

"CONVERT(DECIMAL(10,2),COMP.Total) AS [TOTAL] " &
_
                    "FROM tCompensationList AS COMP
" & _
                        "INNER JOIN tEmployee AS
EMP ON EMP.ID = COMP.ID_Employee " & _
                        "INNER JOIN
tAdjustmentPolicy AS ADJ ON ADJ.ID =
COMP.ID_AdjPolicy " & _
                    "WHERE EMP.ID_Branch = '" &
AttendanceInput.BranchCB.SelectedValue & "' AND "
& _
                        "EMP.ID_Department = '" &
AttendanceInput.DeptCB.SelectedValue & "' AND " &
_
                        "EMP.ID_Designation = '" &
AttendanceInput.DesigCB.SelectedValue & "' AND " &
_
                        "COMP.Date BETWEEN '" &
AttendanceInput.StartDTP.Value.Date & "' AND '" &
AttendanceInput.EndDTP.Value.Date & "' " & _
                    "ORDER BY [NAME], [DATE]"

        SQL.RunQuery(qstring)

        SQL.SQLDA.Fill(dTable)

    End Sub

    Private Sub SetGroups()

        If SQL.SQLDS IsNot Nothing Then

            SQL.SQLDS.Clear()
```

```
        End If

        '-----Get Number of Rows per Employee-----
        qstring = "SELECT " & _
                        "Emp.Name AS Name, " & _
                        "COUNT(*) AS NCount " & _
                  "FROM tCompensationList AS tA "
& _
                        "INNER JOIN tEmployee AS
Emp ON Emp.ID = tA.ID_Employee " & _
                        "INNER JOIN tBranch AS
Branch ON Branch.ID = Emp.ID_Branch " & _
                        "INNER JOIN tDepartment AS
Dept ON Dept.ID = Emp.ID_Department " & _
                        "INNER JOIN tDesignation
AS Desig ON Desig.ID = Emp.ID_Designation " & _
                        "INNER JOIN
tAdjustmentPolicy AS ADJ ON ADJ.ID =
tA.ID_AdjPolicy " & _
                  "WHERE " & _
                        "Branch.ID = '" &
AttendanceInput.BranchCB.SelectedValue & "' AND "
& _
                        "Dept.ID = '" &
AttendanceInput.DeptCB.SelectedValue & "' AND " &
_
                        "Desig.ID = '" &
AttendanceInput.DesigCB.SelectedValue & "' AND " &
_
                        "tA.Date BETWEEN '" &
AttendanceInput.StartDTP.Value.Date & "' AND '" &
AttendanceInput.EndDTP.Value.Date & "' " & _
                  "GROUP BY Emp.Name " & _
                  "ORDER BY Name"

        SQL.RunQuery(qstring)

        SQL.SQLDA.Fill(dTable2)

    End Sub

    Private Sub GetTotalForDate()

        If SQL.SQLDS IsNot Nothing Then

            SQL.SQLDS.Clear()

        End If

        '-----Get Sum for the Day-----
        For i As Integer = 0 To dTable2.Rows.Count
- 1

            qstring = "SELECT " & _
                            "Emp.Name AS Name, " &
_
                            "ta.Date AS Date, " &
_

"CONVERT(DECIMAL(10,2), (SUM(ta.Total ))) AS SUM "
& _
                      "FROM tCompensationList AS
tA " & _
                            "INNER JOIN tEmployee
AS Emp ON Emp.ID = tA.ID_Employee " & _
                            "inner join
tAdjustmentPolicy as Adj on Adj.ID =
ta.ID_AdjPolicy " & _
                      "WHERE " & _
                            "emp.ID_Branch  = '" &
AttendanceInput.BranchCB.SelectedValue & "' AND "
& _
                            "emp.ID_Department =
'" & AttendanceInput.DeptCB.SelectedValue & "' AND
" & _
                            "emp.ID_Designation =
'" & AttendanceInput.DesigCB.SelectedValue & "'
AND " & _
                            "tA.Date BETWEEN '" &
AttendanceInput.StartDTP.Value.Date & "' AND '" &
AttendanceInput.EndDTP.Value.Date & "' " & _
                      "GROUP BY Emp.Name, tA.Date
" & _
                      "ORDER BY Name"

            SQL.RunQuery(qstring)

            SQL.SQLDA.Fill(dTable3)

        Next

    End Sub

    Private Sub GetGrandTotal()

        If SQL.SQLDS IsNot Nothing Then

            SQL.SQLDS.Clear()

        End If

        '-----Get Sum for the Day-----
        For i As Integer = 0 To dTable2.Rows.Count
- 1

            qstring = "SELECT " & _
                            "Emp.Name AS Name, " &
_

"CONVERT(DECIMAL(10,2), (SUM(ta.Total ))) AS TOTAL
" & _
                      "FROM tCompensationList AS
tA " & _
                            "INNER JOIN tEmployee
AS Emp ON Emp.ID = tA.ID_Employee " & _
                            "inner join
tAdjustmentPolicy as Adj on Adj.ID =
ta.ID_AdjPolicy " & _
                      "WHERE " & _
                            "emp.ID_Branch  = '" &
AttendanceInput.BranchCB.SelectedValue & "' AND "
& _
                            "emp.ID_Department =
'" & AttendanceInput.DeptCB.SelectedValue & "' AND
" & _
                            "emp.ID_Designation =
'" & AttendanceInput.DesigCB.SelectedValue & "'
AND " & _
                            "tA.Date BETWEEN '" &
AttendanceInput.StartDTP.Value.Date & "' AND '" &
AttendanceInput.EndDTP.Value.Date & "' " & _
                      "GROUP BY Emp.Name " & _
                      "ORDER BY Name"

            SQL.RunQuery(qstring)

            SQL.SQLDA.Fill(dTable4)
```

```
        Next

    End Sub


    Private Sub CreatePDF()

        '-----Sets Dates for Filename-----
        sDate =
AttendanceInput.StartDTP.Value.Date
        s = sDate.Replace("/", "")

        eDate = AttendanceInput.EndDTP.Value.Date
        e = eDate.Replace("/", "")

        '-----Sets data table-----
        SetDataTable()

        '-----Creating Pdf-----
        Dim filePath As Object
        Dim defaultPath As String
        defaultPath =
Application.StartupPath.ToString & "\Compensation"
& s & "-" & e &
AttendanceInput.BranchCB.SelectedItem("Name") &
".pdf"
        filePath = InputBox("Specify file path.",
"Print Compensation List", defaultPath)
        path = filePath

        pdfWrite =
PdfWriter.GetInstance(pdfCompensation, New
FileStream(path, FileMode.Create))

    End Sub

    Private Sub SetFont()

        '-----Set FONT-----
        bfTimes =
BaseFont.CreateFont(BaseFont.COURIER,
BaseFont.CP1252, False)
        times = New Font(bfTimes, 8)
        times2 = New Font(bfTimes, 8, Font.BOLD)

    End Sub

End Class

Imports iTextSharp.text
Imports iTextSharp.text.pdf
Imports System.IO

Public Class PrintAttendanceSummary

    Dim pdfAttendanceSummary
    Private SQL As New SQLControl
    Private qstring As String
    Private dTable As New DataTable
    Private printString As String
    Dim result1, result2 As Integer
    Dim bfTimes As BaseFont
    Dim times, times2 As Font
    Dim pdfWrite As PdfWriter
    Dim sDate, eDate, s, e, nameD2, text, path As
String
    Dim n, x As Integer

    Public Sub PrintPDFAttendanceSummary()

        Try

            dTable.Clear()

            pdfAttendanceSummary = New
Document(PageSize.LETTER.Rotate())

            SetDataTable()

            CreatePDF()

            SetFont()

            '-----Open PDF-----
            pdfAttendanceSummary.Open()

            '-----Report Header-----
            pdfAttendanceSummary.Add(New
Paragraph("Attendance Summary", times))
            pdfAttendanceSummary.Add(New
Paragraph("Period: " &
AttendanceInput.StartDTP.Value.Date & " to " &
AttendanceInput.EndDTP.Value.Date, times))
            pdfAttendanceSummary.Add(New
Paragraph("Branch: " &
AttendanceInput.BranchCB.SelectedItem("Name").ToSt
ring, times))
            pdfAttendanceSummary.Add(New
Paragraph("Department: " &
AttendanceInput.DeptCB.SelectedItem("Name").ToStri
ng, times))
            pdfAttendanceSummary.Add(New
Paragraph("Designation: " &
AttendanceInput.DesigCB.SelectedItem("Name").ToStr
ing, times))
            pdfAttendanceSummary.Add(New
Paragraph(" "))

            '-----Create PDF Table-----
            Dim pTable1 As PdfPTable

            '-----Size and Position of PDF Table--
---
            pTable1 = New PdfPTable(10)
            pTable1.TotalWidth = 100%
            pTable1.WidthPercentage = 100
            pTable1.TotalWidth =
pdfAttendanceSummary.PageSize.Width -
pdfAttendanceSummary.LeftMargin -
pdfAttendanceSummary.RightMargin

            pTable1.WriteSelectedRows(0, -1,
pdfAttendanceSummary.LeftMargin,
pdfAttendanceSummary.RightMargin,
pdfWrite.DirectContent)

            '-----Table Header-----
            Dim head As PdfPCell

            head = New PdfPCell(New Phrase(" ",
times2))
            head.Colspan = 3
            head.BorderWidthTop = 1
            head.BorderWidthBottom = 0
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            pTable1.AddCell(head)
```

```
            head = New PdfPCell(New Phrase("Leave
Days", times2))
            head.Colspan = 1
            head.BorderWidthTop = 1
            head.BorderWidthBottom = 0
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)

            head = New PdfPCell(New
Phrase("Regular", times2))
            head.Colspan = 1
            head.BorderWidthTop = 1
            head.BorderWidthBottom = 0
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)

            head = New PdfPCell(New Phrase("Rest
Day", times2))
            head.Colspan = 1
            head.BorderWidthTop = 1
            head.BorderWidthBottom = 0
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)

            head = New PdfPCell(New
Phrase("Special Holiday", times2))
            head.Colspan = 1
            head.BorderWidthTop = 1
            head.BorderWidthBottom = 0
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)

            head = New PdfPCell(New Phrase("Legal
Holiday", times2))
            head.Colspan = 1
            head.BorderWidthTop = 1
            head.BorderWidthBottom = 0
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)

            head = New PdfPCell(New
Phrase("Special Rest", times2))
            head.Colspan = 1
            head.BorderWidthTop = 1
            head.BorderWidthBottom = 0
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)

            head = New PdfPCell(New Phrase("Legal
Rest", times2))
            head.Colspan = 1
            head.BorderWidthTop = 1
            head.BorderWidthBottom = 0
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)

            head = New PdfPCell(New Phrase(" ",
times2))
            head.Colspan = 5
            head.BorderWidthTop = 1
            head.BorderWidthBottom = 0
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            pTable1.AddCell(head)

            head = New PdfPCell(New Phrase("WRD",
times2))
            head.Colspan = 1
            head.BorderWidthTop = 1
            head.BorderWidthBottom = 0
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)

            head = New PdfPCell(New Phrase("SH",
times2))
            head.Colspan = 1
            head.BorderWidthTop = 1
            head.BorderWidthBottom = 0
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)

            head = New PdfPCell(New Phrase("WRH",
times2))
            head.Colspan = 1
            head.BorderWidthTop = 1
            head.BorderWidthBottom = 0
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)
```

```
            head = New PdfPCell(New Phrase("WSR",
times2))
            head.Colspan = 1
            head.BorderWidthTop = 1
            head.BorderWidthBottom = 0
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)

            head = New PdfPCell(New Phrase("WRR",
times2))
            head.Colspan = 1
            head.BorderWidthTop = 1
            head.BorderWidthBottom = 0
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)

            head = New PdfPCell(New Phrase(" ",
times2))
            head.Colspan = 2
            head.BorderWidthTop = 0
            head.BorderWidthBottom = 0
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            pTable1.AddCell(head)

            head = New PdfPCell(New
Phrase("Absent", times2))
            head.Colspan = 1
            head.BorderWidthTop = 0
            head.BorderWidthBottom = 0
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)

            head = New PdfPCell(New Phrase("VL",
times2))
            head.Colspan = 1
            head.BorderWidthTop = 0
            head.BorderWidthBottom = 0
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)

            head = New PdfPCell(New Phrase("ND",
times2))
            head.Colspan = 1
            head.BorderWidthTop = 0
            head.BorderWidthBottom = 0
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)

            head = New PdfPCell(New
Phrase("WRDND", times2))
            head.Colspan = 1
            head.BorderWidthTop = 0
            head.BorderWidthBottom = 0
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)

            head = New PdfPCell(New Phrase("SHND",
times2))
            head.Colspan = 1
            head.BorderWidthTop = 0
            head.BorderWidthBottom = 0
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)

            head = New PdfPCell(New
Phrase("WRHND", times2))
            head.Colspan = 1
            head.BorderWidthTop = 0
            head.BorderWidthBottom = 0
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)

            head = New PdfPCell(New
Phrase("WSRND", times2))
            head.Colspan = 1
            head.BorderWidthTop = 0
            head.BorderWidthBottom = 0
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)

            head = New PdfPCell(New
Phrase("WRRND", times2))
            head.Colspan = 1
            head.BorderWidthTop = 0
            head.BorderWidthBottom = 0
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
```

```
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)

            head = New PdfPCell(New
Phrase("Employee Name", times2))
            head.Colspan = 1
            head.BorderWidthTop = 0
            head.BorderWidthBottom = 0
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)

            head = New PdfPCell(New
Phrase("Workhours", times2))
            head.Colspan = 1
            head.BorderWidthTop = 0
            head.BorderWidthBottom = 0
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)

            head = New PdfPCell(New
Phrase("Tardy", times2))
            head.Colspan = 1
            head.BorderWidthTop = 0
            head.BorderWidthBottom = 0
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)

            head = New PdfPCell(New Phrase("SL",
times2))
            head.Colspan = 1
            head.BorderWidthTop = 0
            head.BorderWidthBottom = 0
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)

            head = New PdfPCell(New Phrase("OT",
times2))
            head.Colspan = 1
            head.BorderWidthTop = 0
            head.BorderWidthBottom = 0
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)

            head = New PdfPCell(New Phrase("OTRD",
times2))
            head.Colspan = 1
            head.BorderWidthTop = 0
            head.BorderWidthBottom = 0
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)

            head = New PdfPCell(New Phrase("SHOT",
times2))
            head.Colspan = 1
            head.BorderWidthTop = 0
            head.BorderWidthBottom = 0
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)

            head = New PdfPCell(New Phrase("OTRH",
times2))
            head.Colspan = 1
            head.BorderWidthTop = 0
            head.BorderWidthBottom = 0
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)

            head = New PdfPCell(New Phrase("OTSR",
times2))
            head.Colspan = 1
            head.BorderWidthTop = 0
            head.BorderWidthBottom = 0
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)

            head = New PdfPCell(New Phrase("OTRR",
times2))
            head.Colspan = 1
            head.BorderWidthTop = 0
            head.BorderWidthBottom = 0
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)

            head = New PdfPCell(New
Phrase("Employee Number", times2))
            head.Colspan = 1
            head.BorderWidthTop = 0
```

```
            head.BorderWidthBottom = 1
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)

            head = New PdfPCell(New Phrase(" ",
times2))
            head.Colspan = 1
            head.BorderWidthTop = 0
            head.BorderWidthBottom = 1
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            pTable1.AddCell(head)

            head = New PdfPCell(New Phrase("UT",
times2))
            head.Colspan = 1
            head.BorderWidthTop = 0
            head.BorderWidthBottom = 1
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)

            head = New PdfPCell(New Phrase("PL",
times2))
            head.Colspan = 1
            head.BorderWidthTop = 0
            head.BorderWidthBottom = 1
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)

            head = New PdfPCell(New Phrase("OTND",
times2))
            head.Colspan = 1
            head.BorderWidthTop = 0
            head.BorderWidthBottom = 1
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)

            head = New PdfPCell(New
Phrase("OTRDND", times2))
            head.Colspan = 1
            head.BorderWidthTop = 0
            head.BorderWidthBottom = 1
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)

            head = New PdfPCell(New
Phrase("SHOTND", times2))
            head.Colspan = 1
            head.BorderWidthTop = 0
            head.BorderWidthBottom = 1
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)

            head = New PdfPCell(New
Phrase("OTRHND", times2))
            head.Colspan = 1
            head.BorderWidthTop = 0
            head.BorderWidthBottom = 1
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)

            head = New PdfPCell(New
Phrase("OTSRND", times2))
            head.Colspan = 1
            head.BorderWidthTop = 0
            head.BorderWidthBottom = 1
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)

            head = New PdfPCell(New
Phrase("OTRRND", times2))
            head.Colspan = 1
            head.BorderWidthTop = 0
            head.BorderWidthBottom = 1
            head.BorderWidthLeft = 0
            head.BorderWidthRight = 0
            head.HorizontalAlignment =
Element.ALIGN_CENTER
            head.VerticalAlignment =
Element.ALIGN_MIDDLE
            pTable1.AddCell(head)
            '-----End Table Header-----



            '-----Contents-----
            Dim cell As PdfPCell
            Dim text As String

            For i As Integer = 0 To
dTable.Rows.Count - 1

                cell = New PdfPCell(New Phrase("
", times))
                cell.Colspan = 5
                cell.BorderWidthTop = 0
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
```

```
                cell.BorderWidthRight = 0
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                pTable1.AddCell(cell)

                text = dTable.Rows(i).Item("WRD")
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthTop = 0
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                pTable1.AddCell(cell)

                text = dTable.Rows(i).Item("SH")
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthTop = 0
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                pTable1.AddCell(cell)

                text = dTable.Rows(i).Item("WRH")
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthTop = 0
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                pTable1.AddCell(cell)

                text = dTable.Rows(i).Item("WSR")
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthTop = 0
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                pTable1.AddCell(cell)

                text = dTable.Rows(i).Item("WRR")
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthTop = 0
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                pTable1.AddCell(cell)

                text = dTable.Rows(i).Item("NAME")
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthTop = 0
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                pTable1.AddCell(cell)

                text =
dTable.Rows(i).Item("WORKHOURS")
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthTop = 0
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                pTable1.AddCell(cell)

                text =
dTable.Rows(i).Item("ABSENT")
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthTop = 0
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                pTable1.AddCell(cell)

                text = dTable.Rows(i).Item("VL")
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthTop = 0
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                pTable1.AddCell(cell)

                text = dTable.Rows(i).Item("ND")
                cell = New PdfPCell(New
Phrase(text, times))
```

```
                cell.Colspan = 1
                cell.BorderWidthTop = 0
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                pTable1.AddCell(cell)

                text =
dTable.Rows(i).Item("WRDND")
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthTop = 0
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                pTable1.AddCell(cell)

                text = dTable.Rows(i).Item("SHND")
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthTop = 0
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                pTable1.AddCell(cell)

                text =
dTable.Rows(i).Item("WRHND")
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthTop = 0
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                pTable1.AddCell(cell)

                text =
dTable.Rows(i).Item("WSRND")
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthTop = 0
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                pTable1.AddCell(cell)

                text =
dTable.Rows(i).Item("WRRND")
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthTop = 0
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                pTable1.AddCell(cell)

                text =
dTable.Rows(i).Item("ENUMBER")
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthTop = 0
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                pTable1.AddCell(cell)

                text = " "
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthTop = 0
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                pTable1.AddCell(cell)

                text = dTable.Rows(i).Item("LATE")
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthTop = 0
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                pTable1.AddCell(cell)

                text = dTable.Rows(i).Item("SL")
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthTop = 0
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
```

```
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                pTable1.AddCell(cell)

                text = dTable.Rows(i).Item("OT")
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthTop = 0
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                pTable1.AddCell(cell)

                text = dTable.Rows(i).Item("OTRD")
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthTop = 0
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                pTable1.AddCell(cell)

                text = dTable.Rows(i).Item("SHOT")
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthTop = 0
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                pTable1.AddCell(cell)

                text = dTable.Rows(i).Item("OTRH")
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthTop = 0
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                pTable1.AddCell(cell)

                text = dTable.Rows(i).Item("OTSR")
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthTop = 0
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                pTable1.AddCell(cell)

                text = dTable.Rows(i).Item("OTRR")
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthTop = 0
                cell.BorderWidthBottom = 0
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                pTable1.AddCell(cell)

                text = " "
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 2
                cell.BorderWidthTop = 0
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                pTable1.AddCell(cell)

                text = dTable.Rows(i).Item("UT")
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthTop = 0
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                pTable1.AddCell(cell)

                text = dTable.Rows(i).Item("PL")
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthTop = 0
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                pTable1.AddCell(cell)

                text = dTable.Rows(i).Item("OTND")
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthTop = 0
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 0
```

```
                cell.BorderWidthRight = 0
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                pTable1.AddCell(cell)

                text =
dTable.Rows(i).Item("OTRDND")
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthTop = 0
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                pTable1.AddCell(cell)

                text =
dTable.Rows(i).Item("SHOTND")
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthTop = 0
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                pTable1.AddCell(cell)

                text =
dTable.Rows(i).Item("OTRHND")
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthTop = 0
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                pTable1.AddCell(cell)

                text =
dTable.Rows(i).Item("OTSRND")
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthTop = 0
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                pTable1.AddCell(cell)

                text =
dTable.Rows(i).Item("OTRRND")
                cell = New PdfPCell(New
Phrase(text, times))
                cell.Colspan = 1
                cell.BorderWidthTop = 0
                cell.BorderWidthBottom = 1
                cell.BorderWidthLeft = 0
                cell.BorderWidthRight = 0
                cell.HorizontalAlignment =
Element.ALIGN_CENTER
                cell.VerticalAlignment =
Element.ALIGN_MIDDLE
                pTable1.AddCell(cell)

