

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

UNIVERSITY OF THE PHILIPPINES-MANILA LEARNING  
RESOURCE CENTER TUTORIAL MANAGEMENT  
SYSTEM

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

Submitted by:

Maria Reena I. Quizon  
May 2016

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 “University of the Philippines-Manila Learning Resource Center Tutorial Management System” prepared and submitted by Maria Reena I. Quizon in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science has been examined and is recommended for acceptance.

---

**Avegail D. Carpio, M.Sc.**  
Adviser

### EXAMINERS:

	Approved	Disapproved
1. Gregorio B. Baes, Ph.D. ( <i>candidate</i> )	_____	_____
2. Richard Bryann L. Chua, M.Sc.	_____	_____
3. Perlita E. Gasmien, M.Sc. ( <i>candidate</i> )	_____	_____
4. Marvin John C. Ignacio, M.Sc. ( <i>cand.</i> )	_____	_____
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.

<hr/> <b>Ma. Sheila A. Magboo, M.Sc.</b> Unit Head Mathematical and Computing Sciences Unit Department of Physical Sciences and Mathematics	<hr/> <b>Marcelina B. Lirazan, Ph.D.</b> Chair Department of Physical Sciences and Mathematics
---	---

---

**Leonardo R. Estacio Jr., Ph.D.**  
Dean  
College of Arts and Sciences

## **Abstract**

The University of the Philippines-Manila offers tutoring services at its own Learning Resource Center (LRC). The tutoring service is a peer and/or faculty-assisted small group discussion offered free to all graduating students. Besides the tutorial services, there are also simulated exams to give students a preview of the department exams in Natural Sciences and Mathematics. Scheduling tutorial sessions, applying for simulated examinations, and applying to become student tutors has always been done manually at the LRC. Manual operations lead to inefficiency, misinformation due to human error, and reduced accesibility. To contribute to improving the quality of services offered by the LRC, this project aims to deliver an online information system that is capable of automating these services.

*Keywords:* information system, learning resource center, LRC, schedule, tutor

# Table of Contents

Acceptance Sheet	i
Abstract	ii
List of Figures	iv
List of Tables	xiv
<b>I. Introduction</b>	<b>1</b>
A. Background of the Study . . . . .	1
B. Statement of the Problem . . . . .	3
C. Objectives of the Study . . . . .	4
D. Significance of the Project . . . . .	7
E. Scope and Limitations . . . . .	8
F. Assumptions . . . . .	10
<b>II. Review of Related Literature</b>	<b>11</b>
<b>III. Theoretical Framework</b>	<b>15</b>
<b>IV. Design and Implementation</b>	<b>19</b>
<b>V. Results</b>	<b>55</b>
<b>VI. Discussions</b>	<b>100</b>
<b>VII. Conclusions</b>	<b>104</b>
<b>VIII. Recommendations</b>	<b>106</b>
<b>IX. Bibliography</b>	<b>108</b>
<b>X. Appendix</b>	<b>111</b>
<b>XI. Acknowledgement</b>	<b>192</b>

## List of Figures

1	USE CASE DIAGRAM, for students and student-tutors interacting with the UP Manila Learning Resource Center Tutorial Management System. . . . .	19
2	USE CASE DIAGRAM, for the administrator and faculty members interacting with the UP Manila Learning Resource Center Tutorial Management System. . . . .	20
3	ACTIVITY DIAGRAM, showing the login process for students and student-tutors in UPM-LRC-TMS. . . . .	21
4	ACTIVITY DIAGRAM, showing the login process for the administrator and faculty members in UPM-LRC-TMS . . . . .	22
5	ACTIVITY DIAGRAM, showing the process for resetting the password for all users in UPM-LRC-TMS. . . . .	22
6	ACTIVITY DIAGRAM, showing the application/renewal process for students and student-tutors in UPM-LRC-TMS. . . . .	23
7	ACTIVITY DIAGRAM, showing the endorsement process for faculty members in UPM-LRC-TMS. . . . .	24
8	ACTIVITY DIAGRAM, showing the approval/rejection of the application process for the administrator in UPM-LRC-TMS. . . . .	25
9	ACTIVITY DIAGRAM, showing the tutorial scheduling process for students and student-tutors in UPM-LRC-TMS. . . . .	26
10	ACTIVITY DIAGRAM, showing the tutorial schedule attendance checking process for the administrator in UPM-LRC-TMS. . . . .	27
11	ACTIVITY DIAGRAM, showing the feedback submission process for students and student-tutors in UPM-LRC-TMS. . . . .	28
12	ACTIVITY DIAGRAM, showing the simulated exam application process for student and student-tutors in UPM-LRC-TMS. . . . .	28

13	ACTIVITY DIAGRAM, showing the simulated exam event or the LRC announcement creation process for the administrator in UPM-LRC-TMS. . . . .	29
14	ACTIVITY DIAGRAM, showing the set/edit system academic calendar process for the administrator in UPM-LRC-TMS. . . . .	30
15	ACTIVITY DIAGRAM, showing the set/edit free time schedule process for the administrator and student-tutors in UPM-LRC-TMS. . . . .	30
16	ACTIVITY DIAGRAM, showing the lesson content creation process for the administrator in UPM-LRC-TMS. . . . .	31
17	ACTIVITY DIAGRAM, showing the lesson accessing process for the students and student-tutors in UPM-LRC-TMS. . . . .	31
18	ACTIVITY DIAGRAM, showing the report generation process for the administrator in UPM-LRC-TMS. . . . .	32
19	ACTIVITY DIAGRAM, showing the automatic account generation process for the administrator in UPM-LRC-TMS. . . . .	33
20	ACTIVITY DIAGRAM, showing the search process for all users in UPM-LRC-TMS. . . . .	34
21	ACTIVITY DIAGRAM, showing the block student process for the administrator in UPM-LRC-TMS. . . . .	35
22	ENTITY RELATIONSHIP DIAGRAM, University of the Philippines-Manila Learning Resource Center Tutorial Management System (UPM-LRC-TMS). . . . .	37
23	Model-View-Controller. . . . .	53
24	Home Page. The home page gives a brief preview of the system, and the contact details of the LRC. There are links provided for logging in, and for resetting the password. . . . .	55
25	Login. The login modal prompts the user to enter his/her User Id and Password. . . . .	55

26	Forgot Password. If ever the user forgets his/her password, a modal prompts the user to enter his/her User Id and the email address associated with the account. An email containing a temporary password will be sent. . . . .	56
27	Administrator - Dashboard. All notifications meant for the administrator are shown here. The administrator can also edit the schedule of the availability of the rooms in the LRC in this page. . .	57
28	Administrator - Accounts Overview. This page shows the entire list of generated accounts for the system. The administrator may also choose to generate a PDF containing the list of all accounts generated by the system. . . . .	57
29	Administrator - Tutors Overview. This page shows the list of student-tutors. The administrator may choose to revert all student-tutor accounts to student accounts. . . . .	58
30	Administrator - Blocked Accounts. This page shows the list of accounts blocked from using the system. . . . .	58
31	Administrator - Generate Accounts. This page is where the administrator can generate student accounts. He/she has the option to generate multiple accounts using a preformatted excel file, or just generate one student account. . . . .	59
32	Administrator - Pending Applications. All pending applications for those interested to become student-tutors are displayed here. The administrator may decide to approve or reject an application in this page. . . . .	59
33	Administrator - Approved Applications. All approved applications for those interested to become student-tutors are displayed here. . .	60
34	Administrator - Rejected Applications. All rejected applications for those interested to become student-tutors are displayed here. . . . .	60

35	Administrator - View Application. Details of a specific application are displayed in this modal. . . . .	61
36	Administrator - View Endorsement. Details of a specific endorsement are displayed in this modal. . . . .	61
37	Administrator - Pending Tutorial Sessions. All pending (unfinished) tutorial sessions are displayed here. The administrator may choose to cancel a tutorial session in this page. . . . .	62
38	Administrator - Finished Tutorial Sessions. All finished tutorial sessions are displayed here. . . . .	62
39	Administrator - Cancelled Tutorial Sessions. All cancelled tutorial sessions are displayed here. . . . .	63
40	Administrator - Calendar. This page shows all set calendar dates for the system. There are edit and delete options per entry. . . . .	63
41	Administrator - Create Events. This page is where the administrator can set the dates to be used by the system. Also, the administrator can create simulated exam events and LRC-related announcements. . . . .	64
42	Administrator - Lesson Index. This page shows all subjects where online lesson content may be available. . . . .	64
43	Administrator - View Lessons. This page shows all available online lessons for a specific subject. There are edit and delete options per lesson. . . . .	65
44	Administrator - Create Lesson. The administrator can upload a PDF for online lesson content here. . . . .	65
45	Administrator - Edit Lesson. The administrator can edit an existing online lesson here. . . . .	66
46	Administrator - Mock Exam - View Details. The administrator can view details for a simulated exam in this page. . . . .	66



47	Administrator - Mock Exam - View Student. The administrator can view the list of students who applied for a simulated exam in this page. . . . .	67
48	Administrator - Profile. This is the page where the administrator can view his/her own profile. . . . .	67
49	Administrator - Profile - Change Details. This is the page where the administrator can change his/her profile image and password. . . . .	68
50	Administrator - Report Overview. A summary regarding the number of tutorial sessions and simulated exams that were successfully concluded, and the total number of students served are shown in this page. . . . .	68
51	Administrator - Reports - Number of Students Served. This is where the administrator can generate reports regarding the number of students served within a given date range. . . . .	69
52	Administrator - Reports - Most Frequent Students. This is where the administrator can generate reports regarding the students who avail the LRC services the most, within a given date range. . . . .	69
53	Administrator - Reports - Most Frequent Subjects. This is where the administrator can generate reports regarding the subjects frequently sought assistance for, within a given date range. . . . .	70
54	Administrator - Reports - Generated Accounts Per Batch. This is where the administrator can generate PDF files containing the list of generated accounts per batch. . . . .	70
55	Administrator - Search. This is where the administrator can search for other users. . . . .	71
56	Administrator - Search Results. This is where the administrator views the search results. Clicking on the names of the users will redirect to their respective profile pages. . . . .	71

57	Administrator - View Feedback. The summarized feedback regarding a student-tutor is shown in this page. . . . .	72
58	Administrator - View Feedback. The summarized feedback regarding a student-tutor is shown in this page. This is the page shown instead when the feedback is not yet viewable. . . . .	73
59	Administrator - View Student Profile. This is the page containing the details of a student who is not blocked from using the system. The administrator may block the student in this page. . . . .	73
60	Administrator - View Student Profile. This is the page containing the details of a student who is blocked from using the system. The administrator may unblock the student in this page. . . . .	74
61	Administrator - View Student-Tutor Profile. This is the page containing the details of a student-tutor. The administrator may block or unblock the student in this page. . . . .	74
62	Administrator - View Student-Tutor Details. This is the page containing the subject preference, and the schedule of a student-tutor. . . . .	75
63	PDF - List of Generated Accounts. This is the list of all generated accounts. . . . .	76
64	PDF - List of Generated Accounts, Per Batch. This is the list of generated accounts, per batch. . . . .	76
65	PDF - Report, Number of Students Served. This contains the report regarding the number of students served by the LRC, within a given date range. . . . .	77
66	PDF - Report, Most Frequent Students. This contains the report regarding the students who avail the LRC services the most, within a given date range. . . . .	77
67	PDF - Report, Most Frequent Subjects. This contains the report regarding the subjects frequently sought assistance for, within a given date range. . . . .	78

68	Student - Change Email. If it is the first time the student uses the system, a modal prompts the user to input a valid email address, so that the system can be fully utilized. . . . .	79
69	Student - Change Password. After adding the email address, the student is prompted to change his/her temporary password. This is also for additional security. . . . .	79
70	Student - Dashboard. All notifications meant for a student are shown here. . . . .	80
71	Student - Dashboard - Appointments. All the unfinished tutorial sessions created by a student are shown here. A student may cancel the appointment. . . . .	80
72	Student - Application Reminder. This modal shows the requirements a student needs in order to accomplish the application form. Clicking the upper button lets the user proceed to the application form. On the other hand, clicking the lower button redirects the user back to the dashboard. . . . .	81
73	Student - Application - Step 1 of 3. This is the first page of the application form. This is where the student can input his/her date of birth, and the organizations he/she is currently a member of. . .	81
74	Student - Application - Step 2 of 3. This is the second page of the application form. This is where the student can rank the subjects he/she wants to teach. Also, this is where the student can upload his/her academic standing for all the semesters taken while in UP Manila. . . . .	82
75	Student - Application - Step 3 of 3. This is the third page of the application process. This is where the student can upload a profile image, input his/her free time schedule, and select which faculty member to request and endorsement from. . . . .	82

76	Student - Answer Feedback. If a student has recently attended a tutorial session, he/she is redirected to a feedback form upon logging in to the system. Answering the feedback form redirects the user back to the dashboard. . . . .	83
77	Student - Lesson Index. This page shows all subjects where online lesson content may be available. . . . .	84
78	Student - View Lessons. This page shows all available online lessons for a specific subject. . . . .	84
79	Student - Mock Exam - View Details. The student can view details and apply for a simulated exam in this page. If the student has already applied, he/she can cancel his/her slot instead. . . . .	85
80	Student - Mock Exam - View Students. The student can view the list of students who applied for a simulated exam in this page. . . .	85
81	Student - Profile. This is the page where the student can view his/her own profile. . . . .	86
82	Student - Profile - Change Details. This is the page where the student can change his/her profile image and password. . . . .	86
83	Student - Schedule Tutorial. This is where the student can schedule a tutorial session. At the left side of the screen, the user can select a subject; the list of available tutors who prefer to teach the subject, and their schedules, will appear. . . . .	87
84	Student - Search. This is where the student can search for other users. . . . .	87
85	Student - Search Results. This is where the student views the search results. Clicking on the names of the users will redirect to their respective profile pages. . . . .	88
86	Student - View Administrator Profile. This is where the student can view the administrator's profile. . . . .	88

87	Student - View Student Profile. This is the page containing the details of a student. . . . .	89
88	Student - View Student-Tutor Profile. This is the page containing the details of a student-tutor. . . . .	89
89	Student - View Student-Tutor Details. This is the page containing the subject preference, and the schedule of a student-tutor. . . . .	90
90	Student-Tutor - Dashboard. All notifications meant for a student-tutor are shown here. . . . .	91
91	Student-Tutor - Lesson Index. This page shows all subjects where online lesson content may be available. . . . .	91
92	Student-Tutor - View Lessons. This page shows all available online lessons for a specific subject. . . . .	92
93	Student-Tutor - Mock Exam - View Details. The student-tutor can view details and apply for a simulated exam in this page. If the student-tutor has already applied, he/she can cancel his/her slot instead. . . . .	92
94	Student-Tutor - Mock Exam - View Students. The student-tutor can view the list of students who applied for a simulated exam in this page. . . . .	93
95	Student-Tutor - Profile. This is the page where the student-tutor can view his/her own profile. . . . .	93
96	Student-Tutor - Profile - Change Details. This is the page where the student-tutor can change his/her profile image and password. . . . .	94
97	Student-Tutor - Schedule Tutorial. This is where the student-tutor can schedule a tutorial session. At the left side of the screen, the user can select a subject; the list of available tutors who prefer to teach the subject, and their schedules, will appear. . . . .	94
98	Student-Tutor - Search. This is where the student-tutor can search for other users. . . . .	95

99	Student-Tutor - Search Results. This is where the student-tutor views the search results. Clicking on the names of the users will redirect to their respective profile pages. . . . .	95
100	Student-Tutor - View Administrator Profile. This is where the student-tutor can view the administrator's profile. . . . .	96
101	Student-Tutor - View Student Profile. This is the page containing the details of a student. . . . .	96
102	Student-Tutor - View Student-Tutor Profile. This is the page containing the details of a student-tutor. . . . .	97
103	Student-Tutor - View Student-Tutor Details. This is the page containing the subject preference, and the schedule of a student-tutor. . . . .	97
104	Faculty - Accept Endorsement. This is where the faculty member is redirected to upon clicking the link in the email regarding the endorsement request. This is also where he/she can decide to accept or reject the endorsement request of a student. . . . .	98
105	Faculty - Reject Endorsement. This is the page where the faculty member is redirected to upon rejecting the endorsement request of a student. The student is notified upon rejection of the endorsement. . . . .	98
106	Faculty - Answer Endorsement. This is the page where the faculty member rates the student through the endorsement form, if the faculty accepts the request. . . . .	99
107	Faculty - Finish Endorsement. This is the page where the faculty member is redirected to upon finishing the endorsement of a student. The student is notified upon completion of the endorsement. . . . .	99

## List of Tables

1	APPLICATION has information regarding a student interested in becoming a student-tutor. . . . .	38
2	APPOINTMENT contains information regarding a scheduled tutorial session. . . . .	40
3	CALENDAR contains information regarding the days set on the system calendar. . . . .	41
4	ENDORSEMENT contains information regarding the endorsement of a faculty member for a student applicant . . . . .	42
5	FEEDBACK contains information regarding a submitted feedback form. . . . .	44
6	LESSON contains information regarding a lesson in the online tutorial. . . . .	45
7	MOCKEXAM contains information regarding a simulated exam event. . . . .	46
8	MOCKEXAM_STUDENTS contains the information regarding which students will attend the simulated exam. . . . .	47
9	NOTIFICATIONS contains the information regarding a notification. . . . .	47
10	SCHEDULE contains the schedule of a student. . . . .	48
11	USERS contains all accounts in the system. . . . .	51

# **I. Introduction**

## **A. Background of the Study**

A tutor is someone who gives additional, special, or basic instruction. A tutee is a student of the tutor. Tutoring is an act which facilitates another person's learning process. Its purpose is to help students help themselves, so that they may eventually become independent learners. Students seek tutors when they need additional help in a subject, or when they want to reinforce their learning. A session between student and tutor can occur anywhere, as long as the environment is conducive for learning. Tutors are effective, since they are usually able to combine their knowledge with empathy, honesty, and humor. Thus, they can be appealing to students [1].

Other students volunteer to be tutors, since teaching increases their understanding of the basic skills in a subject, thus they experience higher levels of thinking. Besides improved understanding of the subject matter, the students they tutor, or tutees, are also able to improve their attitudes towards the subject, and enhance self-paced and self-assisted learning. The increased amount of practice during a tutoring session ultimately improves their academic performance and personal growth. Tutoring also has some benefits for the entirety of the college. Better performance among students will increase retention and persistence, increase rate of completion and success, and improve educational climate [1].

Online tutorials are sometimes used in place of human tutors. Made available in the Internet, online tutorials are self-study activities designed to teach a specific learning outcome. There are two main types of online tutorials: recorded and interactive. Recorded tutorials are videos or screencast recordings. They usually present a subject expert imparting information or a demonstration. On the other hand, interactive tutorials are structured collections of navigable web pages. They



may contain any collection of texts, images, videos, audio, or self-test questions. They may also contain screencasts. Online tutorials can be for supplementary learning or for an integral part of a core activity [2]. They are also helpful in ensuring learners have a baseline knowledge of the topic. Moreover, they can present the factual aspects of the subject, freeing up time for face-to-face interaction when it comes to the more complex topics to be discussed. The most important advantage of online tutorials is distance learning. Users can access content anywhere and they can study it at their own pace [3].

The University of the Philippines-Manila offers tutoring services at the Learning Resource Center (LRC). The LRC is tasked to supplement, complement, and coordinate existing learning assistance programs [4]. Its goal is to ensure that academic excellence will be realized in a more caring and nurturing environment [5], especially through fellow students. The tutoring service is a peer and/or faculty-assisted small group discussion offered free to all undergraduate students. The courses they offer are from Chemistry, Biology, Physics, Mathematics, Computer Science, and Communications. There are also simulated exams designed to give students a preview of the department exams in Natural Sciences and Mathematics. A discussion of answers follows the simulated exams, which is handled by a student tutor or a faculty volunteer [6].

Applying for the tutoring services and simulated exams is done manually at the LRC, as well as the application for students who want to volunteer as tutors. Moreover, all tutorials are done face-to-face. Several scheduling systems have been developed before to facilitate appointments in various fields. There are also numerous online tutorials for various subject areas. Such systems are considered to be implemented for the LRC to increase efficiency and accessibility of the tutoring services for the students.

## B. Statement of the Problem

Problems arise from the traditional method of scheduling. First, it requires manpower, which may be student-assistants or LRC staff. Moreover, those who process the tutorial schedule can only work within office hours. The fact that scheduling for a tutorial can be only done in the LRC makes it even more inaccessible. Manual scheduling is inefficient and time-consuming, since it may take a while before student-assistants or the LRC administrator finishes checking the schedules of the tutors and the list of available rooms. Human error also leads to inconsistency and misinformation, especially when notifying the student of the tutorial schedule. There are times when the tutees leave immediately after the tutorial session, which causes the student-assistants to forget that the tutees still have feedback forms to submit. Furthermore, the feedback forms are tallied and summarized traditionally, which is also prone to human error [7].

Manual application also consumes a lot of time, especially when requesting for an endorsement from a faculty member. The faculty member may be busy and may not address the impending endorsement request. This may cause delay for the student. Moreover, the processing of the information may take quite some time too. There may be inconsistency when processing information.

There are also instances when there are no available tutors and the student is in dire need of supplementary learning. Since all of the tutorial sessions are done face-to-face, the student must venture into other websites for additional help in a subject. Sometimes, the content offered in other tutorial sites might not match the level of difficulty given in the lessons from the University. These might not improve the level of learning the student has at all. Besides the availability of the content, the moderator of the current LRC website cannot add an online tutorial module. The currently available learning module in the website is an online simulated examination.

Automation is seen as a solution to these problems. Such a technology utilizes the internet for quicker notification and decision-making for the scheduling of the appointment. It also aims to become more intuitive and user-friendly. This is to ensure that there will be more successful tutorial sessions, and to encourage more students to avail of the tutorial program.

## C. Objectives of the Study

The study aims to develop a system with the following functionalities:

### Web/PC Access

The proposed system should provide a user-friendly interface with HTML5 and PHP: Hypertext Preprocessor as its front end. My Structured Query Language (MySQL) is utilized for back-end processing.

#### 1. General Objectives

- (a) Automatically notify a student if his or her scheduled tutorial request is accepted.
- (b) Automatically notify the administrator whether a student is interested in applying or renewing to become a student-tutor.
- (c) Automatically deactivate student-tutor accounts at the end of every semester.
- (d) Email a faculty member regarding a student's endorsement request.
- (e) Email a randomly generated password if ever a user forgets his/her password.
- (f) Reset contents of the SCHEDULE table for all users every semester.
- (g) Clear contents of the CALENDAR table every academic year.

2. Specific Objectives The following functionalities are sorted according to the types of users:

(a) Student

- i. Login to the system.
- ii. Change profile image and password.
- iii. Schedule a tutorial session.
- iv. Reschedule a tutorial session, if ever the session was cancelled.
- v. Submit feedback regarding a tutor after a tutorial session.
- vi. Apply for simulated examinations and after-exam discussions.
- vii. Apply to become a student-tutor.
- viii. Cancel student-tutor application.
- ix. Use online tutorial module.
- x. Input or update own free time schedule.
- xi. Access profiles of fellow students through the search function.

(b) Student-tutor

- i. Login to the system.
- ii. Change profile image and password.
- iii. Schedule a tutorial session.
- iv. Reschedule a tutorial session, if ever the session was cancelled.
- v. Accept or reject tutorial requests.
- vi. Submit feedback regarding a tutor after a tutorial session.
- vii. View summarized version of feedback every semester.
- viii. Apply for simulated examinations and after-exam discussions.
- ix. Renew to become a student-tutor.
- x. Cancel student-tutor application.
- xi. Use online tutorial module.

- xii. Input or update own free time schedule.
- xiii. Access profiles of fellow students through the search function.

(c) Administrator

- i. Login to the system.
- ii. Change profile image and password.
- iii. View dashboard of scheduled appointments, and if they are accepted or rejected by the tutors.
- iv. Cancel tutorial sessions.
- v. Warn, block or unblock absent tutees from using the system.
- vi. View the list of students blocked from using the system, and the number of times a student was blocked.
- vii. Upload announcements regarding LRC-related events, or schedules of simulated exams or after-exam discussions.
- viii. View list of students who applied for a simulated examination and an after-exam discussion.
- ix. View application form of a student who wants to become a tutor/renews volunteer tutor membership.
- x. View endorsement form submitted by a faculty member.
- xi. Approve application or renewal of a student to become a student-tutor.
- xii. View feedback of all student-tutors by the end of the semester.
- xiii. Upload PDF for a specific lesson in a subject, using the online tutorial module.
- xiv. Upload a Microsoft Office Excel file containing list of freshmen for multiple account generation.
- xv. Create one student-user account, instead of generating multiple accounts at a given time.

- xvi. View and generate PDF file of created accounts for the freshmen for printing.
- xvii. View and generate PDF file for all created accounts, excluding the administrator.
- xviii. Generate reports regarding the number of students served by the LRC per month and per semester.
- xix. Generate reports regarding which student frequents the LRC tutorial services the most.
- xx. Generate reports regarding which subject students frequently seek assistance for.
- xxi. Input or update set dates in the systems academic calendar.
- xxii. Input or update schedule of available rooms.
- xxiii. Access profiles of fellow students through the search function.

## **D. Significance of the Project**

The University of the Philippines-Manila Learning Resource Center Tutorial Management system is intended to provide an application to supply the mechanism necessary to schedule a tutorial, apply for simulated exams and after-exam discussions, apply to become a volunteer tutor, and access tutorial lessons online. It aims to make these procedures more efficient for the students of the University of the Philippines-Manila.

The system promotes accessibility; students can schedule a tutorial anywhere with internet access. Misinformation due to human error can be greatly reduced, since the system handles the most of the decision-making and notifications instead. The feedback given to tutors will be computerized and accurately computed.

These benefits reduce the time and the amount of work done by the LRC

administrator. Moreover, the system may encourage more students to seek additional help from their fellow students to supplement their learning. In the case where there is no available tutor, the students can access the online learning modules for self-paced and distance learning. The online learning module will provide standardization of the presentation of the tutorial content.

The application to become a tutor or to renew membership will become faster and more accessible. Faculty members who are sought endorsement from will be immediately notified, especially when they are not physically present in the campus premises. The accomplished endorsement form will remain confidential between him and the administrator, since it will be sent directly to the latter.

The administrator will be able to view and generate reports immediately. There will also be summary statistics regarding students and the most requested subjects for a tutorial, the number of students served by the LRC every month and every semester, and the individuals who frequently request the services offered by the LRC. The statistics will be displayed in graphs, which will be useful as visual aids for easier interpretation in the reports. It is also a way to gauge which subjects need more tutors.

## **E. Scope and Limitations**

1. The system will not decide which faculty member the student should request an endorsement from.
2. Tutorials, simulated exams, and after-exam discussions outside the Learning Resource Center is not covered by the system.
3. It is up to the LRC administrator to decide which room should be assigned to the students during tutorial sessions. Available rooms during a time frame will not be indicated in the system.

4. The proposed system will cover schedules within the academic calendar of the current academic year.
5. Feedback forms will not be filtered, even if obscenities, vulgarities, or profanities arise. Reporting such events is up to the student.
6. The system is not liable if there is no full coverage of the topics in the online tutorial module.
7. An email will be sent as soon as a student is blocked from using the system.
8. A notification will be sent as soon as a student is approved or rejected to become a volunteer tutor.
9. All reports generated will be restricted to PDF format.
10. The system is not intended to replace face-to-face tutorial sessions, simulated exams, or after-exam discussions.



## **F. Assumptions**

1. The LRC staff members are in charge of the system, the LRC director being the head. They are considered experts in the services offered by the LRC.
2. Students may either be tutor or tutees.
3. If there is a group of students who want to schedule a tutorial session, only one student among them should create the request. That student will be the one responsible for answering the feedback form.
4. Once a faculty member accepts the endorsement request, it is tantamount to his or her signature.
5. Tutors must be officially enrolled for the year.
6. The subjects offered in the tutorial are from Chemistry, Physics, Biology, Mathematics, Computer Science, or Communications.

## II. Review of Related Literature

Shafie et al. (2011) developed a system that would help solve the traditional method of registering student attendance in De Montfort University in Leicester, United Kingdom. The traditional method of registering student attendance yields a high chance of misinformation due to human error. Moreover, there are no records to history of previous attendances, such that these data may be used for statistical analyses. Tabulation of data involves rewriting, which entails additional work. The system allows to register and update student attendance, show time tables for students and instructors, produce statistics with respect to student attendance, and manage system logins [8].

In order to improve the efficiency and quality of the course registration system at Tsinghua University in Beijing, China, an online version of the system was implemented by Peng et al (2012). Since the university offers over three thousand curriculums every year, it caters to over half a million students. The online course registration is an integral part to the university's educational system. The system aims to meet registration requirements, as well as promote fairness for all students. The registration mechanism has three phases: the registration phase, the adjustment phase, and the dropping phase. The registration phase presents all available courses according to the students' current training plan; the adjustment phase allow students to register first-come-first-serve in their chosen courses; the dropping phase is meant for students who want to drop courses. Students now enrolled can manage their performance and arrange exams in the system. Faculty members, on the other hand, can construct their teaching and training plans for a course [9].

The Egyptian urban setting is under a state of deterioration, as stated by Wahed (2015). The author pinpoints that one of the causes of deterioration is the lack of system in issuing building permits. The study outlines the requirements of

developing a building permit issuing system. It is believed that this system would greatly improve staff accountability and efficiency, reduce customer waiting time, and improving business practices and processes. The system outputs a receipt that indicates that the application is received, the statement of validity for the site construction, and a permit file. The permit file can be approved or rejected by authorities [10].

Schedule management of hospitalization is integral for the improvement of health care. According to Iwata et al. (2013), the application of a clinical pathway is one of the important solutions to clinical management. A clinical pathway is a style of schedule management. It is important since several clinical actions should be repeated appropriately for treating diseases. Data warehousing is used to collect and prepare raw data, since data in hospital information systems are histories of clinical actions. The stored data are then transformed into a clinical pathway for doctors and nurses [11].

Another schedule management application was developed using Google App Engine (GAE) and Go, as procedural language released by Google back in 2009. Togashi and Kyulev (2015) created a novel application such that the scheduling management system is similar to Moodle. Each schedule is displayed in a blog-like fashion. The main requirement, according to the authors, is the simplicity of the application, since they view the potential users as people who are not very familiar with computers and information technology [12].

According to Cavus (2014), it is necessary to use technology in the education sector, reorganize it at the very least. There has been changes in the needs of students and educators for the past years. Teachers need to prepare their lectures in the shortest possible time, as students are encouraged to do more self-studying. The author claims that learning management systems such as Moodle are the most convenient to implement. Moreover, it is more cost and time-effective. Learning Management Systems (LMS) have numerous features, such as online tutorials, reg-

istration, and scheduling that would greatly convenience instructors and students alike [13].

Simplifying learning is the main objective in mind when Kozusko et al. (2012) implemented an e-learning program in the study "Enhancing Laboratory Exercises for Biomedical Engineering Courses". Besides the belief the teaching is more effective when combined with e-learning, the program also aims to accommodate the increased amount of college students due to the removal of 13th year in Germany's educational system. The system has three modules, namely, the basic matter layer, the extended learning layer, and the user management system. The basic matter layer presents the tutorial content, including exercises. The extended learning layer presents that additional content needed to supplement the students' learning. On the other hand, the user management system takes into account the learning pace and the evaluation of a student [14].

Koncz et al. (2012) described in their paper their intention to create a solution to support educational processes at the Technical University of Kosice, Slovakia. Their intent led them to create a publicly available e-learning and knowledge base, in the form of a course web site. The web site aims to be collaborative and exploratory among users. Students have unlimited access to the content of the course online, which is advantageous when they are physically dispersed from teachers and classmates. Moreover, the students have better control of their learning process. Joomla, a content management system, and Moodle, a learning management system, were integrated into the application. The web site can manage users, manage student evaluations, and manage assignments and attendance [15].

Unlike majority of the online tutorial websites where only one or several users can manage learning content, Llorens, Nevin, and Mageean (2014) created a site that allows students to create their own content for their fellow students. The project is primarily a platform for mathematics education, which uses the active learning approach. This approach shifts lecturer delivery to active engagement of

the material by the student. The platform consists of a set of quizzes and applications for the study of mathematics; quizzes will adapt and change depending on the students' answers. Appropriate video feedback will be shown in case of incorrect answers. These videos are created by the students themselves, in order to help fellow students. Graphics tablets are used to ease mathematical input [16].

The ViLLE learning environment was used as a collaborative learning environment in the University of Turku in Finland. Karavirta et al. (2015) aims that this system increases student retention in Computer Science courses. ViLLE, besides being collaborative, supports many kinds of exercises, collects interactive data, and collects learner results and progress. The environment supports three kinds of exercises, namely, General Exercises, Programming Exercises, and Mathematical Exercises. General Exercises can be used for any topic; Program Exercises are used when programming is involved; Mathematical Exercises aim to improve the understanding of the topics covered in mathematics. The most important aspect of the exercises is automatic assessment and feedback, which helps students improve their own learning processes. Teachers are also benefited since they do not have to grade all the students' submissions [17].