            Next

            pdfAttendanceSummary.Add(pTable1)

            pdfAttendanceSummary.NewPage()
            pdfAttendanceSummary.Close()

            '-----Opens the PDF-----
            System.Diagnostics.Process.Start(path)

        Catch ex As Exception

            MsgBox(ex.Message)

        End Try

    End Sub

    Private Sub SetDataTable()

        If SQL.SQLDS IsNot Nothing Then

            SQL.SQLDS.Clear()

        End If

        qstring = "SELECT " & _
                        "EMP.Name AS NAME, " & _
                        "EMP.EmployeeNumber AS
ENUMBER, " & _
                        "CONVERT(DECIMAL(10,2),
(SUM(tA.RegularHours)/60.0)) AS [WORKHOURS], " & _
                        "ISNULL(SUM(CASE tA.Absent
WHEN 1 THEN 1 END), 0) AS ABSENT, " & _
                        "SUM(TA.Late) AS LATE, " &
_
                        "SUM(TA.UT) AS UT, " & _
                        "ISNULL(SUM(CASE tA.VL
WHEN 1 THEN 1 END), 0) AS VL, " & _
                        "ISNULL(SUM(CASE tA.SL
WHEN 1 THEN 1 END), 0) AS SL, " & _
                        "ISNULL(SUM(CASE tA.PL
WHEN 1 THEN 1 END), 0) AS PL, " & _
                        "CONVERT(DECIMAL(10,2),
(SUM(TA.ND))/60.00) AS ND, " & _
                        "CONVERT(DECIMAL(10,2),
(SUM(TA.OT))/60.00) AS OT, " & _
                        "CONVERT(DECIMAL(10,2),
(SUM(TA.OTND))/60.00) AS OTND, " & _
                        "CONVERT(DECIMAL(10,2),
(SUM(TA.WRD))/60.00) AS WRD, " & _
                        "CONVERT(DECIMAL(10,2),
(SUM(TA.WRDND))/60.00) AS WRDND, " & _
                        "CONVERT(DECIMAL(10,2),
(SUM(TA.OTRD))/60.00) AS OTRD, " & _
```

```
                        "CONVERT(DECIMAL(10,2),
(SUM(TA.OTRDND))/60.00) AS OTRDND, " & _
                        "CONVERT(DECIMAL(10,2),
(SUM(TA.SH))/60.00) AS SH, " & _
                        "CONVERT(DECIMAL(10,2),
(SUM(TA.SHND))/60.00) AS SHND, " & _
                        "CONVERT(DECIMAL(10,2),
(SUM(TA.SHOT))/60.00) AS SHOT, " & _
                        "CONVERT(DECIMAL(10,2),
(SUM(TA.SHOTND))/60.00) AS SHOTND, " & _
                        "CONVERT(DECIMAL(10,2),
(SUM(TA.WRH))/60.00) AS WRH, " & _
                        "CONVERT(DECIMAL(10,2),
(SUM(TA.WRHND))/60.00) AS WRHND, " & _
                        "CONVERT(DECIMAL(10,2),
(SUM(TA.OTRH))/60.00) AS OTRH, " & _
                        "CONVERT(DECIMAL(10,2),
(SUM(TA.OTRHND))/60.00) AS OTRHND, " & _
                        "CONVERT(DECIMAL(10,2),
(SUM(TA.WSR))/60.00) AS WSR, " & _
                        "CONVERT(DECIMAL(10,2),
(SUM(TA.WSRND))/60.00) AS WSRND, " & _
                        "CONVERT(DECIMAL(10,2),
(SUM(TA.OTSR))/60.00) AS OTSR, " & _
                        "CONVERT(DECIMAL(10,2),
(SUM(TA.OTSRND))/60.00) AS OTSRND, " & _
                        "CONVERT(DECIMAL(10,2),
(SUM(TA.WRR))/60.00) AS WRR, " & _
                        "CONVERT(DECIMAL(10,2),
(SUM(TA.WRRND))/60.00) AS WRRND, " & _
                        "CONVERT(DECIMAL(10,2),
(SUM(TA.OTRR))/60.00) AS OTRR, " & _
                        "CONVERT(DECIMAL(10,2),
(SUM(TA.OTRRND))/60.00) AS OTRRND " & _
                    "FROM tAttendanceComputation AS
TA " & _
                        "INNER JOIN tEmployee AS
EMP ON EMP.ID = TA.ID_Employee " & _
                    "WHERE " & _
                        "EMP.ID_Branch = '" &
AttendanceInput.BranchCB.SelectedValue & "' AND "
& _
                        "EMP.ID_Department = '" &
AttendanceInput.DeptCB.SelectedValue & "' AND " &
_
                        "EMP.ID_Designation = '" &
AttendanceInput.DesigCB.SelectedValue & "' AND " &
_
                        "TA.Date BETWEEN '" &
AttendanceInput.StartDTP.Value.Date & "' AND '" &
AttendanceInput.EndDTP.Value.Date & "' " & _
                    "GROUP BY EMP.Name,
EMP.EmployeeNumber "

        SQL.RunQuery(qstring)

        SQL.SQLDA.Fill(dTable)

    End Sub

    Private Sub CreatePDF()

        '-----Sets Dates for Filename-----
        sDate =
AttendanceInput.StartDTP.Value.Date
        s = sDate.Replace("/", "")

        eDate = AttendanceInput.EndDTP.Value.Date
        e = eDate.Replace("/", "")

        '-----Creating Pdf-----
        Dim filePath As Object
        Dim defaultPath As String
        defaultPath =
Application.StartupPath.ToString & "\AS" & s & "-"
& e &
AttendanceInput.BranchCB.SelectedItem("Name") &
".pdf"
        filePath = InputBox("Specify file path.",
"Print Attendance Summary", defaultPath)
        path = filePath

        pdfWrite =
PdfWriter.GetInstance(pdfAttendanceSummary, New
FileStream(path, FileMode.Create))

    End Sub

    Private Sub SetFont()

        '-----Set FONT-----
        bfTimes =
BaseFont.CreateFont(BaseFont.COURIER,
BaseFont.CP1252, False)
        times = New Font(bfTimes, 8)
        times2 = New Font(bfTimes, 8, Font.BOLD)

    End Sub

End Class

Public Class SpecialHoliday

    Private Duplicate As New CheckDuplicates
    Private Trail As New AuditTrail
    Private SQL As New SQLControl
    Private QString As String 'Query String for
insert or update a special holiday.
    Private AString As String 'Query String for
Audit Trail

    Private Sub SpecialHoliday_Load(ByVal sender
As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load

        SQL.RunQuery("SELECT ID, City from
tLocation")

        SHolidayLocCB.DataSource =
SQL.SQLDS.Tables(0)
        SHolidayLocCB.ValueMember = "ID"
        SHolidayLocCB.DisplayMember = "City"

        If MainWindow.HasRow = True Then

            NSpecialHolidayTB.Text =
MainWindow.NameTB
            SpecialHolidayDTP.Value =
MainWindow.dateTB
            ASHolidayChB.Checked =
MainWindow.isActive
            CommentTB.Text = MainWindow.Comment
            SHolidayLocCB.SelectedIndex =
SHolidayLocCB.FindStringExact(MainWindow.locationT
B)

        End If
```

```
    End Sub


    Private Sub CancelBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles CancelBtn.Click

        Me.Close()

    End Sub

    Private Sub SaveBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles SaveBtn.Click

        Try

            If NSpecialHolidayTB.Text = "" Then

                MsgBox("Enter special holiday
name.", MsgBoxStyle.Information)

            Else

                Dim DString As String

                DString = "SELECT * FROM
tSpecialHoliday WHERE Name = '" &
NSpecialHolidayTB.Text & "' AND Date = '" &
SpecialHolidayDTP.Value.Date & "' AND ID_Location
= '" & SHolidayLocCB.SelectedValue & "' AND
isActive = '" & ASHolidayChB.Checked & "'"

                If Duplicate.IsDuplicated(DString)
= True Then

                    MsgBox("Special Holiday
already exists for the said date!",
MsgBoxStyle.Critical)

                    Exit Sub

                Else

                    If MainWindow.HasRow = False
Then

                        QString = "INSERT INTO
tSpecialHoliday (Name, Date, ID_Location,
isActive, Comment) " & _
                                  "VALUES " & _
                                        "(" & _
                                        "'" &
NSpecialHolidayTB.Text & "', " & _
                                        "'" &
SpecialHolidayDTP.Value.Date & "', " & _
                                        "'" &
SHolidayLocCB.SelectedValue & "', " & _
                                        "'" &
ASHolidayChB.Checked & "', " & _
                                        "'" &
CommentTB.Text & "'" & _
                                        ")"

                        AString = "Inserted " &
NSpecialHolidayTB.Text & " special holiday on " &
SpecialHolidayDTP.Value.Date & "."

                    ElseIf MainWindow.HasRow =
True Then

                        QString = "UPDATE
tSpecialHoliday SET " & _
                                        "Name = '"
& NSpecialHolidayTB.Text & "', " & _
                                        "Date = '"
& SpecialHolidayDTP.Value.Date & "', " & _

"ID_Location = '" & SHolidayLocCB.SelectedValue &
"', " & _
                                        "isActive
= '" & ASHolidayChB.Checked & "', " & _
                                        "Comment =
'" & CommentTB.Text & "' " & _
                                    "WHERE " & _
                                        "ID = '" &
MainWindow.ID & "'"

                        AString = "Updated " &
NSpecialHolidayTB.Text & " special holiday on " &
SpecialHolidayDTP.Value.Date & "."

                    End If

                    SQL.RunQuery(QString)

                    If
Duplicate.IsDuplicated(DString) = True Then

                        Trail.AddTrail(AString)

                        MsgBox("Saved!",
MsgBoxStyle.Information)

                        Me.Close()

                        MainWindow.LoadDataGrid()

                    Else

                        MsgBox("Special holiday
not saved!", MsgBoxStyle.Information)

                    End If

                End If

            End If

        Catch ex As Exception

            MsgBox(ex.Message,
MsgBoxStyle.Critical)

        End Try

    End Sub

    Public Sub SelectSpecialHoliday()

        MainWindow.NewBtn.Visible = True
        MainWindow.EditBtn.Visible = True
        MainWindow.RefreshBtn.Visible = True
        MainWindow.DateTimePicker1.Visible = False
        MainWindow.Label2.Visible = False
        MainWindow.DeleteBtn.Visible = True
```

```
        MainWindow.GroupBox.Visible = True
        MainWindow.GroupBox.Text = "Special
Holiday"

        MainWindow.QString = "EXEC
selectSpecialHoliday"

        MainWindow.LoadDataGrid()


MainWindow.DataGrid.Columns("ID").DisplayIndex = 0
        MainWindow.DataGrid.Columns("Special
Holiday").DisplayIndex = 1

MainWindow.DataGrid.Columns("Date").DisplayIndex =
2

MainWindow.DataGrid.Columns("Location").DisplayInd
ex = 3

MainWindow.DataGrid.Columns("Active").DisplayIndex
= 4

MainWindow.DataGrid.Columns("Comment").DisplayInde
x = 5

        Trail.AddTrail("Viewed Special Holiday.")

    End Sub

    Public Sub DeleteSpecialHoliday()

        Dim result1 As DialogResult =
MessageBox.Show("Are you sure to delete " &
MainWindow.NameTB & "?", _
                "Delete Special Holiday", _
                MessageBoxButtons.YesNo)

        If result1 =
Windows.Forms.DialogResult.Yes Then

            MainWindow.QString = "DELETE from
tSpecialHoliday WHERE ID = '" & MainWindow.ID &
"'"

            MainWindow.LoadDataGrid()

            Trail.AddTrail("Deleted " &
MainWindow.NameTB & " from Special Holiday.")

            SelectSpecialHoliday()

        End If

    End Sub

End Class
Public Class LegalHoliday

    Private Duplicate As New CheckDuplicates
    Private Trail As New AuditTrail
    Private SQL As New SQLControl
    Private QString As String 'Query String for
insert or update a legal holiday.
    Private AString As String 'Query String for
Audit Trail

    Private Sub SaveBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles SaveBtn.Click

        Try

            If NLegalHolidayTB.Text = "" Then

                MsgBox("Enter legal holiday
name.", MsgBoxStyle.Information)

            Else

                Dim DString As String

                DString = "SELECT * FROM
tLegalHoliday WHERE Name = '" &
NLegalHolidayTB.Text & "' AND Date = '" &
LegalHolidayDTP.Value.Date & "' AND isActive = '"
& ALHolidayChB.Checked & "'"

                If Duplicate.IsDuplicated(DString)
= True Then

                    MsgBox("Legal Holiday already
exists for the said date!", MsgBoxStyle.Critical)

                    Exit Sub

                Else

                    If MainWindow.HasRow = False
Then

                        QString = "INSERT INTO
tLegalHoliday (Name, Date, isActive, Comment) " &
_
                                    "VALUES " & _
                                          "(" & _
                                          "'" &
NLegalHolidayTB.Text & "', " & _
                                          "'" &
LegalHolidayDTP.Value.Date & "', " & _
                                          "'" &
ALHolidayChB.Checked & "', " & _
                                          "'" &
CommentTB.Text & "'" & _
                                          ")"

                        AString = "Inserted " &
NLegalHolidayTB.Text & " legal holiday on " &
LegalHolidayDTP.Value.Date & "."

                    ElseIf MainWindow.HasRow =
True Then

                        QString = "UPDATE
tLegalHoliday SET " & _
                                          "Name = '"
& NLegalHolidayTB.Text & "', " & _
                                          "Date = '"
& LegalHolidayDTP.Value.Date & "', " & _
                                          "isActive
= '" & ALHolidayChB.Checked & "', " & _
                                          "Comment =
'" & CommentTB.Text & "' " & _
                                    "WHERE " & _
                                          "ID = '" &
MainWindow.ID & "'"
```

```
                        AString = "Updated " &
NLegalHolidayTB.Text & " legal holiday on " &
LegalHolidayDTP.Value.Date & "."

                    End If

                    SQL.RunQuery(QString)

                    If
Duplicate.IsDuplicated(DString) = True Then

                        Trail.AddTrail(AString)

                        MsgBox("Saved!",
MsgBoxStyle.Information)

                        Me.Close()

                        MainWindow.LoadDataGrid()

                    Else

                        MsgBox("Legal Holiday not
saved!", MsgBoxStyle.Information)

                    End If

                End If

            End If

        Catch ex As Exception

            MsgBox(ex.Message,
MsgBoxStyle.Critical)

        End Try

    End Sub

    Private Sub CancelBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles CancelBtn.Click

        Me.Close()

    End Sub

    Public Sub SelectLegalHoliday()

        MainWindow.NewBtn.Visible = True
        MainWindow.EditBtn.Visible = True
        MainWindow.RefreshBtn.Visible = True
        MainWindow.DateTimePicker1.Visible = False
        MainWindow.DeleteBtn.Visible = True
        MainWindow.Label2.Visible = False

        MainWindow.GroupBox.Visible = True
        MainWindow.GroupBox.Text = "Legal Holiday"

        MainWindow.QString = "EXEC
selectLegalHoliday"

        MainWindow.LoadDataGrid()


MainWindow.DataGrid.Columns("ID").DisplayIndex = 0

        MainWindow.DataGrid.Columns("Legal
Holiday").DisplayIndex = 1

MainWindow.DataGrid.Columns("Date").DisplayIndex =
2

MainWindow.DataGrid.Columns("Active").DisplayIndex
= 3

MainWindow.DataGrid.Columns("Comment").DisplayInde
x = 4

        Trail.AddTrail("Viewed Legal Holiday.")

    End Sub

    Public Sub DeleteLegalHoliday()

        Dim result1 As DialogResult =
MessageBox.Show("Are you sure to delete " &
MainWindow.NameTB & "?", _
                "Delete Legal Holiday", _
                MessageBoxButtons.YesNo)

        If result1 =
Windows.Forms.DialogResult.Yes Then

            MainWindow.QString = "DELETE from
tLegalHoliday WHERE ID = '" & MainWindow.ID & "'"

            MainWindow.LoadDataGrid()

            Trail.AddTrail("Deleted " &
MainWindow.NameTB & " from Legal Holiday.")

            SelectLegalHoliday()

        End If

    End Sub

    Private Sub LegalHoliday_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load

        If MainWindow.HasRow = True Then

            NLegalHolidayTB.Text =
MainWindow.NameTB
            LegalHolidayDTP.Value =
MainWindow.dateTB
            ALHolidayChB.Checked =
MainWindow.isActive
            CommentTB.Text = MainWindow.Comment

        End If

    End Sub

End Class
Imports System.Data.Sql
Imports System.Data.SqlClient

Public Class SQLControl

    Private SQLCon As New SqlConnection
    Private SQLCmd As SqlCommand
    Public SQLDA As SqlDataAdapter
    Public SQLDS As DataSet
```

```
Private connS As String

Public ServerTxt As String
Public DBTxt As String
Public DBUserTxt As String
Public DBPassTxt As String
Public intRows As Integer



Public Function SetConnString() As String

    ServerTxt = My.Settings.ServerTXT
    DBTxt = My.Settings.DBTXT
    DBUserTxt = My.Settings.DBUserTXT
    DBPassTxt = My.Settings.DBPassTXT

    connS = "Server="
    connS += ServerTxt
    connS += ";Database="
    connS += DBTxt
    connS += ";User="
    connS += DBUserTxt
    connS += ";Pwd="
    connS += DBPassTxt

    Return connS

End Function

Public Sub SetSQLConn()

    SQLCon = New SqlConnection(SetConnString)

End Sub

Public Function HasConnection() As Boolean

    SetSQLConn()

    Try

        SQLCon.Open()
        SQLCon.Close()

        Return True

    Catch ex As Exception

        MsgBox(ex.Message)

    End Try

    Return False

End Function

Public Sub RunQuery(ByVal Query As String)

    SetSQLConn()

    Try
        SQLCon.Open()

        SQLCmd = New SqlCommand(Query, SQLCon)

        SQLDA = New SqlDataAdapter(SQLCmd)
        SQLDS = New DataSet
        SQLDA.Fill(SQLDS)

            SQLCon.Close()

        Catch ex As Exception

            MsgBox(ex.Message)

            If SQLCon.State = ConnectionState.Open
Then

                SQLCon.Close()

            End If

        End Try

    End Sub

End Class

Imports System.IO

Public Class MainWindow

    Private Log As New Form1
    Private Trail As New AuditTrail
    Private SQL As New SQLControl
    Public QString As String
    Private VBranch As New Branch
    Private VDepartment As New Department
    Private VDesignation As New Designation
    Private VEStatus As New EmployeeStatus
    Private VLocation As New Location1
    Private VViolation As New Violation
    Private VAdjPolicy As New AjustmentType
    Private VLeaveCredit As New LeaveCredit
    Private VDailySchedule As New DailySchedule
    Private VWeeklySchedule As New WeeklySchedule
    Private VLegalHoliday As New LegalHoliday
    Private VSpecialHoliday As New SpecialHoliday
    Private VEmployee As New Employee
    Private VEmployeeWithViolation As New
EmployeeWithViolation
    Private VEmployeeLeaveCredit As New
EmployeeLeaveCredit
    Private VFCRD As New FileChangeRestDay
    Private VFCSS As New FileChangeShiftSchedule
    Private VFLOA As New FileLeaveofAbsence
    Private VFOT As New FileOverTime
    Private VFUT As New FileUnderTime
    Private VUser As New User
    Private VChangePassword As New ChangePassword
    Private VDAbsenteeism As New
PrintDailyAbsenteeism
    Private VMAbsenteeism As New
PrintMonthlyAbsenteeism
    Private VDOvertime As New PrintDailyOvertime
    Private VMOvertime As New PrintMonthlyOvertime
    Public HasRow As Boolean
    Public NameTB, Code, Rate, Comment, isActive,
ID, Duration, in1, out1, in2, out2, in3, out3,
sunday, monday, tuesday, wednesday, thursday,
friday, saturday, dateTB, locationTB, branchTB,
departmentTB, designationTB, employeestatusTB,
weeklysched, fname, lname, mname, enumber,
violationTB, leavetypeTB, totalnumberTB,
basicpayTB, penaltyTB, rankTB, descTB, reason,
approve, newschedTB, startdateTB, enddateTB,
```

```
printString, COLA, username, isApproved, isAdmin,
password, print As String
    Public in1d, out1d, in2d, out2d, in3d, out3d
As DateTime


    Private Sub ClearStrings()

        NameTB = ""
        Code = ""
        Rate = ""
        Comment = ""
        isActive = ""
        ID = ""
        Duration = ""
        in1 = ""
        out1 = ""
        in2 = ""
        out2 = ""
        in3 = ""
        out3 = ""
        sunday = ""
        monday = ""
        tuesday = ""
        wednesday = ""
        thursday = ""
        friday = ""
        saturday = ""
        dateTB = ""
        locationTB = ""
        branchTB = ""
        departmentTB = ""
        designationTB = ""
        employeestatusTB = ""
        weeklysched = ""
        fname = ""
        lname = ""
        mname = ""
        enumber = ""
        violationTB = ""
        leavetypeTB = ""
        totalnumberTB = ""
        basicpayTB = ""
        penaltyTB = ""
        rankTB = ""
        descTB = ""
        reason = ""
        approve = ""
        newschedTB = ""
        startdateTB = ""
        enddateTB = ""
        printString = ""
        COLA = ""
        username = ""

    End Sub

    Private Sub MainWindow_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load

        Label3.Text = My.Settings.USERTXT

        If Form1.isAdmin Then

            Dim MyNode() As TreeNode
            MyNode =
TreeView1.Nodes.Find("UMRoot", True)
            MyNode(0).Nodes.Add("Users")
            MyNode(0).Nodes.Add("System Audit
Trail")

        End If

    End Sub

    Public Sub SelectOvertime(ByVal Type As
String, ByVal Datei As String)

        NewBtn.Visible = False
        EditBtn.Visible = False
        RefreshBtn.Visible = True
        DeleteBtn.Visible = False
        Label2.Visible = True

        GroupBox.Visible = True
        GroupBox.Text = Type

        QString = "SELECT " & _
                        "tFileOvertime.ID, " & _
                        "tFileOvertime.Date, " & _
                        "Employee.Name AS
[Employee Name], " & _
                        "ENumber.EmployeeNumber AS
[Employee Number], " & _
                        "Branch.Name AS Branch, "
& _
                        "Dept.Name AS Department,
" & _
                        "Desig.Name AS
Designation, " & _
                        "EStatus.Name AS [Employee
Status], " & _
                        "tFileOvertime.Reason, " &
_
                        "tFileOvertime.isApproved
AS Approved, " & _
                        "tFileOvertime.Comment " &
_
                "FROM tFileOvertime " & _
                        "INNER JOIN tEmployee AS
Employee ON Employee.ID =
tFileOvertime.ID_Employee " & _
                        "INNER JOIN tEmployee AS
ENumber ON ENumber.ID = tFileOvertime.ID_Employee
" & _
                        "INNER JOIN tBranch AS
Branch ON Branch.ID = Employee.ID_Branch " & _
                        "INNER JOIN tDepartment AS
Dept ON Dept.ID = Employee.ID_Department " & _
                        "INNER JOIN tDesignation
AS Desig ON Desig.ID = Employee.ID_Designation " &
_
                        "INNER JOIN
tEmployeeStatus AS EStatus ON EStatus.ID =
Employee.ID_EmployeeStatus " & _
                "WHERE tFileOverTime.Date " &
Datei

        LoadDataGrid()

        DataGrid.Columns("ID").DisplayIndex = 0
        DataGrid.Columns("Date").DisplayIndex = 1
        DataGrid.Columns("Employee
Name").DisplayIndex = 2
        DataGrid.Columns("Employee
Number").DisplayIndex = 3
```

```
        DataGrid.Columns("Branch").DisplayIndex =
4

DataGrid.Columns("Department").DisplayIndex = 5

DataGrid.Columns("Designation").DisplayIndex = 6
        DataGrid.Columns("Employee
Status").DisplayIndex = 7
        DataGrid.Columns("Reason").DisplayIndex =
8
        DataGrid.Columns("Approved").DisplayIndex
= 9
        DataGrid.Columns("Comment").DisplayIndex =
10

        Trail.AddTrail("Viewed " & Type & "
Total.")