### III. Theoretical Framework

#### 1. University of the Philippines-Manila Learning Resource Center

Established in 1997, the Learning Resource Center (LRC) of the University of the Philippines-Manila is tasked to supplement, complement, and coordinate existing learning assistance programs. Its specific goal is to upgrade the capability of the student to meet the demands of the University's education. The subjects it focuses on are Communications, Mathematics, and Science [4]. Moreover, it aims to provide alternative avenues for a student's personal growth and psychosocial development through specific activities and programs [5].

The services offered by the LRC are as follows:

##### (a) Tutorials

Offered free to all undergraduate students, the tutorial service is a small group discussion assisted by either a peer tutor or a volunteer faculty member.

##### (b) Simulated examinations

A simulated exam gives a preview of the departmental exams in the natural sciences and mathematics. A discussion of answers will follow the examination, which is handled by either a peer tutor or a volunteer faculty member.

##### (c) Reviewers and books

Reviewers and booklets contain problems and solutions to various subjects under the Department of Physical Sciences and Mathematics. These learning materials are products of the joint effort between student volunteers and faculty members. The materials are offered for a minimal fee.

##### (d) Computer Services

The use of computers in the LRC are for free. There are also printing services offered.

(e) Bridging Program

The Bridging Program is offered for qualified incoming first year students. It is a skills development and enrichment course to bridge the gap between high school exit skills and entry skills for becoming a freshman in UP.

(f) Psychosocial development program

The psychosocial development program is a host of seminars aimed at helping students cope with the challenges of UP life. The activities are primarily organized by the Office of Student Affairs.

(g) Monitoring and evaluation of student performance

The LRC gathers data regarding the students' performance in general education courses [6].

Scheduling a tutorial goes through the following steps:

1. The student fills out and submits a form containing some personal information, the subject, and the date and time preferred for the session. Scheduling is done two days before the appointment.
2. The LRC personnel sends an SMS to a tutor to see if the requested session is accordance with his or her schedule.
3. If the tutor agrees with the schedule, the personnel sends an SMS to the tutee, which contains the details of the appointment.
4. If the tutor is unavailable, the personnel checks with other tutors until one agrees.
5. If there are no tutors available, the tutee is told to reschedule the appointment.
6. The tutor and tutee attend the session.

7. After the tutorial, the personnel gives the tutee a feedback form to rate the tutor.

Applying or renewing to become a student tutor goes through the following steps:

1. The interested student fills out and submits a form containing some personal information, the subjects preferred to teach, and the free time schedule.
2. The student also gives an endorsement form to a preferred faculty member. However, if the student has been a volunteer tutor for more than two semesters, he or she may opt out of requesting an endorsement.
3. The faculty member answers the endorsement form and seals it in an envelope, to be delivered to the LRC personnel by the student.
4. The student submits both the application form and the endorsement form to the LRC personnel. The latter views both forms.
5. Based on the endorsement of the faculty member, the personnel approves or disapproves the application. The application is valid for a semester.

## 2. Information System (IS)

An information system is defined as a software that helps organize and analyze data. Its purpose is to turn raw data into useful information that can be utilized for decision-making [18]. Business firms and other organizations rely on information systems to carry out and manage their operations, and interact with others. Moreover, these systems have quickened the pace and maintained the consistency of such operations. Information and knowledge have become vital economic systems because of information systems [19].

## 3. Database Management System (DBMS)



A database management system (DBMS) is a software for creating and managing databases, providing a systemic way for users to create, retrieve, update, and manage data. The DBMS also serves as an interface between end-users or applications. Its advantages include concurrency, security, and data integrity. The DBMS is secure in such a way that it can limit what data the end-user sees, and protect them from knowing where the data is stored [20]. A DBMS consists of three elements:

- (a) The physical database, which is the collection of files that contain the data.
- (b) The database engine, which is the software that makes it possible to access the database.
- (c) The database scheme, which is the logical structure of the data stored [21].

#### 4. Model-View-Controller (MVC)

Model-View-Controller is a methodology or design pattern for successfully and efficiently relating the user interface to underlying data models. It is heralded as useful for the reuse of object code, since it allows to significantly reduce the time it takes to develop applications with user interfaces. Its three main components are:

- (a) The model, which represents the underlying logical data in a software application. It does not contain any information from and for the user interface.
- (b) The view, which is a collection of classes that contain elements for the user interface.
- (c) The controller, which represents the classes connecting the model and the view [22].

# IV. Design and Implementation

## 1. Use Case Diagrams

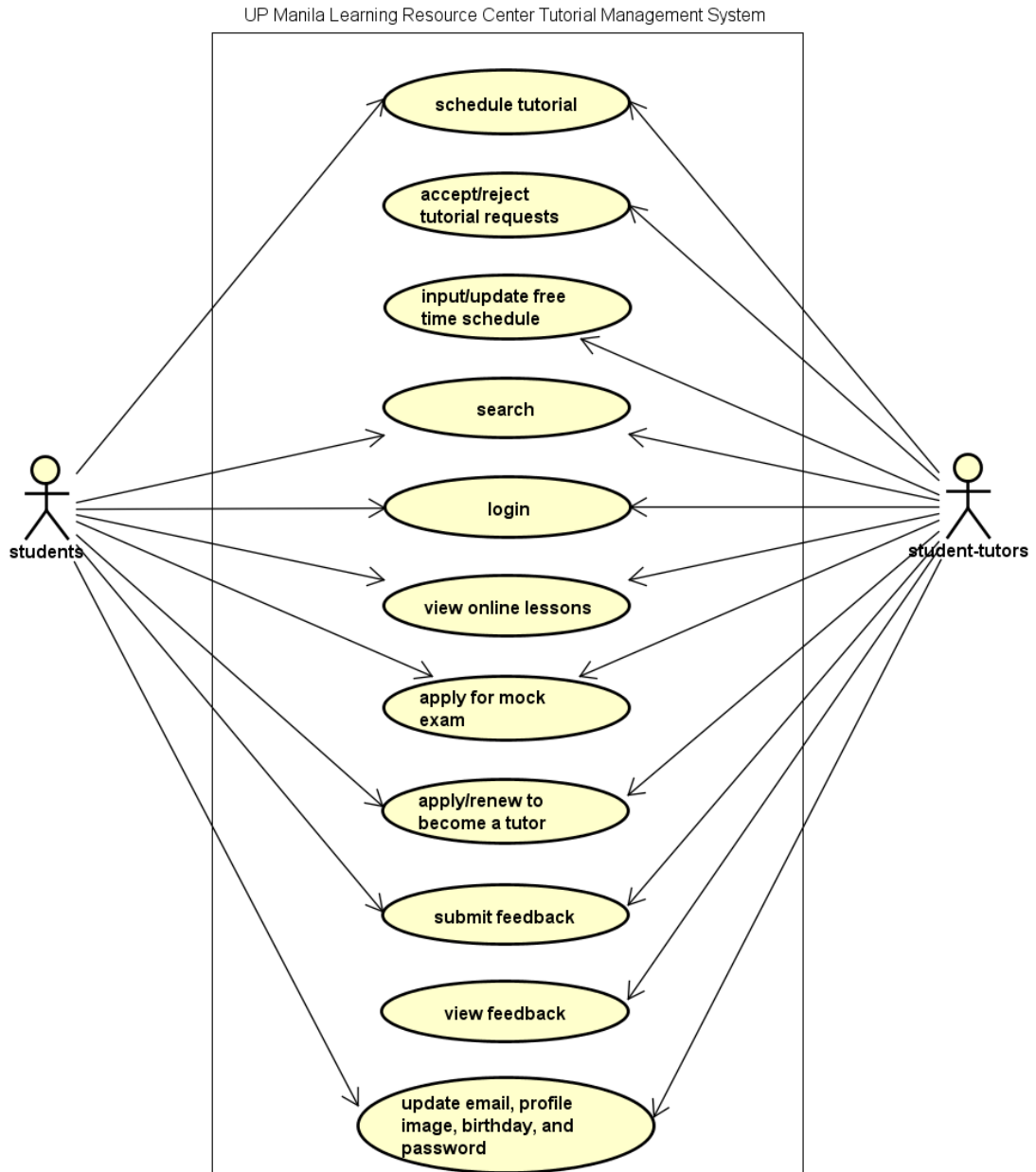


Figure 1: USE CASE DIAGRAM, for students and student-tutors interacting with the UP Manila Learning Resource Center Tutorial Management System.

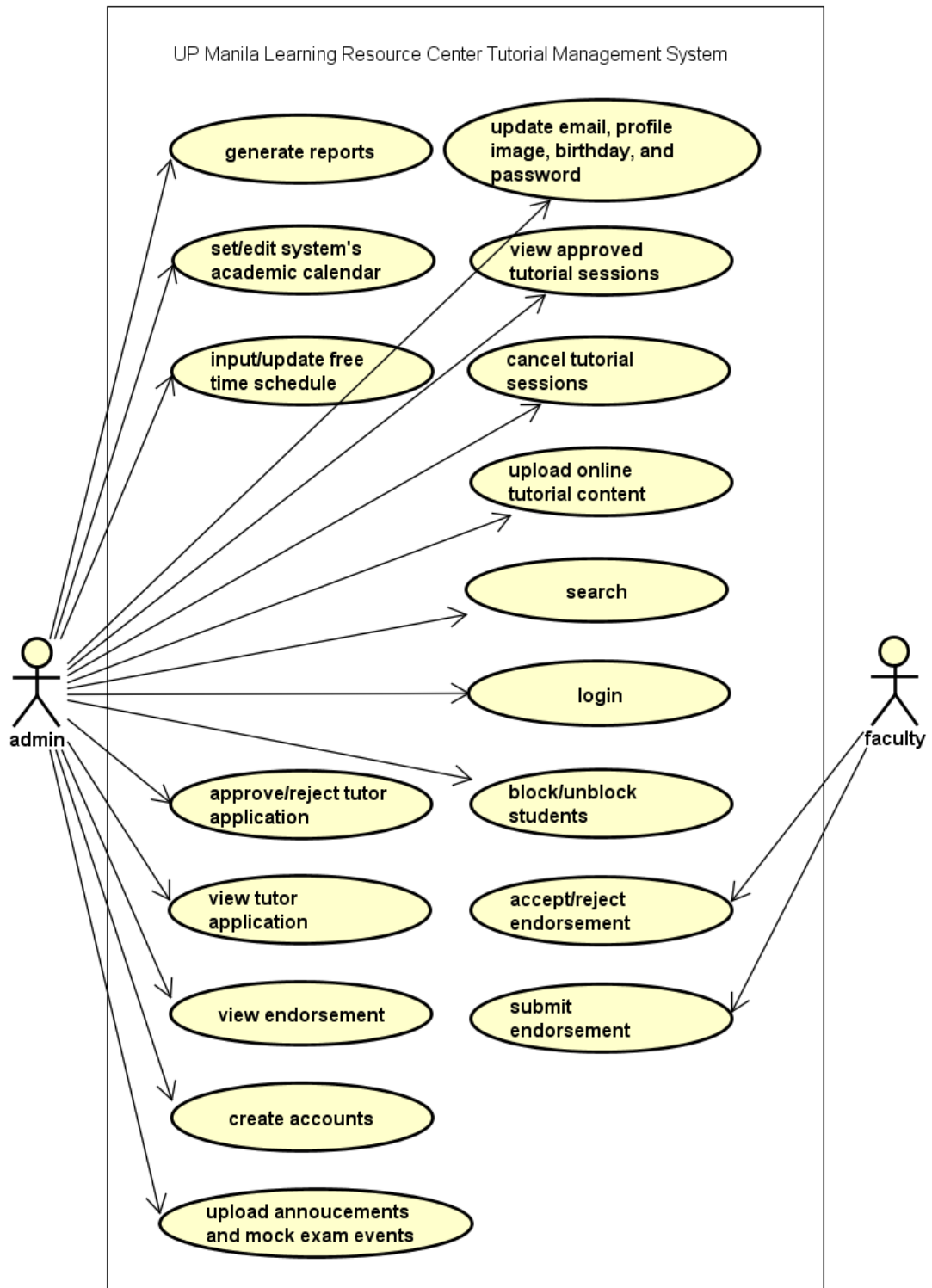


Figure 2: USE CASE DIAGRAM, for the administrator and faculty members interacting with the UP Manila Learning Resource Center Tutorial Management System.

## 2. Activity Diagrams

The activity diagrams showing the various functionalities in the system are in Figures 3-21. They also show the sequences of events when the users interact with the University of the Philippines-Manila Learning Resource Center Tutorial Management System (UPM-LRC-TMS).

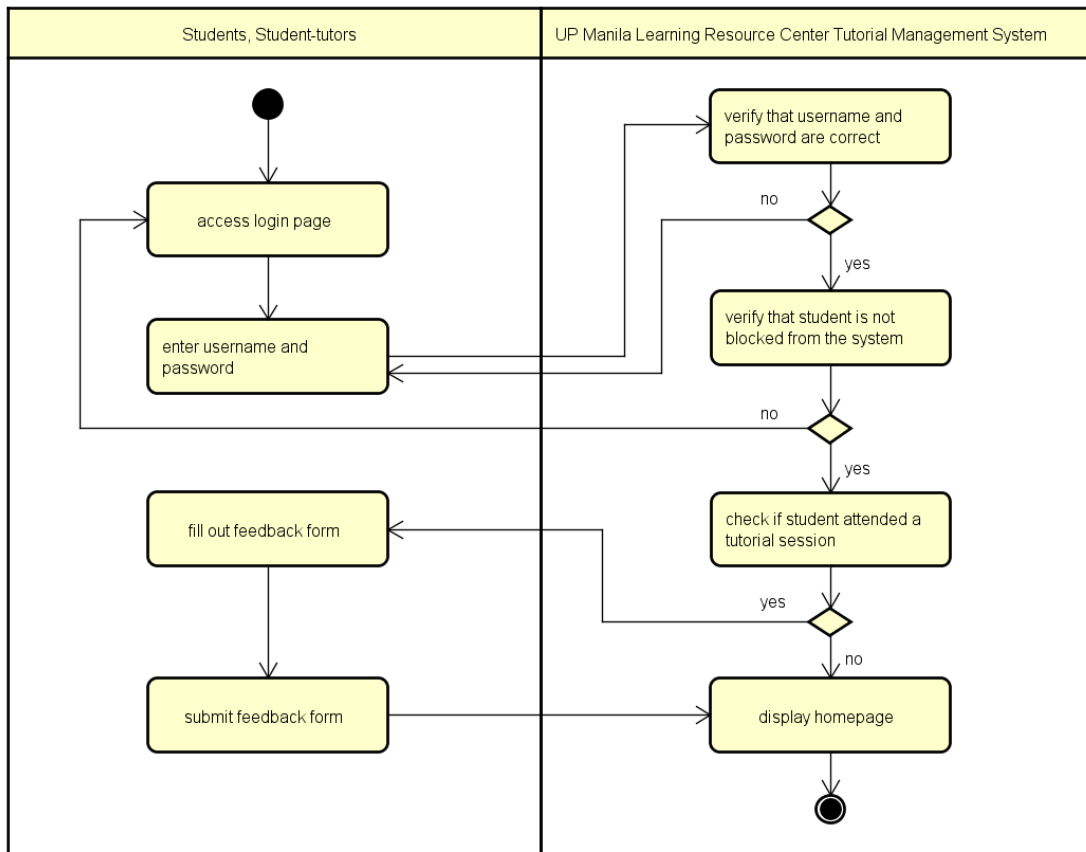


Figure 3: ACTIVITY DIAGRAM, showing the login process for students and student-tutors in UPM-LRC-TMS.

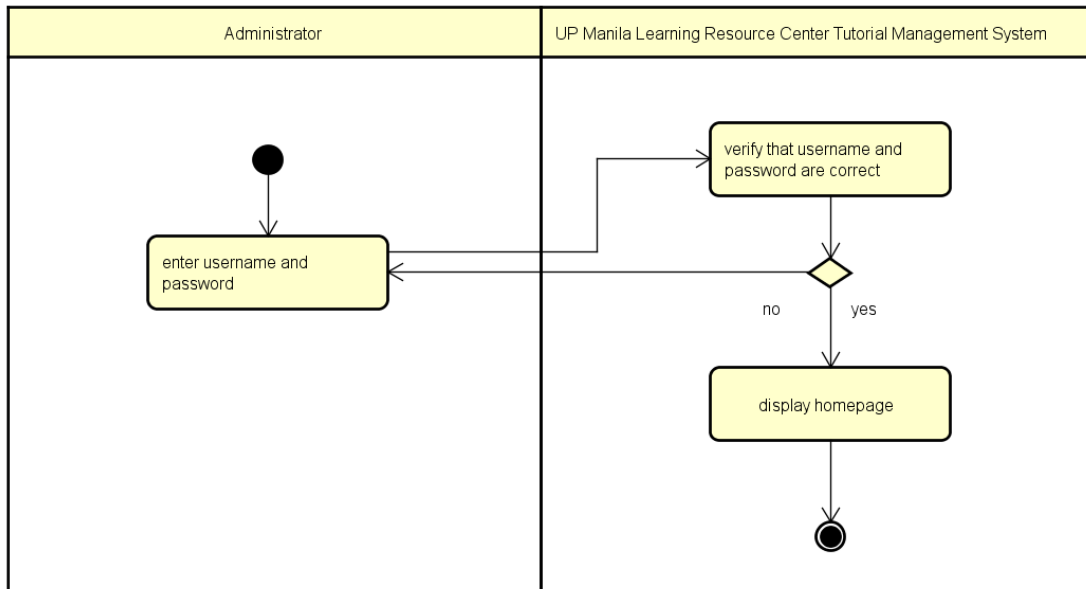


Figure 4: ACTIVITY DIAGRAM, showing the login process for the administrator and faculty members in UPM-LRC-TMS

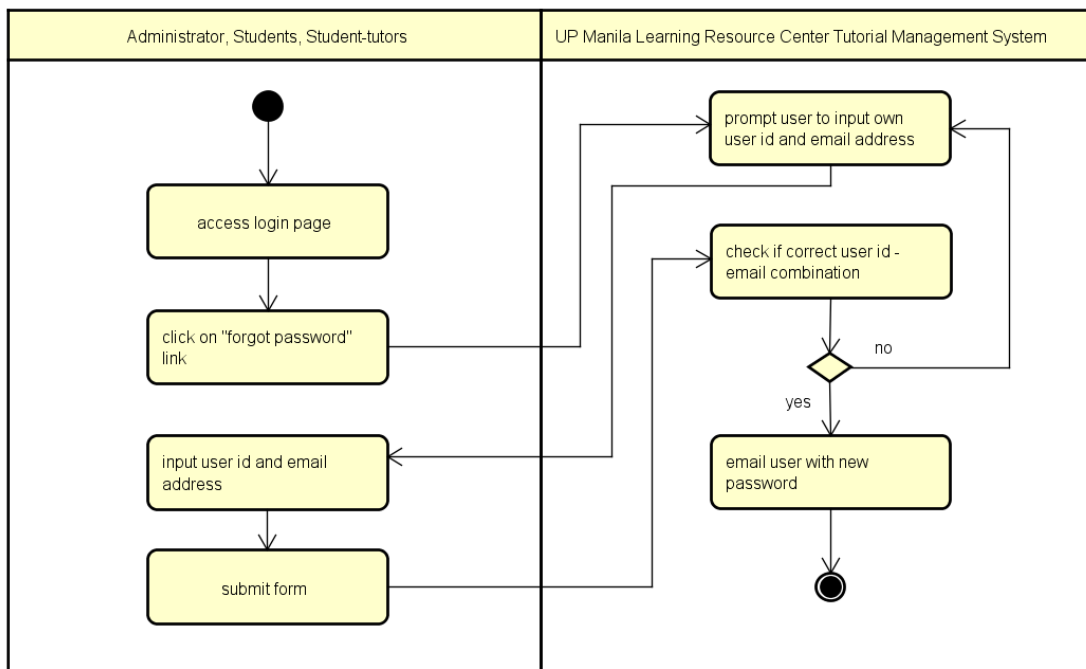


Figure 5: ACTIVITY DIAGRAM, showing the process for resetting the password for all users in UPM-LRC-TMS.

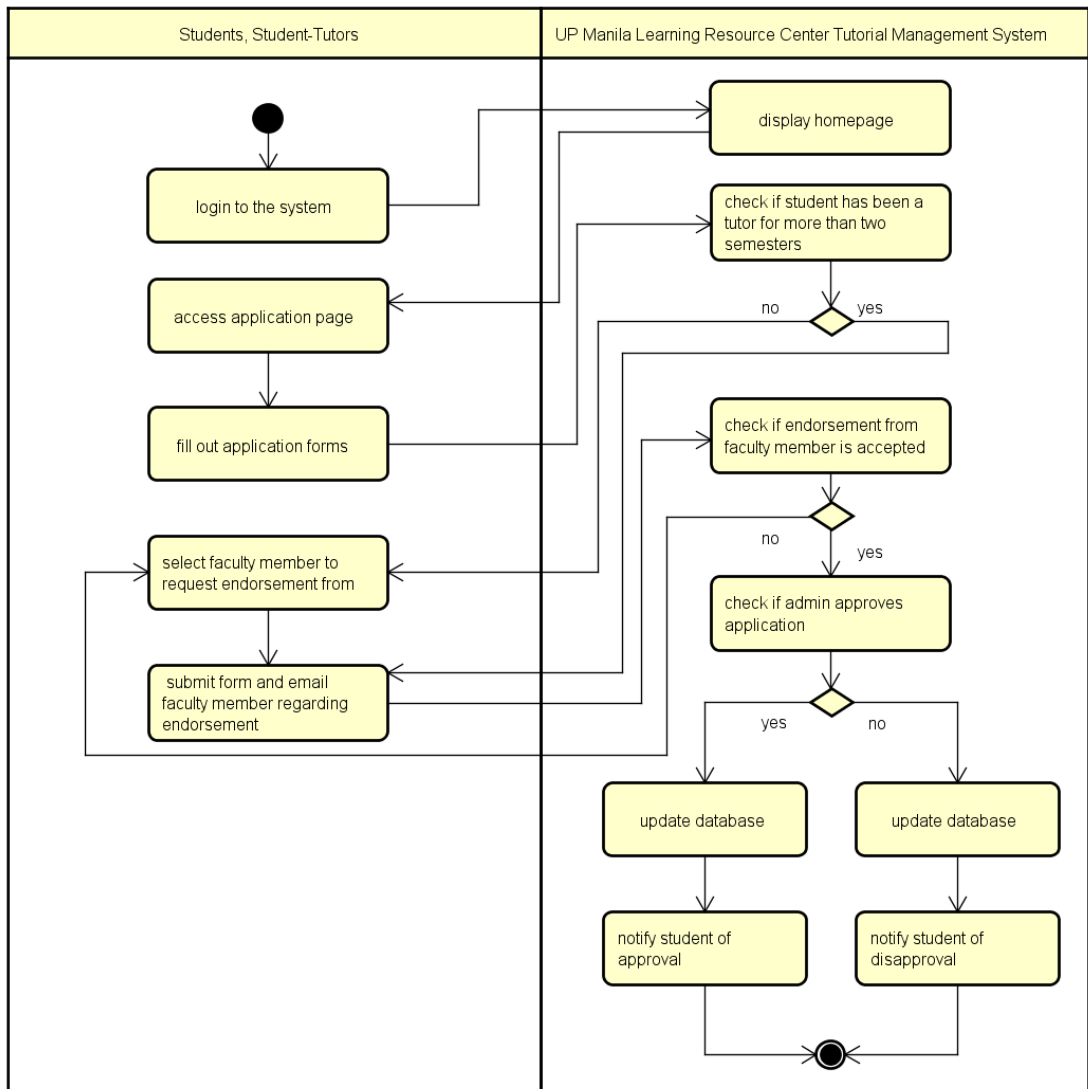


Figure 6: ACTIVITY DIAGRAM, showing the application/renewal process for students and student-tutors in UPM-LRC-TMS.

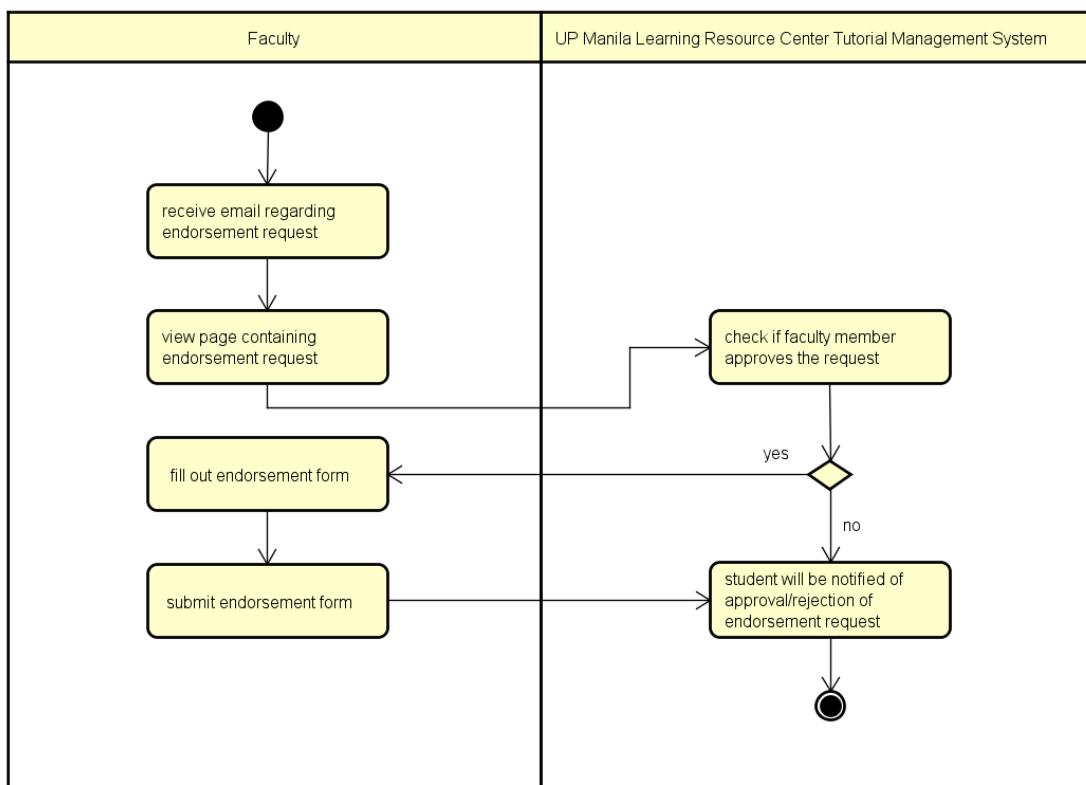


Figure 7: ACTIVITY DIAGRAM, showing the endorsement process for faculty members in UPM-LRC-TMS.

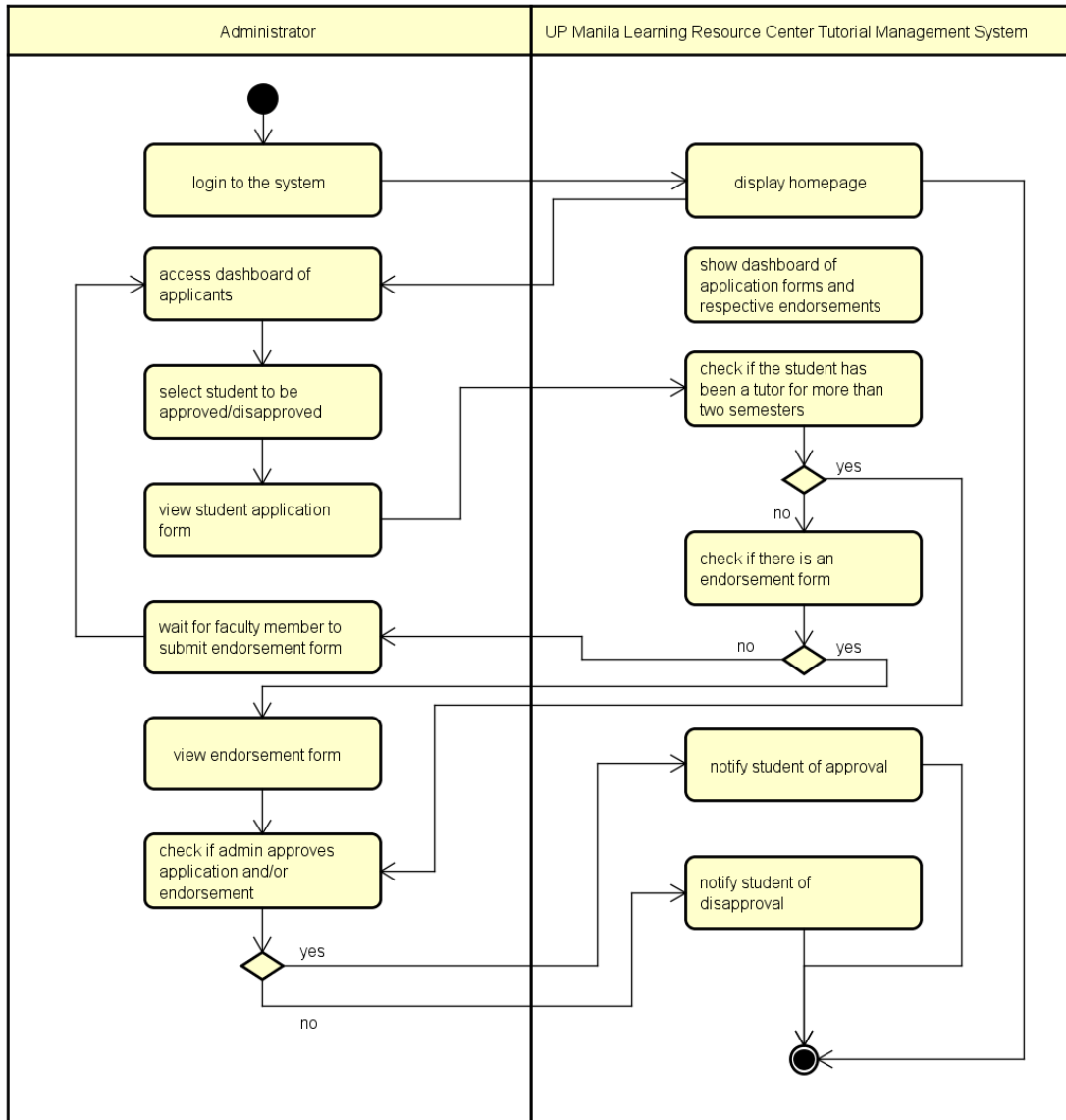


Figure 8: ACTIVITY DIAGRAM, showing the approval/rejection of the application process for the administrator in UPM-LRC-TMS.



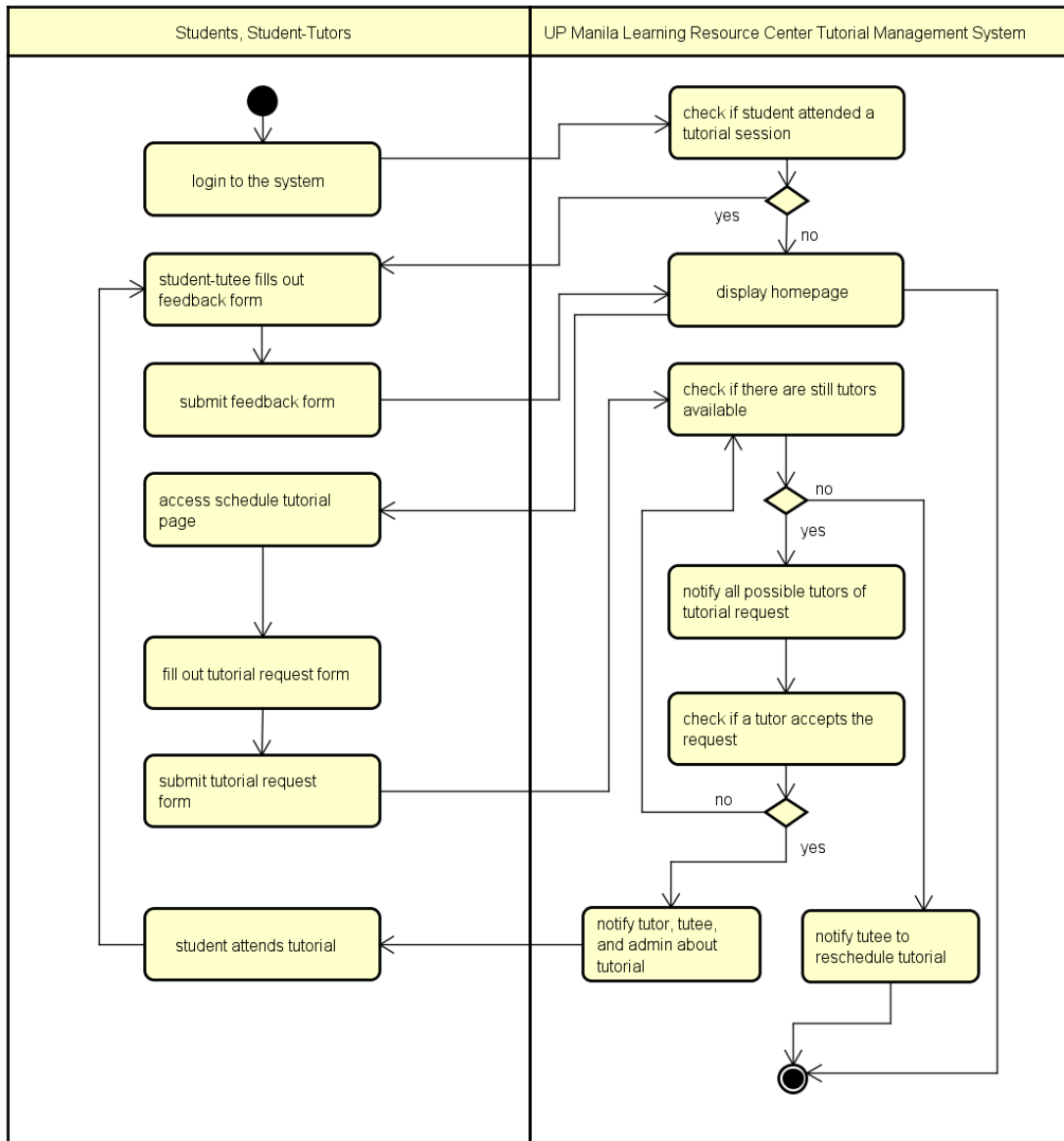


Figure 9: ACTIVITY DIAGRAM, showing the tutorial scheduling process for students and student-tutors in UPM-LRC-TMS.