    End Sub

    Public Sub SelectAbsenteeism(ByVal Type As
String, ByVal Datei As String)

        NewBtn.Visible = False
        EditBtn.Visible = False
        RefreshBtn.Visible = True
        DeleteBtn.Visible = False
        Label2.Visible = True

        GroupBox.Visible = True
        GroupBox.Text = Type

        QString = "SELECT " & _

"tAttendanceComputation.ID, " & _

"tAttendanceComputation.Date, " & _
                        "Employee.Name AS
[Employee Name], " & _
                        "ENumber.EmployeeNumber AS
[Employee Number], " & _
                        "Branch.Name AS Branch, "
& _
                        "Dept.Name AS Department,
" & _
                        "Desig.Name AS
Designation, " & _
                        "EStatus.Name AS [Employee
Status] " & _
                  "FROM tAttendanceComputation " &
_
                        "INNER JOIN tEmployee AS
Employee ON Employee.ID =
tAttendanceComputation.ID_Employee " & _
                        "INNER JOIN tEmployee AS
ENumber ON ENumber.ID =
tAttendanceComputation.ID_Employee " & _
                        "INNER JOIN tBranch AS
Branch ON Branch.ID = Employee.ID_Branch " & _
                        "INNER JOIN tDepartment AS
Dept ON Dept.ID = Employee.ID_Department " & _
                        "INNER JOIN tDesignation
AS Desig ON Desig.ID = Employee.ID_Designation " &
_
                        "INNER JOIN
tEmployeeStatus AS EStatus ON EStatus.ID =
Employee.ID_EmployeeStatus " & _
                  "WHERE " & _
"tAttendanceComputation.Absent = 1 AND " & _

"tAttendanceComputation.Date " & Datei

        LoadDataGrid()

        DataGrid.Columns("ID").DisplayIndex = 0
        DataGrid.Columns("Date").DisplayIndex = 1
        DataGrid.Columns("Employee
Name").DisplayIndex = 2
        DataGrid.Columns("Employee
Number").DisplayIndex = 3
        DataGrid.Columns("Branch").DisplayIndex =
4

DataGrid.Columns("Department").DisplayIndex = 5

DataGrid.Columns("Designation").DisplayIndex = 6
        DataGrid.Columns("Employee
Status").DisplayIndex = 7

        Trail.AddTrail("Viewed " & Type)

    End Sub

    Private Sub TreeView1_AfterSelect(ByVal sender
As System.Object, ByVal e As
System.Windows.Forms.TreeViewEventArgs) Handles
TreeView1.AfterSelect

        Try

            Label3.Text = My.Settings.USERTXT

            If TreeView1.SelectedNode.Text =
"Adjustment Policy" Then


VAdjPolicy.SelectAdjustmentPolicy()

                ToolTip1.SetToolTip(NewBtn, "New
Adjustment Policy")
                ToolTip1.SetToolTip(EditBtn, "Edit
Adjustment Policy")
                ToolTip1.SetToolTip(DeleteBtn,
"Delete Adjustment Policy")
                ToolTip1.SetToolTip(RefreshBtn,
"Refresh Adjustment Policy List")
                ToolTip1.SetToolTip(SearchBox,
"Search Adjustment Policy")

            ElseIf TreeView1.SelectedNode.Text =
"Leave Credit" Then

                VLeaveCredit.SelectLeaveCredit()

                ToolTip1.SetToolTip(NewBtn, "New
Leave Credit")
                ToolTip1.SetToolTip(EditBtn, "Edit
Leave Credit")
                ToolTip1.SetToolTip(DeleteBtn,
"Delete Leave Credit")
                ToolTip1.SetToolTip(RefreshBtn,
"Refresh Leave Credit List")
                ToolTip1.SetToolTip(SearchBox,
"Search Leave Credit")
```

```
            ElseIf TreeView1.SelectedNode.Text =
"Daily Schedule" Then


VDailySchedule.SelectDailySchedule()

                ToolTip1.SetToolTip(NewBtn, "New
Daily Schedule")
                ToolTip1.SetToolTip(EditBtn, "Edit
Daily Schedule")
                ToolTip1.SetToolTip(DeleteBtn,
"Delete Daily Schedule")
                ToolTip1.SetToolTip(RefreshBtn,
"Refresh Daily Schedule List")
                ToolTip1.SetToolTip(SearchBox,
"Search Daily Schedule")

            ElseIf TreeView1.SelectedNode.Text =
"Weekly Schedule" Then


VWeeklySchedule.SelectWeeklySchedule()

                ToolTip1.SetToolTip(NewBtn, "New
Weekly Schedule")
                ToolTip1.SetToolTip(EditBtn, "Edit
Weekly Schedule")
                ToolTip1.SetToolTip(DeleteBtn,
"Delete Weekly Schedule")
                ToolTip1.SetToolTip(RefreshBtn,
"Refresh Weekly Schedule List")
                ToolTip1.SetToolTip(SearchBox,
"Search Weekly Schedule")

            ElseIf TreeView1.SelectedNode.Text =
"Change Password" Then

                VChangePassword.ShowDialog()

            ElseIf TreeView1.SelectedNode.Text =
"Legal Holiday" Then

                VLegalHoliday.SelectLegalHoliday()

                ToolTip1.SetToolTip(NewBtn, "New
Legal Holiday")
                ToolTip1.SetToolTip(EditBtn, "Edit
Legal Holiday")
                ToolTip1.SetToolTip(DeleteBtn,
"Delete Legal Holiday")
                ToolTip1.SetToolTip(RefreshBtn,
"Refresh Legal Holiday List")
                ToolTip1.SetToolTip(SearchBox,
"Search Legal Holiday")

            ElseIf TreeView1.SelectedNode.Text =
"Special Holiday" Then


VSpecialHoliday.SelectSpecialHoliday()

                ToolTip1.SetToolTip(NewBtn, "New
Special Holiday")
                ToolTip1.SetToolTip(EditBtn, "Edit
Special Holiday")
                ToolTip1.SetToolTip(DeleteBtn,
"Delete Special Holiday")
                ToolTip1.SetToolTip(RefreshBtn,
"Refresh Special Holiday List")
                ToolTip1.SetToolTip(SearchBox,
"Search Special Holiday")

            ElseIf TreeView1.SelectedNode.Text =
"Employees" Then

                VEmployee.SelectEmployee()

                ToolTip1.SetToolTip(NewBtn, "New
Employee")
                ToolTip1.SetToolTip(EditBtn, "Edit
Employee")
                ToolTip1.SetToolTip(DeleteBtn,
"Delete Employee")
                ToolTip1.SetToolTip(RefreshBtn,
"Refresh Employee List")
                ToolTip1.SetToolTip(SearchBox,
"Search Employee")

            ElseIf TreeView1.SelectedNode.Text =
"Employee with Violation" Then


VEmployeeWithViolation.SelectEmployeeWithViolation
()

                ToolTip1.SetToolTip(NewBtn, "New
Employee with Violation")
                ToolTip1.SetToolTip(EditBtn, "Edit
Employee with Violation")
                ToolTip1.SetToolTip(DeleteBtn,
"Delete Employee with Violation")
                ToolTip1.SetToolTip(RefreshBtn,
"Refresh Employee with Violation List")
                ToolTip1.SetToolTip(SearchBox,
"Search Employee with Violation")

            ElseIf TreeView1.SelectedNode.Text =
"Vacation Leave" Then


VEmployeeLeaveCredit.SelectEmployeeLeaveCredit("Va
cation Leave")

                EditBtn.Visible = False
                DeleteBtn.Visible = False

                ToolTip1.SetToolTip(NewBtn, "New
Vacation Leave Credit")
                ToolTip1.SetToolTip(RefreshBtn,
"Refresh Vacation Leave Credit List")
                ToolTip1.SetToolTip(SearchBox,
"Search Vacation Leave Credit")

            ElseIf TreeView1.SelectedNode.Text =
"Sick Leave" Then


VEmployeeLeaveCredit.SelectEmployeeLeaveCredit("Si
ck Leave")

                EditBtn.Visible = False
                DeleteBtn.Visible = False

                ToolTip1.SetToolTip(NewBtn, "New
Sick Leave Credit")
                ToolTip1.SetToolTip(RefreshBtn,
"Refresh Sick Leave Credit List")
```

```
                ToolTip1.SetToolTip(SearchBox,
"Search Sick Leave Credit")

            ElseIf TreeView1.SelectedNode.Text =
"Paternity Leave" Then


VEmployeeLeaveCredit.SelectEmployeeLeaveCredit("Pa
ternity Leave")

                EditBtn.Visible = False
                DeleteBtn.Visible = False

                ToolTip1.SetToolTip(NewBtn, "New
Paternity Leave Credit")
                ToolTip1.SetToolTip(RefreshBtn,
"Refresh Paternity Leave Credit List")
                ToolTip1.SetToolTip(SearchBox,
"Search Paternity Leave Credit")

            ElseIf TreeView1.SelectedNode.Text =
"Branch" Then

                VBranch.SelectBranch()

                ToolTip1.SetToolTip(NewBtn, "New
Branch")
                ToolTip1.SetToolTip(EditBtn, "Edit
Branch")
                ToolTip1.SetToolTip(DeleteBtn,
"Delete Branch")
                ToolTip1.SetToolTip(RefreshBtn,
"Refresh Branch List")
                ToolTip1.SetToolTip(SearchBox,
"Search Branch")

            ElseIf TreeView1.SelectedNode.Text =
"Location" Then

                VLocation.SelectLocation()

                ToolTip1.SetToolTip(NewBtn, "New
Location")
                ToolTip1.SetToolTip(EditBtn, "Edit
Location")
                ToolTip1.SetToolTip(DeleteBtn,
"Delete Location")
                ToolTip1.SetToolTip(RefreshBtn,
"Refresh Location List")
                ToolTip1.SetToolTip(SearchBox,
"Search Location")

            ElseIf TreeView1.SelectedNode.Text =
"Department" Then

                VDepartment.SelectDepartment()

                ToolTip1.SetToolTip(NewBtn, "New
Department")
                ToolTip1.SetToolTip(EditBtn, "Edit
Department")
                ToolTip1.SetToolTip(DeleteBtn,
"Delete Department")
                ToolTip1.SetToolTip(RefreshBtn,
"Refresh Department List")
                ToolTip1.SetToolTip(SearchBox,
"Search Department")

            ElseIf TreeView1.SelectedNode.Text =
"Designation" Then

                VDesignation.SelectDesignation()

                ToolTip1.SetToolTip(NewBtn, "New
Designation")
                ToolTip1.SetToolTip(EditBtn, "Edit
Designation")
                ToolTip1.SetToolTip(DeleteBtn,
"Delete Designation")
                ToolTip1.SetToolTip(RefreshBtn,
"Refresh Designation List")
                ToolTip1.SetToolTip(SearchBox,
"Search Designation")

            ElseIf TreeView1.SelectedNode.Text =
"Employee Status" Then

                VEStatus.SelectEmployeeStatus()

                ToolTip1.SetToolTip(NewBtn, "New
Employee Status")
                ToolTip1.SetToolTip(EditBtn, "Edit
Employee Status")
                ToolTip1.SetToolTip(DeleteBtn,
"Delete Employee Status")
                ToolTip1.SetToolTip(RefreshBtn,
"Refresh Employee Status List")
                ToolTip1.SetToolTip(SearchBox,
"Search Employee Status")

            ElseIf TreeView1.SelectedNode.Text =
"Employee Violation" Then

                VViolation.SelectViolation()

                ToolTip1.SetToolTip(NewBtn, "New
Employee Violation")
                ToolTip1.SetToolTip(EditBtn, "Edit
Employee Violation")
                ToolTip1.SetToolTip(DeleteBtn,
"Delete Employee Violation")
                ToolTip1.SetToolTip(RefreshBtn,
"Refresh Employee Violation List")
                ToolTip1.SetToolTip(SearchBox,
"Search Employee Violation")

            ElseIf TreeView1.SelectedNode.Text =
"File Change of Rest Day" Then

                VFCRD.SelectFileChangeRestDay()

                ToolTip1.SetToolTip(NewBtn, "New
File Change of Rest Day")
                ToolTip1.SetToolTip(EditBtn, "Edit
File Change of Rest Day")
                ToolTip1.SetToolTip(DeleteBtn,
"Delete File Change of Rest Day")
                ToolTip1.SetToolTip(RefreshBtn,
"Refresh File Change of Rest Day List")
                ToolTip1.SetToolTip(SearchBox,
"Search File Change of Rest Day")

            ElseIf TreeView1.SelectedNode.Text =
"File Change of Shift Schedule" Then


VFCSS.SelectFileChangeShiftSchedule()
```

```
                ToolTip1.SetToolTip(NewBtn, "New
File Change of Shift Schedule")
                ToolTip1.SetToolTip(EditBtn, "Edit
File Change of Shift Schedule")
                ToolTip1.SetToolTip(DeleteBtn,
"Delete File Change of Shift Schedule")
                ToolTip1.SetToolTip(RefreshBtn,
"Refresh File Change of Shift Schedule List")
                ToolTip1.SetToolTip(SearchBox,
"Search File Change of Shift Schedule")

            ElseIf TreeView1.SelectedNode.Text =
"File Leave of Absence" Then

                VFLOA.SelectFileLeaveOfAbsence()

                ToolTip1.SetToolTip(NewBtn, "New
File Leave of Absence")
                ToolTip1.SetToolTip(EditBtn, "Edit
File Leave of Absence")
                ToolTip1.SetToolTip(DeleteBtn,
"Delete File Leave of Absence")
                ToolTip1.SetToolTip(RefreshBtn,
"Refresh File Leave of Absence List")
                ToolTip1.SetToolTip(SearchBox,
"Search File Leave of Absence")

            ElseIf TreeView1.SelectedNode.Text =
"File Overtime" Then

                VFOT.SelectFileOverTime()

                ToolTip1.SetToolTip(NewBtn, "New
File Overtime")
                ToolTip1.SetToolTip(EditBtn, "Edit
File Overtime")
                ToolTip1.SetToolTip(DeleteBtn,
"Delete File Overtime")
                ToolTip1.SetToolTip(RefreshBtn,
"Refresh File Overtime List")
                ToolTip1.SetToolTip(SearchBox,
"Search File Overtime")

            ElseIf TreeView1.SelectedNode.Text =
"File Undertime" Then

                VFUT.SelectFileUnderTime()

                ToolTip1.SetToolTip(NewBtn, "New
File Undertime")
                ToolTip1.SetToolTip(EditBtn, "Edit
File Undertime")
                ToolTip1.SetToolTip(DeleteBtn,
"Delete File Undertime")
                ToolTip1.SetToolTip(RefreshBtn,
"Refresh File Undertime List")
                ToolTip1.SetToolTip(SearchBox,
"Search File Undertime")

            ElseIf TreeView1.SelectedNode.Text =
"Attendance Input" Then

                AttendanceInput.ShowDialog()

            ElseIf TreeView1.SelectedNode.Text =
"Employees with Violation" Then
VEmployeeWithViolation.SelectEmployeeWithViolation
()

                NewBtn.Visible = False
                EditBtn.Visible = False
                DeleteBtn.Visible = False

                ToolTip1.SetToolTip(RefreshBtn,
"Refresh Employees with Violation List")
                ToolTip1.SetToolTip(SearchBox,
"Search Employees with Violation")

            ElseIf TreeView1.SelectedNode.Text =
"Daily Overtime Total" Then

                DeleteBtn.Visible = False
                PrintButton.Visible = True

                DateTimePicker1.Visible = True
                DateTimePicker1.Format =
DateTimePickerFormat.Custom
                DateTimePicker1.CustomFormat =
"MMM/dd/yyy"

                SelectOvertime("Daily Overtime
Total", "= '" & DateTimePicker1.Value.Date & "'")

                ToolTip1.SetToolTip(PrintButton,
"Print Daily Overtime")
                ToolTip1.SetToolTip(RefreshBtn,
"Refresh Daily Overtime List")
                ToolTip1.SetToolTip(SearchBox,
"Search Daily Overtime")

                print = "Daily Overtime"

            ElseIf TreeView1.SelectedNode.Text =
"Monthly Overtime Total" Then

                PrintButton.Visible = True
                DeleteBtn.Visible = False

                DateTimePicker1.Visible = True
                DateTimePicker1.Format =
DateTimePickerFormat.Custom
                DateTimePicker1.CustomFormat =
"MMM yyy"

                Dim datey As DateTime

                datey =
DateTime.Parse(DateTimePicker1.Value.Date.Month &
" 1, " & DateTimePicker1.Value.Date.Year)

                SelectOvertime("Monthly Overtime
Total", "BETWEEN '" & datey & "' AND '" &
datey.Date.AddMonths(1).AddDays(-1) & "'")

                ToolTip1.SetToolTip(PrintButton,
"Print Monthly Overtime")
                ToolTip1.SetToolTip(RefreshBtn,
"Refresh Monthly Overtime List")
                ToolTip1.SetToolTip(SearchBox,
"Search Monthly Overtime")

                print = "Monthly Overtime"
```

```
            ElseIf TreeView1.SelectedNode.Text =
"Daily Absenteeism Total" Then

                DeleteBtn.Visible = False
                PrintButton.Visible = True

                DateTimePicker1.Visible = True
                DateTimePicker1.Format =
DateTimePickerFormat.Custom
                DateTimePicker1.CustomFormat =
"MMM/dd/yyy"

                SelectAbsenteeism("Daily
Absenteeism Total", "= '" &
DateTimePicker1.Value.Date & "'")

                ToolTip1.SetToolTip(PrintButton,
"Print Daily Absenteeism")
                ToolTip1.SetToolTip(RefreshBtn,
"Refresh Daily Absenteeism List")
                ToolTip1.SetToolTip(SearchBox,
"Search Daily Absenteeism")

                print = "Daily Absenteeism"

            ElseIf TreeView1.SelectedNode.Text =
"Monthly Absenteeism Total" Then

                DeleteBtn.Visible = False
                PrintButton.Visible = True

                DateTimePicker1.Visible = True
                DateTimePicker1.Format =
DateTimePickerFormat.Custom
                DateTimePicker1.CustomFormat =
"MMM yyy"

                Dim datey As DateTime

                datey =
DateTime.Parse(DateTimePicker1.Value.Date.Month &
" 1, " & DateTimePicker1.Value.Date.Year)

                SelectAbsenteeism("Monthly
Absenteeism Total", "BETWEEN '" & datey & "' AND
'" & datey.Date.AddMonths(1).AddDays(-1) & "'")

                ToolTip1.SetToolTip(PrintButton,
"Print Monthly Absenteeism")
                ToolTip1.SetToolTip(RefreshBtn,
"Refresh Monthly Absenteeism List")
                ToolTip1.SetToolTip(SearchBox,
"Search Monthly Absenteeism")

                print = "Monthly Absenteeism"

            ElseIf TreeView1.SelectedNode.Text =
"View Own Audit Trail" Then

                Label2.Visible = False
                NewBtn.Visible = False
                EditBtn.Visible = False
                DeleteBtn.Visible = False

                Trail.SelectViewAuditTrail()

                ToolTip1.SetToolTip(RefreshBtn,
"Refresh Own Audit Trail")
                ToolTip1.SetToolTip(SearchBox,
"Search Own Audit Trail")

            ElseIf TreeView1.SelectedNode.Text =
"System Audit Trail" Then

                Label2.Visible = False
                NewBtn.Visible = False
                EditBtn.Visible = False
                DeleteBtn.Visible = False

                Trail.SelectViewSystemTrail()

                ToolTip1.SetToolTip(RefreshBtn,
"Refresh System Audit Trail")
                ToolTip1.SetToolTip(SearchBox,
"Search System Audit Trail")

            ElseIf TreeView1.SelectedNode.Text =
"Users" Then

                NewBtn.Visible = False
                EditBtn.Visible = False
                DeleteBtn.Visible = False

                VUser.SelectUser()

                ToolTip1.SetToolTip(RefreshBtn,
"Refresh Users List")
                    ToolTip1.SetToolTip(SearchBox,
"Search Users")

            ElseIf TreeView1.SelectedNode.Text =
"Help" Then


System.Diagnostics.Process.Start(Application.Start
upPath.ToString & "\TAMS User Manual.pdf")

            ElseIf TreeView1.SelectedNode.Text =
"Backup Database" Then

                Dim backupPath As Object
                Dim defaultPath As String

                defaultPath =
Application.StartupPath.ToString & "\TAMS"

                backupPath = InputBox("Specify
backup path.", "Backup Database", defaultPath)

                SQL.RunQuery("BACKUP DATABASE " &
My.Settings.DBTXT & " TO DISK = '" & backupPath &
"'")

            ElseIf TreeView1.SelectedNode.Text =
"Restore Database" Then

                Dim restorePath As Object
                Dim defaultPath As String

                defaultPath =
Application.StartupPath.ToString & "\TAMS"

                restorePath = InputBox("Specify
backup path.", "Restore Database", defaultPath)
```

```
                SQL.RunQuery("RESTORE DATABASE " &
My.Settings.DBTXT & " FROM DISK = '" & restorePath
& "'")

            ElseIf TreeView1.SelectedNode.Text =
"Shrink Database" Then

                SQL.RunQuery("DBCC SHRINKDATABASE
(" & My.Settings.DBTXT & ")")

            ElseIf TreeView1.SelectedNode.Text =
"Log Out" Then

                Dim result1 As DialogResult =
MessageBox.Show("Are you sure?", _
                "Log Out", _
                MessageBoxButtons.YesNo)

                If result1 =
Windows.Forms.DialogResult.Yes Then

                    Trail.AddTrail("Logged Out.")