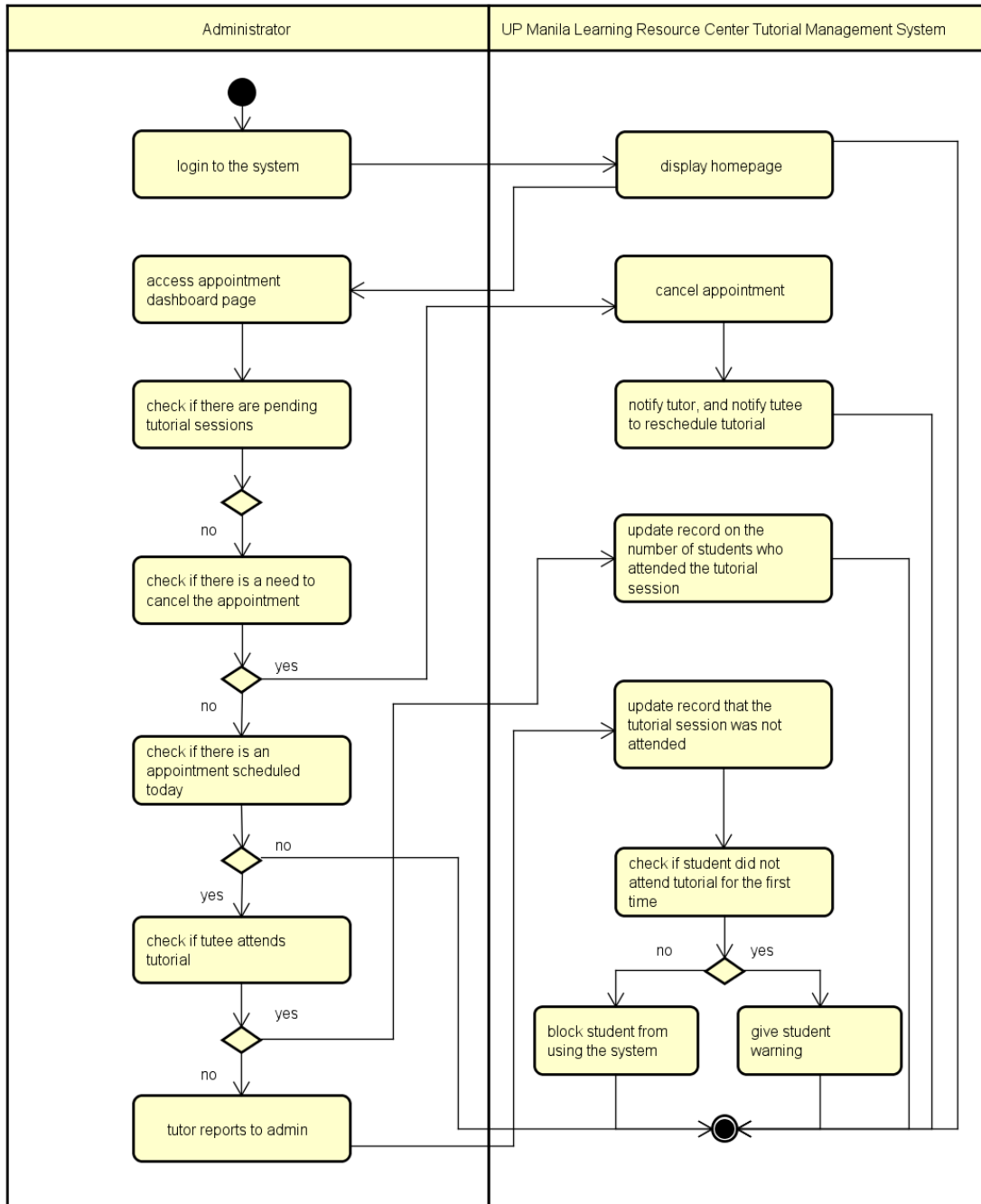


Figure 10: ACTIVITY DIAGRAM, showing the tutorial schedule attendance checking process for the administrator in UPM-LRC-TMS.

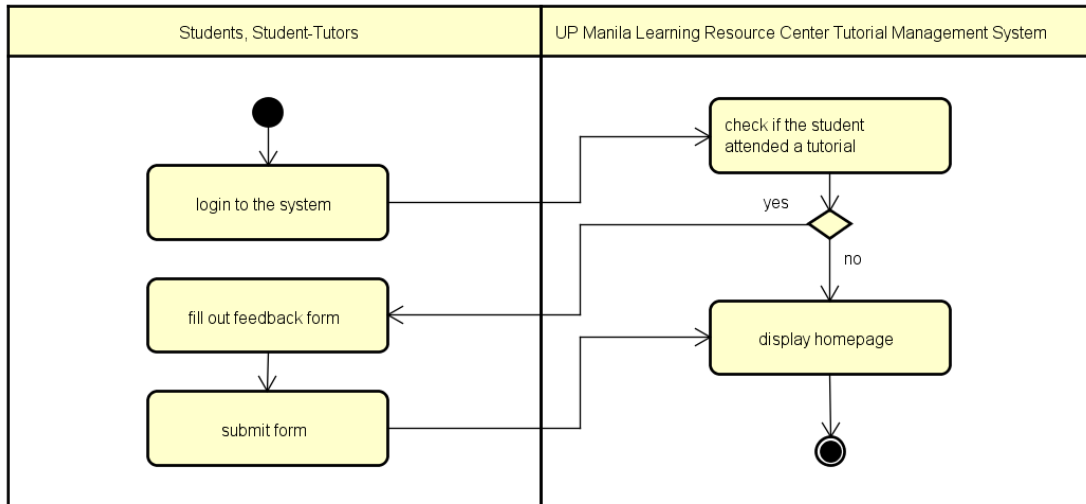


Figure 11: ACTIVITY DIAGRAM, showing the feedback submission process for students and student-tutors in UPM-LRC-TMS.

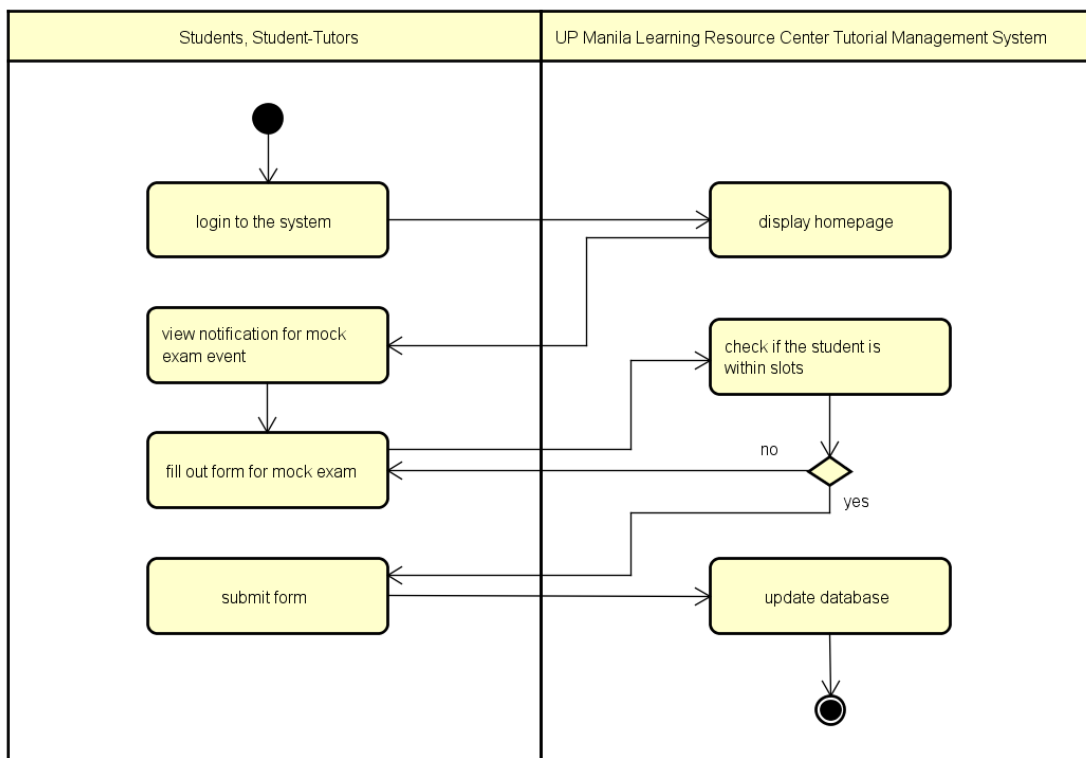


Figure 12: ACTIVITY DIAGRAM, showing the simulated exam application process for student and student-tutors in UPM-LRC-TMS.

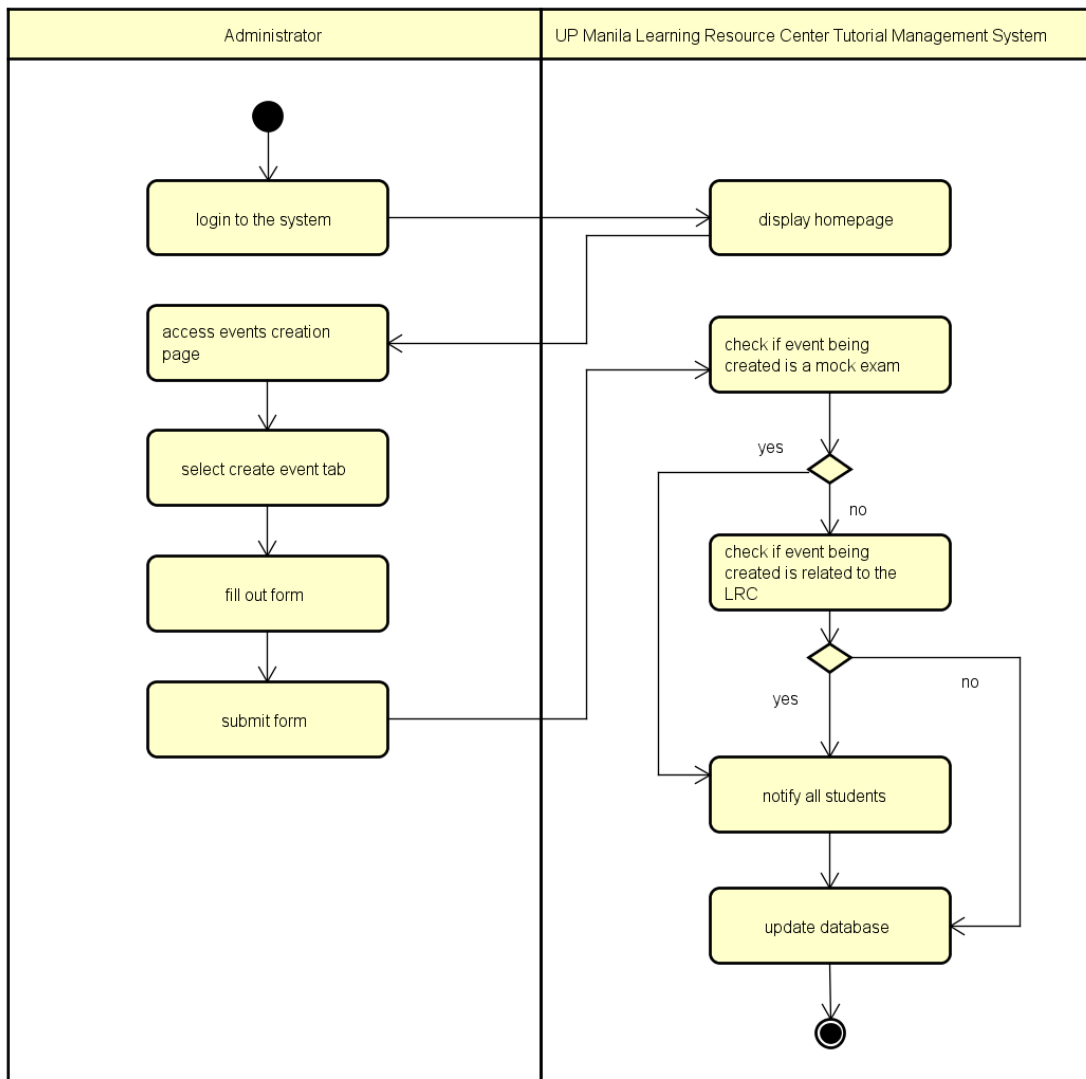


Figure 13: ACTIVITY DIAGRAM, showing the simulated exam event or the LRC announcement creation process for the administrator in UPM-LRC-TMS.

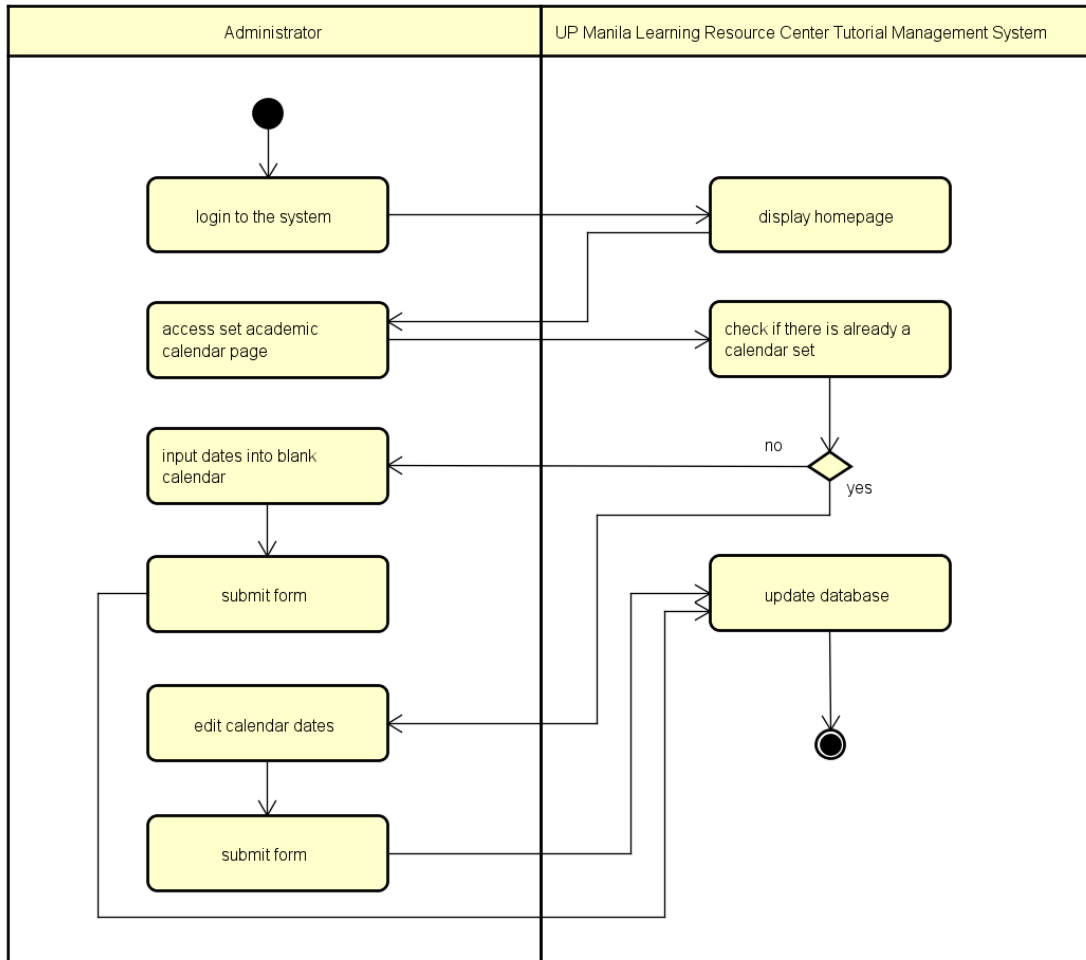


Figure 14: ACTIVITY DIAGRAM, showing the set/edit system academic calendar process for the administrator in UPM-LRC-TMS.

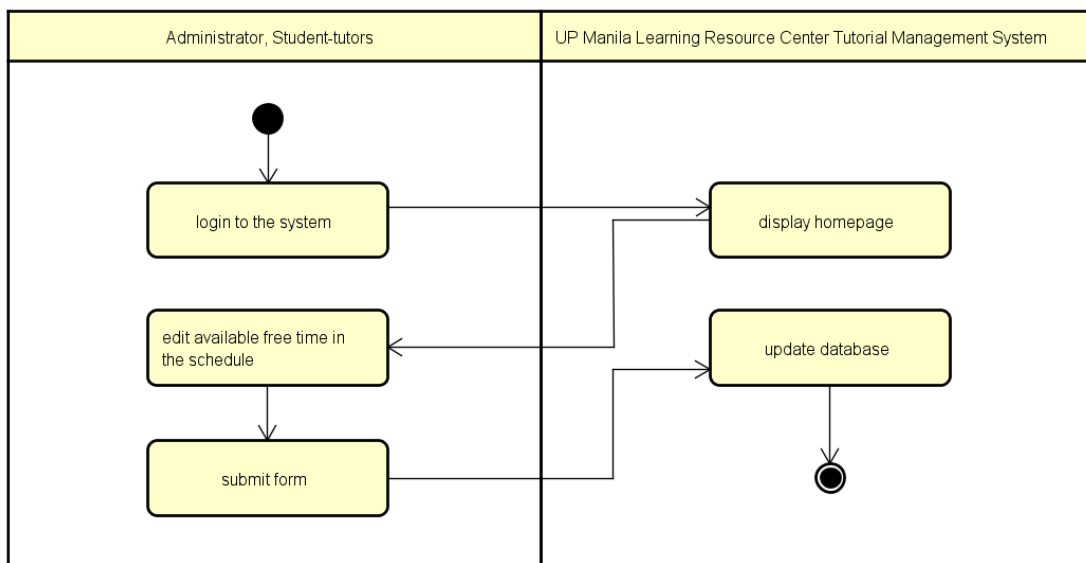


Figure 15: ACTIVITY DIAGRAM, showing the set/edit free time schedule process for the administrator and student-tutors in UPM-LRC-TMS.

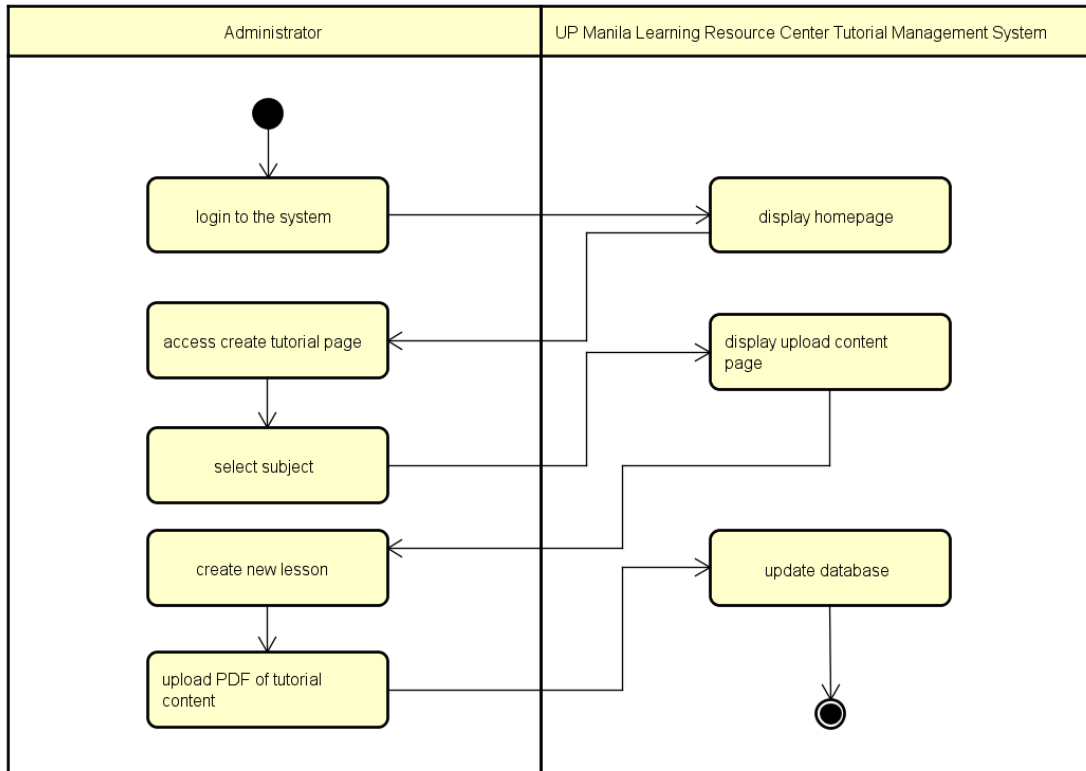


Figure 16: ACTIVITY DIAGRAM, showing the lesson content creation process for the administrator in UPM-LRC-TMS.

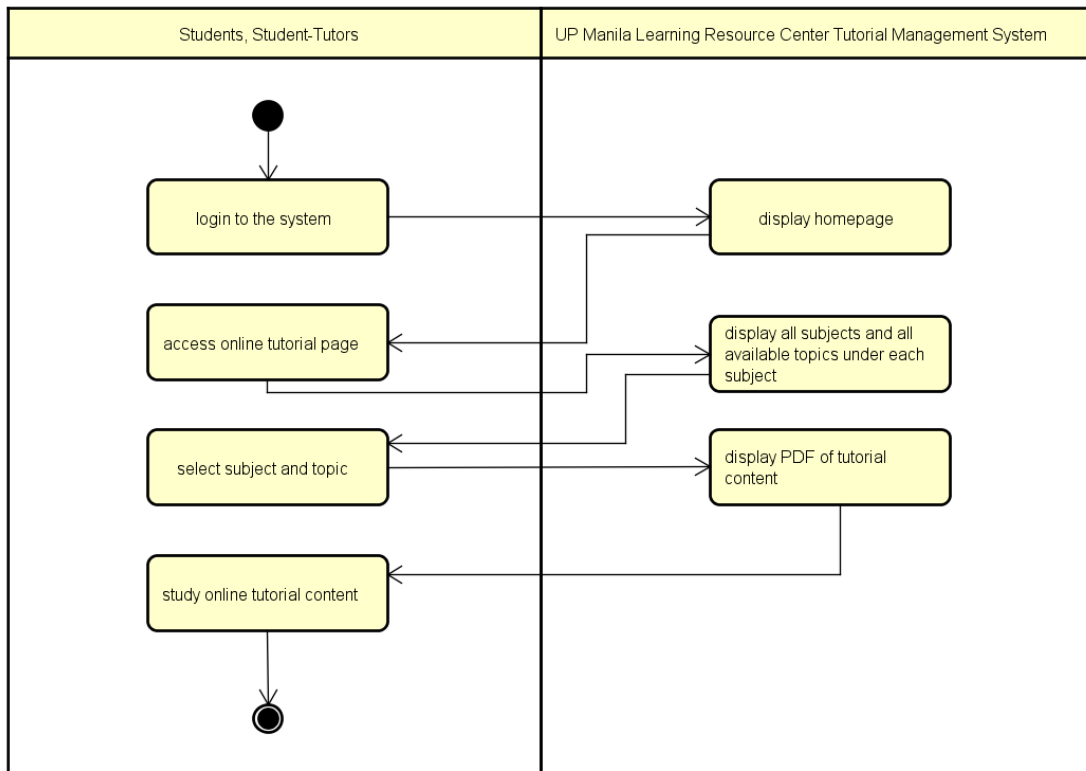


Figure 17: ACTIVITY DIAGRAM, showing the lesson accessing process for the students and student-tutors in UPM-LRC-TMS.

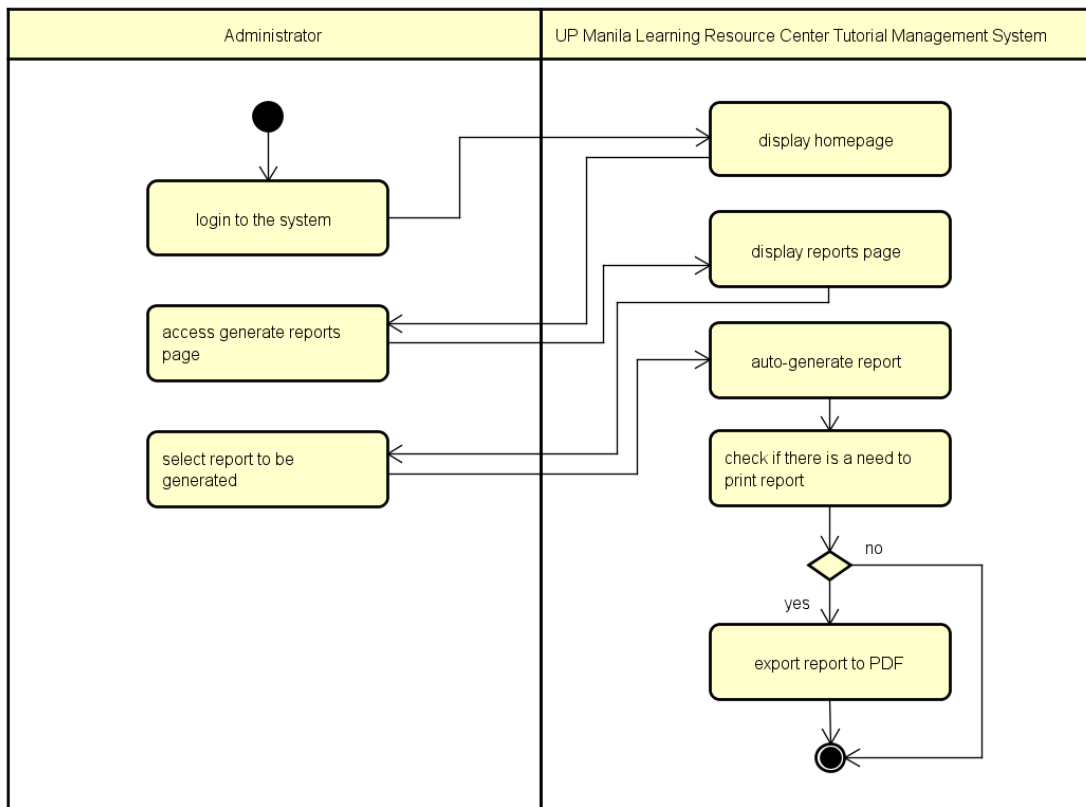


Figure 18: ACTIVITY DIAGRAM, showing the report generation process for the administrator in UPM-LRC-TMS.

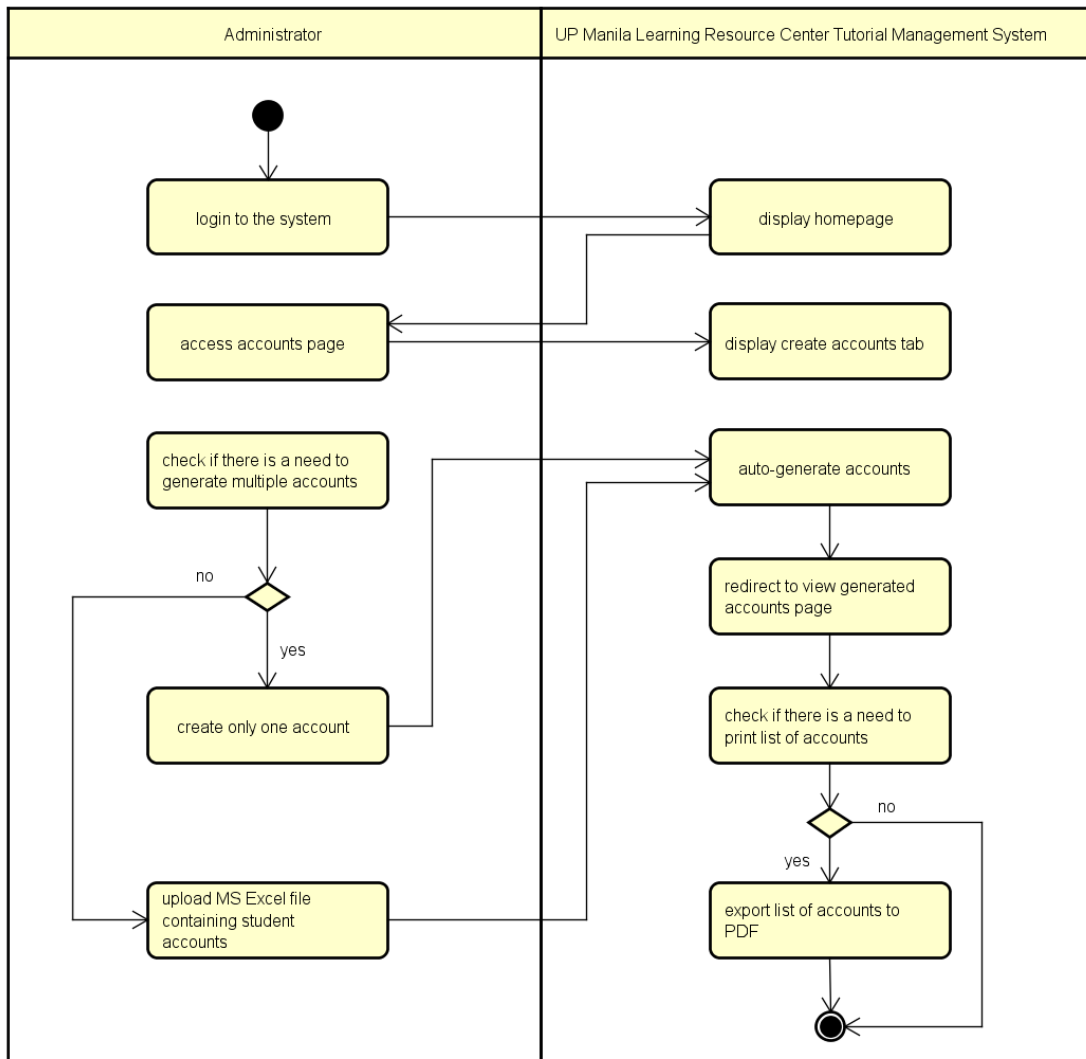


Figure 19: ACTIVITY DIAGRAM, showing the automatic account generation process for the administrator in UPM-LRC-TMS.



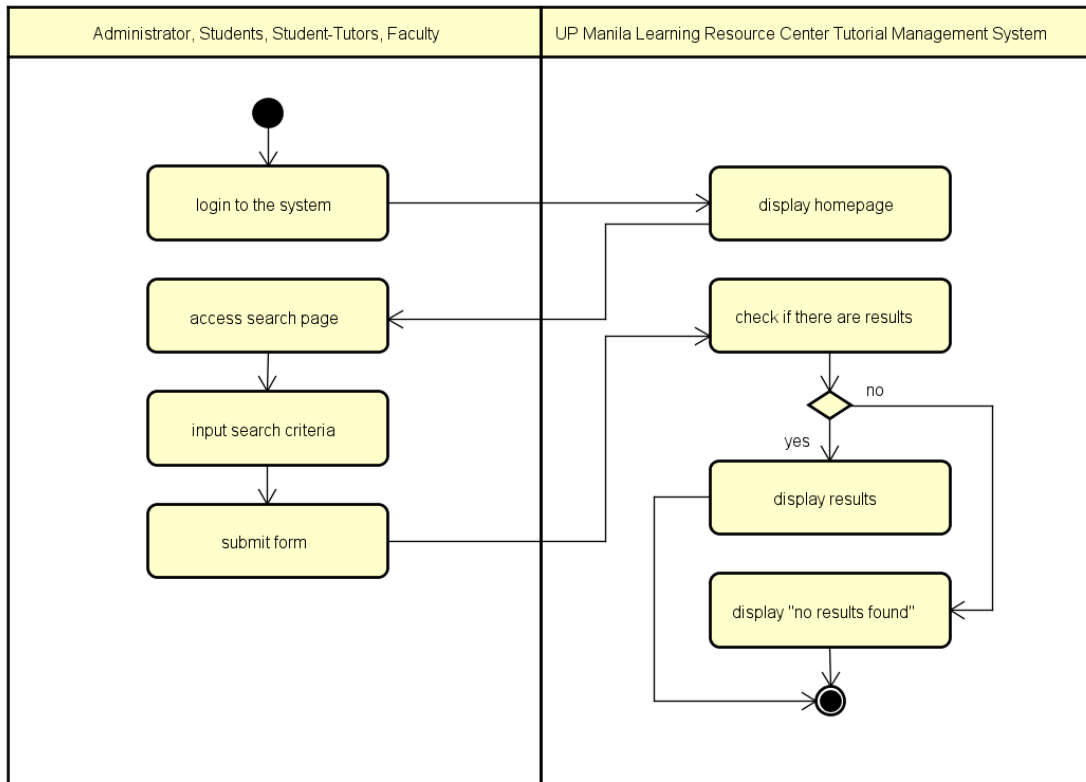


Figure 20: ACTIVITY DIAGRAM, showing the search process for all users in UPM-LRC-TMS.