                    Application.Restart()

                End If

            Else

                GroupBox.Visible = False
                PictureBox1.Visible = True

            End If

        Catch ex As Exception

            MsgBox(ex.Message,
MsgBoxStyle.Critical)

        End Try

    End Sub

    Public Sub LoadDataGrid()

        Dim bSource As New BindingSource
        Dim dTable As New DataTable

        If SQL.SQLDS IsNot Nothing Then

            SQL.SQLDS.Clear()

        End If

        SQL.RunQuery(QString)
        SQL.SQLDA.Fill(dTable)
        bSource.DataSource = dTable
        DataGrid.DataSource = bSource
        SQL.SQLDA.Update(dTable)

    End Sub

    Private Sub NewBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles NewBtn.Click

        Try

            HasRow = False

            If TreeView1.SelectedNode.Text =
"Adjustment Policy" Then

                AjustmentType.Show()

            ElseIf TreeView1.SelectedNode.Text =
"Leave Credit" Then

                LeaveCredit.Show()

            ElseIf TreeView1.SelectedNode.Text =
"Daily Schedule" Then

                DailySchedule.Show()

            ElseIf TreeView1.SelectedNode.Text =
"Weekly Schedule" Then

                WeeklySchedule.Show()

            ElseIf TreeView1.SelectedNode.Text =
"Legal Holiday" Then

                LegalHoliday.Show()

            ElseIf TreeView1.SelectedNode.Text =
"Special Holiday" Then

                SpecialHoliday.Show()

            ElseIf TreeView1.SelectedNode.Text =
"Employees" Then

                Employee.Show()

            ElseIf TreeView1.SelectedNode.Text =
"Employee with Violation" Then

                EmployeeWithViolation.Show()

            ElseIf TreeView1.SelectedNode.Text =
"Vacation Leave" Then

                leavetypeTB = "Vacation Leave"

                EmployeeLeaveCredit.Show()

            ElseIf TreeView1.SelectedNode.Text =
"Sick Leave" Then

                leavetypeTB = "Sick Leave"

                EmployeeLeaveCredit.Show()

            ElseIf TreeView1.SelectedNode.Text =
"Paternity Leave" Then

                leavetypeTB = "Paternity Leave"

                EmployeeLeaveCredit.Show()

            ElseIf TreeView1.SelectedNode.Text =
"Branch" Then

                Branch.Show()

            ElseIf TreeView1.SelectedNode.Text =
"Location" Then
```

```
                Location1.Show()

            ElseIf TreeView1.SelectedNode.Text =
"Department" Then

                Department.Show()

            ElseIf TreeView1.SelectedNode.Text =
"Designation" Then

                Designation.Show()

            ElseIf TreeView1.SelectedNode.Text =
"Employee Status" Then

                EmployeeStatus.Show()

            ElseIf TreeView1.SelectedNode.Text =
"Employee Violation" Then

                Violation.Show()

            ElseIf TreeView1.SelectedNode.Text =
"File Change of Rest Day" Then

                FileChangeRestDay.Show()

            ElseIf TreeView1.SelectedNode.Text =
"File Change of Shift Schedule" Then

                FileChangeShiftSchedule.Show()

            ElseIf TreeView1.SelectedNode.Text =
"File Leave of Absence" Then

                FileLeaveofAbsence.Show()

            ElseIf TreeView1.SelectedNode.Text =
"File Overtime" Then

                FileOverTime.Show()

            ElseIf TreeView1.SelectedNode.Text =
"File Undertime" Then

                FileUnderTime.Show()

            ElseIf TreeView1.SelectedNode.Text =
"Employees with Violation" Then

            ElseIf TreeView1.SelectedNode.Text =
"Users" Then

                User.Show()

            ElseIf TreeView1.SelectedNode.Text =
"Daily Overtime Total" Then

            ElseIf TreeView1.SelectedNode.Text =
"Monthly Overtime Total" Then

            ElseIf TreeView1.SelectedNode.Text =
"Daily Absenteeism Total" Then

            ElseIf TreeView1.SelectedNode.Text =
"Monthly Absenteeism Total" Then

            Else

            End If

        Catch ex As Exception

            MsgBox(ex.Message,
MsgBoxStyle.Critical)

        End Try

    End Sub

    Private Sub RefreshBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles RefreshBtn.Click

        LoadDataGrid()

    End Sub

    Private Sub EditBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles EditBtn.Click

        Try

            HasRow = True

            If TreeView1.SelectedNode.Text =
"Adjustment Policy" Then

                AjustmentType.Show()

            ElseIf TreeView1.SelectedNode.Text =
"Leave Credit" Then

                LeaveCredit.Show()

            ElseIf TreeView1.SelectedNode.Text =
"Daily Schedule" Then

                DailySchedule.Show()

            ElseIf TreeView1.SelectedNode.Text =
"Weekly Schedule" Then

                WeeklySchedule.Show()

            ElseIf TreeView1.SelectedNode.Text =
"Legal Holiday" Then

                LegalHoliday.Show()

            ElseIf TreeView1.SelectedNode.Text =
"Special Holiday" Then

                SpecialHoliday.Show()

            ElseIf TreeView1.SelectedNode.Text =
"Employees" Then

                Employee.Show()

            ElseIf TreeView1.SelectedNode.Text =
"Employee with Violation" Then

                EmployeeWithViolation.Show()
```

```
            ElseIf TreeView1.SelectedNode.Text =
"Vacation Leave" Then

                EmployeeLeaveCredit.Show()

            ElseIf TreeView1.SelectedNode.Text =
"Sick Leave" Then

                EmployeeLeaveCredit.Show()

            ElseIf TreeView1.SelectedNode.Text =
"Paternity Leave" Then

                EmployeeLeaveCredit.Show()

            ElseIf TreeView1.SelectedNode.Text =
"Branch" Then

                Branch.Show()

            ElseIf TreeView1.SelectedNode.Text =
"Location" Then

                Location1.Show()

            ElseIf TreeView1.SelectedNode.Text =
"Department" Then

                Department.Show()

            ElseIf TreeView1.SelectedNode.Text =
"Designation" Then

                Designation.Show()

            ElseIf TreeView1.SelectedNode.Text =
"Employee Status" Then

                EmployeeStatus.Show()

            ElseIf TreeView1.SelectedNode.Text =
"Employee Violation" Then

                Violation.Show()

            ElseIf TreeView1.SelectedNode.Text =
"File Change of Rest Day" Then

                FileChangeRestDay.Show()

            ElseIf TreeView1.SelectedNode.Text =
"File Change of Shift Schedule" Then

                FileChangeShiftSchedule.Show()

            ElseIf TreeView1.SelectedNode.Text =
"File Leave of Absence" Then

                FileLeaveofAbsence.Show()

            ElseIf TreeView1.SelectedNode.Text =
"File Overtime" Then

                FileOverTime.Show()

            ElseIf TreeView1.SelectedNode.Text =
"Users" Then

                User.Show()

            ElseIf TreeView1.SelectedNode.Text =
"File Undertime" Then

                FileUnderTime.Show()

            Else

            End If

        Catch ex As Exception

            MsgBox(ex.Message,
MsgBoxStyle.Critical)

        End Try

    End Sub

    Private Sub DataGrid_CellContentClick_1(ByVal
sender As System.Object, ByVal e As
System.Windows.Forms.DataGridViewCellEventArgs)
Handles DataGrid.CellContentClick

        Try

            If e.ColumnIndex >= 0 Then

                Dim row As DataGridViewRow

                row = Me.DataGrid.Rows(e.RowIndex)

                If TreeView1.SelectedNode.Text =
"Adjustment Policy" Then

                    ClearStrings()

                    NameTB = row.Cells("Adjustment
Policy").Value.ToString
                    Code = row.Cells("Adjustment
Policy Code").Value.ToString
                    Rate =
row.Cells("Rate").Value.ToString
                    Comment =
row.Cells("Comment").Value.ToString
                    isActive =
row.Cells("Active").Value.ToString
                    ID =
row.Cells("ID").Value.ToString

                ElseIf TreeView1.SelectedNode.Text
= "Leave Credit" Then

                    ClearStrings()

                    NameTB = row.Cells("Leave
Credit").Value.ToString
                    Code = row.Cells("Leave Credit
Code").Value.ToString
                    Comment =
row.Cells("Comment").Value.ToString
                    isActive =
row.Cells("Active").Value.ToString
                    ID =
row.Cells("ID").Value.ToString
```

```
                ElseIf TreeView1.SelectedNode.Text
= "Daily Schedule" Then

                    ClearStrings()

                    NameTB = row.Cells("Daily
Schedule").Value.ToString
                    Comment =
row.Cells("Comment").Value.ToString
                    isActive =
row.Cells("Active").Value.ToString
                    ID =
row.Cells("ID").Value.ToString
                    Duration =
row.Cells("Duration").Value.ToString
                    in1 = row.Cells("Time In
1").Value.ToString
                    out1 = row.Cells("Time Out
1").Value.ToString
                    in2 = row.Cells("Time In
2").Value.ToString
                    out2 = row.Cells("Time Out
2").Value.ToString
                    in3 = row.Cells("Time In
3").Value.ToString
                    out3 = row.Cells("Time Out
3").Value.ToString

                    in1d =
DateTime.Parse(DateTime.Today & " " & in1)
                    out1d =
DateTime.Parse(DateTime.Today & " " & out1)
                    in2d =
DateTime.Parse(DateTime.Today & " " & in2)
                    out2d =
DateTime.Parse(DateTime.Today & " " & out2)
                    in3d =
DateTime.Parse(DateTime.Today & " " & in3)
                    out3d =
DateTime.Parse(DateTime.Today & " " & out3)

                    'MsgBox(in1d)

                ElseIf TreeView1.SelectedNode.Text
= "Weekly Schedule" Then

                    ClearStrings()

                    ID =
row.Cells("ID").Value.ToString
                    NameTB = row.Cells("Weekly
Schedule").Value.ToString
                    sunday =
row.Cells("Sunday").Value.ToString
                    monday =
row.Cells("Monday").Value.ToString
                    tuesday =
row.Cells("Tuesday").Value.ToString
                    wednesday =
row.Cells("Wednesday").Value.ToString
                    thursday =
row.Cells("Thursday").Value.ToString
                    friday =
row.Cells("Friday").Value.ToString
                    saturday =
row.Cells("Saturday").Value.ToString
                    isActive =
row.Cells("Active").Value.ToString
                    Comment =
row.Cells("Comment").Value.ToString

                ElseIf TreeView1.SelectedNode.Text
= "Legal Holiday" Then

                    ClearStrings()

                    ID =
row.Cells("ID").Value.ToString
                    NameTB = row.Cells("Legal
Holiday").Value.ToString
                    dateTB =
row.Cells("Date").Value.ToString
                    isActive =
row.Cells("Active").Value.ToString
                    Comment =
row.Cells("Comment").Value.ToString

                ElseIf TreeView1.SelectedNode.Text
= "Users" Then

                    ClearStrings()
                    ID =
row.Cells("ID").Value.ToString
                    lname =
row.Cells("LastName").Value.ToString
                    fname =
row.Cells("FirstName").Value.ToString
                    mname =
row.Cells("MiddleName").Value.ToString
                    isActive =
row.Cells("isActive").Value.ToString
                    Comment =
row.Cells("Comment").Value.ToString
                    isApproved =
row.Cells("isApproved").Value.ToString
                    isAdmin =
row.Cells("isAdmin").Value.ToString
                    username =
row.Cells("Username").Value.ToString

                ElseIf TreeView1.SelectedNode.Text
= "Special Holiday" Then

                    ClearStrings()

                    ID =
row.Cells("ID").Value.ToString
                    NameTB = row.Cells("Special
Holiday").Value.ToString
                    dateTB =
row.Cells("Date").Value.ToString
                    isActive =
row.Cells("Active").Value.ToString
                    locationTB =
row.Cells("Location").Value.ToString

                ElseIf TreeView1.SelectedNode.Text
= "Employees" Then

                    ClearStrings()

                    ID =
row.Cells("ID").Value.ToString
                    lname = row.Cells("Last
Name").Value.ToString
                    fname = row.Cells("First
Name").Value.ToString
```

```
                    mname = row.Cells("Middle
Name").Value.ToString
                    branchTB =
row.Cells("Branch").Value.ToString
                    departmentTB =
row.Cells("Department").Value.ToString
                    designationTB =
row.Cells("Designation").Value.ToString
                    weeklysched =
row.Cells("Weekly Schedule").Value.ToString
                    employeestatusTB =
row.Cells("Employee Status").Value.ToString
                    isActive =
row.Cells("Active").Value.ToString
                    Comment =
row.Cells("Comment").Value.ToString
                    enumber = row.Cells("Employee
Number").Value.ToString

                ElseIf TreeView1.SelectedNode.Text
= "Employee with Violation" Then

                    ClearStrings()

                    ID =
row.Cells("ID").Value.ToString
                    NameTB = row.Cells("Employee
Name").Value.ToString
                    dateTB =
row.Cells("Date").Value.ToString
                    violationTB =
row.Cells("Violation").Value.ToString
                    Comment =
row.Cells("Comment").Value.ToString

                ElseIf TreeView1.SelectedNode.Text
= "Vacation Leave" Then

                    ClearStrings()

                    ID =
row.Cells("ID").Value.ToString
                    NameTB = row.Cells("Employee
Name").Value.ToString
                    leavetypeTB = row.Cells("Leave
Type").Value.ToString
                    totalnumberTB =
row.Cells("Total Number").Value.ToString
                    Comment =
row.Cells("Comment").Value.ToString

                ElseIf TreeView1.SelectedNode.Text
= "Sick Leave" Then

                    ClearStrings()

                    ID =
row.Cells("ID").Value.ToString
                    NameTB = row.Cells("Employee
Name").Value.ToString
                    leavetypeTB = row.Cells("Leave
Type").Value.ToString
                    totalnumberTB =
row.Cells("Total Number").Value.ToString
                    Comment =
row.Cells("Comment").Value.ToString

                ElseIf TreeView1.SelectedNode.Text
= "Paternity Leave" Then

                    ClearStrings()

                    ID =
row.Cells("ID").Value.ToString
                    NameTB = row.Cells("Employee
Name").Value.ToString
                    leavetypeTB = row.Cells("Leave
Type").Value.ToString
                    totalnumberTB =
row.Cells("Total Number").Value.ToString
                    Comment =
row.Cells("Comment").Value.ToString

                ElseIf TreeView1.SelectedNode.Text
= "Branch" Then

                    ClearStrings()

                    ID =
row.Cells("ID").Value.ToString
                    NameTB =
row.Cells("Branch").Value.ToString
                    locationTB =
row.Cells("Location").Value.ToString
                    isActive =
row.Cells("Active").Value.ToString
                    Comment =
row.Cells("Comment").Value.ToString

                ElseIf TreeView1.SelectedNode.Text
= "Location" Then

                    ClearStrings()

                    ID =
row.Cells("ID").Value.ToString
                    locationTB =
row.Cells("Location").Value.ToString
                    isActive =
row.Cells("Active").Value.ToString
                    Comment =
row.Cells("Comment").Value.ToString

                ElseIf TreeView1.SelectedNode.Text
= "Department" Then

                    ClearStrings()

                    ID =
row.Cells("ID").Value.ToString
                    departmentTB =
row.Cells("Department").Value.ToString
                    isActive =
row.Cells("Active").Value.ToString
                    Comment =
row.Cells("Comment").Value.ToString

                ElseIf TreeView1.SelectedNode.Text
= "Designation" Then

                    ClearStrings()

                    ID =
row.Cells("ID").Value.ToString
                    designationTB =
row.Cells("Designation").Value.ToString
                    basicpayTB = row.Cells("Basic
Pay").Value.ToString
```

```
                    COLA =
row.Cells("COLA").Value.ToString
                    isActive =
row.Cells("Active").Value.ToString
                    Comment =
row.Cells("Comment").Value.ToString

                ElseIf TreeView1.SelectedNode.Text
= "Employee Status" Then

                    ClearStrings()

                    ID =
row.Cells("ID").Value.ToString
                    isActive =
row.Cells("Active").Value.ToString
                    Comment =
row.Cells("Comment").Value.ToString
                    employeestatusTB =
row.Cells("Employee Status").Value.ToString

                ElseIf TreeView1.SelectedNode.Text
= "Employee Violation" Then

                    ClearStrings()

                    ID =
row.Cells("ID").Value.ToString
                    isActive =
row.Cells("Active").Value.ToString
                    Comment =
row.Cells("Comment").Value.ToString
                    penaltyTB =
row.Cells("Penalty").Value.ToString
                    rankTB = row.Cells("Offense
Rank").Value.ToString
                    descTB =
row.Cells("Description").Value.ToString
                    NameTB =
row.Cells("Violation").Value.ToString

                ElseIf TreeView1.SelectedNode.Text
= "File Change of Rest Day" Then

                    ClearStrings()

                    ID =
row.Cells("ID").Value.ToString
                    approve =
row.Cells("Approved").Value.ToString
                    Comment =
row.Cells("Comment").Value.ToString
                    NameTB = row.Cells("Employee
Name").Value.ToString
                    dateTB =
row.Cells("Date").Value.ToString
                    reason =
row.Cells("Reason").Value.ToString

                ElseIf TreeView1.SelectedNode.Text
= "File Change of Shift Schedule" Then

                    ClearStrings()

                    ID =
row.Cells("ID").Value.ToString
                    approve =
row.Cells("Approved").Value.ToString
                    Comment =
row.Cells("Comment").Value.ToString
                    NameTB = row.Cells("Employee
Name").Value.ToString
                    dateTB =
row.Cells("Date").Value.ToString
                    reason =
row.Cells("Reason").Value.ToString
                    newschedTB =
row.Cells("Schedule").Value.ToString

                ElseIf TreeView1.SelectedNode.Text
= "File Leave of Absence" Then

                    ClearStrings()

                    ID =
row.Cells("ID").Value.ToString
                    approve =
row.Cells("Approved").Value.ToString
                    Comment =
row.Cells("Comment").Value.ToString
                    NameTB = row.Cells("Employee
Name").Value.ToString
                    dateTB = row.Cells("Filing
Date").Value.ToString
                    reason =
row.Cells("Reason").Value.ToString
                    leavetypeTB = row.Cells("Leave
Type").Value.ToString
                    startdateTB = row.Cells("Start
Date").Value.ToString
                    enddateTB = row.Cells("End
Date").Value.ToString

                ElseIf TreeView1.SelectedNode.Text
= "File Overtime" Then

                    ClearStrings()

                    ID =
row.Cells("ID").Value.ToString
                    approve =
row.Cells("Approved").Value.ToString
                    Comment =
row.Cells("Comment").Value.ToString
                    NameTB = row.Cells("Employee
Name").Value.ToString
                    dateTB =
row.Cells("Date").Value.ToString
                    reason =
row.Cells("Reason").Value.ToString

                ElseIf TreeView1.SelectedNode.Text
= "File Undertime" Then

                    ClearStrings()

                    ID =
row.Cells("ID").Value.ToString
                    approve =
row.Cells("Approved").Value.ToString
                    Comment =
row.Cells("Comment").Value.ToString
                    NameTB = row.Cells("Employee
Name").Value.ToString
                    dateTB =
row.Cells("Date").Value.ToString
```

```
                    reason =
row.Cells("Reason").Value.ToString

                ElseIf TreeView1.SelectedNode.Text
= "Employees with Violation" Then

                ElseIf TreeView1.SelectedNode.Text
= "Daily Overtime Total" Then

                ElseIf TreeView1.SelectedNode.Text
= "Monthly Overtime Total" Then

                ElseIf TreeView1.SelectedNode.Text
= "Daily Absenteeism Total" Then

                ElseIf TreeView1.SelectedNode.Text
= "Monthly Absenteeism Total" Then

                ElseIf TreeView1.SelectedNode.Text
= "View Own Audit Trail" Then

                Else

                End If

            End If

        Catch ex As Exception

            MsgBox(ex.Message,
MsgBoxStyle.Critical)

        End Try

    End Sub

    Private Sub DeleteBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles DeleteBtn.Click

        Try

            If TreeView1.SelectedNode.Text =
"Adjustment Policy" Then


VAdjPolicy.DeleteAdjustmentPolicy()

            ElseIf TreeView1.SelectedNode.Text =
"Leave Credit" Then

                VLeaveCredit.DeleteLeaveCredit()

            ElseIf TreeView1.SelectedNode.Text =
"Daily Schedule" Then


VDailySchedule.DeleteDailySchedule()

            ElseIf TreeView1.SelectedNode.Text =
"Weekly Schedule" Then


VWeeklySchedule.DeleteWeeklySchedule()

            ElseIf TreeView1.SelectedNode.Text =
"Legal Holiday" Then

                VLegalHoliday.DeleteLegalHoliday()

            ElseIf TreeView1.SelectedNode.Text =
"Special Holiday" Then


VSpecialHoliday.DeleteSpecialHoliday()

            ElseIf TreeView1.SelectedNode.Text =
"Employees" Then

                VEmployee.DeleteEmployee()

            ElseIf TreeView1.SelectedNode.Text =
"Employee with Violation" Then


VEmployeeWithViolation.DeleteEmployeeWithViolation
()

            ElseIf TreeView1.SelectedNode.Text =
"Branch" Then

                VBranch.DeleteBranch()

            ElseIf TreeView1.SelectedNode.Text =
"Location" Then

                VLocation.DeleteLocation()

            ElseIf TreeView1.SelectedNode.Text =
"Department" Then

                VDepartment.DeleteDepartment()

            ElseIf TreeView1.SelectedNode.Text =
"Designation" Then

                VDesignation.DeleteDesignation()

            ElseIf TreeView1.SelectedNode.Text =
"Employee Status" Then

                VEStatus.DeleteEmployeeStatus()

            ElseIf TreeView1.SelectedNode.Text =
"Employee Violation" Then

                VViolation.DeleteViolation()

            ElseIf TreeView1.SelectedNode.Text =
"Users" Then

                VUser.DeleteUser()

            ElseIf TreeView1.SelectedNode.Text =
"File Change of Rest Day" Then

                VFCRD.DeleteFileChangeRestDay()

            ElseIf TreeView1.SelectedNode.Text =
"File Change of Shift Schedule" Then


VFCSS.DeleteFileChangeShiftSchedule()

            ElseIf TreeView1.SelectedNode.Text =
"File Leave of Absence" Then

                VFLOA.DeleteFileLeaveOfAbsence()
```

```
            ElseIf TreeView1.SelectedNode.Text =
"File Overtime" Then

                VFOT.DeleteFileOverTime()

            ElseIf TreeView1.SelectedNode.Text =
"File Undertime" Then

                VFUT.DeleteFileUnderTime()

            Else

            End If

            ClearStrings()

        Catch ex As Exception

            MsgBox(ex.Message,
MsgBoxStyle.Critical)

        End Try

    End Sub

    Private Sub DateTimePicker1_ValueChanged(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
DateTimePicker1.ValueChanged

        Try

            If TreeView1.SelectedNode.Text =
"Daily Overtime Total" Then

                SelectOvertime("Daily Overtime
Total", "= '" & DateTimePicker1.Value.Date & "'")

            ElseIf TreeView1.SelectedNode.Text =
"Monthly Overtime Total" Then

                Dim datey As DateTime

                datey =
DateTime.Parse(DateTimePicker1.Value.Date.Month &
" 1, " & DateTimePicker1.Value.Date.Year)

                SelectOvertime("Monthly Overtime
Total", "BETWEEN '" & datey & "' AND '" &
datey.Date.AddMonths(1).AddDays(-1) & "'")

            ElseIf TreeView1.SelectedNode.Text =
"Daily Absenteeism Total" Then

                SelectAbsenteeism("Daily
Absenteeism Total", "= '" &
DateTimePicker1.Value.Date & "'")

            ElseIf TreeView1.SelectedNode.Text =
"Monthly Absenteeism Total" Then

                Dim datey As DateTime

                datey =
DateTime.Parse(DateTimePicker1.Value.Date.Month &
" 1, " & DateTimePicker1.Value.Date.Year)

                SelectAbsenteeism("Monthly
Absenteeism Total", "BETWEEN '" & datey & "' AND
'" & datey.Date.AddMonths(1).AddDays(-1) & "'")

            End If

        Catch ex As Exception

            MsgBox(ex.Message,
MsgBoxStyle.Critical)

        End Try

    End Sub

    Private Sub SearchBox_TextChanged(ByVal sender
As System.Object, ByVal e As System.EventArgs)
Handles SearchBox.TextChanged

        Try

            LoadDataGrid()

            DataGrid.CurrentCell = Nothing

            For Each row As DataGridViewRow In
Me.DataGrid.Rows

                Dim list As New ArrayList

                For Each cell As DataGridViewCell
In row.Cells

                    If
cell.Value.ToString.ToLower.Contains(SearchBox.Tex
t) Then

                        list.Add(True)

                    Else

                        list.Add(False)

                    End If

                Next

                If Not list.Contains(True) Then

                    row.Visible = False

                End If

            Next

        Catch ex As Exception

            MsgBox(ex.Message,
MsgBoxStyle.Critical)

        End Try

    End Sub

    Private Sub PrintButton_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles PrintButton.Click

        Try
```

```
            If print = "Daily Absenteeism" Then

                VDAbsenteeism.PrintDAbsenteeism()
                Trail.AddTrail("Printed Daily
Absenteeism")

            ElseIf print = "Monthly Absenteeism"
Then

                VMAbsenteeism.PrintMAbsenteeism()
                Trail.AddTrail("Printed Monthly
Absenteeism")

            ElseIf print = "Daily Overtime" Then

                VDOvertime.PrintDOvertime()
                Trail.AddTrail("Printed Daily
Overtime")

            ElseIf print = "Monthly Overtime" Then

                VMOvertime.PrintMOvertime()
                Trail.AddTrail("Printed Monthly
Overtime")

            End If

        Catch ex As Exception

            MsgBox(ex.Message,
MsgBoxStyle.Information)

        End Try

    End Sub

End Class
Public Class Form1

    Private Trail As New AuditTrail
    Private SQL As New SQLControl
    Public AuthUser As String
    Public isAdmin As Boolean

    Private Sub LinkLabel1_LinkClicked(ByVal
sender As System.Object, ByVal e As
System.Windows.Forms.LinkLabelLinkClickedEventArgs
) Handles DBSet.LinkClicked

        DBSettings.Show()

    End Sub

    Private Sub CancelBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles CancelBtn.Click

        Me.Close()

    End Sub

    Private Sub LinkLabel1_LinkClicked_1(ByVal
sender As System.Object, ByVal e As
System.Windows.Forms.LinkLabelLinkClickedEventArgs
) Handles LinkLabel1.LinkClicked

        User.Show()

    End Sub

    Private Sub OKBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles OKBtn.Click

        Try

            If SQL.HasConnection = True Then

                isAdmin = False

                If IsAuthenticated() = True Then

                    MsgBox("Login successful.",
MsgBoxStyle.Information)

                    AuthUser = UsernameTB.Text

                    If isAdmin = True Then



                        MainWindow.Show()
                        MainWindow.Focus()
                        Me.Close()

                    Else



                        MainWindow.Show()
                        MainWindow.Focus()
                        Me.Close()

                    End If

                End If

            End If

        Catch ex As Exception

            MsgBox(ex.Message,
MsgBoxStyle.Critical)

        End Try

    End Sub

    Private Function IsAuthenticated() As Boolean

        If SQL.SQLDS IsNot Nothing Then

            SQL.SQLDS.Clear()

        End If

        SQL.RunQuery("SELECT Count(Username) As
UserCount " & _
                     "FROM tUser " & _
                     "WHERE Username='" &
UsernameTB.Text & "' " & _
                     " AND Password='" &
PwdTB.Text & "' COLLATE
SQL_Latin1_General_CP1_CS_AS AND isApproved = 1")
```

```
        If
SQL.SQLDS.Tables(0).Rows(0).Item("UserCount") = 1
Then

            Trail.AddTrail("Logged In.")

            SQL.RunQuery("SELECT isAdmin, ID FROM
tUser WHERE Username = '" & UsernameTB.Text & "'")

            If
SQL.SQLDS.Tables(0).Rows(0).Item("isAdmin").ToStri
ng = True Then

                isAdmin = True

            Else : isAdmin = False

            End If

            Return True

        End If

        MsgBox("Invalid user credentials or
account not yet approved.", MsgBoxStyle.Critical,
"LOGIN FAILED")

        Return False

    End Function

    Private Sub Form1_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load

        UsernameTB.Text = My.Settings.USERTXT
        PwdTB.Text = My.Settings.PASSTXT

    End Sub

    Private Sub Form1_FormClosed(ByVal sender As
Object, ByVal e As
System.Windows.Forms.FormClosedEventArgs) Handles
MyBase.FormClosed

        My.Settings.USERTXT = UsernameTB.Text
        My.Settings.PASSTXT = PwdTB.Text
        My.Settings.Save()

    End Sub


End Class

Public Class DBSettings

    Private SQL As New SQLControl
    Public ServerTxt As String
    Public DBTxt As String
    Public DBUserTxt As String
    Public DBPassTxt As String


    Private Sub CancelBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles CancelBtn.Click
        Me.Close()

    End Sub

    Private Sub SaveBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles SaveBtn.Click

        My.Settings.ServerTXT = ServerTB.Text
        My.Settings.DBTXT = DBTB.Text
        My.Settings.DBUserTXT = DBUserTB.Text
        My.Settings.DBPassTXT = DBPwdTB.Text
        My.Settings.Save()

        If SQL.HasConnection = True Then

            MsgBox("Connection success!")

            Me.Close()

        End If

        'MsgBox(My.Settings.ServerTXT)



    End Sub

    Private Sub DBSettings_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load

        ServerTB.Text = My.Settings.ServerTXT
        DBTB.Text = My.Settings.DBTXT
        DBUserTB.Text = My.Settings.DBUserTXT
        DBPwdTB.Text = My.Settings.DBPassTXT

    End Sub

    Private Sub DBSettings_FormClosed(ByVal sender
As Object, ByVal e As
System.Windows.Forms.FormClosedEventArgs) Handles
MyBase.FormClosed

        My.Settings.ServerTXT = ServerTB.Text
        My.Settings.DBTXT = DBTB.Text
        My.Settings.DBUserTXT = DBUserTB.Text
        My.Settings.DBPassTXT = DBPwdTB.Text
        My.Settings.Save()



    End Sub
End Class
Public Class CheckDuplicates

    Private SQL As New SQLControl

    Public Function IsDuplicated(ByVal DupString
As String)

        SQL.RunQuery(DupString)

        If SQL.SQLDS.Tables(0).Rows.Count > 0 Then

            Return True

        Else : Return False

        End If
```

```
    End Function

End Class

Public Class AttendanceInput

    Private SQL As New SQLControl
    Private HRow As Boolean
    Private qstring As String
    Private bSource As New BindingSource
    Private dTable As New DataTable
    Private printString As String
    Private absent As String
    Private dayWeek As String
    Private Trail As New AuditTrail
    Private LeaveType As String
    Private ENumber, SchedID As String
    Private Regular As String
    Private ins, outs, innd, outnd As String
    Private datei As DateTime
    Private COLA, BasicPay As String
    Private IDAttendanceComp As String
    Private DTPCol, DTPCol2, DTPCol3, DTPCol4,
DTPCol5, DTPCol6 As New CalendarColumn
    Private PdfDTR As New PrintDTR
    Private PdfCompensation As New
PrintCompensation
    Private PdfAttendanceSummary As New
PrintAttendanceSummary
    Private PayslipPDF As New PrintPDFPayslip

    Private Sub AttendanceInput_Load(ByVal sender
As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load

        Try



            SQL.RunQuery("SELECT ID, Name FROM
tBranch WHERE isActive = 1")

            BranchCB.DataSource =
SQL.SQLDS.Tables(0)
            BranchCB.ValueMember = "ID"
            BranchCB.DisplayMember = "Name"

            SQL.RunQuery("SELECT ID, Name FROM
tDepartment WHERE isActive = 1")

            DeptCB.DataSource =
SQL.SQLDS.Tables(0).Copy
            DeptCB.ValueMember = "ID"
            DeptCB.DisplayMember = "Name"

            SQL.RunQuery("SELECT ID, Name FROM
tDesignation WHERE isActive = 1")

            DesigCB.DataSource =
SQL.SQLDS.Tables(0).Copy
            DesigCB.ValueMember = "ID"
            DesigCB.DisplayMember = "Name"

            dTable.Clear()
            dTable.Columns.Clear()
            SetTable()

        Catch ex As Exception
            MsgBox(ex.Message,
MsgBoxStyle.Critical)

        End Try

    End Sub

    Private Sub CancelBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles CancelBtn.Click
        'Closes the form
        Me.Close()

    End Sub

    Private Sub SetTable()



        'Set the Data Table

        Dim dcName = New DataColumn("Date",
GetType(Date))

        Dim day = New DataColumn("Day",
GetType(String))

        Dim sched = New DataColumn("Schedule",
GetType(String))

        Dim isSpecial = New DataColumn("Special
Holiday", GetType(Boolean))

        Dim isLegal = New DataColumn("Legal
Holiday", GetType(Boolean))

        Dim isAbsent = New DataColumn("Absent",
GetType(Boolean))

        Dim isLeave = New DataColumn("Leave",
GetType(Boolean))

        Dim isOT = New DataColumn("Overtime
Filed", GetType(Boolean))

        Dim isRD = New DataColumn("Rest Day",
GetType(Boolean))

        dTable.Columns.Add(dcName)
        dTable.Columns.Add(day)
        dTable.Columns.Add(sched)
        dTable.Columns.Add(isSpecial)
        dTable.Columns.Add(isLegal)
        dTable.Columns.Add(isAbsent)
        dTable.Columns.Add(isLeave)
        dTable.Columns.Add(isOT)
        dTable.Columns.Add(isRD)

    End Sub

    Private Sub SetDataGridColumn()

        DataGridView1.Columns("Employee
Name").DisplayIndex = 0
        DataGridView1.Columns("Employee
Name").ReadOnly = True
```

```
        DataGridView1.Columns("Employee
Number").DisplayIndex = 1
        DataGridView1.Columns("Employee
Number").ReadOnly = True

        DataGridView1.Columns("Date").DisplayIndex
= 2
        DataGridView1.Columns("Date").ReadOnly =
True

        DataGridView1.Columns("Day").DisplayIndex
= 3
        DataGridView1.Columns("Day").ReadOnly =
True


DataGridView1.Columns("Schedule").DisplayIndex = 4
        DataGridView1.Columns("Schedule").ReadOnly
= True

        DataGridView1.Columns("Time In
1").DisplayIndex = 5
        DataGridView1.Columns("Time Out
1").DisplayIndex = 6
        DataGridView1.Columns("Time In
2").DisplayIndex = 7
        DataGridView1.Columns("Time Out
2").DisplayIndex = 8
        DataGridView1.Columns("Time In
3").DisplayIndex = 9
        DataGridView1.Columns("Time Out
3").DisplayIndex = 10


DataGridView1.Columns("Absent").DisplayIndex = 11

        DataGridView1.Columns("Rest
Day").DisplayIndex = 12
        DataGridView1.Columns("Rest Day").ReadOnly
= True


DataGridView1.Columns("Leave").DisplayIndex = 13
        DataGridView1.Columns("Leave").ReadOnly =
True

        DataGridView1.Columns("Special
Holiday").DisplayIndex = 14
        DataGridView1.Columns("Special
Holiday").ReadOnly = True

        DataGridView1.Columns("Legal
Holiday").DisplayIndex = 15
        DataGridView1.Columns("Legal
Holiday").ReadOnly = True

        DataGridView1.Columns("Overtime
Filed").DisplayIndex = 16
        DataGridView1.Columns("Overtime
Filed").ReadOnly = True


    End Sub

    Private Sub GenerateBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles GenerateBtn.Click

        Try

            dTable.Clear()

            'Generate employee names and employee
numbers to dTable
            GenerateEmployeeNamesAndNumbers()

            'Bind dTable to binding source
            BindToSource()

            'Add Dates to dTable
            AddDatesAndDays()

            For j As Integer = 0 To
Me.DataGridView1.Rows.Count - 1


Me.DataGridView1.Rows(j).Cells("Absent").Value =
False

                'Check data if already in the
database..
                If CheckDataInDB(j) = True Then

                    'Fetch from database
                    FetchData(j)

                End If

                'Add other details
                GenerateDetails(j)

            Next

            SetDataGridColumn()

        Catch ex As Exception

            MsgBox(ex.Message,
MsgBoxStyle.Critical)

        End Try



    End Sub

    Private Sub GenerateEmployeeNamesAndNumbers()

        If SQL.SQLDS IsNot Nothing Then

            SQL.SQLDS.Clear()

        End If

        qstring = "SELECT Name AS [Employee Name],
" & _
                        "EmployeeNumber AS
[Employee Number] " & _
                  "FROM tEmployee " & _
                  "WHERE ID_Branch = '" &
BranchCB.SelectedValue & "' " & _
                        "AND ID_Department = '" &
DeptCB.SelectedValue & "' " & _
                        "AND ID_Designation = '" &
DesigCB.SelectedValue & "' " & _
                        "AND isActive = 1"

        SQL.RunQuery(qstring)
```

```
        For i As Integer = 0 To
EndDTP.Value.Date.Subtract(StartDTP.Value.Date).Da
ys

            SQL.SQLDA.Fill(dTable)

        Next

    End Sub

    Private Sub BindToSource()

        bSource.DataSource = dTable
        Me.DataGridView1.DataSource = bSource

    End Sub

    Private Sub AddDatesAndDays()


Me.DataGridView1.Sort(Me.DataGridView1.Columns("Em
ployee Name"),
System.ComponentModel.ListSortDirection.Ascending)

        DTPCol.Name = "Time In 1"
        Me.DataGridView1.Columns.Add(DTPCol)

        DTPCol2.Name = "Time Out 1"
        Me.DataGridView1.Columns.Add(DTPCol2)

        DTPCol3.Name = "Time In 2"
        Me.DataGridView1.Columns.Add(DTPCol3)

        DTPCol4.Name = "Time Out 2"
        Me.DataGridView1.Columns.Add(DTPCol4)

        DTPCol5.Name = "Time In 3"
        Me.DataGridView1.Columns.Add(DTPCol5)

        DTPCol6.Name = "Time Out 3"
        Me.DataGridView1.Columns.Add(DTPCol6)

        Dim a As Integer

        a =
EndDTP.Value.Date.Subtract(StartDTP.Value.Date).Da
ys

        For i As Integer = 0 To
SQL.SQLDS.Tables(0).Rows.Count - 1

            For x As Integer = 0 To a

                DataGridView1.Rows((i * a) + x +
i).Cells("Date").Value =
StartDTP.Value.Date.AddDays(x)
                DataGridView1.Rows((i * a) + x +
i).Cells("Time In 1").Value =
StartDTP.Value.AddDays(x)
                DataGridView1.Rows((i * a) + x +
i).Cells("Time Out 1").Value =
StartDTP.Value.AddDays(x)
                DataGridView1.Rows((i * a) + x +
i).Cells("Time In 2").Value =
StartDTP.Value.AddDays(x)
                DataGridView1.Rows((i * a) + x +
i).Cells("Time Out 2").Value =
StartDTP.Value.AddDays(x)
                DataGridView1.Rows((i * a) + x +
i).Cells("Time In 3").Value =
StartDTP.Value.AddDays(x)
                DataGridView1.Rows((i * a) + x +
i).Cells("Time Out 3").Value =
StartDTP.Value.AddDays(x)
                DataGridView1.Rows((i * a) + x +
i).Cells("Day").Value =
StartDTP.Value.AddDays(x).DayOfWeek.ToString

            Next

        Next



    End Sub

    Private Function CheckDataInDB(ByVal i As
Integer) As Boolean

        qstring = "SELECT ID FROM tEmployee WHERE
EmployeeNumber = '" &
Me.DataGridView1.Rows(i).Cells("Employee
Number").Value.ToString & "'"

        SQL.RunQuery(qstring)

        ENumber =
SQL.SQLDS.Tables(0).Rows(0).Item("ID")

        qstring = "SELECT * FROM
tAttendanceComputation WHERE ID_Employee = '" &
ENumber & "' AND Date = '" &
Me.DataGridView1.Rows(i).Cells("Date").Value.ToStr
ing & "'"

        SQL.RunQuery(qstring)