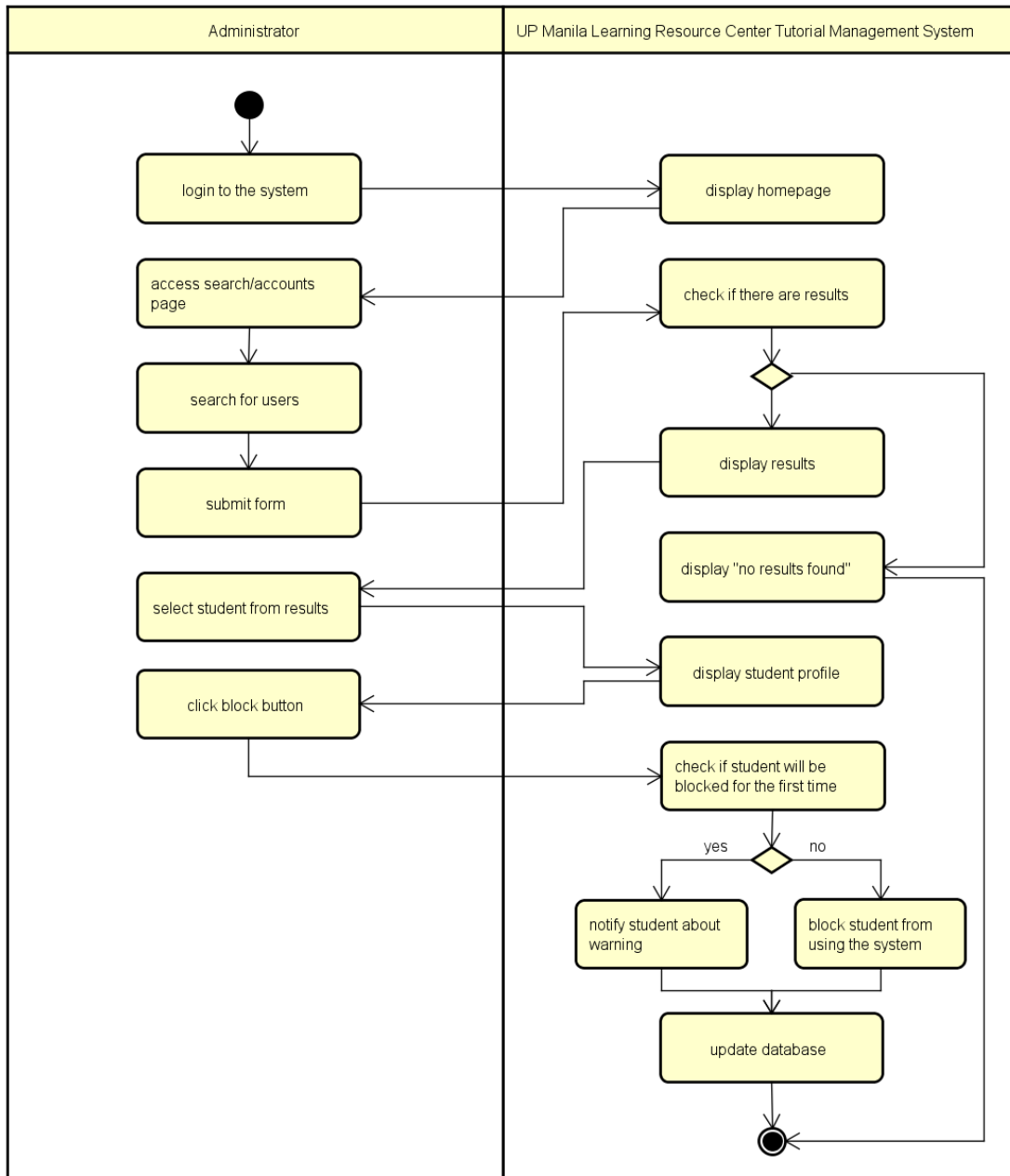


Figure 21: ACTIVITY DIAGRAM, showing the block student process for the administrator in UPM-LRC-TMS.

### 3. Entity Relationship Diagram

The entity relationship diagram is shown in Figure 22. This includes all entities and their relationships with each other. Students can schedule appointments, apply for simulated exams and after-exam discussions, submit feedback regarding a tutor, apply to become a student-tutor, and access on-line tutorial content. Student-tutors can do the same things as student users do. Additionally, they can accept or reject tutorial requests, and view feedback given to them after every semester. The administrator can announce schedules for simulated exams and after-exam discussions, generate student accounts, approve applications from those who are interested to become student-tutors, create content for the online tutorial modules, set the system academic calendar, set the schedule whenever there are available rooms, and block students from using the system. Faculty members can approve or reject endorsement requests, as well as answer and submit endorsement forms for students who are interested to become student-tutors. Simulated exams and after-exam discussions contain the list of students who will attend these sessions.

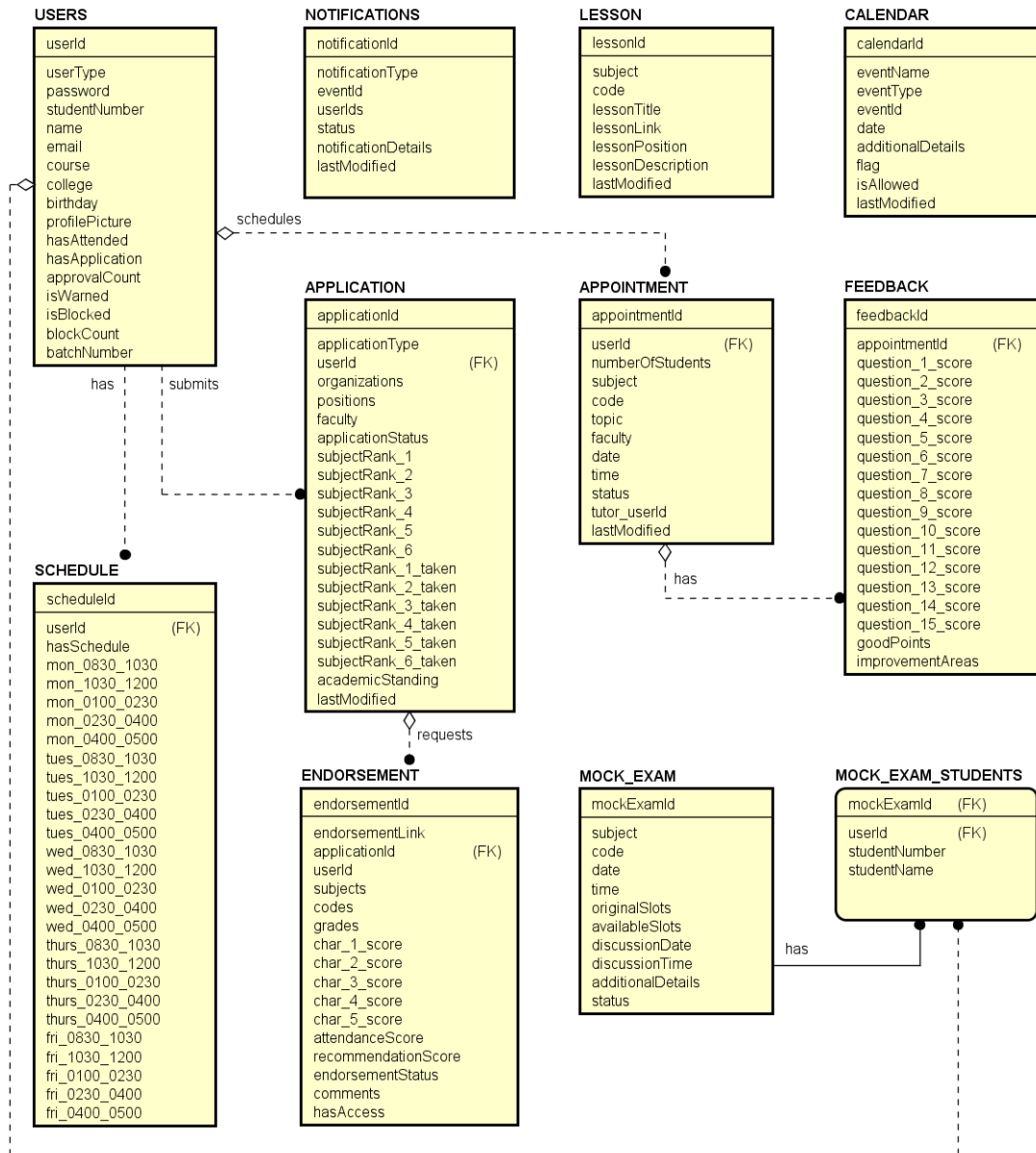


Figure 22: ENTITY RELATIONSHIP DIAGRAM, University of the Philippines-Manila Learning Resource Center Tutorial Management System (UPM-LRC-TMS).

#### 4. Data Dictionary

Table 1: APPLICATION has information regarding a student interested in becoming a student-tutor.

Field	Type	Definition	
applicationId	INT (5)	ID that refers to a specific application	Primary Key
userId	INT (5)	ID that references a specific student-user	Foreign Key
applicationType	SMALLINT (1)	status that determines whether the application is new (0) or a renewal (1)	
organizations	VARCHAR (500)	the organization(s) a student belongs to	
positions	VARCHAR (300)	the respective organization position(s) of the student	
faculty	VARCHAR (50)	the faculty member the student wants to request an endorsement from	
applicationStatus	SMALLINT (1)	status that determines whether the application is disapproved (0), approved (1), or pending (2)	
subjectRank_1	VARCHAR (16)	the first subject that the student prefers; the subject the student prefers the most	
subjectRank_2	VARCHAR (16)	the second subject that the student prefers	
subjectRank_3	VARCHAR (16)	the third subject that the student prefers	

subjectRank_4	VARCHAR (16)	the fourth subject that the student prefers	
subjectRank_5	VARCHAR (16)	the fifth subject that the student prefers	
subjectRank_6	VARCHAR (16)	the sixth subject that the student prefers; the subject the student prefers the least	
subjectRank_1_taken	VARCHAR (50)	the subjects that student has already taken for the subject ranked first	
subjectRank_2_taken	VARCHAR (50)	the subjects that student has already taken for the subject ranked second	
subjectRank_3_taken	VARCHAR (50)	the subjects that student has already taken for the subject ranked third	
subjectRank_4_taken	VARCHAR (50)	the subjects that student has already taken for the subject ranked fourth	
subjectRank_5_taken	VARCHAR (50)	the subjects that student has already taken for the subject ranked fifth	
subjectRank_6_taken	VARCHAR (50)	the subjects that student has already taken for the subject ranked sixth	
academicStanding	VARCHAR (100)	file name of the PDF file containing screenshots of the SAIS pages containing the students grades in all semesters	
lastModified	TIMESTAMP	time of submission or change of the application	

Table 2: APPOINTMENT contains information regarding a scheduled tutorial session.

Field	Type	Definition	
appointmentId	INT (5)	ID that refers to a specific tutorial session	Primary Key
userId	INT (5)	ID that references a specific student-user	Foreign Key
numberOfStudents	INT (2)	the number of students attending the tutorial session	
subject	VARCHAR (16)	subject covered in the tutorial session	
code	VARCHAR (10)	course code of the selected subject for the tutorial session	
topic	VARCHAR (50)	specific topic(s) to be covered in the tutorial session	
faculty	VARCHAR (50)	faculty member who teaches the subject	
date	VARCHAR (10)	date of the tutorial session	
time	VARCHAR (12)	time of the tutorial session	
status	SMALLINT (1)	status that determines whether the requested tutorial session is cancelled (0), approved (1), pending (2), or finished (3)	
tutor_userId	INT (5)	ID that references which student-tutor accepts the tutorial request	Foreign Key
lastModified	TIMESTAMP	time of submission or change of the tutorial schedule	

Table 3: CALENDAR contains information regarding the days set on the system calendar.

Field	Type	Definition	
calendarId	INT (8)	ID that refers to a specific day in the system calendar	Primary Key
eventName	VARCHAR (50)	name of the event	
eventType	VARCHAR (30)	event type, which may be any of the following: beginning of semester (SEM_BEGIN), end of semester (SEM_END), LRC-related announcement (LRC_EVENT), simulated examination (MOCK_EXAM), national holiday (HOLIDAY), or other events (OTHER)	
date	DATE	date of the event	
additionalDetails	VARCHAR(300)	additional details regarding the event	
flag	SMALLINT(1)	flag to determine whether the event marks the beginning of the semester (1), end of the semester (2), finished (5)	
isAllowed	SMALLINT(1)	boolean which determines whether a student-user can schedule a tutorial session on a certain day	



Table 4: ENDORSEMENT contains information regarding the endorsement of a faculty member for a student applicant

Field	Type	Definition	
endorsementId	INT (5)	ID that refers to a specific endorsement given by a faculty member	Primary Key
applicationId	INT (5)	ID that refers to a specific application	Foreign Key
userId	INT (5)	ID that references a specific faculty-user	Foreign Key
char_1_score	SMALLINT (1)	score given regarding the first characteristic of the applicant, as indicated in the form	
char_2_score	SMALLINT (1)	score given regarding the second characteristic of the applicant, as indicated in the form	
char_3_score	SMALLINT (1)	score given regarding the third characteristic of the applicant, as indicated in the form	
char_4_score	SMALLINT (1)	score given regarding the fourth characteristic of the applicant, as indicated in the form	
char_5_score	SMALLINT (1)	score given regarding the fifth characteristic of the applicant, as indicated in the form	

subjects	VARCHAR (300)	subjects the student-user took under the faculty member endorsing	
codes	VARCHAR (50)	course codes of the subjects the student-user took under the faculty member endorsing	
grades	VARCHAR (100)	grades acquired in the subjects the student-user took under the faculty member endorsing	
comments	VARCHAR (300)	additional comments regarding the student being endorsed	
attendanceScore	SMALLINT (1)	score that determines how well the applicant attends the faculty members classes	
recommendationScore	SMALLINT (1)	score that determines whether the applicant is recommended to become a tutor	
status	SMALLINT (1)	status that determines whether the endorsement request is rejected (0), accepted (1), or pending (2)	
hasAccess	SMALLINT (1)	boolean that determines if the faculty member can still access the endorsement page (1), or not (0)	

Table 5: FEEDBACK contains information regarding a submitted feedback form.

Field	Type	Definition	
feedbackId	INT (5)	ID that refers to a specific feedback form submitted	Primary Key
appointmentId	INT (5)	ID that refers to a specific appointment	Foreign Key
question_1_score	SMALLINT (1)	score of the first question in the feedback form	
question_2_score	SMALLINT (1)	score of the second question in the feedback form	
question_3_score	SMALLINT (1)	score of the third question in the feedback form	
question_4_score	SMALLINT (1)	score of the fourth question in the feedback form	
question_5_score	SMALLINT (1)	score of the fifth question in the feedback form	
question_6_score	SMALLINT (1)	score of the sixth question in the feedback form	
question_7_score	SMALLINT (1)	score of the seventh question in the feedback form	
question_8_score	SMALLINT (1)	score of the eighth question in the feedback form	
question_9_score	SMALLINT (1)	score of the ninth question in the feedback form	
question_10_score	SMALLINT (1)	score of the tenth question in the feedback form	
question_11_score	SMALLINT (1)	score of the eleventh question in the feedback form	

question_12_score	SMALLINT (1)	score of the twelfth question in the feedback form	
question_13_score	SMALLINT (1)	score of the thirteenth question in the feedback form	
question_14_score	SMALLINT (1)	score of the fourteenth question in the feedback form	
question_15_score	SMALLINT (1)	score of the fifteenth question in the feedback form	
goodPoints	VARCHAR (300)	positive feedback regarding the tutorial session	
improvementAreas	VARCHAR (300)	areas of improvement regarding the tutorial session	

Table 6: LESSON contains information regarding a lesson in the online tutorial.

Field	Type	Definition	
lessonId	INT (5)	ID that refers to a specific lesson page	Primary Key
subject	VARCHAR (16)	subject of the lesson	
code	VARCHAR (10)	course code of the subject	
lessonTitle	VARCHAR (50)	title of the lesson	
lessonLink	VARCHAR (50)	file name of the PDF file of the uploaded lesson	
lessonPosition	INT (2)	position of the lesson in a subjects lesson flow	
lessonDescription	VARCHAR (300)	description of the lesson	
lastModified	TIMESTAMP	time of submission or change of the lesson details	

Table 7: MOCKEXAM contains information regarding a simulated exam event.

Field	Type	Definition	
mockExamId	INT (5)	ID that refers to a specific simulated exam event	Primary Key
subject	VARCHAR (16)	subject covered in the simulated exam	
code	VARCHAR (10)	course code of the subject	
date	VARCHAR (10)	date of the simulated exam	
time	VARCHAR (12)	time of the simulated exam	
originalSlots	INT (3)	original number of slots available for the simulated exam	
availableSlots	INT (3)	available number of slots available for the simulated exam	
discussionDate	date	date of the after-exam discussion	
discussionTime	VARCHAR (8)	time of the after-exam discussion	
additionalDetails	VARCHAR (300)	additional details regarding the simulated exam event	
status	SMALLINT (1)	boolean which determines if the simulated exam is cancelled (0), available (2), or finished (3)	

Table 8: MOCKEXAM\_STUDENTS contains the information regarding which students will attend the simulated exam.

Field	Type	Definition	
mockExamId	INT (5)	ID that refers to a specific simulated exam event	Foreign Key
userId	INT (5)	Id that references a specific student-user	Foreign Key
studentNumber	INT (9)	student number of the student	
studentName	VARCHAR (50)	name of the student	

Table 9: NOTIFICATIONS contains the information regarding a notification.

Field	Type	Definition	
notificationId	INT (7)	ID that refers to a specific notification; is a concatenation of the notification type and the event number	Primary Key
notificationType	INT (3)	determines kind of notification	
eventId	INT (5)	specific ID which refers to the event in the notification	
userIds	VARCHAR (100)	indicates which users a specific notification is meant for	
notificationDetails	VARCHAR (500)	additional details regarding the notification	
status	SMALLINT (1)	boolean which determines whether the notification is inactive (0), or active (1)	
lastModified	TIMESTAMP	time of creation of change of the notification	

Table 10: SCHEDULE contains the schedule of a student.

Field	Type	Definition	
scheduleId	INT (5)	ID that refers to a specific schedule	Primary Key
userId	INT (5)	ID that references a specific user	Foreign Key
hasSchedule	SMALLINT (1)	boolean that determines if the user has input his/her schedule already	
mon_0830_1030	SMALLINT (1)	boolean that determines the time frame is free (8:30 10:30) every Monday	
mon_1030_1200	SMALLINT (1)	boolean that determines the time frame is free (10:30 12:00) every Monday	
mon_0100_0230	SMALLINT (1)	boolean that determines the time frame is free (1:00 2:30) every Monday	
mon_0230_0400	SMALLINT (1)	boolean that determines the time frame is free (2:30 4:00) every Monday	
mon_0400_0500	SMALLINT (1)	boolean that determines the time frame is free (4:00 5:00) every Monday	
tues_0830_1030	SMALLINT (1)	boolean that determines the time frame is free (8:30 10:30) every Tuesday	
tues_1030_1200	SMALLINT (1)	boolean that determines the time frame is free (10:30 12:00) every Tuesday	

tues_0100_0230	SMALLINT (1)	boolean that determines the time frame is free (1:00 2:30) every Tuesday	
tues_0230_0400	SMALLINT (1)	boolean that determines the time frame is free (2:30 4:00) every Tuesday	
tues_0400_0500	SMALLINT (1)	boolean that determines the time frame is free (4:00 5:00) every Tuesday	
wed_0830_1030	SMALLINT (1)	boolean that determines the time frame is free (8:30 10:30) every Wednesday	
wed_1030_1200	SMALLINT (1)	boolean that determines the time frame is free (10:30 12:00) every Wednesday	
wed_0100_0230	SMALLINT (1)	boolean that determines the time frame is free (1:00 2:30) every Wednesday	
wed_0230_0400	SMALLINT (1)	boolean that determines the time frame is free (2:30 4:00) every Wednesday	
wed_0400_0500	SMALLINT (1)	boolean that determines the time frame is free (4:00 5:00) every Wednesday	
thurs_0830_1030	SMALLINT (1)	boolean that determines the time frame is free (8:30 10:30) every Thursday	



thurs_1030_1200	SMALLINT (1)	boolean that determines the time frame is free (10:30 12:00) every Thursday	
thurs_0100_0230	SMALLINT (1)	boolean that determines the time frame is free (1:00 2:30) every Thursday	
thurs_0230_0400	SMALLINT (1)	boolean that determines the time frame is free (2:30 4:00) every Thursday	
thurs_0400_0500	SMALLINT (1)	boolean that determines the time frame is free (4:00 5:00) every Thursday	
fri_0830_1030	SMALLINT (1)	boolean that determines the time frame is free (8:30 10:30) every Friday	
fri_1030_1200	SMALLINT (1)	boolean that determines the time frame is free (10:30 12:00) every Friday	
fri_0100_0230	SMALLINT (1)	boolean that determines the time frame is free (1:00 2:30) every Friday	
fri_0230_0400	SMALLINT (1)	boolean that determines the time frame is free (2:30 4:00) every Friday	
fri_0400_0500	SMALLINT (1)	boolean that determines the time frame is free (4:00 5:00) every Friday	

Table 11: USERS contains all accounts in the system.

Field	Type	Definition	
userId	INT (9)	ID that references a specific user	Primary Key
userType	SMALLINT (1)	determines the kind of user, whether system-administrator (1), student (2), or student-tutor (3)	
password	VARCHAR (30)	the users password	
studentNumber	INT (9)	ID that references either a student or a student-tutor	
name	VARCHAR (50)	name of the user	
email	VARCHAR (50)	email address of the user	
course	VARCHAR (50)	course of a student or a student-tutor	
college	VARCHAR (4)	college where a student or a student-tutor belongs to	
birthday	VARCHAR (10)	birthday of the user	
profilePicture	VARCHAR (100)	file name of the image file of the user	
hasAttended	VARCHAR (5)	appointment ID which determines the tutorial session the user recently attended; if set, the user will be redirected to the feedback submission page	
hasApplication	SMALLINT (1)	boolean which determines if a student-user has submitted an application form (1), or not (0)	

approvalCount	INT (2)	the number of times a student has been approved to become a student-tutor	
isWarned	SMALLINT (1)	boolean that determines whether the student or student-tutor has a warning (1), or not (0)	
isBlocked	SMALLINT (1)	boolean that determines whether the student or student-tutor is blocked from using the system (1), or not (0)	
blockCount	INT (3)	the number of times a student or student-tutor is blocked from using the system	
batchNumber	INT (5)	indicates which batch of accounts the user is in; used for report generation	

## 5. System Architecture

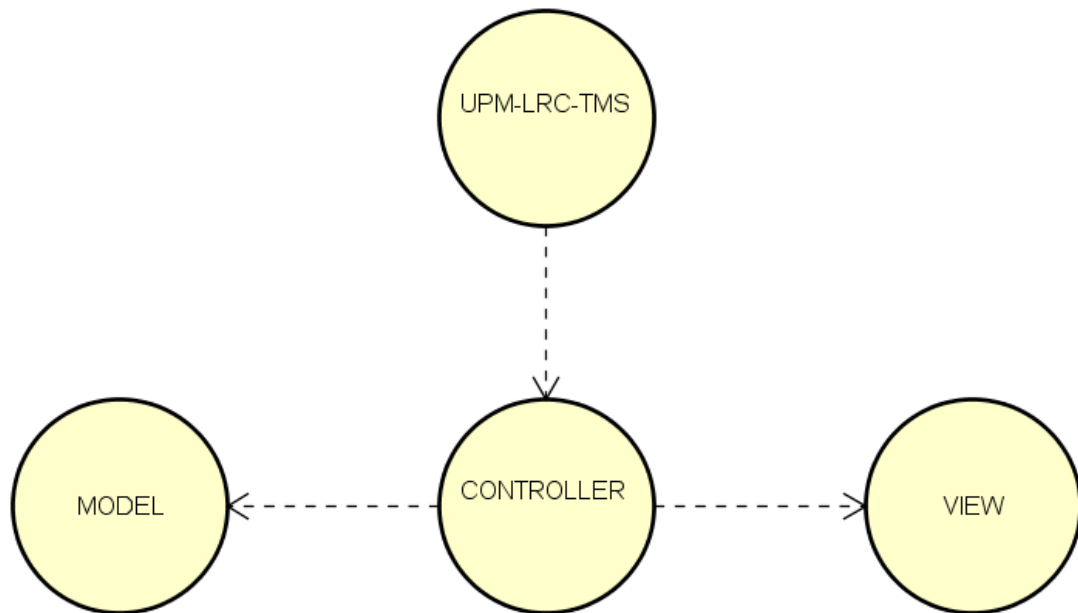


Figure 23: Model-View-Controller.

The University of the Philippines-Manila Learning Resource Center Tutorial Management System (UPM-LRC-TMS) follows the Model-View-Controller design from the CodeIgniter framework. Model-View-Controller is a methodology or design pattern for successfully and efficiently relating the user interface to underlying data models. It is heralded as useful for the reuse of object code, since it allows to significantly reduce the time it takes to develop applications with user interfaces. Its three main components are:

- (a) The model, which represents the underlying logical data in a software application. It does not contain any information from and for the user interface.
- (b) The view, which is a collection of classes that contain elements for the user interface.
- (c) The controller, which represents the classes connecting the model and the view [22].

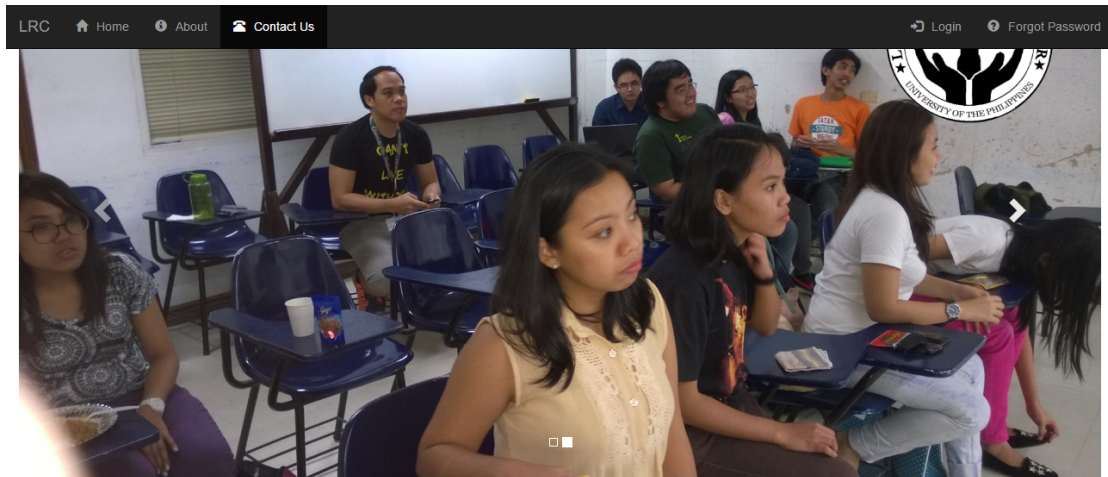
## 6. Technical Architecture

These are the minimum requirements for the University of the Philippines-Manila Learning Resource Center Tutorial Management System:

- Web Server: Apache
- Database Server: MySQL
- MySQL Administration Tool: phpMyAdmin
- PHP Framework: CodeIgniter
- Internet browser
- Stable internet connection

## V. Results

Users: Administrator, Student, Student-tutor



University of the Philippines-Manila's Learning Resource Center now has its own Tutorial Management System.  
This system is designed to provide an online, user-friendly interface for everything that's usually done at the LRC.

Figure 24: Home Page. The home page gives a brief preview of the system, and the contact details of the LRC. There are links provided for logging in, and for resetting the password.

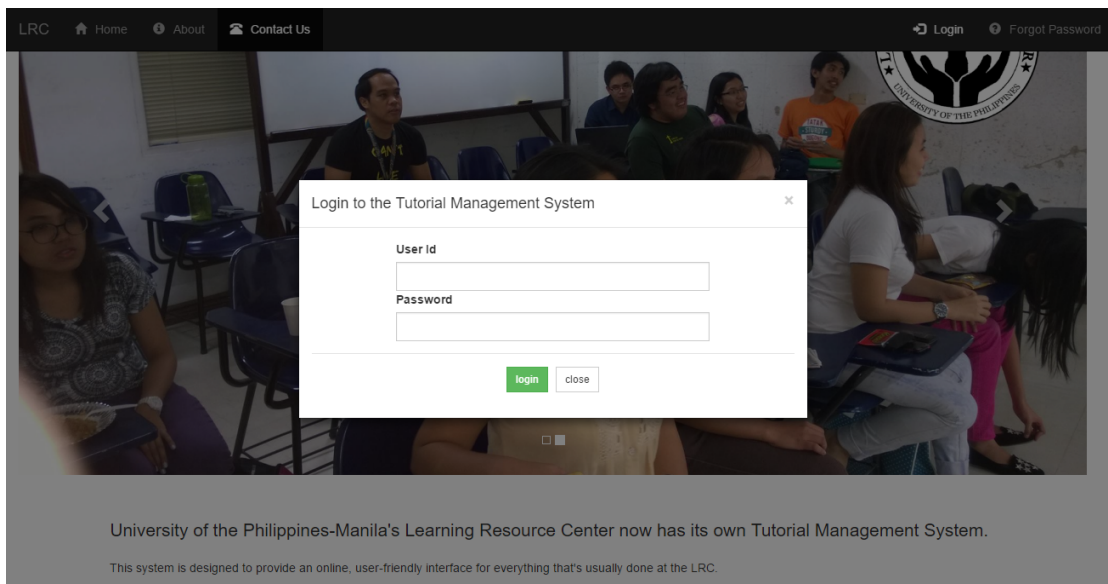


Figure 25: Login. The login modal prompts the user to enter his/her User Id and Password.

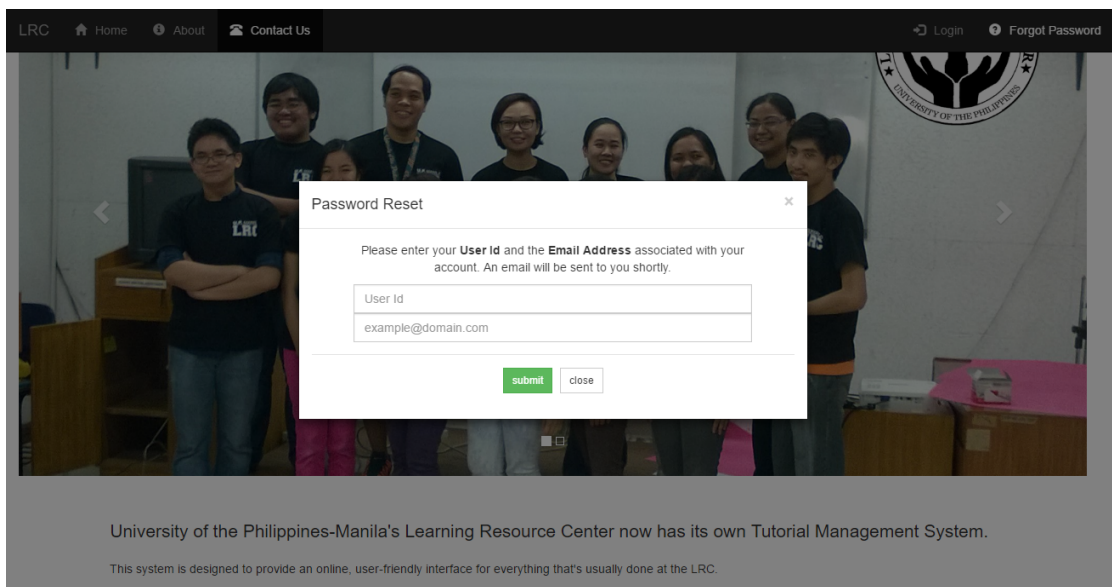


Figure 26: Forgot Password. If ever the user forgets his/her password, a modal prompts the user to enter his/her User Id and the email address associated with the account. An email containing a temporary password will be sent.