        If SQL.SQLDS.Tables(0).Rows.Count > 0 Then

            Return True

        Else : Return False

        End If


    End Function

    Private Sub FetchData(ByVal s As Integer)

        qstring = "SELECT DS.Name AS Schedule, " &
_
"tAttendanceComputation.A_In1 AS [Time In 1], " &
_
"tAttendanceComputation.A_Out1 AS [Time Out 1], "
& _
"tAttendanceComputation.A_In2 AS [Time In 2], " &
_
"tAttendanceComputation.A_Out2 AS [Time Out 2], "
& _
```

```
"tAttendanceComputation.A_In3 AS [Time In 3], " &
_
"tAttendanceComputation.A_Out3 AS [Time Out 3], "
& _
"tAttendanceComputation.Absent, " & _
"tAttendanceComputation.isRestDay AS [Rest Day], "
& _
"tAttendanceComputation.isLeave AS Leave, " & _
"tAttendanceComputation.isSpecial AS [Special
Holiday], " & _
"tAttendanceComputation.isLegal AS [Legal
Holiday], " & _
"tAttendanceComputation.isOvertime AS [Overtime
Filed] " & _
                "FROM tAttendanceComputation " &
_
                        "INNER JOIN tDailySchedule
AS DS ON DS.ID = tAttendanceComputation.ID_DS " &
_
                        "INNER JOIN tEmployee AS
Emp ON Emp.ID = tAttendanceComputation.ID_Employee
" & _
                "WHERE Emp.EmployeeNumber = '" &
Me.DataGridView1.Rows(s).Cells("Employee
Number").Value.ToString & "' " & _
                        "AND Date = '" &
Me.DataGridView1.Rows(s).Cells("Date").Value.ToStr
ing & "'"

        SQL.RunQuery(qstring)


Me.DataGridView1.Rows(s).Cells("Schedule").Value =
SQL.SQLDS.Tables(0).Rows(0).Item("Schedule").ToStr
ing
        Me.DataGridView1.Rows(s).Cells("Time In
1").Value = SQL.SQLDS.Tables(0).Rows(0).Item("Time
In 1").ToString
        Me.DataGridView1.Rows(s).Cells("Time Out
1").Value = SQL.SQLDS.Tables(0).Rows(0).Item("Time
Out 1").ToString
        Me.DataGridView1.Rows(s).Cells("Time In
2").Value = SQL.SQLDS.Tables(0).Rows(0).Item("Time
In 2").ToString
        Me.DataGridView1.Rows(s).Cells("Time Out
2").Value = SQL.SQLDS.Tables(0).Rows(0).Item("Time
Out 2").ToString
        Me.DataGridView1.Rows(s).Cells("Time In
3").Value = SQL.SQLDS.Tables(0).Rows(0).Item("Time
In 3").ToString
        Me.DataGridView1.Rows(s).Cells("Time Out
3").Value = SQL.SQLDS.Tables(0).Rows(0).Item("Time
Out 3").ToString

Me.DataGridView1.Rows(s).Cells("Absent").Value =
SQL.SQLDS.Tables(0).Rows(0).Item("Absent").ToStrin
g
        Me.DataGridView1.Rows(s).Cells("Rest
Day").Value =
SQL.SQLDS.Tables(0).Rows(0).Item("Rest
Day").ToString
Me.DataGridView1.Rows(s).Cells("Leave").Value =
SQL.SQLDS.Tables(0).Rows(0).Item("Leave").ToString
        Me.DataGridView1.Rows(s).Cells("Special
Holiday").Value =
SQL.SQLDS.Tables(0).Rows(0).Item("Special
Holiday").ToString
        Me.DataGridView1.Rows(s).Cells("Legal
Holiday").Value =
SQL.SQLDS.Tables(0).Rows(0).Item("Legal
Holiday").ToString
        Me.DataGridView1.Rows(s).Cells("Overtime
Filed").Value =
SQL.SQLDS.Tables(0).Rows(0).Item("Overtime
Filed").ToString

    End Sub

    Private Sub GenerateDetails(ByVal f As
Integer)

        'Original Schedule
        qstring = "SELECT " & _
                        "sun.Name AS Sunday, " & _
                        "mon.Name AS Monday, " & _
                        "tue.Name AS Tuesday, " &
_
                        "wed.Name AS Wednesday, "
& _
                        "thu.Name AS Thursday, " &
_
                        "fri.Name AS Friday, " & _
                        "sat.Name AS Saturday " &
_
                "FROM tEmployee " & _
                        "INNER JOIN
tWeeklySchedule AS week ON week.ID =
tEmployee.ID_WeeklySchedule " & _
                        "INNER JOIN tDailySchedule
AS sun ON sun.ID = week.ID_SundayDS " & _
                        "INNER JOIN tDailySchedule
AS mon ON mon.ID = week.ID_MondayDS " & _
                        "INNER JOIN tDailySchedule
AS tue ON tue.ID = week.ID_TuesdayDS " & _
                        "INNER JOIN tDailySchedule
AS wed ON wed.ID = week.ID_WednesdayDS " & _
                        "INNER JOIN tDailySchedule
AS thu ON thu.ID = week.ID_ThursdayDS " & _
                        "INNER JOIN tDailySchedule
AS fri ON fri.ID = week.ID_FridayDS " & _
                        "INNER JOIN tDailySchedule
AS sat ON sat.ID = week.ID_SaturdayDS " & _
                "WHERE tEmployee.EmployeeNumber
= '" & DataGridView1.Rows(f).Cells("Employee
Number").Value & "'"

        SQL.RunQuery(qstring)


Me.DataGridView1.Rows(f).Cells("Schedule").Value =
SQL.SQLDS.Tables(0).Rows(0).Item(Me.DataGridView1.
Rows(f).Cells("Day").Value).ToString

        'Schedule from Filed Change of Shift
Schedule (Approved)
        qstring = "SELECT " & _
                        "DS.Name AS Schedule " & _
                "FROM tFileChangeShiftSchedule "
& _
```

```
                        "INNER JOIN tDailySchedule
AS DS ON DS.ID = tFileChangeShiftSchedule.ID_DS "
& _
                        "INNER JOIN tEmployee AS
Emp ON Emp.ID =
tFileChangeShiftSchedule.ID_Employee " & _
                    "WHERE Emp.EmployeeNumber = '" &
Me.DataGridView1.Rows(f).Cells("Employee
Number").Value.ToString & "' " & _
                        "AND Date = '" &
Me.DataGridView1.Rows(f).Cells("Date").Value.ToStr
ing & "' " & _
                        "AND isApproved = '1'"

        SQL.RunQuery(qstring)

        If SQL.SQLDS.Tables(0).Rows.Count > 0 Then


Me.DataGridView1.Rows(f).Cells("Schedule").Value =
SQL.SQLDS.Tables(0).Rows(0).Item("Schedule").ToStr
ing

        End If

        'Rest Day from Filed Change of Rest Day

        qstring = "SELECT * FROM
tFileChangeRestDay " & _
                        "INNER JOIN tEmployee AS
Emp ON Emp.ID = tFileChangeRestDay.ID_Employee " &
_
                    "WHERE Emp.EmployeeNumber = '" &
Me.DataGridView1.Rows(f).Cells("Employee
Number").Value.ToString & "' " & _
                        "AND Date = '" &
Me.DataGridView1.Rows(f).Cells("Date").Value.ToStr
ing & "' " & _
                        "AND isApproved = '1'"

        SQL.RunQuery(qstring)

        If SQL.SQLDS.Tables(0).Rows.Count > 0 Then


Me.DataGridView1.Rows(f).Cells("Schedule").Value =
"Rest Day"

        End If

        'Rest Day
        If
Me.DataGridView1.Rows(f).Cells("Schedule").Value =
"Rest Day" Then

            Me.DataGridView1.Rows(f).Cells("Rest
Day").Value = True

        Else :
Me.DataGridView1.Rows(f).Cells("Rest Day").Value =
False

        End If

        'Leave

        qstring = "SELECT LOA.Code FROM
tFileLeaveOfAbsence " & _
                        "INNER JOIN tEmployee AS
Emp ON Emp.ID = tFileLeaveOfAbsence.ID_Employee "
& _
                        "INNER JOIN tLeaveType AS
LOA ON LOA.ID = tFileLeaveOfAbsence.ID_LeaveType "
& _
                    "WHERE ('" &
DataGridView1.Rows(f).Cells("Date").Value & "'
BETWEEN tFileLeaveOfAbsence.StartDate AND
tFileLeaveOfAbsence.EndDate) AND " & _
                        "isApproved = 1 AND " & _
                        "Emp.EmployeeNumber = '" &
DataGridView1.Rows(f).Cells("Employee
Number").Value & "'"

        SQL.RunQuery(qstring)

        If SQL.SQLDS.Tables(0).Rows.Count > 0 Then


DataGridView1.Rows(f).Cells("Leave").Value = True

            LeaveType =
SQL.SQLDS.Tables(0).Rows(0).Item("Code").ToString

        Else :
DataGridView1.Rows(f).Cells("Leave").Value = False

        End If

        'Special Holiday

        qstring = "SELECT Loc.City, " & _
                                    "Branch.Name,
" & _

"tSpecialHoliday.Name " & _
                                "FROM
tSpecialHoliday " & _
                                    "INNER JOIN
tLocation AS Loc ON Loc.ID =
tSpecialHoliday.ID_Location " & _
                                    "INNER JOIN
tBranch AS Branch ON Branch.ID_Location = Loc.ID "
& _
                                "WHERE Date = '" &
DataGridView1.Rows(f).Cells("Date").Value & "' AND
" & _
                                    "Branch.ID =
'" & BranchCB.SelectedValue & "'"

        SQL.RunQuery(qstring)

        If SQL.SQLDS.Tables(0).Rows.Count > 0 Then

            DataGridView1.Rows(f).Cells("Special
Holiday").Value = True

        Else :
DataGridView1.Rows(f).Cells("Special
Holiday").Value = False

        End If

        'Legal Holiday

        qstring = "SELECT * FROM tLegalHoliday
WHERE Date = '" &
DataGridView1.Rows(f).Cells("Date").Value & "'"
```

```
        SQL.RunQuery(qstring)

        If SQL.SQLDS.Tables(0).Rows.Count > 0 Then

            DataGridView1.Rows(f).Cells("Legal
Holiday").Value = True

        Else : DataGridView1.Rows(f).Cells("Legal
Holiday").Value = False

        End If

        'Overtime Filed

        qstring = "SELECT Emp.ID " & _
                  "FROM tFileOvertime " & _
                        "INNER JOIN tEmployee AS
Emp ON Emp.ID = tFileOvertime.ID_Employee " & _
                  "WHERE tFileOvertime.Date = '" &
DataGridView1.Rows(f).Cells("Date").Value & "' AND
" & _
                        "tFileOvertime.isApproved
= 1 AND " & _
                        "Emp.EmployeeNumber = '" &
DataGridView1.Rows(f).Cells("Employee
Number").Value & "'"

        SQL.RunQuery(qstring)

        If SQL.SQLDS.Tables(0).Rows.Count > 0 Then

            DataGridView1.Rows(f).Cells("Overtime
Filed").Value = True

        Else :
DataGridView1.Rows(f).Cells("Overtime
Filed").Value = False

        End If

    End Sub

    Private Sub SaveBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles SaveBtn.Click

        Try

            For x As Integer = 0 To
Me.DataGridView1.Rows.Count - 1

                'Set details

                SetDetails(x)

                If CheckDataInDB(x) = True Then

                    'Update in DB
                    qstring = UpdateAttendance(x)

                Else

                    'Insert in DB
                    qstring = InsertAttendance(x)

                End If

                'Insert or Update Attendance in DB

                SQL.RunQuery(qstring)

                '-----ADDING COMPENSATION
                'Get COLA and BasicPay
                BasicPay = GetBasicPay()

                'MsgBox(BasicPay)

                COLA = GetCOLA()

                'MsgBox(COLA)

                'Get ID AttendanceComputation
                AddCompensation(x)

            Next

        Catch ex As Exception

            MsgBox(ex.Message,
MsgBoxStyle.Critical)

        End Try

    End Sub

    Private Sub SetDetails(ByVal h As Integer)

        Dim out3a, out3b As String

        datei =
Me.DataGridView1.Rows(h).Cells("Date").Value

        'S_In1

        qstring = "SELECT TimeIn_1, TimeOut_3 FROM
tDailySchedule WHERE Name = '" &
Me.DataGridView1.Rows(h).Cells("Schedule").Value.T
oString & "'"

        'MsgBox(qstring)

        SQL.RunQuery(qstring)

        ins = datei.Date & " " &
SQL.SQLDS.Tables(0).Rows(0).Item("TimeIn_1").ToStr
ing

        'S_Out3
        out3a =
SQL.SQLDS.Tables(0).Rows(0).Item("TimeOut_3").ToSt
ring
        outs = datei.Date & " " &
SQL.SQLDS.Tables(0).Rows(0).Item("TimeOut_3").ToSt
ring

        'ND_In
        qstring = "SELECT TimeIn_1, TimeOut_3 FROM
tDailySchedule WHERE Name = 'Night Differential'"

        SQL.RunQuery(qstring)

        innd = datei.Date & " " &
SQL.SQLDS.Tables(0).Rows(0).Item("TimeIn_1").ToStr
ing

        'ND_Out
```

```
        out3b =
SQL.SQLDS.Tables(0).Rows(0).Item("TimeOut_3").ToSt
ring
        outnd = datei.Date.AddDays(1) & " " &
SQL.SQLDS.Tables(0).Rows(0).Item("TimeOut_3").ToSt
ring

        If out3a = out3b Then

            outs = outnd

        End If

        'isRegular

        If Me.DataGridView1.Rows(h).Cells("Rest
Day").Value = False And
Me.DataGridView1.Rows(h).Cells("Leave").Value =
False And Me.DataGridView1.Rows(h).Cells("Special
Holiday").Value = False And
Me.DataGridView1.Rows(h).Cells("Legal
Holiday").Value = False Then

            Regular = "True"

        Else : Regular = "False"

        End If

        'ID_DS

        qstring = "SELECT ID FROM tDailySchedule
WHERE Name = '" &
Me.DataGridView1.Rows(h).Cells("Schedule").Value.T
oString & "'"

        SQL.RunQuery(qstring)

        SchedID =
SQL.SQLDS.Tables(0).Rows(0).Item("ID").ToString

    End Sub

    Private Function UpdateAttendance(ByVal g As
Integer)

        qstring = "UPDATE tAttendanceComputation
SET " & _
                          "A_In1 = '" &
Me.DataGridView1.Rows(g).Cells("Time In
1").Value.ToString & "', " & _
                          "A_Out1 = '" &
Me.DataGridView1.Rows(g).Cells("Time Out
1").Value.ToString & "', " & _
                          "A_In2 = '" &
Me.DataGridView1.Rows(g).Cells("Time In
2").Value.ToString & "', " & _
                          "A_Out2 = '" &
Me.DataGridView1.Rows(g).Cells("Time Out
2").Value.ToString & "', " & _
                          "A_In3 = '" &
Me.DataGridView1.Rows(g).Cells("Time In
3").Value.ToString & "', " & _
                          "A_Out3 = '" &
Me.DataGridView1.Rows(g).Cells("Time Out
3").Value.ToString & "', " & _
                          "Absent = '" &
Me.DataGridView1.Rows(g).Cells("Absent").Value.ToS
tring & "', " & _
                          "isRestDay = '" &
Me.DataGridView1.Rows(g).Cells("Rest
Day").Value.ToString & "', " & _
                          "isLeave = '" &
Me.DataGridView1.Rows(g).Cells("Leave").Value.ToSt
ring & "', " & _
                          "isSpecial = '" &
Me.DataGridView1.Rows(g).Cells("Special
Holiday").Value.ToString & "', " & _
                          "isLegal = '" &
Me.DataGridView1.Rows(g).Cells("Legal
Holiday").Value.ToString & "', " & _
                          "isOvertime = '" &
Me.DataGridView1.Rows(g).Cells("Overtime
Filed").Value.ToString & "', " & _
                          "isRegular = '" & Regular
& "', " & _
                          "S_In1 = '" & ins & "', "
& _
                          "S_Out3 = '" & outs & "',
" & _
                          "ND_In = '" & innd & "', "
& _
                          "ND_Out = '" & outnd & "'"

        If
Me.DataGridView1.Rows(g).Cells("Leave").Value.ToSt
ring = "True" Then

            qstring &= ", " & LeaveType & " = '1'"

        End If

        qstring &= " WHERE ID_Employee = '" &
ENumber & "' AND Date = '" & datei.Date & "'"

        Return qstring

    End Function

    Private Function InsertAttendance(ByVal e As
Integer)

        Dim add, add2 As String

        If
Me.DataGridView1.Rows(e).Cells("Leave").Value.ToSt
ring = "True" Then

            add = ", " & LeaveType

            add2 = ", '1'"

        Else : add = ""

            add2 = ""

        End If

        qstring = "INSERT INTO
tAttendanceComputation " & _
                          "(ID_Employee, Date,
ID_DS, S_In1, A_In1, A_Out1, A_In2, A_Out2, A_In3,
S_Out3, A_Out3, ND_Out, ND_In, isRestDay, isLeave,
isSpecial, isLegal, isOvertime, isRegular, Absent"
& add & ") " & _
```

```
                        "VALUES " & _
                        "(" & _
                        "'" & ENumber & "', " & _
                        "'" & datei & "', " & _
                        "'" & SchedID & "', " & _
                        "'" & ins & "', " & _
                        "'" &
Me.DataGridView1.Rows(e).Cells("Time In 1").Value
& "', " & _
                        "'" &
Me.DataGridView1.Rows(e).Cells("Time Out 1").Value
& "', " & _
                        "'" &
Me.DataGridView1.Rows(e).Cells("Time In 2").Value
& "', " & _
                        "'" &
Me.DataGridView1.Rows(e).Cells("Time Out 2").Value
& "', " & _
                        "'" &
Me.DataGridView1.Rows(e).Cells("Time In 3").Value
& "', " & _
                        "'" & outs & "', " & _
                        "'" &
Me.DataGridView1.Rows(e).Cells("Time Out 3").Value
& "', " & _
                        "'" & outnd & "', " & _
                        "'" & innd & "', " & _
                        "'" &
Me.DataGridView1.Rows(e).Cells("Rest Day").Value &
"', " & _
                        "'" &
Me.DataGridView1.Rows(e).Cells("Leave").Value &
"', " & _
                        "'" &
Me.DataGridView1.Rows(e).Cells("Special
Holiday").Value & "', " & _
                        "'" &
Me.DataGridView1.Rows(e).Cells("Legal
Holiday").Value & "', " & _
                        "'" &
Me.DataGridView1.Rows(e).Cells("Overtime
Filed").Value & "', " & _
                        "'" & Regular & "', " & _
                        "'" &
Me.DataGridView1.Rows(e).Cells("Absent").Value &
"'" & _
                        add2 & ")"

        Return qstring

    End Function

    Private Function GetCOLA()

        SQL.RunQuery("SELECT COLA FROM
tDesignation WHERE ID = '" & DesigCB.SelectedValue
& "'")

        Return
SQL.SQLDS.Tables(0).Rows(0).Item("COLA")

    End Function

    Private Function GetBasicPay()

        SQL.RunQuery("SELECT BasicPay FROM
tDesignation WHERE ID = '" & DesigCB.SelectedValue
& "'")

        Return
SQL.SQLDS.Tables(0).Rows(0).Item("BasicPay")

    End Function

    Private Sub AddCompensation(ByVal q As
Integer)

        qstring = "SELECT ID, " & _
                            "ID_Employee, Date, "
& _
                            "RegularHours AS
Regular, " & _
                            "OT, " & _
                            "ND, " & _
                            "OTND, " & _
                            "WRD, " & _
                            "WRDND, " & _
                            "OTRD, " & _
                            "OTRDND, " & _
                            "SH, " & _
                            "SHND, " & _
                            "SHOT, " & _
                            "SHOTND, " & _
                            "WRH, " & _
                            "WRHND, " & _
                            "OTRH, " & _
                            "OTRHND, " & _
                            "WSR, " & _
                            "WSRND, " & _
                            "OTSR, " & _
                            "OTSRND, " & _
                            "WRR, " & _
                            "WRRND, " & _
                            "OTRR, " & _
                            "OTRRND " & _
                    "FROM tAttendanceComputation
WHERE ID_Employee = '" & ENumber & "' AND Date =
'" & Me.DataGridView1.Rows(q).Cells("Date").Value
& "'"

        'MsgBox(qstring)

        SQL.RunQuery(qstring)

        IDAttendanceComp =
SQL.SQLDS.Tables(0).Rows(0).Item("ID").ToString


'MsgBox(SQL.SQLDS.Tables(0).Rows(0).Item(4).ToStri
ng & " Row Count")

        'arraylist

        Dim CValues As New ArrayList
        Dim CNames As New ArrayList

        For y As Integer = 3 To 26

            If Not
Double.Parse(SQL.SQLDS.Tables(0).Rows(0).Item(y).T
oString) = 0.0 Then


CValues.Add(SQL.SQLDS.Tables(0).Rows(0).Item(y).To
String)

CNames.Add(SQL.SQLDS.Tables(0).Columns(y).ColumnNa
me.ToString)
```

```
            End If

        Next

        For g As Integer = 0 To CNames.Count - 1

            Dim IDAdjPolicy, Rate, Duration As
String

            IDAdjPolicy = ""
            Rate = ""
            Duration = ""

            Duration = CValues.Item(g).ToString

            qstring = "SELECT ID, Rate FROM
tAdjustmentPolicy WHERE Code = '" &
CNames.Item(g).ToString & "'"

            SQL.RunQuery(qstring)

            IDAdjPolicy =
SQL.SQLDS.Tables(0).Rows(0).Item("ID").ToString

            Rate =
SQL.SQLDS.Tables(0).Rows(0).Item("Rate").ToString

            qstring = "SELECT * FROM
tCompensationList WHERE ID_Employee = '" & ENumber
& "' AND ID_AttendanceComputation = '" &
IDAttendanceComp & "'"

            SQL.RunQuery(qstring)

            'check if it has rows, if there are
then delete

            If Not SQL.SQLDS.Tables Is Nothing
Then

                qstring = "DELETE FROM
tCompensationList WHERE ID_Employee = '" & ENumber
& "' AND ID_AttendanceComputation = '" &
IDAttendanceComp & "'"

                SQL.RunQuery(qstring)

            End If

            'insert new rows

            qstring = "INSERT INTO
tCompensationList (ID_Employee, ID_Designation,
BasicPay, COLA, Date, Duration, ID_AdjPolicy,
Rate, ID_AttendanceComputation) VALUES " & _
                            "(" & _
                            "'" & ENumber & "', " &
_
                            "'" &
DesigCB.SelectedValue & "', " & _
                            "'" & BasicPay & "', "
& _
                            "'" & COLA & "', " & _
                            "'" &
Me.DataGridView1.Rows(q).Cells("Date").Value & "',
" & _
                            "'" & Duration & "', "
& _
                            "'" & IDAdjPolicy & "',
" & _
                            "'" & Rate & "'," & _
                            "'" & IDAttendanceComp
& "'" & _
                            ")"

            SQL.RunQuery(qstring)

        Next

    End Sub

    Private Sub PrintDTRBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles PrintDTRBtn.Click

        'ReportForm.Show()
        PdfDTR.PrintPDFDTR()
        Trail.AddTrail("Printed DTR")

    End Sub

    Private Sub PrintASBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles PrintASBtn.Click

        'PrintAS.Show()

PdfAttendanceSummary.PrintPDFAttendanceSummary()
        Trail.AddTrail("Printed Attendance
Summary")


    End Sub

    Private Sub PrintComBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles PrintComBtn.Click

        'Print.Show()
        PdfCompensation.PrintPDFCompensation()
        Trail.AddTrail("Printed Compensation")

    End Sub

    Private Sub PrintPayslipBtn_Click(ByVal sender
As System.Object, ByVal e As System.EventArgs)
Handles PrintPayslipBtn.Click

        'PrintPS.Show()
        PayslipPDF.PDFPayslip()
        Trail.AddTrail("Printed Payslip")

    End Sub

End Class
Public Class User

    Private Duplicate As New CheckDuplicates
    Private Trail As New AuditTrail
    Private SQL As New SQLControl
    Private QString As String 'Query String for
insert or update a user.
    Private AString As String 'Query String for
Audit Trail
```

```
    Private Sub CancelBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles CancelBtn.Click

        Me.Close()

    End Sub

    Private Function CheckUsername()

        SQL.RunQuery("SELECT Username FROM tUser
WHERE Username = '" & NUUsrTB.Text & "'")

        If SQL.SQLDS.Tables(0).Rows.Count > 0 Then

            Return True

        Else : Return False

        End If

    End Function

    Private Function CheckName()

        SQL.RunQuery("SELECT * FROM tUser WHERE
LastName = '" & LNameTB.Text & "' AND FirstName =
'" & FNameTB.Text & "' AND MiddleName = '" & "'")

        If SQL.SQLDS.Tables(0).Rows.Count > 0 Then

            Return True

        Else : Return False

        End If

    End Function

    Private Sub SaveBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles SaveBtn.Click

        Try
            If MainWindow.HasRow Then
                'Edit

                SQL.RunQuery("UPDATE tUser SET " &
_
                                        "LastName
= '" & LNameTB.Text & "', " & _
                                        "FirstName
= '" & FNameTB.Text & "', " & _

"MiddleName = '" & MNameTB.Text & "', " & _
                                        "isActive
= '" & AUserChB.Checked & "', " & _
                                        "isAdmin =
'" & AdminChB.Checked & "', " & _
                                        "Username
= '" & NUUsrTB.Text & "', " & _
                                        "Password
= '" & NUPwdTB.Text & "', " & _
                                        "Comment =
'" & CommentTB.Text & "', " & _

"isApproved = '" & ApproveChB.Checked & "'" & _
                                    "WHERE ID = '" &
MainWindow.ID & "'")

                If Duplicate.IsDuplicated("SELECT
* FROM tUser WHERE Username = '" & NUUsrTB.Text &
"' AND Password = '" & NUPwdTB.Text & "'") Then

                    Trail.AddTrail("Updated a
user.")

                    MsgBox("Changes saved!",
MsgBoxStyle.Information)

                    Me.Close()

                    MainWindow.LoadDataGrid()

                Else

                    MsgBox("Changes not saved!",
MsgBoxStyle.Information)

                End If

            Else

                'New

                If LNameTB.Text = "" Or
FNameTB.Text = "" Or NUUsrTB.Text = "" Or
NUPwdTB.Text = "" Then

                    MsgBox("Please complete
fields.", MsgBoxStyle.Information)

                Else

                    If CheckName() = True Then

                        MsgBox("Account already
exists for " & LNameTB.Text & ", " & FNameTB.Text
& " " & MNameTB.Text & ".", MsgBoxStyle.Critical)

                    ElseIf CheckUsername() = True
Then

                        MsgBox("Username already
taken.", MsgBoxStyle.Critical)

                    Else

                        SQL.RunQuery("INSERT INTO
tUser " & _

"(LastName, FirstName, MiddleName, isActive,
isAdmin, Username, Password, Comment, isApproved)
" & _
                                    "VALUES " & _
                                           "(" & _

"'" & LNameTB.Text & "', " & _

"'" & FNameTB.Text & "', " & _

"'" & MNameTB.Text & "', " & _

"'" & AUserChB.Checked.ToString & "', " & _

"'" & AdminChB.Checked.ToString & "', " & _

"'" & NUUsrTB.Text & "', " & _
```

```
"'" & NUPwdTB.Text & "', " & _

"'" & CommentTB.Text & "', " & _

"'0'" & _
                                        ")")

                        If CheckName() Then


Trail.AddTrail("Inserted a user.")

                            MsgBox("User saved!",
MsgBoxStyle.Information)

                            Me.Close()

                        Else

                            MsgBox("User not
saved!", MsgBoxStyle.Information)

                        End If


                    End If


                End If


            End If


        Catch ex As Exception

            MsgBox(ex.Message,
MsgBoxStyle.Critical)

        End Try

    End Sub

    Public Sub DeleteUser()

        'MsgBox(MainWindow.ID)

        Dim result1 As DialogResult =
MessageBox.Show("Are you sure to delete " &
MainWindow.NameTB & "?", _
                "Delete User", _
                MessageBoxButtons.YesNo)

        If result1 =
Windows.Forms.DialogResult.Yes Then

            MainWindow.QString = "DELETE from
tUser WHERE ID = '" & MainWindow.ID & "'"

            Trail.AddTrail("Deleted " &
MainWindow.NameTB & " from Users.")

            MainWindow.LoadDataGrid()

            SelectUser()

        End If

    End Sub

    Public Sub SelectUser()

        MainWindow.NewBtn.Visible = True
        MainWindow.EditBtn.Visible = True
        MainWindow.RefreshBtn.Visible = True
        MainWindow.DateTimePicker1.Visible = False
        MainWindow.Label2.Visible = False
        MainWindow.DeleteBtn.Visible = True

        MainWindow.GroupBox.Visible = True
        MainWindow.GroupBox.Text = "User"

        MainWindow.QString = "SELECT ID, LastName,
FirstName, MiddleName, Username, isActive,
isAdmin, isApproved, Comment FROM tUser"

        MainWindow.LoadDataGrid()

        Trail.AddTrail("Viewed Users.")

    End Sub

    Private Sub User_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load

        ApproveChB.Visible = False

        If MainWindow.HasRow Then

            ApproveChB.Visible = True
            LNameTB.Enabled = False
            FNameTB.Enabled = False
            MNameTB.Enabled = False
            NUUsrTB.Enabled = False
            NUPwdTB.Visible = False


            LNameTB.Text = MainWindow.lname
            FNameTB.Text = MainWindow.fname
            MNameTB.Text = MainWindow.mname
            NUUsrTB.Text = MainWindow.username
            NUPwdTB.Text = MainWindow.password
            CommentTB.Text = MainWindow.Comment
            AUserChB.Checked = MainWindow.isActive
            AdminChB.Checked = MainWindow.isAdmin
            ApproveChB.Checked =
MainWindow.isApproved

        End If

    End Sub

End Class


Public Class ChangePassword

    Private SQL As New SQLControl
    Private Dupe As New CheckDuplicates
    Dim p1, p2 As String

    Private Sub CancelBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles CancelBtn.Click

        Me.Close()
```

```
    End Sub

    Private Sub OKBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles OKBtn.Click

        If PasswordTB.Text = "" Then

            If Label1.Text = "Enter current
password." Then

                MsgBox("Enter current password.",
MsgBoxStyle.Information)

            ElseIf Label1.Text = "Enter new
password." Then

                MsgBox("Enter new password.",
MsgBoxStyle.Information)

            Else

                MsgBox("Confirm new password.",
MsgBoxStyle.Information)

            End If

        Else

            If Label1.Text = "Enter current
password." Then

                If Dupe.IsDuplicated("SELECT *
FROM tUser WHERE username = '" &
My.Settings.USERTXT & "' and password = '" &
PasswordTB.Text & "'") Then

                    Label1.Text = "Enter new
password."

                    PasswordTB.Text = ""

                Else

                    MsgBox("Wrong password.",
MsgBoxStyle.Information)

                End If

            ElseIf Label1.Text = "Enter new
password." Then

                p1 = PasswordTB.Text

                Label1.Text = "Confirm new
password."

                PasswordTB.Text = ""

            Else

                If p1.Equals(PasswordTB.Text) Then

                    MsgBox(p1)
                    MsgBox(p2)
                    MsgBox("UPDATE tUser SET
Password = '" & PasswordTB.Text & "' WHERE
username = '" & My.Settings.USERTXT & "'")

                    SQL.RunQuery("UPDATE tUser SET
Password = '" & PasswordTB.Text & "' WHERE
username = '" & My.Settings.USERTXT & "'")

                    If Dupe.IsDuplicated("SELECT *
FROM tUser WHERE username = '" &
My.Settings.USERTXT & "' and password = '" & p1 &
"'") Then

                        MsgBox("Password
changed!", MsgBoxStyle.Information)

                        Me.Close()

                    Else

                        MsgBox("Password not
changed!", MsgBoxStyle.Information)

                    End If

                Else

                    MsgBox("Passwords do not
match.", MsgBoxStyle.Information)

                    Label1.Text = "Enter new
password."

                End If

            End If

        End If

    End Sub

    Private Sub ChangePassword_Load(ByVal sender
As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load

    End Sub
End Class
Public Class AuditTrail

    Private SQL As New SQLControl

    Public Sub AddTrail(ByVal Action As String)

        SQL.RunQuery("INSERT INTO tAuditTrail
(ID_Username, Action, DateModified) VALUES ('" &
GetUserID() & "', '" & Action & "', '" &
DateTime.Now.ToString & "')")

    End Sub

    Public Function GetUserID()

        SQL.RunQuery("SELECT ID FROM tUser WHERE
Username = '" & My.Settings.USERTXT & "'")

        Return
SQL.SQLDS.Tables(0).Rows(0).Item("ID").ToString

    End Function

    Public Sub SelectViewAuditTrail()
```

```
        MainWindow.GroupBox.Visible = True
        MainWindow.GroupBox.Text = "View Own Audit
Trail"

        MainWindow.QString = "SELECT Action,
DateModified AS [Date Modified] FROM tAuditTrail
WHERE ID_Username = '" & GetUserID() & "' ORDER BY
DateModified DESC"

        SQL.RunQuery(MainWindow.QString)

        MainWindow.LoadDataGrid()

        AddTrail("Viewed Own Audit Trail.")

    End Sub

    Public Sub SelectViewSystemTrail()

        MainWindow.GroupBox.Visible = True
        MainWindow.GroupBox.Text = "View System
Audit Trail"

        MainWindow.QString = "SELECT
USERS.Username AS [Username], TRAIL.Action AS
Action, TRAIL.DateModified AS [Date Modified] FROM
tAuditTrail AS TRAIL INNER JOIN tUser AS USERS ON
USERS.ID = TRAIL.ID_Username ORDER BY DateModified
DESC"


        SQL.RunQuery(MainWindow.QString)

        MainWindow.LoadDataGrid()

        AddTrail("Viewed System Audit Trail.")

    End Sub

End Class

Public Class LeaveCredit

    Private Duplicate As New CheckDuplicates
    Private Trail As New AuditTrail
    Private SQL As New SQLControl
    Private QString As String
    Private AString As String 'Query String for
Audit Trail

    Private Sub CancelBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles CancelBtn.Click

        Me.Close()

    End Sub

    Private Sub SaveBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles SaveBtn.Click


        Try

            If NLeaveTypeTB.Text = "" Or
CLeaveTypeTB.Text = "" Then

                MsgBox("Please complete fields.",
MsgBoxStyle.Information)

            Else

                Dim DString As String

                DString = "SELECT * FROM
tLeaveType WHERE Name = '" & NLeaveTypeTB.Text &
"' AND Code = '" & CLeaveTypeTB.Text & "' AND
isActive = '" & ALeaveTypeChB.Checked & "'"

                If Duplicate.IsDuplicated(DString)
Then

                    MsgBox("Leave Credit already
exists!", MsgBoxStyle.Critical)

                Else

                    If MainWindow.HasRow = False
Then

                        QString = "INSERT INTO
tLeaveType (Name, Code, isActive, Comment) VALUES
" & _
                                            "(" & _
                                            "'" &
NLeaveTypeTB.Text & "', " & _
                                            "'" &
CLeaveTypeTB.Text & "', " & _
                                            "'" &
ALeaveTypeChB.Checked & "', " & _
                                            "'" &
CommentTB.Text & "'" & _
                                            ")"

                        AString = "Inserted " &
NLeaveTypeTB.Text & " leave credit."

                    ElseIf MainWindow.HasRow =
True Then

                        QString = "UPDATE
tLeaveType SET " & _
                                            "Name = '"
& NLeaveTypeTB.Text & "', " & _
                                            "Code = '"
& CLeaveTypeTB.Text & "', " & _
                                            "isActive
= '" & ALeaveTypeChB.Checked & "', " & _
                                            "Comment =
'" & CommentTB.Text & "' " & _
                                        "WHERE ID = '" &
MainWindow.ID & "'"

                        AString = "Updated " &
NLeaveTypeTB.Text & " leave credit."

                    End If

                    SQL.RunQuery(QString)

                    If
Duplicate.IsDuplicated(DString) Then

                        Trail.AddTrail(AString)
```

```
                        MsgBox("Saved!",
MsgBoxStyle.Information)

                        Me.Close()

                        MainWindow.LoadDataGrid()

                    Else

                        MsgBox("Leave credit not
saved!", MsgBoxStyle.Information)

                    End If

                End If

            End If

        Catch ex As Exception

            MsgBox(ex.Message)

        End Try


    End Sub

    Public Sub SelectLeaveCredit()

        MainWindow.NewBtn.Visible = True
        MainWindow.EditBtn.Visible = True
        MainWindow.RefreshBtn.Visible = True
        MainWindow.DateTimePicker1.Visible = False
        MainWindow.DeleteBtn.Visible = True
        MainWindow.Label2.Visible = False

        MainWindow.GroupBox.Visible = True
        MainWindow.GroupBox.Text = "Leave Credit"

        MainWindow.QString = "EXEC
selectLeaveCredit"

        MainWindow.LoadDataGrid()


MainWindow.DataGrid.Columns("ID").DisplayIndex = 0
        MainWindow.DataGrid.Columns("Leave
Credit").DisplayIndex = 1
        MainWindow.DataGrid.Columns("Leave Credit
Code").DisplayIndex = 2

MainWindow.DataGrid.Columns("Active").DisplayIndex
= 3

MainWindow.DataGrid.Columns("Comment").DisplayInde
x = 4

        Trail.AddTrail("Viewed Leave Credit.")

    End Sub

    Public Sub DeleteLeaveCredit()

        Dim result1 As DialogResult =
MessageBox.Show("Are you sure to delete " &
MainWindow.NameTB & "?", _
                "Delete Leave Credit", _
                MessageBoxButtons.YesNo)

        If result1 =
Windows.Forms.DialogResult.Yes Then

            MainWindow.QString = "DELETE from
tLeaveType WHERE ID = '" & MainWindow.ID & "'"

            MainWindow.LoadDataGrid()

            Trail.AddTrail("Deleted " &
MainWindow.NameTB & " from Leave Credit.")

            SelectLeaveCredit()

        End If

    End Sub

    Private Sub LeaveCredit_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load

        If MainWindow.HasRow = True Then

            NLeaveTypeTB.Text = MainWindow.NameTB
            CommentTB.Text = MainWindow.Comment
            CLeaveTypeTB.Text = MainWindow.Code
            ALeaveTypeChB.Checked =
MainWindow.isActive

        End If

    End Sub

End Class
Public Class AjustmentType

    Private Duplicate As New CheckDuplicates
    Private Trail As New AuditTrail
    Private SQL As New SQLControl
    Private QString As String
    Private AString As String 'Query String for
Audit Trail

    Private Sub CancelBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles CancelBtn.Click

        Me.Close()

    End Sub

    Private Sub SaveBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles SaveBtn.Click

        Try

            If NAdjustmentTypeTB.Text = "" Or
CAdjustmentTypeTB.Text = "" Or RateTB.Text = ""
Then

                MsgBox("Please complete fields.",
MsgBoxStyle.Information)

            Else

                Dim DString As String
```

```
                DString = "SELECT * FROM
tAdjustmentPolicy WHERE Name = '" &
NAdjustmentTypeTB.Text & "' AND Code = '" &
CAdjustmentTypeTB.Text & "' AND Rate = '" &
RateTB.Text & "' AND isActive = '" &
AAdjTypeChB.Checked & "'"

                If Duplicate.IsDuplicated(DString)
Then

                    MsgBox("Adjustment Policy
already exists!", MsgBoxStyle.Critical)

                Else

                    If MainWindow.HasRow = False
Then

                        QString = "INSERT INTO
tAdjustmentPolicy (Name, Code, Rate, isActive,
Comment) VALUES " & _
                                            "(" & _
                                            "'" &
NAdjustmentTypeTB.Text & "', " & _
                                            "'" &
CAdjustmentTypeTB.Text & "', " & _
                                            "'" &
RateTB.Text & "', " & _
                                            "'" &
AAdjTypeChB.Checked & "', " & _
                                            "'" &
CommentTB.Text & "'" & _
                                            ")"
                        AString = "Inserted " &
NAdjustmentTypeTB.Text & " adjustment policy."

                    ElseIf MainWindow.HasRow =
True Then

                        QString = "UPDATE
tAdjustmentPolicy SET " & _
                                            "Name = '"
& NAdjustmentTypeTB.Text & "', " & _
                                            "Code = '"
& CAdjustmentTypeTB.Text & "', " & _
                                            "Rate = '"
& RateTB.Text & "', " & _
                                            "isActive
= '" & AAdjTypeChB.Checked & "', " & _
                                            "Comment =
'" & CommentTB.Text & "' " & _
                                    "WHERE ID = '" &
MainWindow.ID & "'"

                        AString = "Updated " &
NAdjustmentTypeTB.Text & " adjustment policy."

                    End If

                    If
Duplicate.IsDuplicated(DString) Then

                        SQL.RunQuery(QString)

                        Trail.AddTrail(AString)

                        MsgBox("Saved!",
MsgBoxStyle.Information)

                        Me.Close()

                        MainWindow.LoadDataGrid()

                    Else

                        MsgBox("Adjustment type
not saved!", MsgBoxStyle.Information)

                    End If

                End If

            End If

        Catch ex As Exception

            MsgBox(ex.Message)

        End Try

    End Sub

    Private Sub AjustmentType_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load

        If MainWindow.HasRow = True Then

            NAdjustmentTypeTB.Text =
MainWindow.NameTB
            CAdjustmentTypeTB.Text =
MainWindow.Code
            RateTB.Text = MainWindow.Rate
            CommentTB.Text = MainWindow.Comment
            AAdjTypeChB.Checked =
MainWindow.isActive

        End If

    End Sub

    Public Sub SelectAdjustmentPolicy()

        MainWindow.NewBtn.Visible = True
        MainWindow.EditBtn.Visible = True
        MainWindow.RefreshBtn.Visible = True
        MainWindow.DeleteBtn.Visible = True
        MainWindow.DateTimePicker1.Visible = False
        MainWindow.Label2.Visible = False

        MainWindow.GroupBox.Visible = True
        MainWindow.GroupBox.Text = "Adjustment
Policy"

        MainWindow.QString = "EXEC
selectAdjustmentPolicy"

        MainWindow.LoadDataGrid()


MainWindow.DataGrid.Columns("ID").DisplayIndex = 0
        MainWindow.DataGrid.Columns("Adjustment
Policy").DisplayIndex = 1
        MainWindow.DataGrid.Columns("Adjustment
Policy Code").DisplayIndex = 2

MainWindow.DataGrid.Columns("Rate").DisplayIndex =
3
```

```
MainWindow.DataGrid.Columns("Active").DisplayIndex
= 4

MainWindow.DataGrid.Columns("Comment").DisplayInde
x = 5

        Trail.AddTrail("Viewed Adjustment
Policy.")