## User: Administrator

The screenshot shows the Administrator Dashboard. At the top, there is a navigation bar with links for Home, Search, Applications, Tutorial Sessions, Online Lessons, Calendar, Reports, Accounts, and a user profile for Administrator ADMIN. The main content area is divided into two columns. The left column features a 'Notifications' section with a green header, followed by a 'Schedule' section. The 'Schedule' section contains a grid with columns for days of the week (M, T, W, T, F) and rows for time slots (8:30-10:30, 10:30-12:00, 1:00-2:30, 2:30-4:00, 4:00-5:00). A legend indicates that green squares represent 'available' and red squares represent 'not available'. Below the grid is a 'save schedule' button and an 'about' link. The right column displays two notification cards. The first card, dated May 17 2016, 2:05 pm, states 'A student has filed an application.' The second card, dated May 17 2016, 1:05 pm, states 'A student has requested a new tutorial session.'

Figure 27: Administrator - Dashboard. All notifications meant for the administrator are shown here. The administrator can also edit the schedule of the availability of the rooms in the LRC in this page.

The screenshot shows the Administrator Accounts Overview page. At the top, there is a navigation bar with links for Home, Search, Applications, Tutorial Sessions, Online Lessons, Calendar, Reports, Accounts, and a user profile for Administrator ADMIN. The main content area is divided into two columns. The left column features an 'Overview' section with a green header and three links: 'View Tutors', 'View Blocked Students', and 'Generate Accounts'. The right column features a 'generate pdf for all accounts' button at the top. Below this, there is a list of generated accounts. The first account is 'Abad, Earl Andrea General' with a Student Number of 2012 - 45085 and a User Id of 10001. The second account is 'Abdulla, Abdel Jeffri Caeg Jr.' with a Student Number of 2012 - 24745 and a User Id of 10002. Each account entry includes a 'view feedback' button. The third account is partially visible at the bottom of the screen.

Figure 28: Administrator - Accounts Overview. This page shows the entire list of generated accounts for the system. The administrator may also choose to generate a PDF containing the list of all accounts generated by the system.



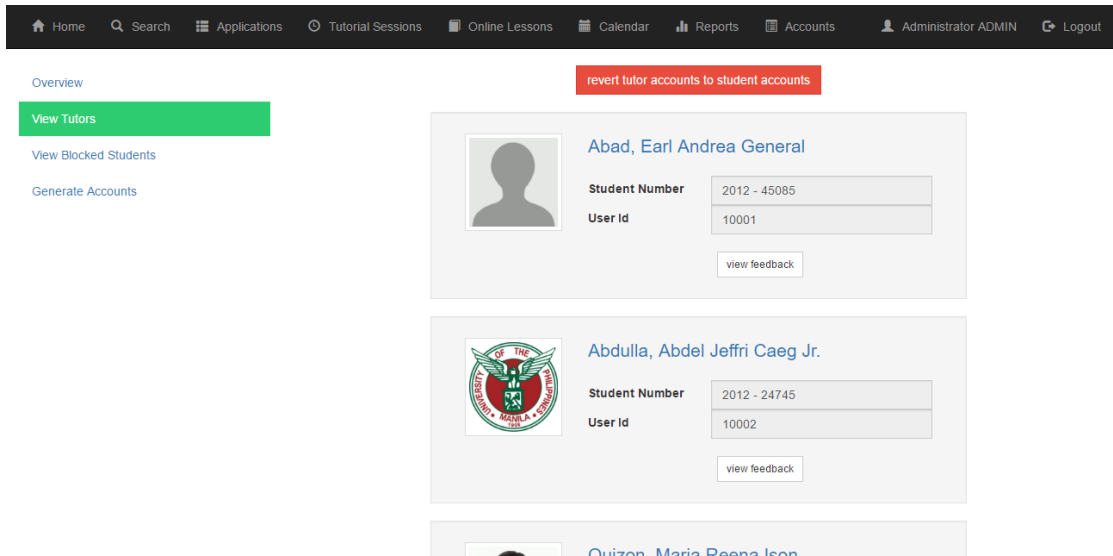


Figure 29: Administrator - Tutors Overview. This page shows the list of student-tutors. The administrator may choose to revert all student-tutor accounts to student accounts.

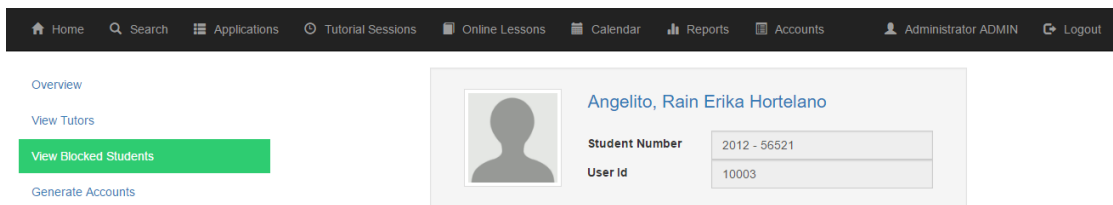


Figure 30: Administrator - Blocked Accounts. This page shows the list of accounts blocked from using the system.

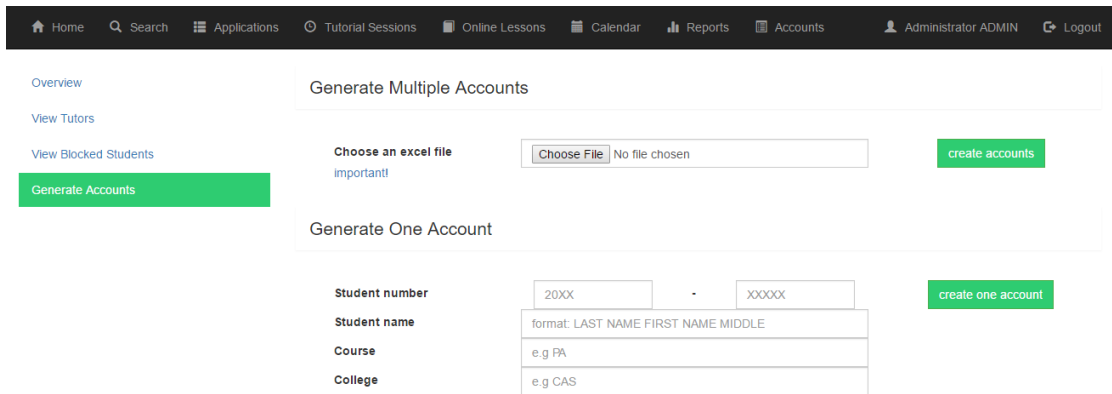


Figure 31: Administrator - Generate Accounts. This page is where the administrator can generate student accounts. He/she has the option to generate multiple accounts using a preformatted excel file, or just generate one student account.

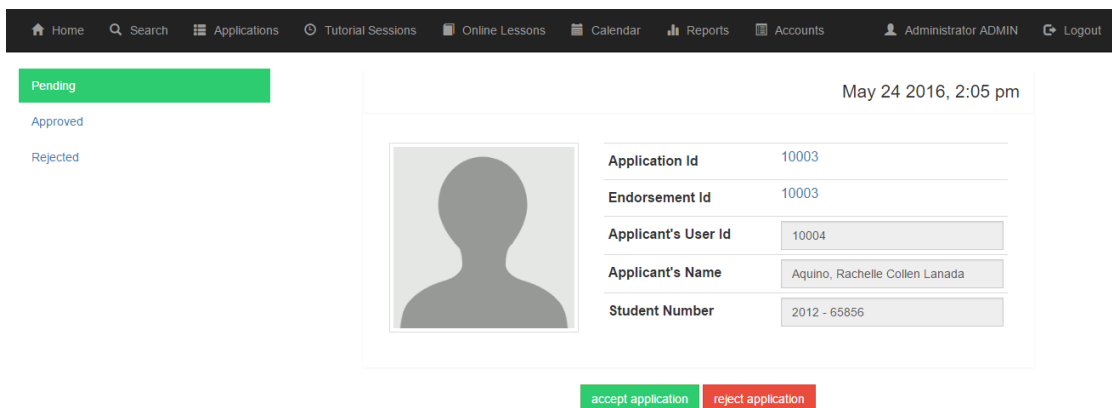


Figure 32: Administrator - Pending Applications. All pending applications for those interested to become student-tutors are displayed here. The administrator may decide to approve or reject an application in this page.

Home Search Applications Tutorial Sessions Online Lessons Calendar Reports Accounts Administrator ADMIN Logout

Pending  
**Approved**  
 Rejected

May 17 2016, 2:05 pm

Application Id 10000  
 Endorsement Id 10000  
 Applicant's User Id 10002  
 Applicant's Name Abdulla, Abdel Jeffri Caeg Jr.  
 Student Number 2012 - 24745

May 24 2016, 9:05 am

Application Id 10001  
 Endorsement Id 10001  
 Applicant's User Id 10001

Figure 33: Administrator - Approved Applications. All approved applications for those interested to become student-tutors are displayed here.

Home Search Applications Tutorial Sessions Online Lessons Calendar Reports Accounts Administrator ADMIN Logout

Pending  
 Approved  
**Rejected**

May 24 2016, 2:05 pm

Application Id 10003  
 Endorsement Id 10003  
 Applicant's User Id 10004  
 Applicant's Name Aquino, Rachelle Collen Lanada  
 Student Number 2012 - 65856

Figure 34: Administrator - Rejected Applications. All rejected applications for those interested to become student-tutors are displayed here.

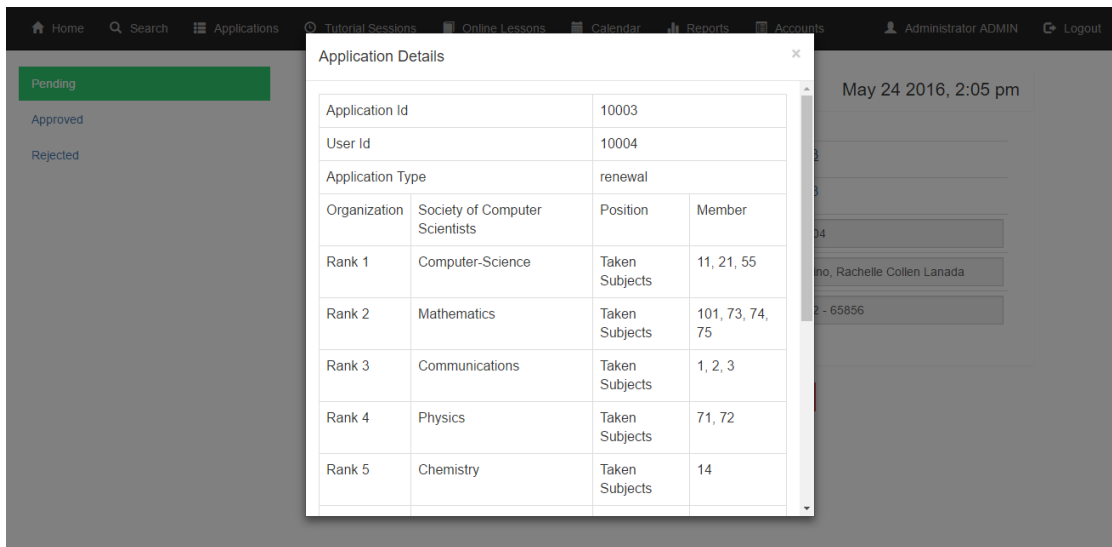


Figure 35: Administrator - View Application. Details of a specific application are displayed in this modal.

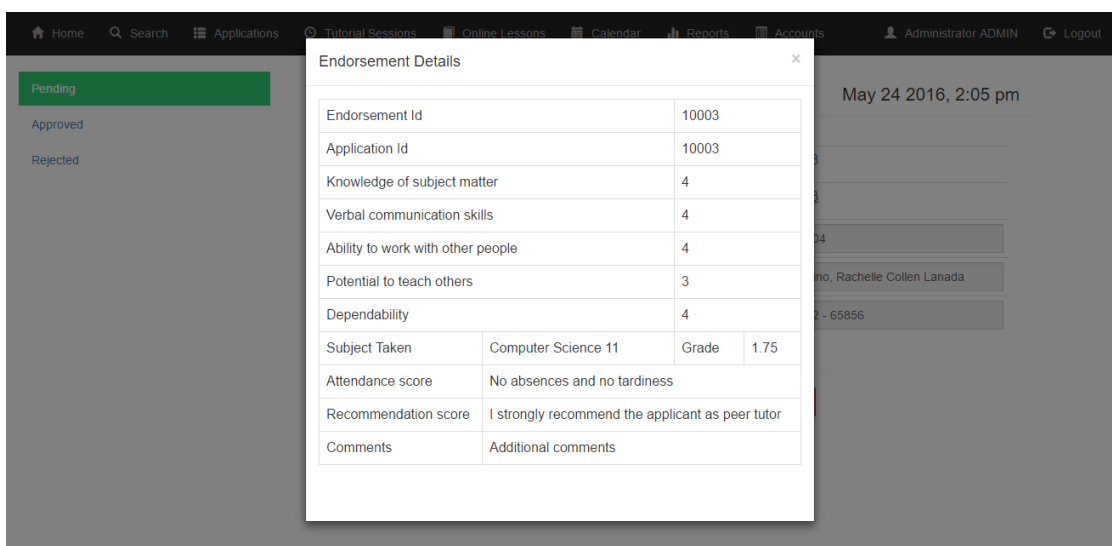


Figure 36: Administrator - View Endorsement. Details of a specific endorsement are displayed in this modal.

Home Search Applications Tutorial Sessions Online Lessons Calendar Reports Accounts Administrator ADMIN Logout

Pending

Finished

Cancelled

May 24 2016, 3:05 pm

Tutee's User Id	10004
Tutor's User Id	10001
Subject	Mathematics 14
Tutorial Date	2016-05-24
Tutorial Time	4:00-5:00

Number of students who attended:  tutee attended tutee did not attend

cancel appointment

Figure 37: Administrator - Pending Tutorial Sessions. All pending (unfinished) tutorial sessions are displayed here. The administrator may choose to cancel a tutorial session in this page.

Home Search Applications Tutorial Sessions Online Lessons Calendar Reports Accounts Administrator ADMIN Logout

Pending

Finished

Cancelled

May 17 2016, 1:05 pm

Tutee's User Id	10002
Tutor's User Id	10035
Subject	Computer Science 11
Tutorial Date	2016-05-17
Tutorial Time	8:30-10:30

Figure 38: Administrator - Finished Tutorial Sessions. All finished tutorial sessions are displayed here.

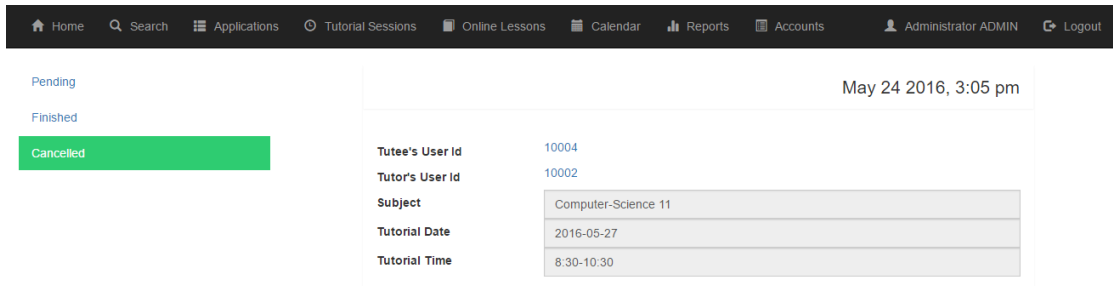


Figure 39: Administrator - Cancelled Tutorial Sessions. All cancelled tutorial sessions are displayed here.

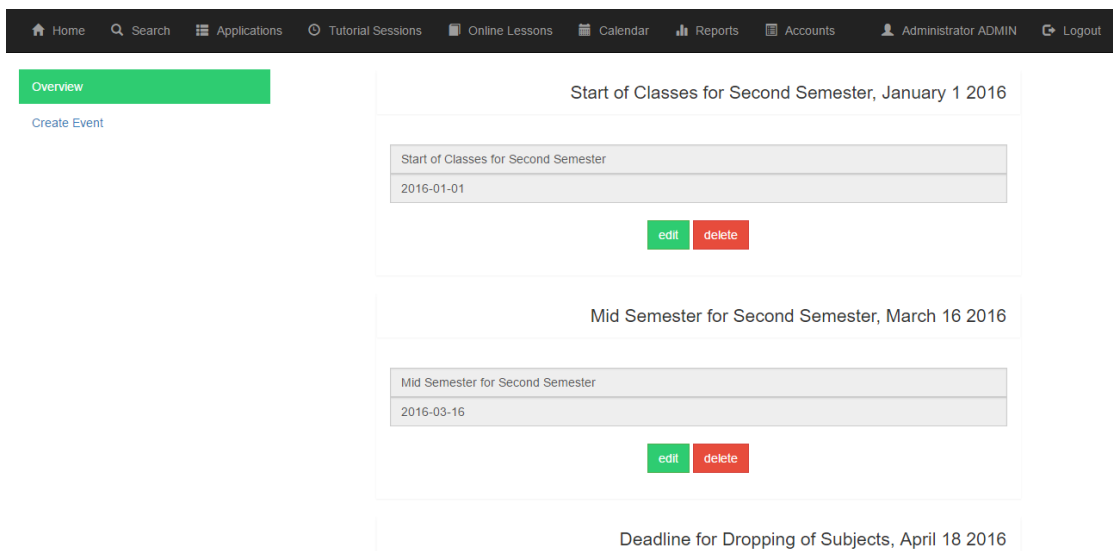


Figure 40: Administrator - Calendar. This page shows all set calendar dates for the system. There are edit and delete options per entry.

Home Search Applications Tutorial Sessions Online Lessons Calendar Reports Accounts Administrator ADMIN Logout

Overview

Create Event

### Create an event

Event Name

Event Date

Event Type

Additional Details

Subject

Mock exam time

Discussion date

Discussion time

Slots

close

add entry

Figure 41: Administrator - Create Events. This page is where the administrator can set the dates to be used by the system. Also, the administrator can create simulated exam events and LRC-related announcements.

Home Search Applications Tutorial Sessions Online Lessons Calendar Reports Accounts Administrator ADMIN Logout

Chemistry

Mathematics

Physics

Biology

Computer Science

Communications

Create Lesson

### Chemistry subjects

Chemistry 14
Chemistry 18
Chemistry 27
Chemistry 31
Chemistry 40
Chemistry 156
Chemistry 157
NSI part

Figure 42: Administrator - Lesson Index. This page shows all subjects where online lesson content may be available.

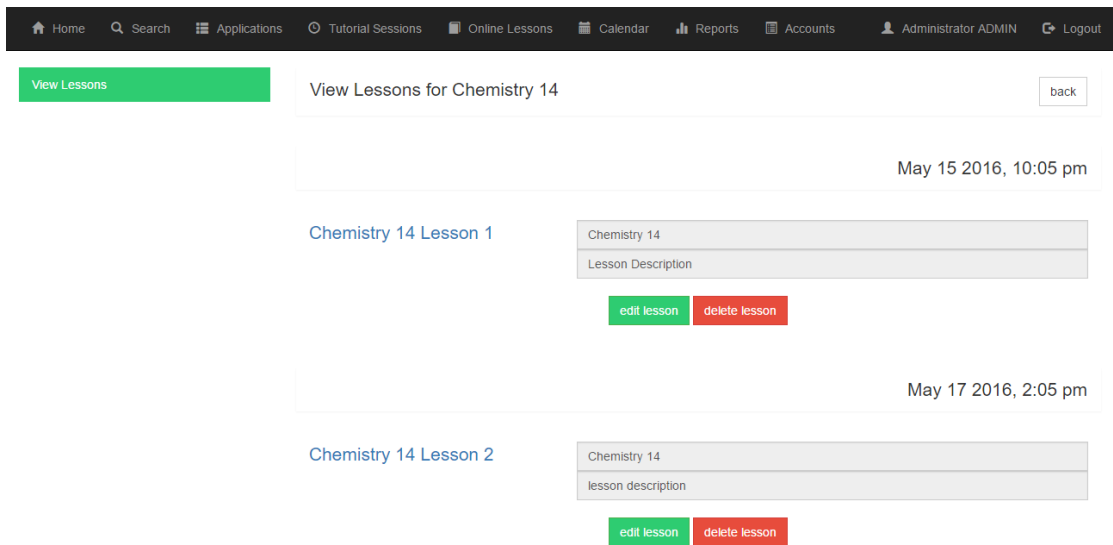


Figure 43: Administrator - View Lessons. This page shows all available online lessons for a specific subject. There are edit and delete options per lesson.

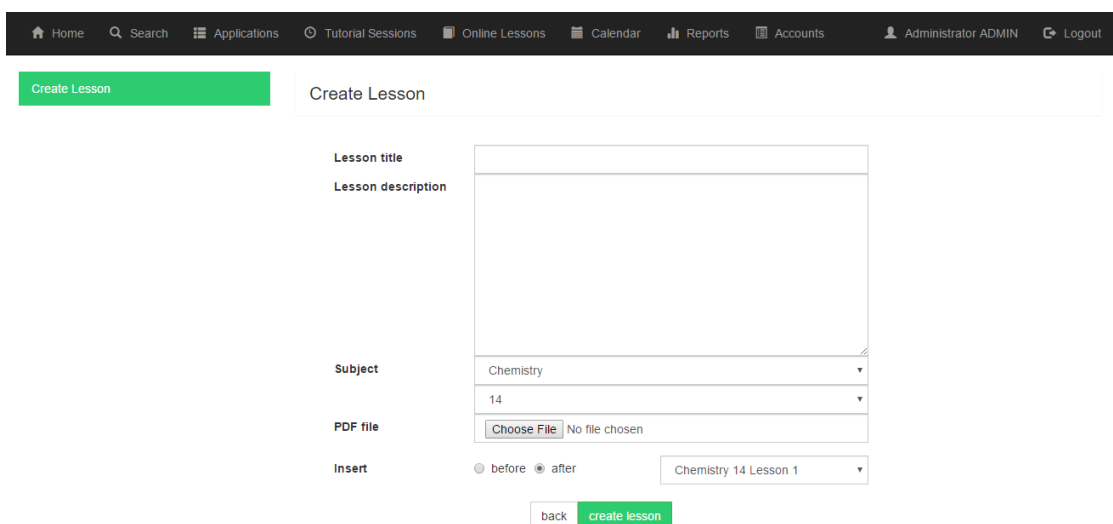


Figure 44: Administrator - Create Lesson. The administrator can upload a PDF for online lesson content here.



Figure 45: Administrator - Edit Lesson. The administrator can edit an existing online lesson here.

<b>Subject</b>	Computer Science 11
<b>Date</b>	2016-05-19
<b>Time</b>	10:00:00
<b>Total Slots</b>	5
<b>Available Slots</b>	4
<b>Discussion Schedule</b>	2016-05-23 12:30:00
<b>Additional Details</b>	details

Figure 46: Administrator - Mock Exam - View Details. The administrator can view details for a simulated exam in this page.

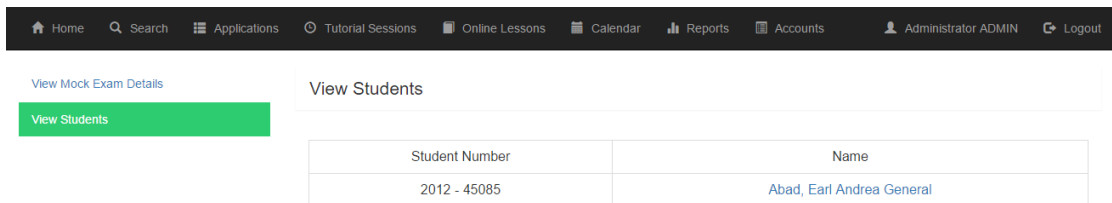


Figure 47: Administrator - Mock Exam - View Student. The administrator can view the list of students who applied for a simulated exam in this page.

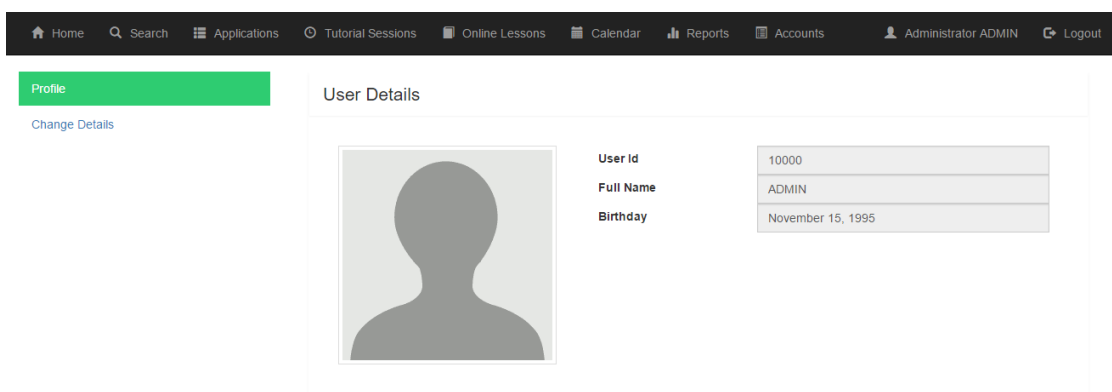


Figure 48: Administrator - Profile. This is the page where the administrator can view his/her own profile.

Figure 49: Administrator - Profile - Change Details. This is the page where the administrator can change his/her profile image and password.

Figure 50: Administrator - Report Overview. A summary regarding the number of tutorial sessions and simulated exams that were successfully concluded, and the total number of students served are shown in this page.

date range	reports
2016-05-01:2016-05-08	0
2016-05-9:2016-05-16	0
2016-05-17:2016-05-24	2
2016-05-25:2016-06-01	0

Figure 51: Administrator - Reports - Number of Students Served. This is where the administrator can generate reports regarding the number of students served within a given date range.

date range	reports
2016-05-01:2016-05-08	no reports
2016-05-9:2016-05-16	no reports
2016-05-17:2016-05-24	Abdulla, Abdel Jeffri Caeg Jr.-1#
2016-05-25:2016-06-01	no reports

Figure 52: Administrator - Reports - Most Frequent Students. This is where the administrator can generate reports regarding the students who avail the LRC services the most, within a given date range.

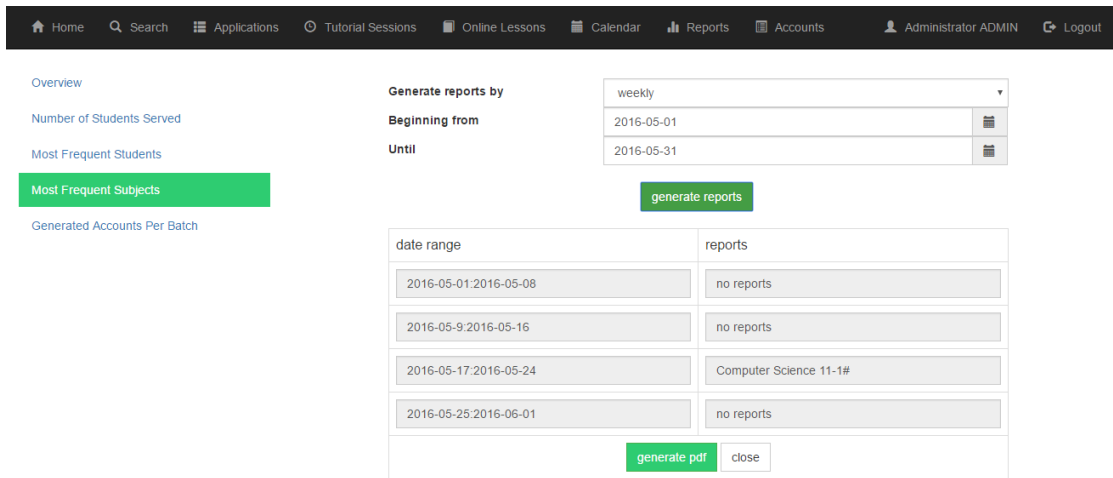


Figure 53: Administrator - Reports - Most Frequent Subjects. This is where the administrator can generate reports regarding the subjects frequently sought assistance for, within a given date range.

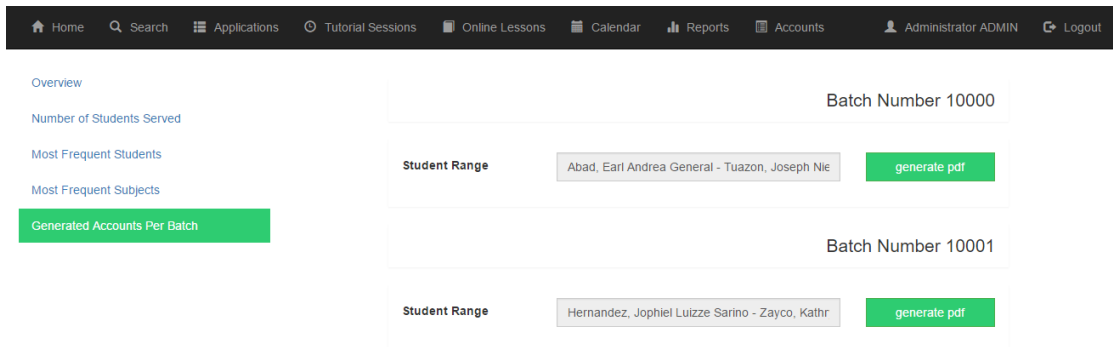


Figure 54: Administrator - Reports - Generated Accounts Per Batch. This is where the administrator can generate PDF files containing the list of generated accounts per batch.

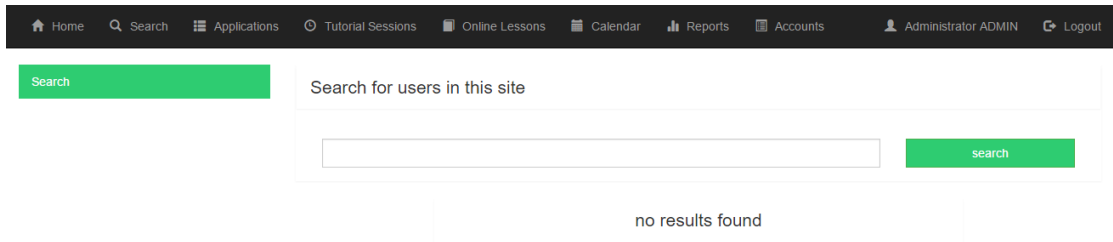


Figure 55: Administrator - Search. This is where the administrator can search for other users.

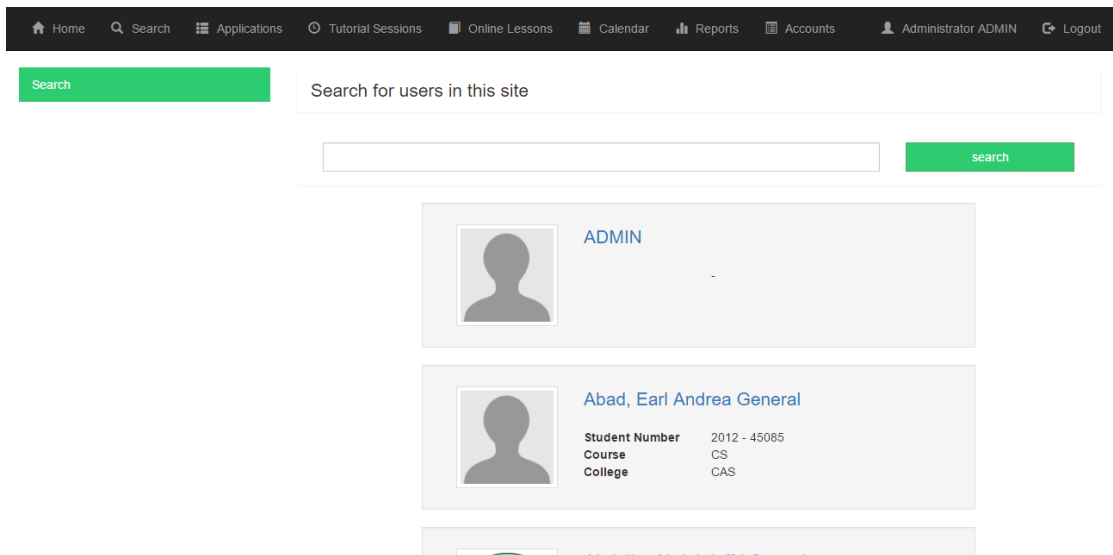


Figure 56: Administrator - Search Results. This is where the administrator views the search results. Clicking on the names of the users will redirect to their respective profile pages.