    End Sub

    Public Sub DeleteAdjustmentPolicy()

        Dim result1 As DialogResult =
MessageBox.Show("Are you sure to delete " &
MainWindow.NameTB & "?", _
                "Delete Adjustment Policy", _
                MessageBoxButtons.YesNo)

        If result1 =
Windows.Forms.DialogResult.Yes Then

            MainWindow.QString = "DELETE from
tAdjustmentPolicy WHERE ID = '" & MainWindow.ID &
"'"

            MainWindow.LoadDataGrid()

            Trail.AddTrail("Deleted " &
MainWindow.NameTB & " from Adjustment Policy.")

            SelectAdjustmentPolicy()

        End If

    End Sub

End Class
Public Class WeeklySchedule

    Private Duplicate As New CheckDuplicates
    Private Trail As New AuditTrail
    Private SQL As New SQLControl
    Private QString As String 'Query String for
insert or update weekly schedule
    Private AString As String 'Query String for
Audit Trail

    Private Sub WeeklySchedule_Load(ByVal sender
As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load

        SQL.RunQuery("SELECT ID, Name from
tDailySchedule WHERE isActive = '1'")

        SundayDSCB.DataSource =
SQL.SQLDS.Tables(0)
        SundayDSCB.ValueMember = "ID"
        SundayDSCB.DisplayMember = "Name"

        SQL.RunQuery("SELECT ID, Name from
tDailySchedule WHERE isActive = '1'")

        MondayDSCB.DataSource =
SQL.SQLDS.Tables(0)
        MondayDSCB.ValueMember = "ID"
        MondayDSCB.DisplayMember = "Name"

        SQL.RunQuery("SELECT ID, Name from
tDailySchedule WHERE isActive = '1'")

        TuesdayDSCB.DataSource =
SQL.SQLDS.Tables(0)
        TuesdayDSCB.ValueMember = "ID"
        TuesdayDSCB.DisplayMember = "Name"

        SQL.RunQuery("SELECT ID, Name from
tDailySchedule WHERE isActive = '1'")

        WednesdayDSCB.DataSource =
SQL.SQLDS.Tables(0)
        WednesdayDSCB.ValueMember = "ID"
        WednesdayDSCB.DisplayMember = "Name"

        SQL.RunQuery("SELECT ID, Name from
tDailySchedule WHERE isActive = '1'")

        ThursdayDSCB.DataSource =
SQL.SQLDS.Tables(0)
        ThursdayDSCB.ValueMember = "ID"
        ThursdayDSCB.DisplayMember = "Name"

        SQL.RunQuery("SELECT ID, Name from
tDailySchedule WHERE isActive = '1'")

        FridayDSCB.DataSource =
SQL.SQLDS.Tables(0)
        FridayDSCB.ValueMember = "ID"
        FridayDSCB.DisplayMember = "Name"

        SQL.RunQuery("SELECT ID, Name from
tDailySchedule WHERE isActive = '1'")

        SaturdayDSCB.DataSource =
SQL.SQLDS.Tables(0)
        SaturdayDSCB.ValueMember = "ID"
        SaturdayDSCB.DisplayMember = "Name"

        If MainWindow.HasRow = True Then

            NWeeklyScheduleTB.Text =
MainWindow.NameTB
            AWeeklyScheduleChB.Checked =
MainWindow.isActive
            CommentTB.Text = MainWindow.Comment

            SundayDSCB.SelectedIndex =
SundayDSCB.FindStringExact(MainWindow.sunday)
            MondayDSCB.SelectedIndex =
MondayDSCB.FindStringExact(MainWindow.monday)
            TuesdayDSCB.SelectedIndex =
TuesdayDSCB.FindStringExact(MainWindow.tuesday)
            WednesdayDSCB.SelectedIndex =
WednesdayDSCB.FindStringExact(MainWindow.wednesday
)
            ThursdayDSCB.SelectedIndex =
ThursdayDSCB.FindStringExact(MainWindow.thursday)
            FridayDSCB.SelectedIndex =
FridayDSCB.FindStringExact(MainWindow.friday)
            SaturdayDSCB.SelectedIndex =
SaturdayDSCB.FindStringExact(MainWindow.saturday)

        End If

    End Sub
```

```
    Private Sub SaveBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles SaveBtn.Click

        Try

            If NWeeklyScheduleTB.Text = "" Then

                MsgBox("Enter Weekly Schedule
name.", MsgBoxStyle.Information)

            Else

                Dim DString As String

                DString = "SELECT * FROM
tWeeklySchedule " & _
                          "WHERE " & _
                                "Name = '" &
NWeeklyScheduleTB.Text & "' AND " & _
                                "ID_SundayDS = '"
& SundayDSCB.SelectedValue & "' AND " & _
                                "ID_MondayDS = '"
& MondayDSCB.SelectedValue & "' AND " & _
                                "ID_TuesdayDS = '"
& TuesdayDSCB.SelectedValue & "' AND " & _
                                "ID_WednesdayDS =
'" & WednesdayDSCB.SelectedValue & "' AND " & _
                                "ID_ThursdayDS =
'" & ThursdayDSCB.SelectedValue & "' AND " & _
                                "ID_FridayDS = '"
& FridayDSCB.SelectedValue & "' AND " & _
                                "ID_SaturdayDS =
'" & SaturdayDSCB.SelectedValue & "' AND " & _
                                "isActive = '" &
AWeeklyScheduleChB.Checked & "'"

                If Duplicate.IsDuplicated(DString)
= True Then

                    MsgBox("Weekly Schedule
already exists!", MsgBoxStyle.Critical)

                    Exit Sub

                Else

                    If MainWindow.HasRow = False
Then

                        QString = "INSERT INTO
tWeeklySchedule " & _
                                            "(Name,
ID_SundayDS, ID_MondayDS, ID_TuesdayDS,
ID_WednesdayDS, ID_ThursdayDS, ID_FridayDS,
ID_SaturdayDS, isActive, Comment) VALUES " & _
                                            "(" & _
                                            "'" &
NWeeklyScheduleTB.Text & "', " & _
                                            "'" &
SundayDSCB.SelectedValue & "', " & _
                                            "'" &
MondayDSCB.SelectedValue & "', " & _
                                            "'" &
TuesdayDSCB.SelectedValue & "', " & _
                                            "'" &
WednesdayDSCB.SelectedValue & "', " & _
                                            "'" &
ThursdayDSCB.SelectedValue & "', " & _
                                            "'" &
FridayDSCB.SelectedValue & "', " & _
                                            "'" &
SaturdayDSCB.SelectedValue & "', " & _
                                            "'" &
AWeeklyScheduleChB.Checked & "', " & _
                                            "'" &
CommentTB.Text & "'" & _
                                            ")"

                        AString = "Inserted " &
NWeeklyScheduleTB.Text & " weekly schedule."

                    ElseIf MainWindow.HasRow =
True Then

                        QString = "UPDATE
tWeeklySchedule SET " & _
                                            "Name = '"
& NWeeklyScheduleTB.Text & "', " & _

"ID_SundayDS = '" & SundayDSCB.SelectedValue & "',
" & _

"ID_MondayDS = '" & MondayDSCB.SelectedValue & "',
" & _

"ID_TuesdayDS = '" & TuesdayDSCB.SelectedValue &
"', " & _

"ID_WednesdayDS = '" & WednesdayDSCB.SelectedValue
& "', " & _

"ID_ThursdayDS = '" & ThursdayDSCB.SelectedValue &
"', " & _

"ID_FridayDS = '" & FridayDSCB.SelectedValue & "',
" & _

"ID_SaturdayDS = '" & SaturdayDSCB.SelectedValue &
"', " & _
                                            "isActive
= '" & AWeeklyScheduleChB.Checked & "', " & _
                                            "Comment =
'" & CommentTB.Text & "' " & _
                                    "WHERE ID = '" &
MainWindow.ID & "'"

                        AString = "Updated " &
NWeeklyScheduleTB.Text & " weekly schedule"

                    End If

                    SQL.RunQuery(QString)

                    If
Duplicate.IsDuplicated(DString) = True Then

                        Trail.AddTrail(AString)

                        MsgBox("Saved!",
MsgBoxStyle.Information)

                        Me.Close()

                        MainWindow.LoadDataGrid()

                    Else
```

```
                        MsgBox("Weekly Schedule
not saved!", MsgBoxStyle.Information)

                    End If

                End If

            End If

        Catch ex As Exception

            MsgBox(ex.Message,
MsgBoxStyle.Critical)

        End Try

    End Sub

    Public Sub SelectWeeklySchedule()

        MainWindow.NewBtn.Visible = True
        MainWindow.EditBtn.Visible = True
        MainWindow.RefreshBtn.Visible = True
        MainWindow.DateTimePicker1.Visible = False
        MainWindow.DeleteBtn.Visible = True
        MainWindow.Label2.Visible = False

        MainWindow.GroupBox.Visible = True
        MainWindow.GroupBox.Text = "Weekly
Schedule"

        MainWindow.QString = "EXEC
selectWeeklySchedule"

        MainWindow.LoadDataGrid()


MainWindow.DataGrid.Columns("ID").DisplayIndex = 0
        MainWindow.DataGrid.Columns("Weekly
Schedule").DisplayIndex = 1

MainWindow.DataGrid.Columns("Sunday").DisplayIndex
= 2

MainWindow.DataGrid.Columns("Monday").DisplayIndex
= 3

MainWindow.DataGrid.Columns("Tuesday").DisplayInde
x = 4

MainWindow.DataGrid.Columns("Wednesday").DisplayIn
dex = 5

MainWindow.DataGrid.Columns("Thursday").DisplayInd
ex = 6

MainWindow.DataGrid.Columns("Friday").DisplayIndex
= 7

MainWindow.DataGrid.Columns("Saturday").DisplayInd
ex = 8

MainWindow.DataGrid.Columns("Active").DisplayIndex
= 9

MainWindow.DataGrid.Columns("Comment").DisplayInde
x = 10

        Trail.AddTrail("Viewed Weekly Schedule.")
    End Sub

    Public Sub DeleteWeeklySchedule()

        Dim result1 As DialogResult =
MessageBox.Show("Are you sure to delete " &
MainWindow.NameTB & "?", _
                "Delete Weekly Schedule", _
                MessageBoxButtons.YesNo)

        If result1 =
Windows.Forms.DialogResult.Yes Then

            MainWindow.QString = "DELETE from
tWeeklySchedule WHERE ID = '" & MainWindow.ID &
"'"

            MainWindow.LoadDataGrid()

            Trail.AddTrail("Deleted " &
MainWindow.NameTB & " from Weekly Schedule.")

            SelectWeeklySchedule()

        End If

    End Sub

    Private Sub CancelBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles CancelBtn.Click

        Me.Close()

    End Sub

End Class
Public Class DailySchedule

    Private Duplicate As New CheckDuplicates
    Private Trail As New AuditTrail
    Private SQL As New SQLControl
    Private QString As String 'Query String for
insert or update a daily schedule.
    Private AString As String 'Query String for
Audit Trail

    Private Sub CancelBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles CancelBtn.Click

        Me.Close()

    End Sub

    Private Sub SaveBtn_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles SaveBtn.Click

        Try


            Dim out1, in2, out2, in3 As String

            If OUT1TimePicker.Checked = True Then

                out1 =
OUT1TimePicker.Value.ToShortTimeString
```

```
            Else

                out1 = Nothing

            End If

            If IN2TimePicker.Checked = True Then

                in2 =
IN2TimePicker.Value.ToShortTimeString

            Else

                in2 = Nothing

            End If

            If OUT2TimePicker.Checked = True Then

                out2 =
OUT2TimePicker.Value.ToShortTimeString

            Else

                out2 = Nothing

            End If

            If IN3TimePicker.Checked = True Then

                in3 =
IN2TimePicker.Value.ToShortTimeString

            Else

                in3 = Nothing

            End If

            If NDailyScheduleTB.Text = "" Or
NDurationTB.Text = "" Then

                MsgBox("Please complete fields.",
MsgBoxStyle.Information)

            Else

                Dim DString As String

                DString = "SELECT * FROM
tDailySchedule " & _
                          "WHERE " & _
                                "Name = '" &
NDailyScheduleTB.Text & "' AND " & _
                                "Hour = '" &
NDurationTB.Text & "' AND " & _
                                "TimeIn_1 = '" &
IN1TimePicker.Value.ToShortTimeString & "' AND " &
_
                                "TimeOut_1 = '" &
out1 & "' AND " & _
                                "TimeIn_2 = '" &
in2 & "' AND " & _
                                "TimeOut_2 = '" &
out2 & "' AND " & _
                                "TimeIn_3 = '" &
in3 & "' AND " & _
                                "TimeOut_3 = '" &
OUT3TimePicker.Value.ToShortTimeString & "' AND "
& _
                                "isActive = '" &
ADailyScheduleChB.Checked & "'"

                If Duplicate.IsDuplicated(DString)
= True Then

                    MsgBox("Daily Schedule Name
already exists!", MsgBoxStyle.Critical)

                    Exit Sub

                Else

                    If MainWindow.HasRow = False
Then

                        QString = "INSERT INTO
tDailySchedule " & _
                                  "(Name,
Hour, TimeIn_1, TimeOut_1, TimeIn_2, TimeOut_2,
TimeIn_3, TimeOut_3, isActive, Comment) " & _
                                  "VALUES "
& _
                                  "(" & _
                                  "'" &
NDailyScheduleTB.Text & "', " & _
                                  "'" &
NDurationTB.Text & "', " & _
                                  "'" &
IN1TimePicker.Value.ToShortTimeString & "', " & _
                                  "'" & out1
& "', " & _
                                  "'" & in2
& "', " & _
                                  "'" & out2
& "', " & _
                                  "'" & in3
& "', " & _
                                  "'" &
OUT3TimePicker.Value.ToShortTimeString & "', " & _
                                  "'" &
ADailyScheduleChB.Checked & "', " & _
                                  "'" &
CommentTB.Text & "'" & _
                                  ")"

                        AString = "Inserted " &
NDailyScheduleTB.Text & " daily schedule."

                    Else

                        QString = "UPDATE
tDailySchedule SET " & _
                                  "Name = '"
& NDailyScheduleTB.Text & "', " & _
                                  "Hour = '"
& NDurationTB.Text & "', " & _
                                  "TimeIn_1
= '" & IN1TimePicker.Value.ToShortTimeString & "',
" & _
                                  "TimeOut_1
= '" & out1 & "', " & _
                                  "TimeIn_2
= '" & in2 & "', " & _
                                  "TimeOut_2
= '" & out2 & "', " & _
```

```
                                        "TimeIn_3
= '" & in3 & "', " & _
                                        "TimeOut_3
= '" & OUT3TimePicker.Value.ToShortTimeString &
"', " & _
                                        "isActive
= '" & ADailyScheduleChB.Checked & "', " & _
                                        "Comment =
'" & CommentTB.Text & "' " & _
                                    "WHERE " & _
                                        "ID = '" &
MainWindow.ID & "'"

                        AString = "Updated " &
NDailyScheduleTB.Text & " daily schedule."

                    End If

                    SQL.RunQuery(QString)

                    If
Duplicate.IsDuplicated(DString) = True Then

                        Trail.AddTrail(AString)

                        MsgBox("Saved!",
MsgBoxStyle.Information)

                        Me.Close()

                        MainWindow.LoadDataGrid()

                    Else

                        MsgBox("Daily schedule not
saved!", MsgBoxStyle.Information)

                    End If

                End If

            End If

        Catch ex As Exception

            MsgBox(ex.Message,
MsgBoxStyle.Critical)

        End Try

    End Sub

    Private Sub DailySchedule_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load

        IN1TimePicker.CustomFormat = "hh:mm tt"
        IN1TimePicker.Format =
DateTimePickerFormat.Custom
        IN1TimePicker.ShowUpDown = True
        IN1TimePicker.Checked = True
        IN1TimePicker.ResetText()

        OUT1TimePicker.CustomFormat = "hh:mm tt"
        OUT1TimePicker.Format =
DateTimePickerFormat.Custom
        OUT1TimePicker.ShowUpDown = True
        OUT1TimePicker.ShowCheckBox = True
        OUT1TimePicker.ResetText()

        IN2TimePicker.CustomFormat = "hh:mm tt"
        IN2TimePicker.Format =
DateTimePickerFormat.Custom
        IN2TimePicker.ShowUpDown = True
        IN2TimePicker.ShowCheckBox = True
        IN2TimePicker.ResetText()

        OUT2TimePicker.CustomFormat = "hh:mm tt"
        OUT2TimePicker.Format =
DateTimePickerFormat.Custom
        OUT2TimePicker.ShowUpDown = True
        OUT2TimePicker.ShowCheckBox = True
        OUT2TimePicker.ResetText()

        IN3TimePicker.CustomFormat = "hh:mm tt"
        IN3TimePicker.Format =
DateTimePickerFormat.Custom
        IN3TimePicker.ShowUpDown = True
        IN3TimePicker.ShowCheckBox = True
        IN3TimePicker.ResetText()

        OUT3TimePicker.CustomFormat = "hh:mm tt"
        OUT3TimePicker.Format =
DateTimePickerFormat.Custom
        OUT3TimePicker.ShowUpDown = True
        OUT3TimePicker.ResetText()

        If MainWindow.HasRow = True Then

            NDailyScheduleTB.Text =
MainWindow.NameTB
            NDurationTB.Text = MainWindow.Duration
            ADailyScheduleChB.Checked =
MainWindow.isActive
            CommentTB.Text = MainWindow.Comment

            IN1TimePicker.Value = New
DateTime(MainWindow.in1d.Year,
MainWindow.in1d.Month, MainWindow.in1d.Day,
MainWindow.in1d.Hour, MainWindow.in1d.Minute,
MainWindow.in1d.Second)

            OUT1TimePicker.Value = New
DateTime(MainWindow.out1d.Year,
MainWindow.out1d.Month, MainWindow.out1d.Day,
MainWindow.out1d.Hour, MainWindow.out1d.Minute,
MainWindow.out1d.Second)

            IN2TimePicker.Value = New
DateTime(MainWindow.in2d.Year,
MainWindow.in2d.Month, MainWindow.in2d.Day,
MainWindow.in2d.Hour, MainWindow.in2d.Minute,
MainWindow.in2d.Second)

            OUT2TimePicker.Value = New
DateTime(MainWindow.out2d.Year,
MainWindow.out2d.Month, MainWindow.out2d.Day,
MainWindow.out2d.Hour, MainWindow.out2d.Minute,
MainWindow.out2d.Second)

            IN3TimePicker.Value = New
DateTime(MainWindow.in3d.Year,
MainWindow.in3d.Month, MainWindow.in3d.Day,
MainWindow.in3d.Hour, MainWindow.in3d.Minute,
MainWindow.in3d.Second)

            OUT3TimePicker.Value = New
DateTime(MainWindow.out3d.Year,
```

```
MainWindow.out3d.Month, MainWindow.out3d.Day,
MainWindow.out3d.Hour, MainWindow.out3d.Minute,
MainWindow.out3d.Second)

        End If

    End Sub

    Public Sub SelectDailySchedule()

        MainWindow.NewBtn.Visible = True
        MainWindow.EditBtn.Visible = True
        MainWindow.RefreshBtn.Visible = True
        MainWindow.DateTimePicker1.Visible = False
        MainWindow.DeleteBtn.Visible = True
        MainWindow.Label2.Visible = False

        MainWindow.GroupBox.Visible = True
        MainWindow.GroupBox.Text = "Daily
Schedule"

        MainWindow.QString = "EXEC
selectDailySchedule"

        MainWindow.LoadDataGrid()


MainWindow.DataGrid.Columns("ID").DisplayIndex = 0
        MainWindow.DataGrid.Columns("Daily
Schedule").DisplayIndex = 1

MainWindow.DataGrid.Columns("Duration").DisplayInd
ex = 2
        MainWindow.DataGrid.Columns("Time In
1").DisplayIndex = 3
        MainWindow.DataGrid.Columns("Time Out
1").DisplayIndex = 4
        MainWindow.DataGrid.Columns("Time In
2").DisplayIndex = 5
        MainWindow.DataGrid.Columns("Time Out
2").DisplayIndex = 6
        MainWindow.DataGrid.Columns("Time In
3").DisplayIndex = 7
        MainWindow.DataGrid.Columns("Time Out
3").DisplayIndex = 8

MainWindow.DataGrid.Columns("Active").DisplayIndex
= 9

MainWindow.DataGrid.Columns("Comment").DisplayInde
x = 10

        Trail.AddTrail("Viewed Daily Schedule.")

    End Sub

    Public Sub DeleteDailySchedule()

        Dim result1 As DialogResult =
MessageBox.Show("Are you sure to delete " &
MainWindow.NameTB & "?", _
                "Delete Daily Schedule", _
                MessageBoxButtons.YesNo)

        If result1 =
Windows.Forms.DialogResult.Yes Then

            MainWindow.QString = "DELETE from
tDailySchedule WHERE ID = '" & MainWindow.ID & "'"

            MainWindow.LoadDataGrid()

            Trail.AddTrail("Deleted " &
MainWindow.NameTB & " from Daily Schedule.")

            SelectDailySchedule()

        End If

    End Sub

End Class
```

XI. Acknowledgement

"They say that good things take time, but really great things happen in a blink of an eye."

That quotation above really applied to me very well. I never thought this would take a really long time. But greater things did really happen in a blink of an eye specially that I am writing this now.

To all the people who supported me all through this, THANK YOU. I want to thank my family, without them this wouldn't have been possible. Especially to my mom for the never ending nagging me of finishing the project. Haha. And to my adviser, Mr. Bernie Terrado, thank you for all the things you taught me. No words can express how much thankful I am right now.