View Feedback

Viewing feedback for User 10035

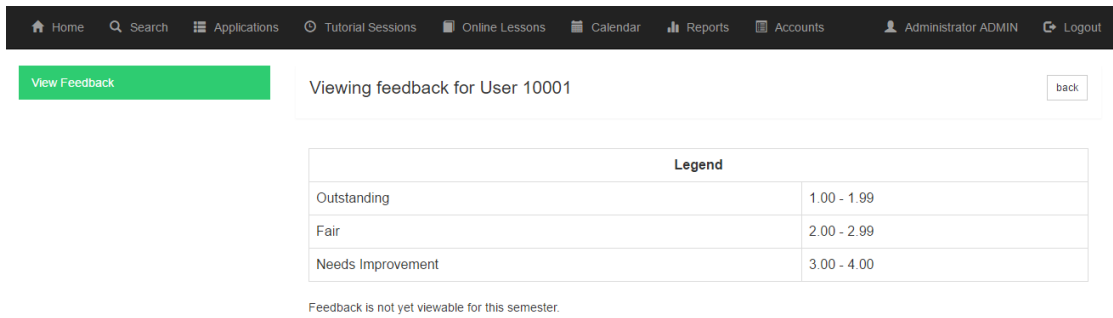
back

Legend	
Outstanding	1.00 - 1.99
Fair	2.00 - 2.99
Needs Improvement	3.00 - 4.00

Question	Score
The tutorial program has met my positive expectation.	2
The tutorial program helped me understand the topics in this course.	2
The LRC staff fulfilled my request in a timely manner.	2
I am satisfied with my tutorial.	2
I am more likely to recommend the LRC tutorial to others.	2
The tutor encouraged my interest in the course topic.	2
The tutor effectively managed the tutorial interaction.	2
The tutor developed good rapport with the tutee.	2
The tutor respects the tutees and responds to their needs.	2
The tutor's manner of teaching is easy to understand.	2
The tutor gave examples to help the tutee understand the topic.	2
The tutor felt comfortable in asking questions.	2
The tutor encouraged the tutee to participate actively during the tutorial.	2
The tutor was prompt and professional.	2
The tutor was well prepared for the tutorial.	2
<b>Overall Score</b>	<b>2</b>

Comments	
Positive comment #1	comments
Improvement area #1	comments

Figure 57: Administrator - View Feedback. The summarized feedback regarding a student-tutor is shown in this page.



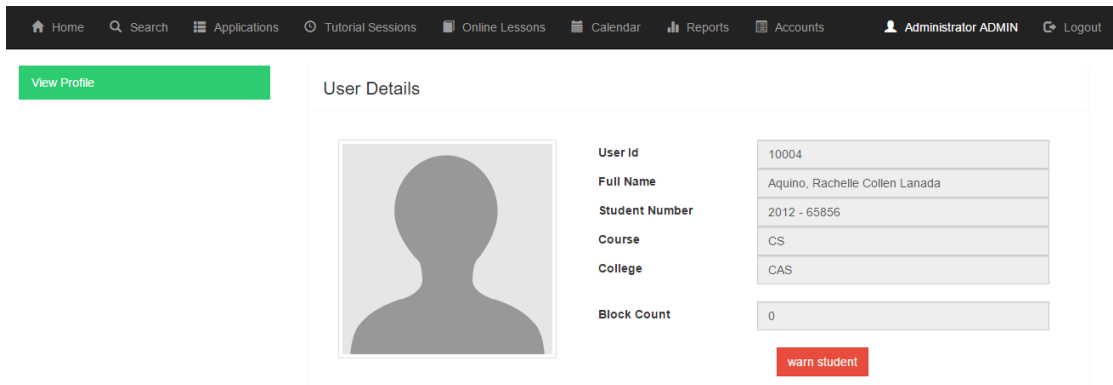
View Feedback

Viewing feedback for User 10001 back

Legend	
Outstanding	1.00 - 1.99
Fair	2.00 - 2.99
Needs Improvement	3.00 - 4.00


Feedback is not yet viewable for this semester.

Figure 58: Administrator - View Feedback. The summarized feedback regarding a student-tutor is shown in this page. This is the page shown instead when the feedback is not yet viewable.



View Profile

User Details



**User Id** 10004

**Full Name** Aquino, Rachele Collen Lanada

**Student Number** 2012 - 65856

**Course** CS

**College** CAS

**Block Count** 0

warn student

Figure 59: Administrator - View Student Profile. This is the page containing the details of a student who is not blocked from using the system. The administrator may block the student in this page.



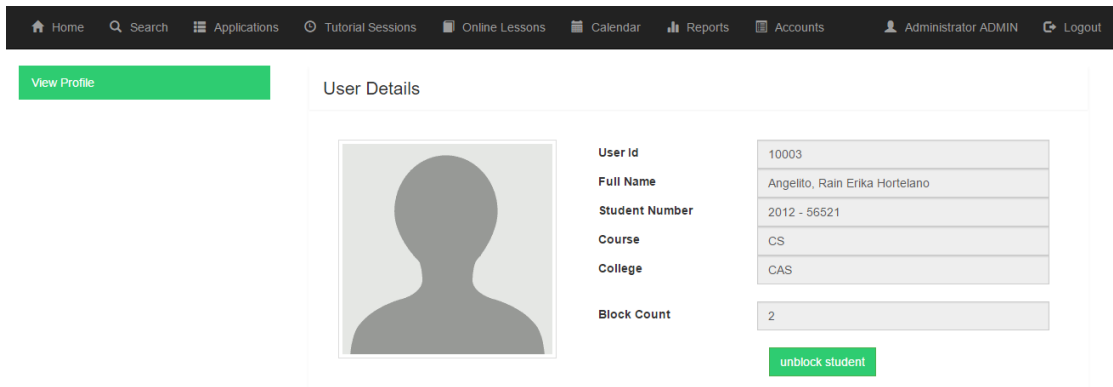


Figure 60: Administrator - View Student Profile. This is the page containing the details of a student who is blocked from using the system. The administrator may unblock the student in this page.

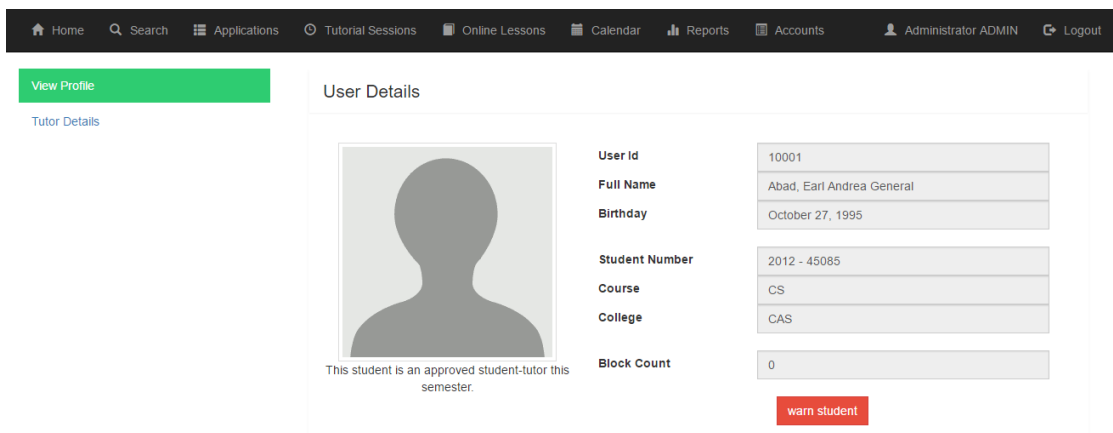


Figure 61: Administrator - View Student-Tutor Profile. This is the page containing the details of a student-tutor. The administrator may block or unblock the student in this page.

Home Search Applications Tutorial Sessions Online Lessons Calendar Reports Accounts Administrator ADMIN Logout

View Profile  
Tutor Details

### Tutor Details - Abad, Earl Andrea General

#### Tutor's Subject Preference

Rank 1 (most preferred)	Mathematics
Rank 2	Computer-Science
Rank 3	Communications
Rank 4	Physics
Rank 5	Chemistry
Rank 6 (least preferred)	Biology

#### User Schedule

available not available

Time	Monday	Tuesday	Wednesday	Thursday	Friday
8:30-10:30	available	available	available	not available	available
10:30-12:00	available	available	available	not available	available
1:00-2:30	available	not available	not available	not available	available
2:30-4:00	available	available	not available	not available	available
4:00-5:00	available	available	not available	available	available

Figure 62: Administrator - View Student-Tutor Details. This is the page containing the subject preference, and the schedule of a student-tutor.




**University of the Philippines  
Manila**  
**Learning Resource Center**  
2F Gusaling Joaquin Gonzales, Padre Faura St.  
Ermita, Manila 1000  
Landline: 336 5937 | Mobile No: 09279152834  
Email Address: <http://www.upm.edu.ph/lrc>




**List of all generated accounts by the Learning Resource Center Tutorial Management System**

Student Number	Name	Course	College	User Id
201245085	Abad, Earl Andrea General	CS	CAS	10001
201224745	Abdulla, Abdel Jeffri Caeg Jr.	CS	CAS	10002
201256521	Angelito, Rain Erika Hortelano	CS	CAS	10003
201265856	Aquino, Rachelle Collen Lanada	CS	CAS	10004
201243645	Arellano, Darlene Valdez	CS	CAS	10005
201201421	Arevalo, Adriel Sanchez	CS	CAS	10006
201213554	Arzadon, Christine Joyce Opena	CS	CAS	10007

Figure 63: PDF - List of Generated Accounts. This is the list of all generated accounts.



**University of the Philippines  
Manila**  
**Learning Resource Center**  
2F Gusaling Joaquin Gonzales, Padre Faura St.  
Ermita, Manila 1000  
Landline: 336 5937 | Mobile No: 09279152834  
Email Address: <http://www.upm.edu.ph/lrc>



**List of all generated accounts by the Learning Resource Center Tutorial Management System**

Student Number	Name	Course	College	User Id	Password
201245085	Abad, Earl Andrea General	CS	CAS	10001	
201224745	Abdulla, Abdel Jeffri Caeg Jr.	CS	CAS	10002	
201256521	Angelito, Rain Erika Hortelano	CS	CAS	10003	8b88aff
201265856	Aquino, Rachelle Collen Lanada	CS	CAS	10004	f672135
201243645	Arellano, Darlene Valdez	CS	CAS	10005	9fabdff
201201421	Arevalo, Adriel Sanchez	CS	CAS	10006	d8c41f8
201213554	Arzadon, Christine Joyce Opena	CS	CAS	10007	3e029c1

Figure 64: PDF - List of Generated Accounts, Per Batch. This is the list of generated accounts, per batch.

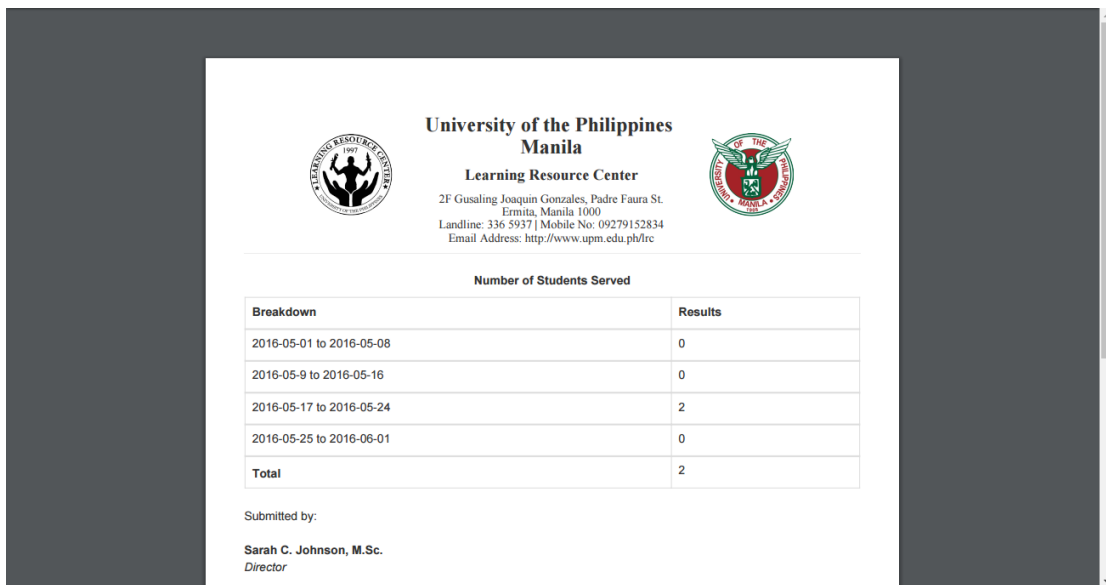


Figure 65: PDF - Report, Number of Students Served. This contains the report regarding the number of students served by the LRC, within a given date range.

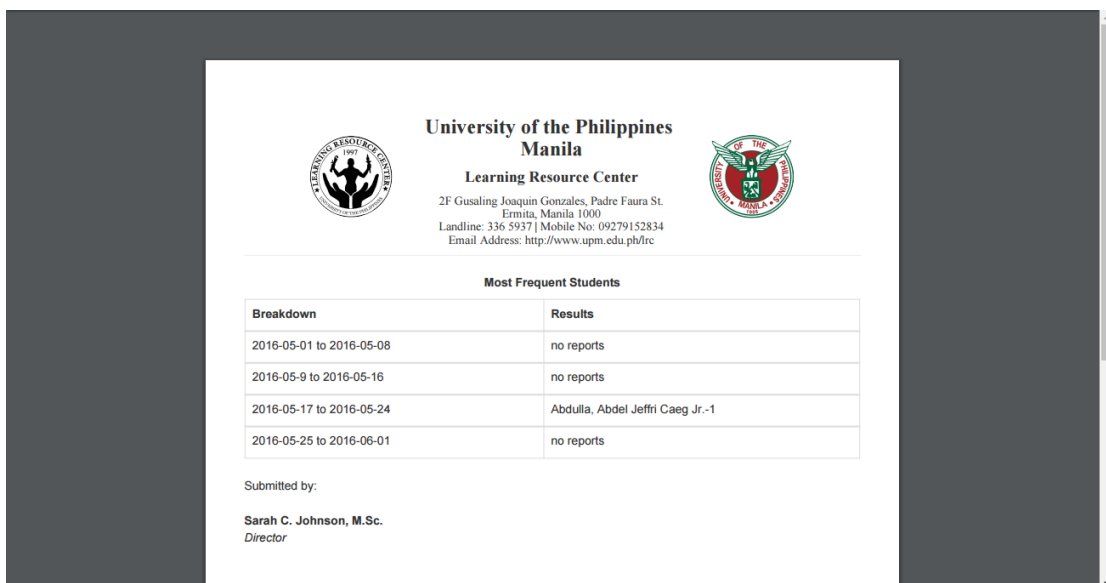


Figure 66: PDF - Report, Most Frequent Students. This contains the report regarding the students who avail the LRC services the most, within a given date range.

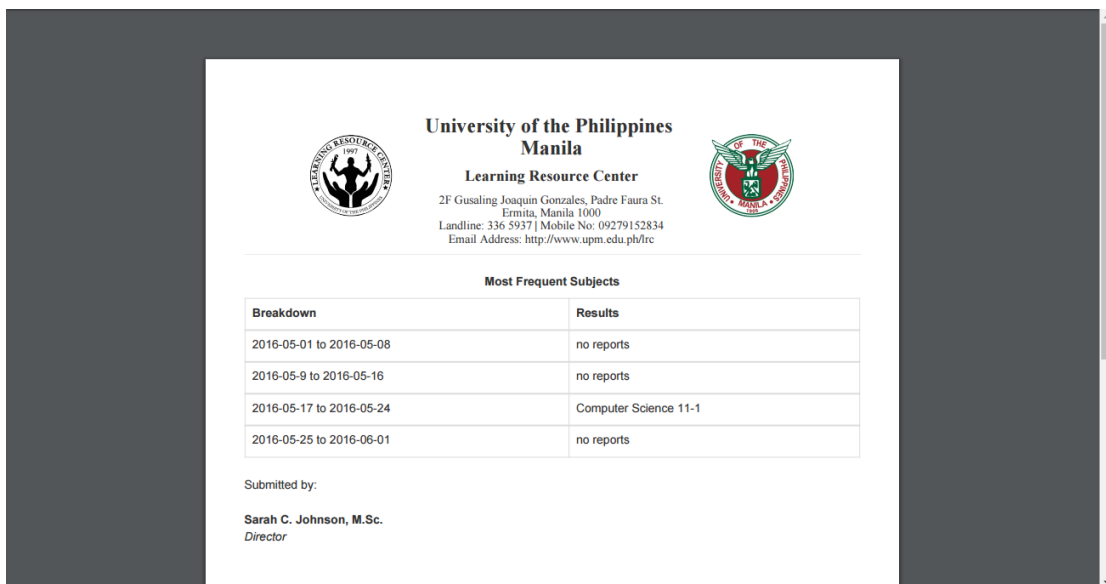


Figure 67: PDF - Report, Most Frequent Subjects. This contains the report regarding the subjects frequently sought assistance for, within a given date range.

## User: Student

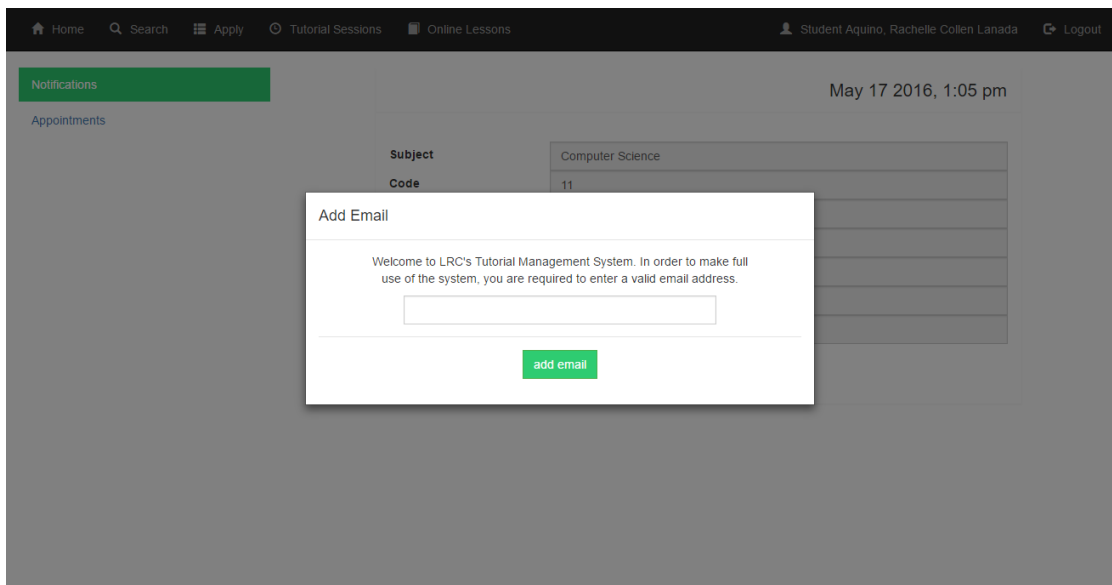


Figure 68: Student - Change Email. If it is the first time the student uses the system, a modal prompts the user to input a valid email address, so that the system can be fully utilized.

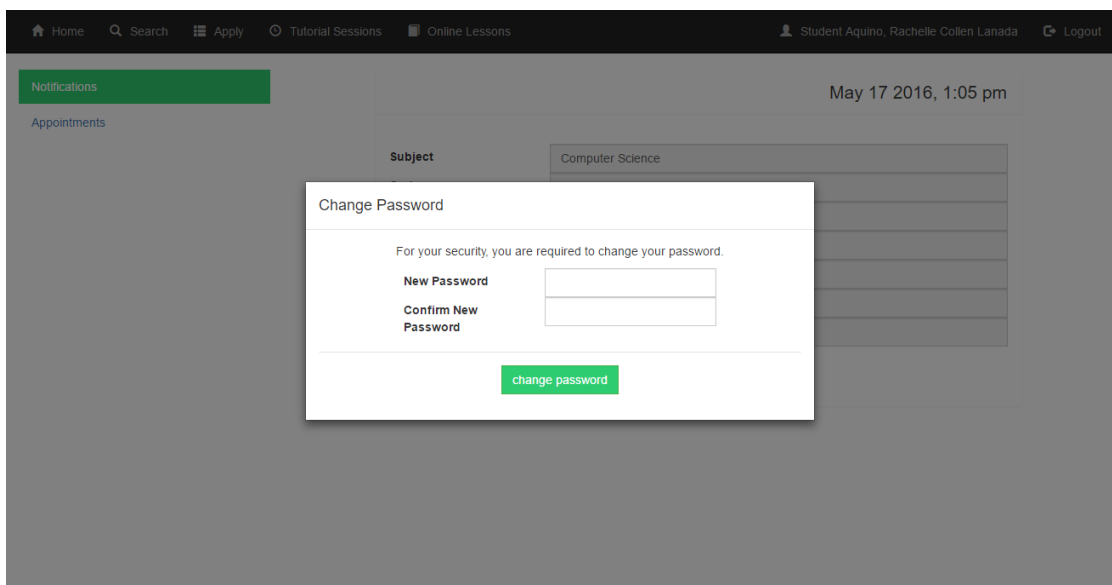


Figure 69: Student - Change Password. After adding the email address, the student is prompted to change his/her temporary password. This is also for additional security.

The screenshot shows the Student Dashboard with a navigation bar at the top containing Home, Search, Tutorial Sessions, View Feedback, Online Lessons, and a user profile for Student-tutor Abad, Earl Andrea General. The main content area is divided into two sections. On the left, under the 'Notifications' header, there is a sub-section for 'Appointments' which contains a 'Schedule' grid. The grid has columns for days of the week (M, T, W, T, F) and rows for time slots (8:30-10:30, 10:30-12:00, 1:00-2:30, 2:30-4:00, 4:00-5:00). Green squares indicate availability, and red squares indicate unavailability. A 'save schedule' button and an 'about' link are below the grid. A text box below the grid explains that this is where a student-tutor sets their availability. On the right, a notification card dated 'May 24 2016, 2:05 pm' displays details for a tutorial session: User Id (10004), Tutee Name (Aquino, Rachele Collen Lanada), Number of students (3), Subject (Mathematics), Code (14), Topic (Math topic), Date (2016-05-26), and Time (4:00-5:00). Below the details are two buttons: 'accept tutorial request' and 'reject tutorial request'. A second notification card dated 'May 17 2016, 1:05 pm' is partially visible below, showing the Subject as 'Computer Science'.

Figure 70: Student - Dashboard. All notifications meant for a student are shown here.

The screenshot shows the Student Dashboard with a navigation bar at the top containing Home, Search, Apply, Tutorial Sessions, Online Lessons, and a user profile for Student Aquino, Rachele Collen Lanada. The main content area is divided into two sections. On the left, under the 'Notifications' header, there is a sub-section for 'Appointments' which is highlighted in green. On the right, a notification card dated 'May 24 2016, 2:05 pm' displays details for a tutorial session: Subject (Mathematics 14), Tutorial Date (2016-05-26), and Tutorial Time (4:00-5:00). Below the details is a red button labeled 'cancel appointment'.

Figure 71: Student - Dashboard - Appointments. All the unfinished tutorial sessions created by a student are shown here. A student may cancel the appointment.

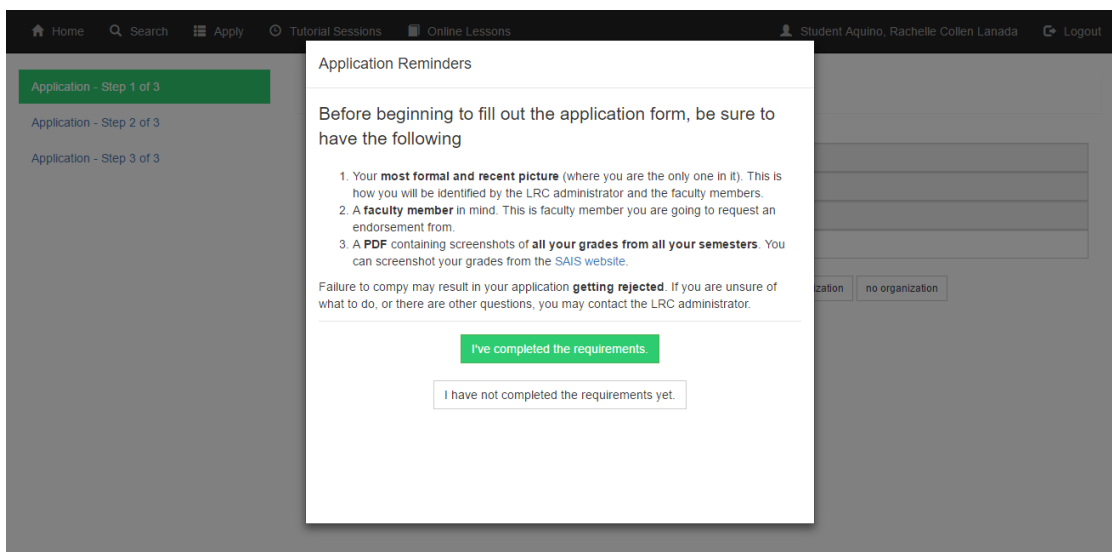


Figure 72: Student - Application Reminder. This modal shows the requirements a student needs in order to accomplish the application form. Clicking the upper button lets the user proceed to the application form. On the other hand, clicking the lower button redirects the user back to the dashboard.

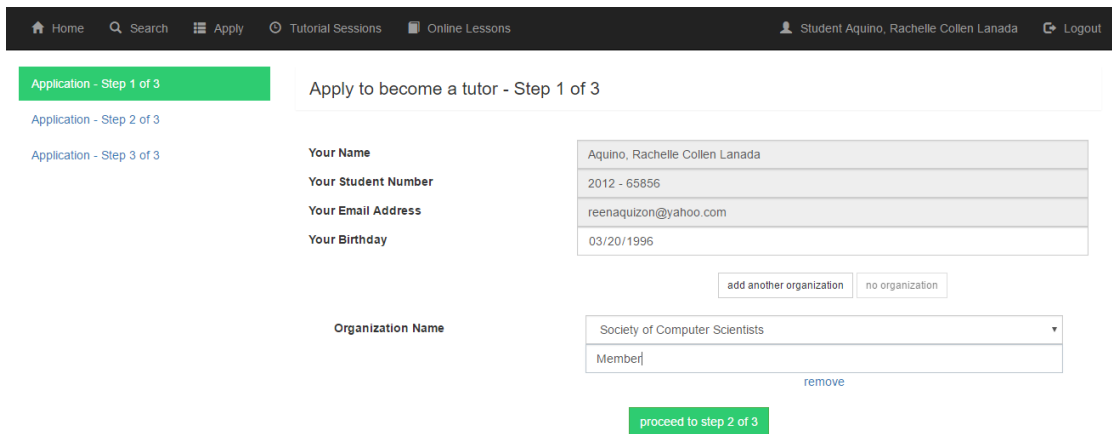


Figure 73: Student - Application - Step 1 of 3. This is the first page of the application form. This is where the student can input his/her date of birth, and the organizations he/she is currently a member of.



Figure 74: Student - Application - Step 2 of 3. This is the second page of the application form. This is where the student can rank the subjects he/she wants to teach. Also, this is where the student can upload his/her academic standing for all the semesters taken while in UP Manila.

Figure 75: Student - Application - Step 3 of 3. This is the third page of the application process. This is where the student can upload a profile image, input his/her free time schedule, and select which faculty member to request and endorsement from.

LRC [Home](#)

[Answer Feedback](#)

### Feedback regarding your last tutorial session

**Tutor's Name** Abad, Earl Andrea General

**Subject During Tutorial** Mathematics 14

**Tutorial Date** 2016-05-24

**Tutorial Time** 4:00-5:00

Question	Strongly agree	Agree	Disagree	Strongly disagree
The tutorial program has met my positive expectation.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The tutorial program helped me understand the topics in this course.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The LRC staff fulfilled my request in a timely manner.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I am satisfied with my tutorial.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I am more likely to recommend the LRC tutorial to others.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The tutor encouraged my interest in the course topic.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The tutor effectively managed the tutorial interaction.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The tutor developed good rapport with the tutee.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The tutor respects the tutees and responds to their needs.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The tutor's manner of teaching is easy to understand.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The tutor gave examples to help the tutee understand the topic.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The tutor felt comfortable in asking questions.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The tutor encouraged the tutee to participate actively during the tutorial.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The tutor was prompt and professional.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The tutor was well prepared for the tutorial.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Good points</b>	<input type="text"/>			
<b>Areas for improvement</b>	<input type="text"/>			

[submit feedback](#)

Figure 76: Student - Answer Feedback. If a student has recently attended a tutorial session, he/she is redirected to a feedback form upon logging in to the system. Answering the feedback form redirects the user back to the dashboard.

The screenshot shows a student portal interface. At the top, there is a navigation bar with links for Home, Search, Apply, Tutorial Sessions, and Online Lessons. The user is identified as 'Student Aquino, Rachele Collen Lanada' with a Logout option. On the left, a sidebar lists various subjects: Chemistry (highlighted in green), Mathematics, Physics, Biology, Computer Science, and Communications. The main content area is titled 'Chemistry subjects' and contains a table with the following entries:

Chemistry 14
Chemistry 18
Chemistry 27
Chemistry 31
Chemistry 40
Chemistry 156
Chemistry 157
NSI part

Figure 77: Student - Lesson Index. This page shows all subjects where online lesson content may be available.

The screenshot shows the 'View Lessons for Chemistry 14' page. The navigation bar is the same as in Figure 77. The sidebar now highlights 'View Lessons'. The main content area shows a 'back' button in the top right corner. Below it, the date and time 'May 15 2016, 10:05 pm' are displayed. The first lesson is 'Chemistry 14 Lesson 1', which includes a table with the following details:

Chemistry 14
Lesson Description

Below this, the date and time 'May 17 2016, 2:05 pm' are shown. The second lesson is 'Chemistry 14 Lesson 2', which includes a table with the following details:

Chemistry 14
lesson description

Figure 78: Student - View Lessons. This page shows all available online lessons for a specific subject.

View Mock Exam Details

[View Students](#)

### About this Mock Exam

<b>Subject</b>	Computer Science 11
<b>Date</b>	2016-05-19
<b>Time</b>	10:00:00
<b>Total Slots</b>	5
<b>Available Slots</b>	4
<b>Discussion Schedule</b>	2016-05-23
	12:30:00
<b>Additional Details</b>	details

[apply for mock exam](#)

Figure 79: Student - Mock Exam - View Details. The student can view details and apply for a simulated exam in this page. If the student has already applied, he/she can cancel his/her slot instead.

[View Mock Exam Details](#)

[View Students](#)

### View Students

Student Number	Name
2012 - 45085	<a href="#">Abad, Earl Andrea General</a>

Figure 80: Student - Mock Exam - View Students. The student can view the list of students who applied for a simulated exam in this page.

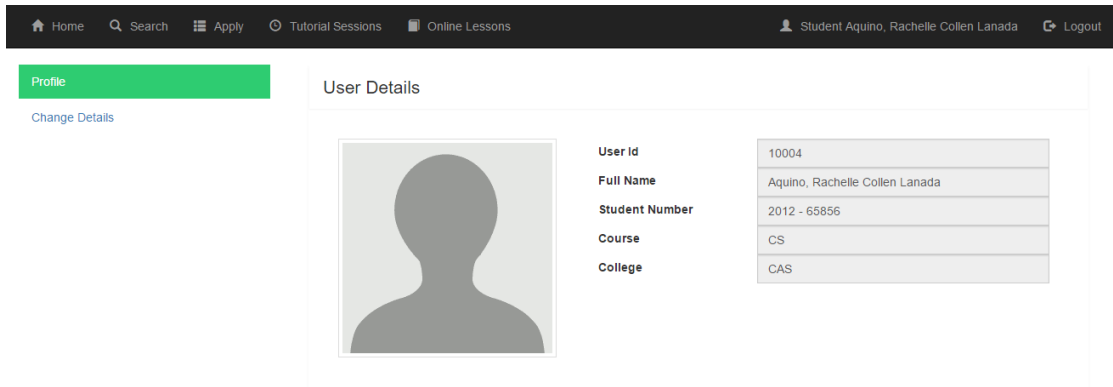


Figure 81: Student - Profile. This is the page where the student can view his/her own profile.

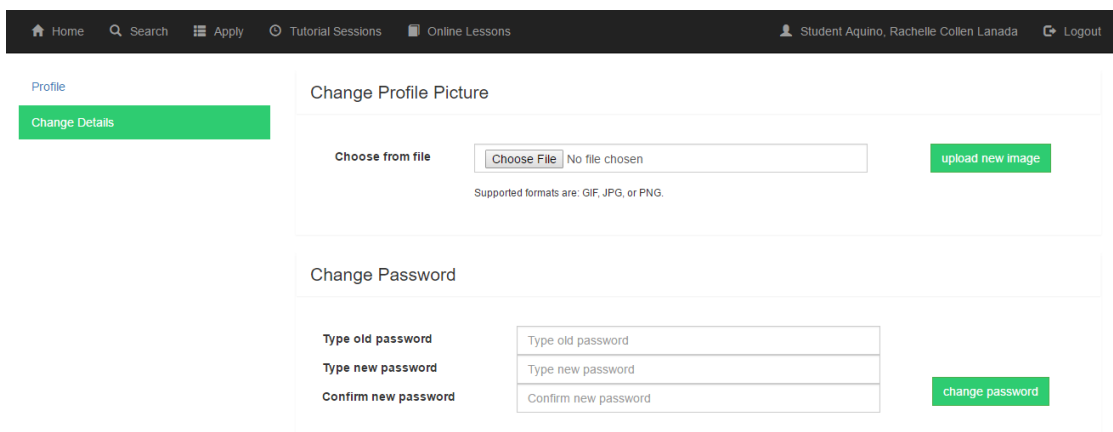


Figure 82: Student - Profile - Change Details. This is the page where the student can change his/her profile image and password.

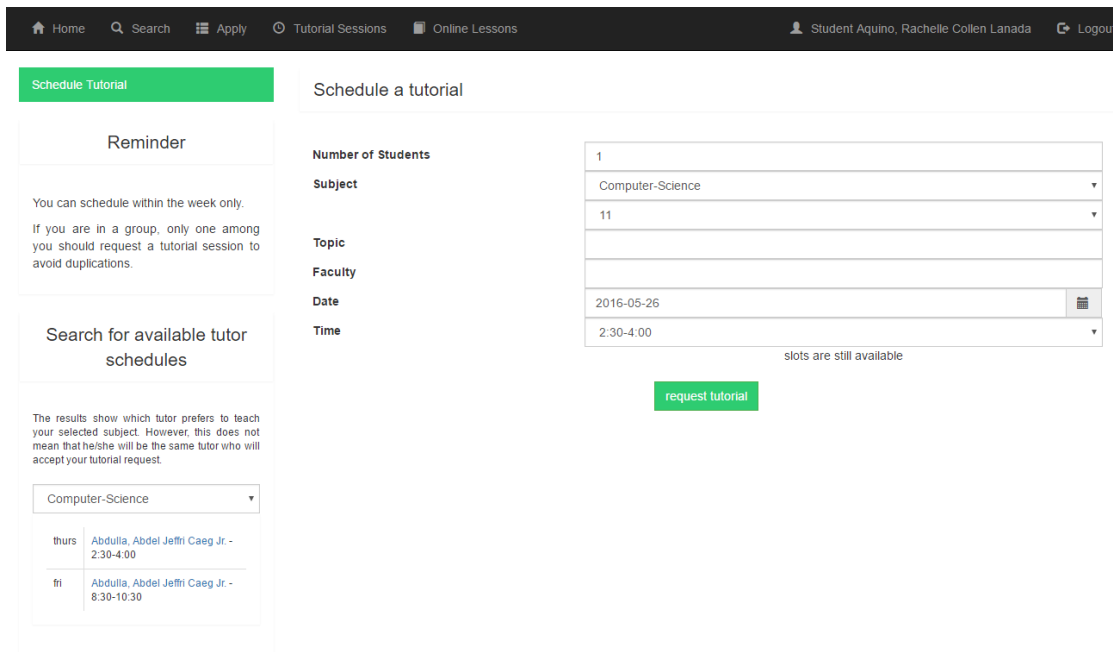


Figure 83: Student - Schedule Tutorial. This is where the student can schedule a tutorial session. At the left side of the screen, the user can select a subject; the list of available tutors who prefer to teach the subject, and their schedules, will appear.

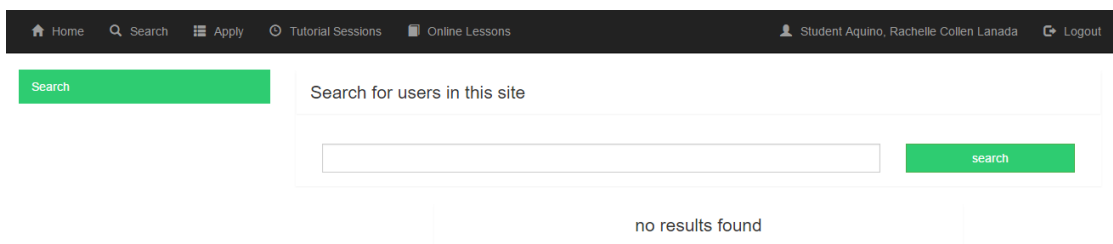


Figure 84: Student - Search. This is where the student can search for other users.

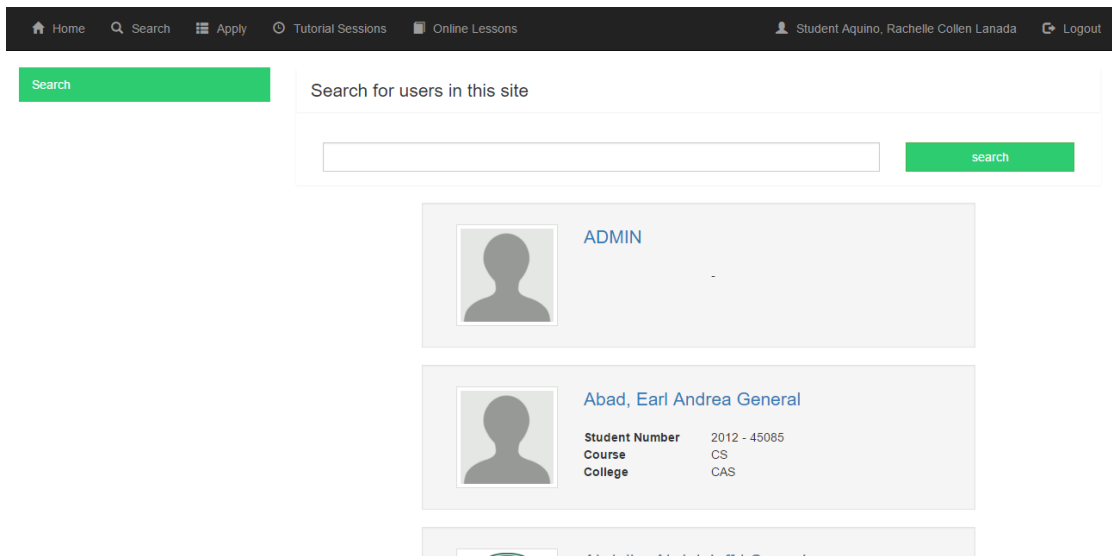


Figure 85: Student - Search Results. This is where the student views the search results. Clicking on the names of the users will redirect to their respective profile pages.

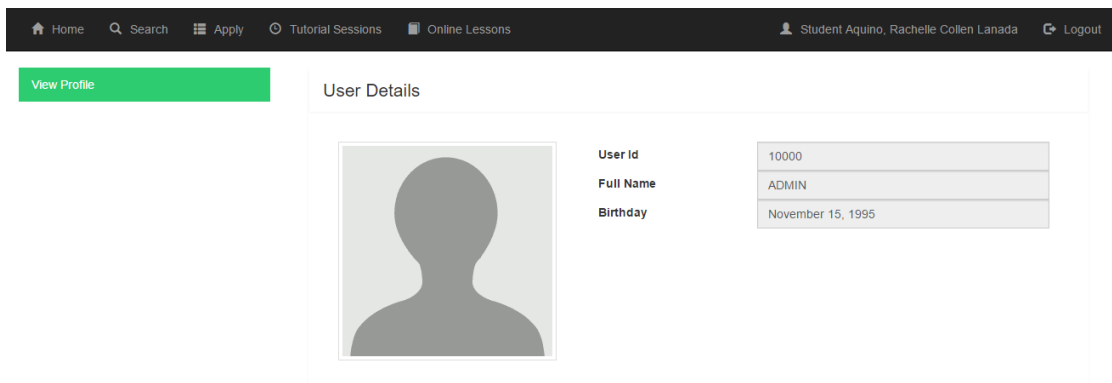


Figure 86: Student - View Administrator Profile. This is where the student can view the administrator's profile.

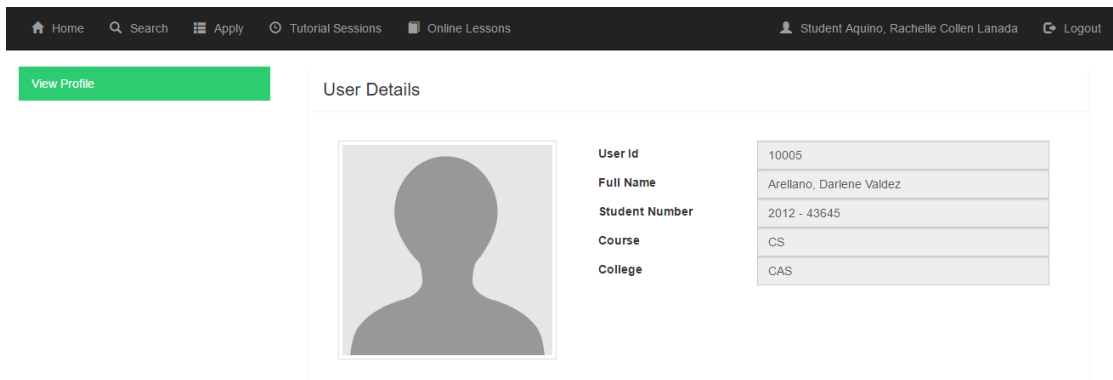


Figure 87: Student - View Student Profile. This is the page containing the details of a student.

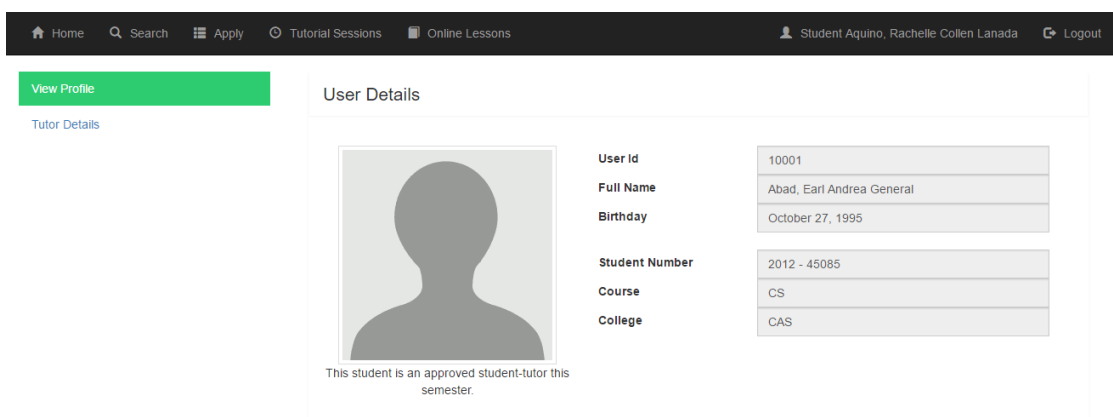


Figure 88: Student - View Student-Tutor Profile. This is the page containing the details of a student-tutor.



Home Search Apply Tutorial Sessions Online Lessons Student Aquino, Rachele Collen Lanada Logout

View Profile  
Tutor Details

### Tutor Details - Abad, Earl Andrea General

#### Tutor's Subject Preference

Rank 1 (most preferred)	Mathematics
Rank 2	Computer-Science
Rank 3	Communications
Rank 4	Physics
Rank 5	Chemistry
Rank 6 (least preferred)	Biology

#### User Schedule

available not available

Time	Monday	Tuesday	Wednesday	Thursday	Friday
8:30-10:30	available	available	available	not available	available
10:30-12:00	available	available	available	not available	available
1:00-2:30	available	not available	not available	not available	available
2:30-4:00	available	available	not available	not available	available
4:00-5:00	available	available	not available	available	available

Figure 89: Student - View Student-Tutor Details. This is the page containing the subject preference, and the schedule of a student-tutor.

## User: Student-tutor

The dashboard shows a navigation bar with links for Home, Search, Tutorial Sessions, View Feedback, and Online Lessons. The user is identified as 'Student-tutor Abad, Earl Andrea General'. The 'Notifications' section is active, displaying a 'Schedule' grid and a list of notifications.

**Schedule Grid:**

Time	M	T	W	T	F
8:30-10:30	available	available	available	not available	available
10:30-12:00	available	available	available	not available	available
1:00-2:30	available	not available	not available	not available	available
2:30-4:00	available	available	not available	not available	available
4:00-5:00	available	available	available	available	available

**Notifications:**

- Notification 1 (May 24 2016, 2:05 pm):**
  - User Id: 10004
  - Tutee Name: Aquino, Rachelle Collen Lanada
  - Number of students: 3
  - Subject: Mathematics
  - Code: 14
  - Topic: Math topic
  - Date: 2016-05-26
  - Time: 4:00-5:00
  - Actions: [accept tutorial request](#) [reject tutorial request](#)
- Notification 2 (May 17 2016, 1:05 pm):**
  - Subject: Computer Science
  - Code: 11

Figure 90: Student-Tutor - Dashboard. All notifications meant for a student-tutor are shown here.

The 'Lesson Index' page features a navigation bar and a 'Chemistry' section. The 'Chemistry subjects' list includes the following items:

- Chemistry 14
- Chemistry 18
- Chemistry 27
- Chemistry 31
- Chemistry 40
- Chemistry 156
- Chemistry 157
- NSI part

Figure 91: Student-Tutor - Lesson Index. This page shows all subjects where online lesson content may be available.

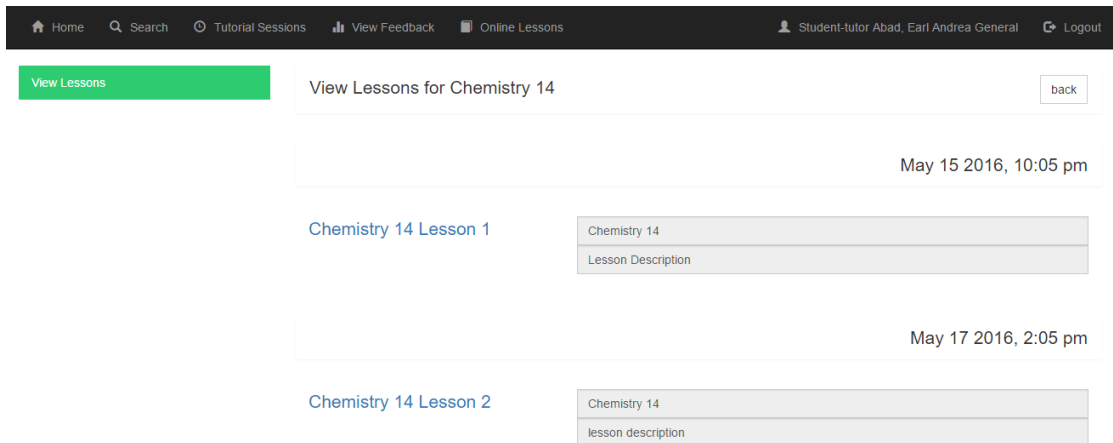


Figure 92: Student-Tutor - View Lessons. This page shows all available online lessons for a specific subject.

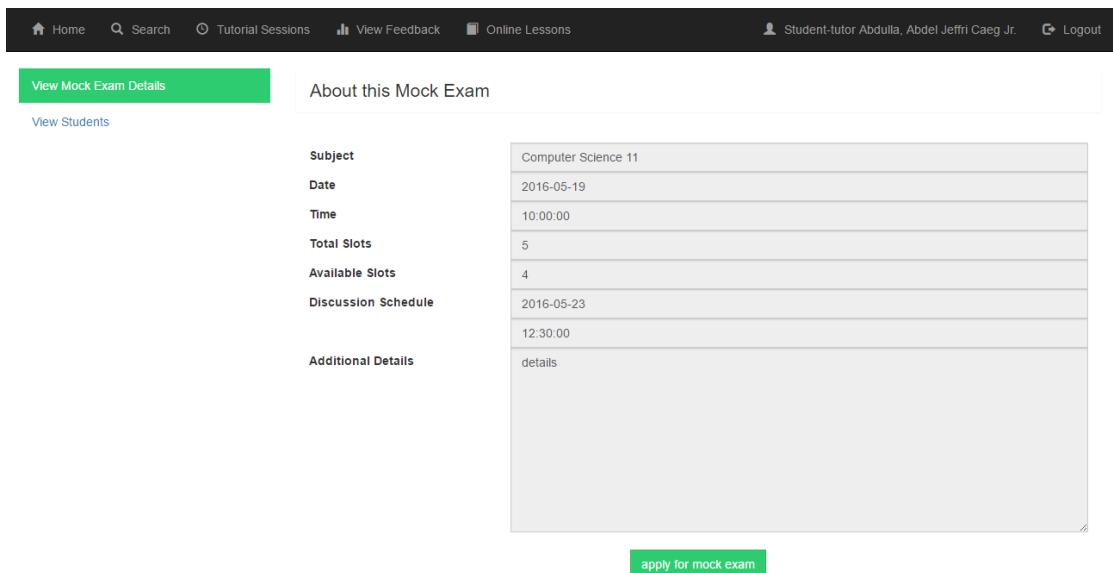


Figure 93: Student-Tutor - Mock Exam - View Details. The student-tutor can view details and apply for a simulated exam in this page. If the student-tutor has already applied, he/she can cancel his/her slot instead.

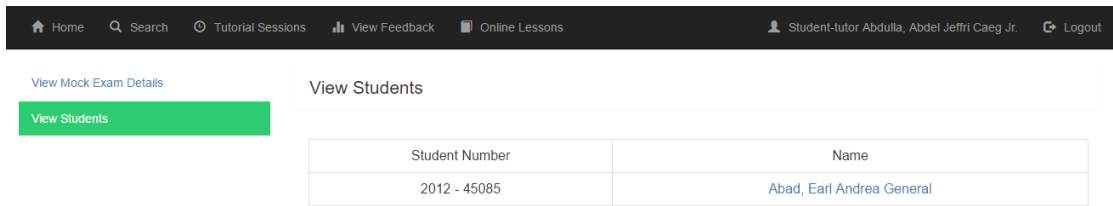


Figure 94: Student-Tutor - Mock Exam - View Students. The student-tutor can view the list of students who applied for a simulated exam in this page.

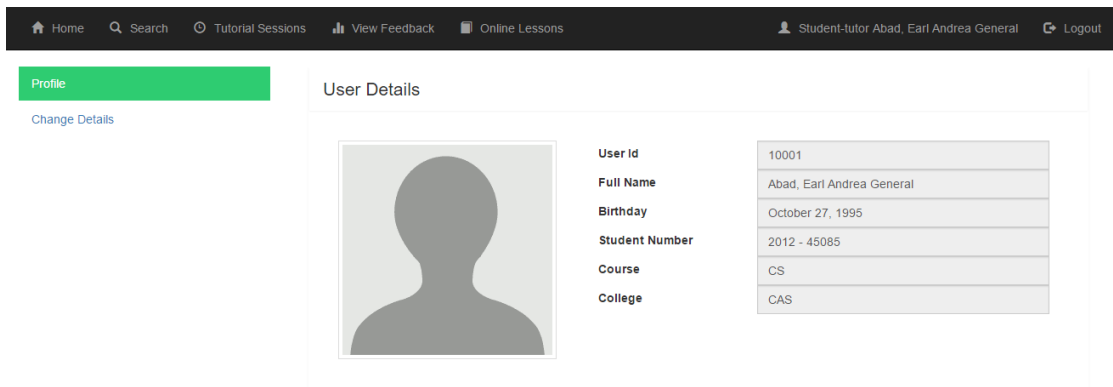


Figure 95: Student-Tutor - Profile. This is the page where the student-tutor can view his/her own profile.

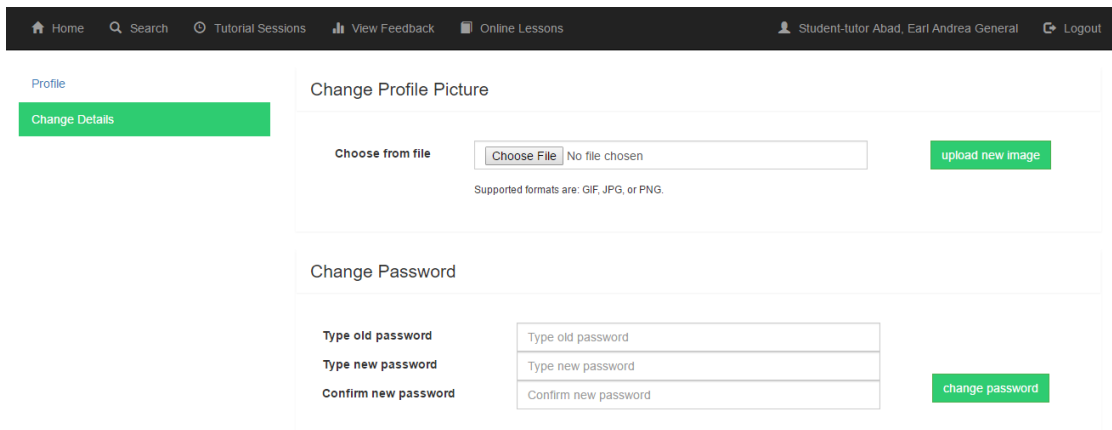


Figure 96: Student-Tutor - Profile - Change Details. This is the page where the student-tutor can change his/her profile image and password.

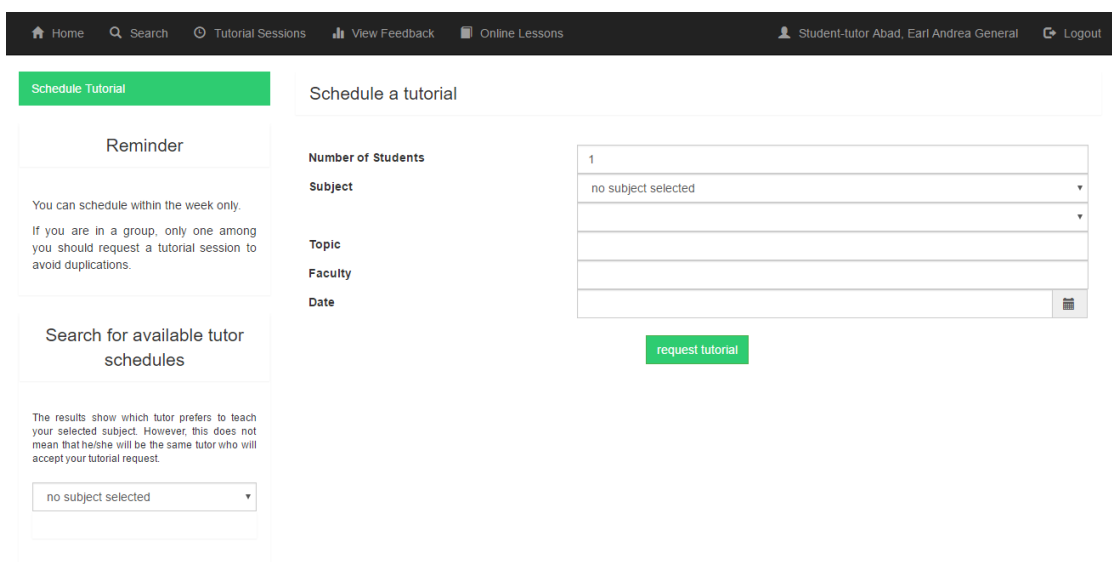


Figure 97: Student-Tutor - Schedule Tutorial. This is where the student-tutor can schedule a tutorial session. At the left side of the screen, the user can select a subject; the list of available tutors who prefer to teach the subject, and their schedules, will appear.

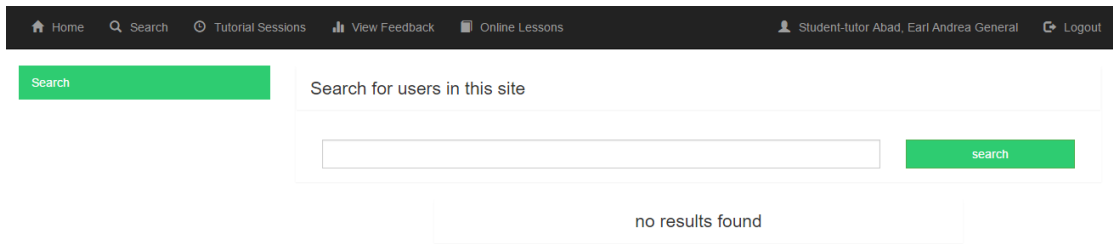


Figure 98: Student-Tutor - Search. This is where the student-tutor can search for other users.

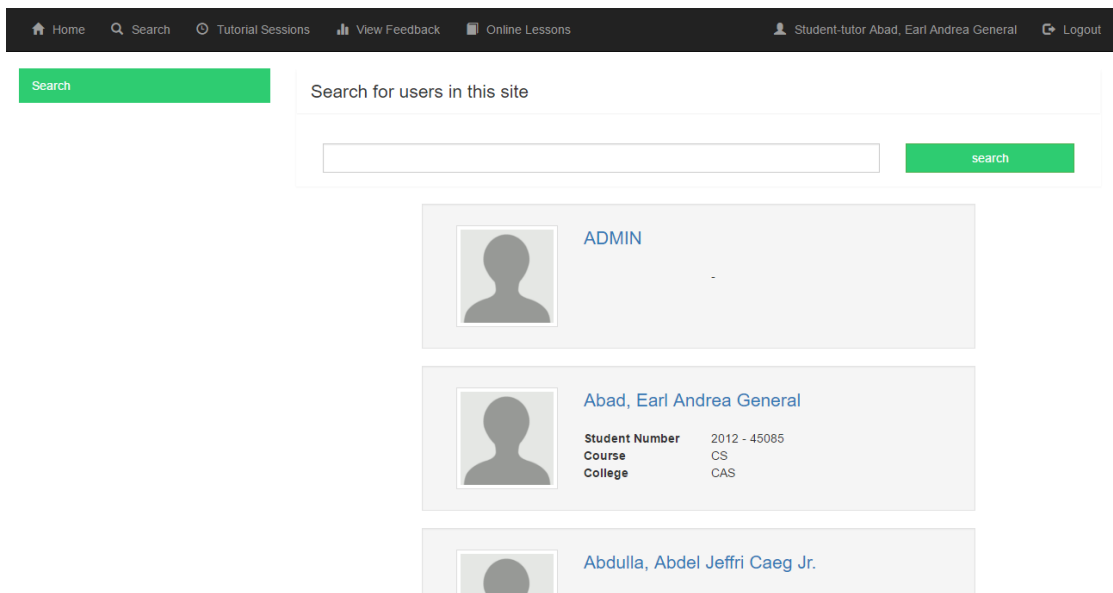


Figure 99: Student-Tutor - Search Results. This is where the student-tutor views the search results. Clicking on the names of the users will redirect to their respective profile pages.

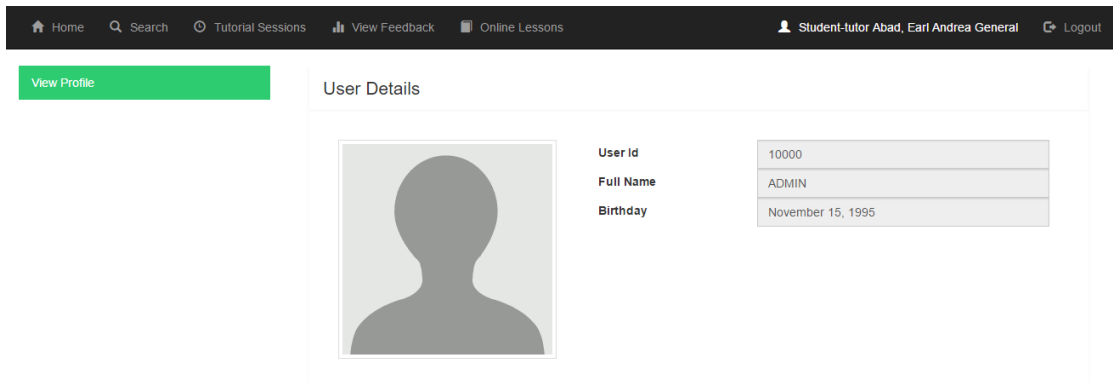


Figure 100: Student-Tutor - View Administrator Profile. This is where the student-tutor can view the administrator's profile.

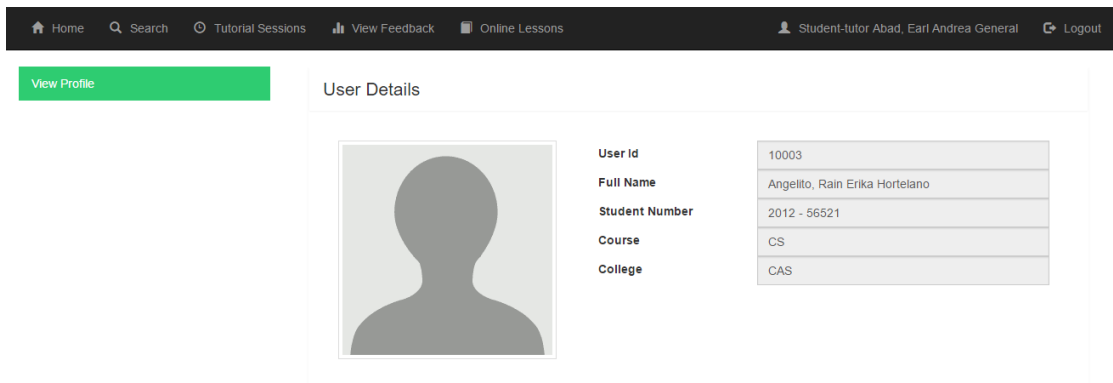
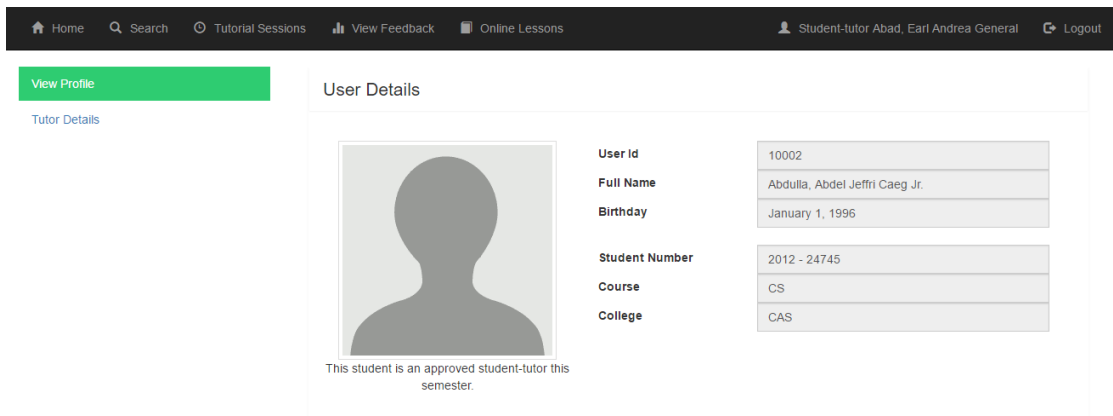


Figure 101: Student-Tutor - View Student Profile. This is the page containing the details of a student.



Home Search Tutorial Sessions View Feedback Online Lessons Student-tutor Abad, Earl Andrea General Logout

View Profile  
Tutor Details

### User Details

**User Id**: 10002

**Full Name**: Abdulla, Abdel Jeffri Caeg Jr.

**Birthday**: January 1, 1996

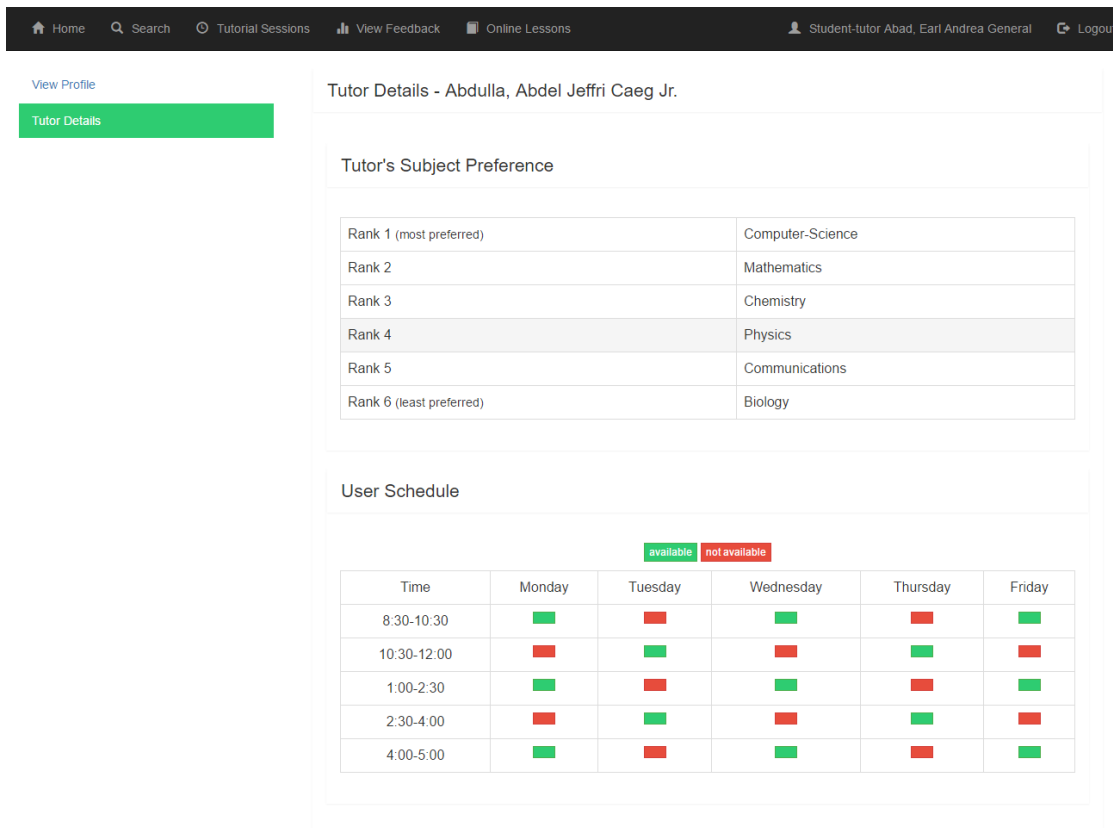
**Student Number**: 2012 - 24745

**Course**: CS

**College**: CAS

This student is an approved student-tutor this semester.

Figure 102: Student-Tutor - View Student-Tutor Profile. This is the page containing the details of a student-tutor.



Home Search Tutorial Sessions View Feedback Online Lessons Student-tutor Abad, Earl Andrea General Logout

View Profile  
Tutor Details

### Tutor Details - Abdulla, Abdel Jeffri Caeg Jr.

#### Tutor's Subject Preference

Rank 1 (most preferred)	Computer-Science
Rank 2	Mathematics
Rank 3	Chemistry
Rank 4	Physics
Rank 5	Communications
Rank 6 (least preferred)	Biology

#### User Schedule

Time	available		not available			
	Monday	Tuesday	Wednesday	Thursday	Friday	
8:30-10:30	available	not available	available	not available	available	
10:30-12:00	not available	available	not available	available	not available	
1:00-2:30	available	not available	available	not available	available	
2:30-4:00	not available	available	not available	available	not available	
4:00-5:00	available	not available	available	not available	available	

Figure 103: Student-Tutor - View Student-Tutor Details. This is the page containing the subject preference, and the schedule of a student-tutor.



## User: Faculty

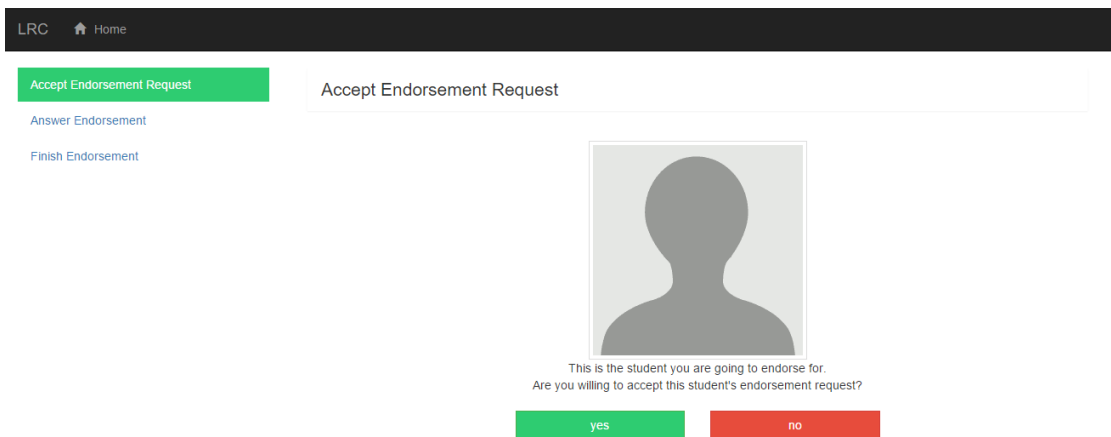


Figure 104: Faculty - Accept Endorsement. This is where the faculty member is redirected to upon clicking the link in the email regarding the endorsement request. This is also where he/she can decide to accept or reject the endorsement request of a student.

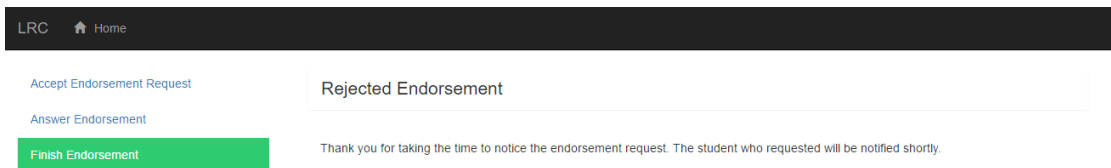



Figure 105: Faculty - Reject Endorsement. This is the page where the faculty member is redirected to upon rejecting the endorsement request of a student. The student is notified upon rejection of the endorsement.

LRC Home

Accept Endorsement Request

**Answer Endorsement**

Finish Endorsement



**Student's Name** Aquino, Rachele Collen Lanada

**Student Number** 2012 - 65856

Characteristic	Poor (1)	Fair (2)	Average (3)	Above average (4)	Superior (5)
Knowledge of subject matter	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Verbal communication skills	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Ability to work with other people	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Potential to teach others	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Dependability	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>

**Subjects Taken Under You**

**Subject**

Computer Science	▼
11	
1.75	▼

**Attendance** no absences and no tardiness

**Your Recommendation** I strongly recommend the applicant as peer tutor

**Additional Comments**

Additional comments

Figure 106: Faculty - Answer Endorsement. This is the page where the faculty member rates the student through the endorsement form, if the faculty accepts the request.

LRC Home

Accept Endorsement Request

Answer Endorsement

**Finish Endorsement**

Thank you for taking the time to answer the endorsement request. The student who requested will be notified shortly.

Figure 107: Faculty - Finish Endorsement. This is the page where the faculty member is redirected to upon finishing the endorsement of a student. The student is notified upon completion of the endorsement.

## VI. Discussions

The University of the Philippines Manila Learning Resource Center Tutorial Management System (UPM-LRC-TMS) is a web-based system for scheduling tutorial sessions, filing/renewing student-tutor applications, and creating/accessing online lessons in PDF format. Only the LRC administrator and the students of the university have accounts; the administrator generates the accounts. Guests can only view the main page of the website. If the user is a faculty member whom an endorsement is requested from, the faculty member can also view the endorsement-related pages. Registered users are the administrator, the student, and the student-tutor.

All users can access their own profiles. The details that are changeable in the profile are their profile images, their birthdays, and their passwords. Using the search functionality in the website enables all users to access each other's profiles. When the administrator views a student or a student-tutor account, there is an option to block the user in their profile pages.

The user is prompted to input his/her own email address the first time he/she logs in the system. If the email address is legitimate, he/she will be able to login to the system and be prompted to change the temporary password for additional security.

In the login page, if ever a user forgets the password to his/her account, a link is provided where he/she can enter the User ID and the email address associated with the account. If the User Id-email combination is correct, an email will be sent to the user, containing the temporary password. Upon login, the user will be prompted to immediately change the password, also for security purposes. If all of these are done, the user will be redirected to the dashboard, where all notifications are posted. If ever the user is a student or a student-tutor, he/she can also view the tutorial sessions that are currently scheduled.

In the system's main dashboard page is a side panel containing a schedule. The schedule should indicate the time slots when the LRC has available rooms. The administrator must update this as much as needed to prevent conflicts for all users. There is also one for each student-tutor, but for the student-tutors, editing their schedules will affect which tutorial requests they are going to receive, as the schedule indicates the time slots they are free.

The system can only have one administrator. The administrator is in charge of managing the accounts, either by creating accounts, blocking or unblocking student users from using the system, resetting all student-tutors to regular student accounts, or generating a PDF file containing the list of all student accounts. If a student or student-tutor is blocked from using the system, the user will receive an email about the event. However, the user can still request to the administrator to be unblocked from using the system. To prevent being blocked, the user has to attend the tutorial sessions he/she scheduled, unless they are cancelled.

Regarding the tutorial sessions, the administrator will be notified if there is a new tutorial schedule. He/she can check the attendance of the student/s on the day of the tutorial session, or cancel the tutorial if ever the need arises (e.g.: a tutor informed the administrator that he/she cannot attend the tutorial session). If ever the student-tutee did not attend the tutorial, the administrator can immediately flag the tutorial session as unsuccessful and warn/block the user. The records about past tutorial sessions can also be viewed.

When students file/renew their applications to become a student-tutor, the administrator is immediately notified. Only the administrator can view the application form and the endorsement form of the interested student. The details provided in the forms serve as basis for the decision of the administrator to approve or reject the application. Records regarding past applications can also be viewed.

While student-users can access the content of the online lessons, the administrator is the only one who can upload content for the online lessons as well. The file type of the lesson content is restricted to PDF. The details of a lesson can be edited or deleted by the administrator at any given time.

The system is heavily date and time-based, meaning, certain actions can only be performed on certain dates. Such actions are as follows: scheduling of tutorial sessions, the viewing of feedback regarding a student-tutor, and the generation of reports. It is imperative for the administrator to set the system's academic calendar at once. The events page can also be used for LRC-related announcements, especially when there are simulated exams scheduled. All events set in the system's academic calendar can be edited, marked as finished, or deleted.

For faster generation of reports, there is also a page where the administrator can select the date range of the reports and the type of the report to be generated. The generated report is table containing the breakdown of the results. For convenience, there is an option to export the results to a formatted PDF file.

Student and student-tutors can file or renew their applications to become a student-tutor. Filing an application requires them to also request an endorsement from a faculty member. Only faculty members with known emails can be requested from. The faculty member has the right to accept or reject an endorsement request, especially if he/she personally knows the requesting student. If ever the request is accepted, the faculty can rate the student. The details of the endorsement form is confidential between the faculty member and the administrator. On the other hand, if the request is rejected, the student can change the faculty member to request an endorsement from, or cancel the application.

Both student and student-tutors can schedule tutorial sessions. However, only student-tutors can accept or reject tutorial requests. Both types of accounts can submit feedback regarding their tutor if ever they attended a tutorial session.

Student-tutors and administrators cannot view the summarized feedback unless it is the end of the semester, as indicated in the system's academic calendar.

If ever there are upcoming simulated examinations, all student and student-tutors are notified of its details. They can decide whether they want to apply for the exam, or, if ever they have already applied, they can also decide to cancel their slot. All users can view the list of students who applied for the exam. The administrator can cancel the event, or flag the event as finished.

## VII. Conclusions

The University of the Philippines-Manila Learning Resource Center Tutorial Management System (UPM-LRC-TMS) is able to provide a systematic and automated approach to several operations normally done at the LRC.

Scheduling tutorial sessions, summarizing and viewing of feedback for a student-tutor, filing/renewing student-tutor applications and requesting endorsements, announcing LRC-related events (especially simulated examinations), providing on-line lessons, generating accounts, and generating reports.

Students can now schedule tutorial requests online; the administrator and tutors will be automatically notified of these requests. The tutors can quickly decide whether to accept or reject them. The administrator can check whether the students are attending their scheduled tutorial sessions. To solve the problem of forgetting to answer feedback regarding the tutor, the student-user is redirected to answering the feedback form upon a return login to the system. Student-tutors can view their feedback once the semester ends.

If ever there are no student-tutors available, online lessons are available. The content is provided by the administrator. The files can easily be saved, as they are all in PDF format.

For the students interested in becoming a student-tutor, they can now file their applications online. If ever they need a faculty member for the endorsement, the system can notify the faculty member through email. This is helpful, especially if the faculty member is not physically present in the university premises. The administrator can decide to approve or reject the application immediately.

Announcements regarding the LRC, especially simulated examination events, can be easily disseminated to all users. Students or student-tutors can also apply

for simulated examinations, as well as cancel their slots if ever they applied already. They will also be notified about the details of the after-exam discussions.

Given the population of the students of the University of the Philippines-Manila, it is tedious to create accounts for all students. To automate this operation, the administrator can upload an MS Excel file containing relevant details of the students. The file is provided by the Office of the University Registrar every academic year. In order to be able to provide the details of the accounts to the students, the system can generate a PDF file listing all generated accounts. There is still an option for the administrator to create only one account.

Every semester, the LRC generates reports declaring the number of students that were served by the LRC. The system, besides being able to generate reports on how many students availed the LRC tutorial sessions and simulated examinations, can also generate reports on which student avails these services the most, and which subjects the students frequently seek assistance for. All reports can be exported to a PDF file.



## VIII. Recommendations

The University of the Philippines-Manila Learning Resource Center Tutorial Management System (UPM-LRC-TMS) provides a portal where students can efficiently avail online the services being offered by the LRC. LRC administrators are also benefited from this, since they can now easily track LRC-related operations that were normally done manually. However, this system is not a substitute for face-to-face interaction, especially when it comes to tutorial sessions.

Inasmuch as the dashboard's notification system is informative, there is a need to improve the customization of the messages. It is recommended that the messages should suit more specifically to a user when they are displayed (i.e. notifying the user by name). The search function of the system can also be extended to find not only users, but also online lessons and LRC-related events.

The PDF format of the online lessons allow easier access to the students, but the LRC director requests that the online lessons be expanded to be able to upload tutorial videos. This is because there is a lack of student-tutors currently. The list of subjects that the system recognizes is limited, there should also be a functionality where the administrator can add or remove subjects and the course codes under them.

As of now, the current method of setting the academic calendar is quite tedious. Since PDF files of the university's academic calendar are available online, it would be better to implement within the system a parser for PDF files. The administrator, instead of entering dates one by one, can just upload the PDF file of the academic calendar for the current year, and let the system extract and set important dates.

The current format for generating reports is via table. To improve visualization, it is also recommended to implement charting for all reports. The charts

should also be exportable to PDF format. Another type of report should be implemented, which is determining which faculty member is most referred to when it comes to filling up tutorial session requests. This is to determine who among the faculty members the students have a difficult time with. Therefore, there should also be a list of faculty members and their email addresses. The administrator should be able to update the list, especially if there are newly hired faculty members.

All events done in the system occur whenever a user interacts with it. There is no automated way of maintaining the database, especially when there is a limited number of available User Ids to be allocated. In future improvements of the system, there should be an automated maintenance period, where the system checks and blocks inactive accounts, and resets the schedules of all users. The database should be refreshed periodically, and the notifications table should be truncated every beginning of the semester. This is to prevent overloading the users with unnecessary records. Instead of the administrator reverting the student-tutor to student accounts every end of the semester, the system should automatically do it.

## IX. Bibliography

- [1] “What is tutoring.” <http://ltcconline.net/green1/courses/TutorTraining/whatIsTutoring.htm>. Accessed: 2015-11-28.
- [2] “What is an online tutorial?.” [http://www.bristol.ac.uk/esu/media/tutorials/design-principles/page\\_02.htm](http://www.bristol.ac.uk/esu/media/tutorials/design-principles/page_02.htm). Accessed: 2015-12-03.
- [3] “Why create online tutorials?.” [http://www.bristol.ac.uk/esu/media/tutorials/design-principles/page\\_03.htm](http://www.bristol.ac.uk/esu/media/tutorials/design-principles/page_03.htm). Accessed: 2015-12-03.
- [4] “Learning resource center — university of the philippines manila.” <http://www.upm.edu.ph/node/256>. Accessed: 2015-11-28.
- [5] “About us — learning resource center.” <http://lrc.upm.edu.ph/?q=node/6>. Accessed: 2015-11-28.
- [6] “Services — learning resource center.” <http://lrc.upm.edu.ph/?q=node/33>. Accessed: 2015-11-28.
- [7] “Manual versus automated appointment scheduling.” [https://www.appointment-plus.com/pdf/resources/manual\\_vs\\_automated\\_appointment\\_scheduling.pdf](https://www.appointment-plus.com/pdf/resources/manual_vs_automated_appointment_scheduling.pdf). Accessed: 2015-11-28.
- [8] E. Shafie, A. Al-Ajlan, K. Aldrawiesh, A. Bajahzar, and Y. Al-Saawy, “On-line registration system,” in *Advanced Computer Control (ICACC), 2011 3rd International Conference on*, pp. 174–179, Jan 2011.
- [9] Y. Peng, N. Liu, Y. Li, and Z. Shao, “Design and implementation of the online course registration system at tsinghua university,” in *Systems and Informatics (ICSAI), 2012 International Conference on*, pp. 1179–1182, May 2012.
- [10] M. M. A. Wahed, “Requirements for establishing an information system to manage issuing building permits,” *HBRC Journal*, 2015.

- [11] H. Iwata, S. Tsumoto, and S. Hirano, “Clinical schedule management based on granularity-based mining,” in *Cybernetics (CYBCONF), 2013 IEEE International Conference on*, pp. 237–242, June 2013.
- [12] N. Togashi and V. Klyuev, “A novel approach for web development: A schedule management system using gae/go,” in *Awareness Science and Technology (iCAST), 2015 IEEE 7th International Conference on*, pp. 55–59, Sept 2015.
- [13] N. Cavus, “Distance learning and learning management systems,” *Procedia - Social and Behavioral Sciences*, vol. 191, pp. 872 – 877, 2015. The Proceedings of 6th World Conference on Educational Sciences.
- [14] J. Kozusko, A. Abdel-Haq, J. Kus, U. Morgenstern, and A. Bartels, “Enhanced laboratory exercises for biomedical engineering courses,” in *Education and e-Learning Innovations (ICEELI), 2012 International Conference on*, pp. 1–6, July 2012.
- [15] P. Koncz, A. Lukacova, and J. Paralic, “Course web site as an integrated solution for e-learning, collaboration and publicly available knowledge base,” in *Emerging eLearning Technologies Applications (ICETA), 2012 IEEE 10th International Conference on*, pp. 197–202, Nov 2012.
- [16] M. Llorens, E. Nevin, and E. Mageean, “Work in progress: Online resource platform for mathematics education,” in *Information Systems and Technologies (CISTI), 2014 9th Iberian Conference on*, pp. 1–4, June 2014.
- [17] V. Karavirta, R. Haavisto, E. Kaila, M.-J. Laakso, T. Rajala, and T. Salakoski, “Interactive learning content for introductory computer science course using the ville exercise framework,” in *Learning and Teaching in Computing and Engineering (LaTiCE), 2015 International Conference on*, pp. 9–16, April 2015.
- [18] “What are information systems? - definition and types - video and lesson transcript — study.com.” <http://study.com/academy/lesson/>

- `what-are-information-systems-definition-types-quiz.html`. Accessed: 2015-11-29.
- [19] “Information system.” <http://www.britannica.com/topic/information-system>. Accessed: 2015-11-29.
- [20] “What is database management system (dbms)? - definition from whatis.com.” <http://searchsqlserver.techtarget.com/definition/database-management-system>. Accessed: 2015-11-29.
- [21] “What is a database management system? - purpose and function - video and lesson transcript — study.com.” <http://study.com/academy/lesson/what-is-a-database-management-system-purpose-and-function.html>. Accessed: 2015-11-29.
- [22] “What is model-view-controller (mvc) - definition from whatis.com.” <http://whatis.techtarget.com/definition/model-view-controller-MVC>. Accessed: 2015-11-29.

# X. Appendix

## Application.php

```
<?php if ( ! defined("BASEPATH")) exit("No
direct script access allowed");

/**
 * @author Quizon, Maria Reena I.
 * filename: Application.php
 * description: Controller for the application
           pages in LRC_TMS
 * start date: Feburary 5, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

class Application extends CI_Controller{
function __construct(){
    parent::__construct();

    $this->load->helper("form");
    $this->load->helper("url");

    $this->load->library("form_validation");
    $this->load->library("Email_Controller");
    $this->load->library("
Notification_Controller");
    $this->load->library("Validation_Controller
");

    $this->load->model("General_Model");
    $this->load->model("Application_Model");

    $this->validation_controller->
check_validation();
}

public function index($msg = NULL, $action =
NULL) {

    $data["msg"] = $msg;
    $data["action"] = $action;

    $this->load->view("templates/navigation");

    if ($action == NULL || $action ==
APPLICATION_INCOMPLETE_1) {
        $this->load->view("apply/apply_1", $data)
    ;
    }
    elseif ($action == APPLICATION_PROCEED_1) {
        $this->load->view("apply/apply_2", $data)
    ;
    }
    elseif ($action == APPLICATION_PROCEED_2 ||
$action == EMAIL_FAILED) {
        $userId = $this->session->userdata("
userId");
        $profilePicture = $this->session->
userdata("profilePicture");
        $schedules = $this->General_Model->
get_user_entry("schedule", $userId);

        $path = base_url("./uploads/".$userId."/".
"profilePicture");
        $hasPicture = 0; if (isset(
"profilePicture")) { $hasPicture = 1; }

        $data["path"] = $path;
        $data["hasPicture"] = $hasPicture;
        $data["schedules"] = $schedules;
        $data["schedule"] = $this->load->view("
templates/schedule", $data, TRUE);
        $this->load->view("apply/apply_3", $data)
    ;
    }
    elseif ($action == APPLICATION_PROCEED_3) {
        redirect(base_url("/dashboard"));
    }
}
}
```

```
public function process_application_1() {

    $msg = DEFAULT_STRING;
    $action = DEFAULT_STRING;

    $status = EVENT_PENDING;
    $applicationType = APPLICATION_FIRST;
    $applicationId = $this->General_Model->
get_tableId("application");

    if ($this->session->userdata("approvalCount
") > DEFAULT_INTEGER) {
        $applicationType = APPLICATION_RENEWAL;
    }

    $organizations = implode("#", $this->input
->post("organizations"));
    $positions = implode("#", $this->input->
post("positions"));

    $userData = array(
        "userId" => $this->session->userdata("
userId"),
        "birthday" => $this->input->post("
birthday")
    );
    $applicationData = array(
        "applicationId" => $applicationId,
        "applicationType" => $applicationType,
        "organizations" => $organizations,
        "positions" => $positions,
        "status" => $status
    );

    $this->General_Model->update_table_entry("
user", $userData);
    $this->session->set_userdata(
$applicationData);

    $action = APPLICATION_PROCEED_1;
    $this->index($msg, $action);
}

public function process_application_2() {

    $msg = DEFAULT_STRING;
    $action = DEFAULT_STRING;
    $academicStanding = DEFAULT_STRING;

    $path = "./uploads/".$this->session->
userdata("userId");

    if(!is_dir($path)) {
        mkdir($path, 0755, TRUE);
    }

    $config = array(
        "upload_path" => $path,
        "allowed_types" => "pdf",
        "max_size" => "2048000",
        "overwrite" => TRUE
    );

    $this->load->library("upload", $config);

    $applicationData = array();
    for ($counter = 1; $counter < 7; $counter
++) {
        ${"subjectRank_". $counter} = $this->input
->post("subjectRank_". $counter);
        ${"subjectRank_". $counter."_taken"} =
implode("#", $this->input->post("
subjectRank_". $counter."_taken"));
        $applicationData = array_merge(
$applicationData, array("subjectRank_".
$counter => ${"subjectRank_". $counter}, "
subjectRank_". $counter."_taken" => ${"
subjectRank_". $counter."_taken"}));
    }
}
```

```

if (!$this->upload->do_upload("
academicStanding")) {
    $academicStanding = NULL;
}
else {
    $academicStanding = $this->upload->data("
file_name");
}

$applicationData = array_merge(
$applicationData, array("academicStanding"
=> $academicStanding));
$this->session->set_userdata(
$applicationData);

$action = APPLICATION_PROCEED_2;

$this->index($msg, $action);
}

public function process_application_3() {

$msg = DEFAULT_STRING;
$action = DEFAULT_STRING;
$profilePicture = DEFAULT_STRING;

$hasSchedule = EVENT_ACTIVE;
$hasApplication = EVENT_ACTIVE;
$userId = $this->session->userdata("userId"
);

$path = "./uploads/".$userId;

if(!is_dir($path)) {
    mkdir($path, 0755, TRUE);
}

$config = array(
    "upload_path" => $path,
    "allowed_types" => "gif|jpg|png",
    "max_size" => "2048000",
    "overwrite" => TRUE
);

$this->load->library("upload", $config);

$hasPicture = $this->input->post("
hasPicture");
if ($hasPicture == DEFAULT_INTEGER) {
    if (!$this->upload->do_upload("
profilePicture")) {
        $profilePicture = NULL;
    }
    else {
        $profilePicture = $this->upload->data("
file_name");
    }
}
else {
    $profilePicture = $this->input->post("
profilePicture");
}

$applicationData = array(
    "applicationId" => $this->session->
userdata("applicationId"),
    "userId" => $this->session->userdata("
userId"),
    "applicationType" => $this->session->
userdata("applicationType"),
    "organizations" => $this->session->
userdata("organizations"),
    "positions" => $this->session->userdata("
positions"),
    "faculty" => $this->input->post("faculty"
),
    "applicationStatus" => $this->session->
userdata("status"),
    "academicStanding" => $this->session->
userdata("academicStanding"),
);
$scheduleData = array(
    "userId" => $userId,
    "hasSchedule" => $hasSchedule
);

$userData = array(
    "userId" => $userId,
    "profilePicture" => $profilePicture,
    "hasApplication" => $hasApplication
);

$dayArray = array("mon", "tues", "wed", "
thurs", "fri");
$timeArray = array("0", "0830_1030", "1030
_1200", "0100_0230", "0230_0400", "0400
_0500");

for ($counter = 1; $counter < 7; $counter
++) {
    $applicationData = array_merge(
$applicationData, array("subjectRank_" .
$counter => $this->session->userdata("
subjectRank_" . $counter));
    $applicationData = array_merge(
$applicationData, array("subjectRank_" .
$counter . "_taken" => $this->session->
userdata("subjectRank_" . $counter . "_taken"
)));
}

for ($row = 0; $row < 5; $row++) {
    for ($column = 0; $column < 6; $column++)
    {
        if ($column > 0) {
            $scheduleData = array_merge(
$scheduleData, array($dayArray[$row] . "_" .
$timeArray[$column] => $this->input->post(
$dayArray[$row] . "_" . $timeArray[$column]));
        }
    }
}

$emailType = EMAIL_ENDORSEMENT;
$email = $this->input->post("faculty");

$this->session->set_userdata(array("
hasApplication" => $hasApplication));
$this->session->set_userdata(array("
profilePicture" => $profilePicture));

$this->General_Model->update_table_entry("
schedule", $scheduleData);
$this->General_Model->update_table_entry("
user", $userData);

$endorsementLink = $this->email_controller
->create_email($email, $emailType);
if (!$endorsementLink) {
    $msg = EMAIL_FAILED;
    $action = EMAIL_FAILED;
}
else {
    $status = EVENT_PENDING;
    $notificationDetails = DEFAULT_STRING;
    $applicationId = $this->session->userdata
("applicationId");
    $endorsementId = $this->General_Model->
get_tableId("endorsement");

    $endorsementData = array(
        "userId" => $userId,
        "applicationId" => $applicationId,
        "endorsementId" => $endorsementId,
        "endorsementLink" => $endorsementLink,
        "endorsementStatus" => $status
    );

    if ($this->session->userdata("
applicationType") == 0) {
        $notificationDetails =
APPLICATION_NEW_MSG;
    }
    else {
        $notificationDetails =
APPLICATION_RENEWAL_MSG;
    }

    $notificationData = $this->
notification_controller->

```

```

create_notification(APPLICATION_NEW,
$applicationId, ADMIN_ID,
$notificationDetails);
$this->General_Model->create_table_entry(
"notification", $notificationData);
$this->General_Model->create_table_entry(
"application", $applicationData);
$this->General_Model->create_table_entry(
"endorsement", $endorsementData);

$msg = EMAIL_SUCCESSFUL;
$action = APPLICATION_PROCEED_3;
}
$this->index($msg, $action);
}

public function get_faculty() {
$xml = simplexml_load_file(base_url("/
application/views/apply/faculty.xml"));
$facultyString = "<br/><div class = 'col
col-xs-6'>";
$facultyString .= "<select name = 'faculty'
class = 'form-control'>";
foreach ($xml->faculty as $faculty) {
$facultyString .= "<option value = '".
$faculty->email.">".$faculty->name."</
option>";
}
$facultyString .= "</select></div>";
$facultyString .= "<div class = 'col col-xs
-6'>";
$facultyString .= "<input type = 'submit'
name = 'change' class = 'btn btn-success'
value = 'change faculty'/'>";
$facultyString .= "<button class = 'cancel
btn btn-default'>close</button></div>";
echo json_encode($facultyString);
}

public function change_application() {
$msg = DEFAULT_STRING;
$applicationId = DEFAULT_INTEGER;
$endorsementId = $this->uri->segment(3);
$userId = $this->session->userdata("userId"
);
$applications = $this->Application_Model->
get_application($endorsementId);

foreach ($applications as $application) {
$applicationId = $application->
applicationId;
}

if (isset($_POST["change"])) {
$hasAccess = EVENT_ACTIVE;
$emailType = EMAIL_ENDORSEMENT;

$userIds = $userId;
$email = $this->input->post("faculty");

$endorsementLink = $this->
email_controller->create_email($email,
$emailType);
if (!$endorsementLink) {
$msg = EMAIL_FAILED;
}
else {
$endorsementStatus = EVENT_PENDING;
$applicationData = array(
"applicationId" => $applicationId,
"faculty" => $email
);
$endorsementData = array(
"endorsementId" => $endorsementId,
"endorsementLink" => $endorsementLink
,
"endorsementStatus" =>
$endorsementStatus,
"hasAccess" => $hasAccess
);
}
}

$notificationData_1 = $this->
notification_controller->
create_notification(
APPLICATION_CHANGED_STUDENT, $applicationId
, $userIds, APPLICATION_CHANGED_STUDENT_MSG
);
$notificationData_2 = $this->
notification_controller->
update_notification(FACULTY_REJECTED,
$endorsementId, $userIds, EVENT_INACTIVE);

$this->General_Model->
update_table_entry("application",
$applicationData);
$this->General_Model->
update_table_entry("endorsement",
$endorsementData);
$this->General_Model->
update_table_entry("notification",
$notificationData_2);
$this->General_Model->
create_table_entry("notification",
$notificationData_1);

$msg = EMAIL_SUCCESSFUL;
}
if (isset($_POST["cancel"])) {
$applicationStatus = EVENT_INACTIVE;
$adminId = ADMIN_ID;
$userIds_1 = $userId."#".$adminId;
$userIds_2 = $userId;

$applicationData = array(
"applicationId" => $applicationId,
"applicationStatus" =>
$applicationStatus
);
$userData = array(
"userId" => $userId,
"hasApplication" => $applicationStatus
);

$notificationData_1 = $this->
notification_controller->
create_notification(
APPLICATION_CANCELLED_STUDENT,
$applicationId, $userIds_1,
APPLICATION_CANCELLED_STUDENT_MSG);
$notificationData_2 = $this->
notification_controller->
update_notification(FACULTY_REJECTED,
$endorsementId, $userIds_2, EVENT_INACTIVE)
;

$this->General_Model->update_table_entry(
"user", $userData);
$this->General_Model->update_table_entry(
"application", $applicationData);
$this->General_Model->update_table_entry(
"notification", $notificationData_2);
$this->General_Model->create_table_entry(
"notification", $notificationData_1);
$this->session->set_userdata("
hasApplication", $applicationStatus);
}

redirect(base_url("/dashboard"));
}
?>

```

## Dashboard.php - Application

```

<?php if ( ! defined("BASEPATH")) exit("No
direct script access allowed");

```

```

/**

```



```

* @author Quizon, Maria Reena I.
* filename: Dashboard.php
* description: Controller for the application
  dashboard page in LRC_TMS
* start date: February 25, 2016
* end date:
* last updated:
* version number: 1.0
*/

class Dashboard extends CI_Controller {
function __construct(){
parent::__construct();

$this->load->helper("url");
$this->load->library("form_validation");
$this->load->library("Notification_Controller");
$this->load->library("Validation_Controller");
};

$this->load->model("General_Model");
$this->load->model("Application_Model");

$this->validation_controller->
check_validation();
}

public function index($msg = NULL) {

$data["msg"] = $msg;
$this->load->view("templates/navigation");
$applications = $this->Application_Model->
get_applications();
$endorsements = $this->General_Model->
get_table("endorsement");

if ($applications != DEFAULT_INTEGER) {
$data["applications"] = $applications;
}

if ($endorsements != DEFAULT_INTEGER) {
$data["endorsements"] = $endorsements;
}

$this->load->view("apply/dashboard", $data);
}

public function get_application() {

$userId = DEFAULT_INTEGER;
$applicationId = $_GET["applicationId"];
$applications = $this->General_Model->
get_table_entry("application",
$applicationId);

$applicationString = "<table class = 'table
table-bordered table-responsive'>";

$dayArray = array("0", "mon", "tues", "wed",
"thurs", "fri");
$timeArray_1 = array("0830_1030", "1030_1200",
"0100_0230", "0230_0400", "0400_0500");
$timeArray_2 = array("8:30-10:30", "
10:30-12:00", "1:00-2:30", "2:30-4:00", "
4:00-5:00");

foreach ($applications as $application) {
$userId = $application->userId;
$applicationType = $application->
applicationType;
$academicStanding = $application->
academicStanding;
$path = base_url("/uploads/".$userId)."/".
$academicStanding;

if ($applicationType == APPLICATION_NEW) {
$applicationType = "new application";
}
else {
$applicationType = "renewal";
}

$applicationString .= "<tr><td colspan =
'2'>Application Id</td><td colspan =
'2'>". $application->applicationId."</td
></tr>";
$applicationString .= "<tr><td colspan =
'2'>User Id</td><td colspan = '2'>".
$application->userId."</td></tr>";
$applicationString .= "<tr><td colspan =
'2'>Application Type</td><td colspan =
'2'>". $applicationType."</td></tr>";

if (strpos($application->organizations, "#")
!= false) {

$organizations = explode("#", $application
->organizations);
$positions = explode("#", $application->
positions);

for ($index = 0; $index < sizeof(
$organizations); $index++) {
$applicationString .= "<tr><td>
Organization</td><td>". $organizations[
$index]."</td>";
$applicationString .= "<td>Position</td><
td>". $positions[$index]."</td></tr>";
}
}
else {
$applicationString .= "<tr><td>Organization
</td><td>". $application->organizations.
"</td>";
$applicationString .= "<td>Position</td><td
>". $application->positions."</td></tr>"
;
}

for ($counter = 1; $counter < 7; $counter++)
{
$applicationString .= "<tr><td>Rank ".
$counter."</td><td>". $application->{"
subjectRank_". $counter}. "</td>";
$applicationString .= "<td>Taken Subjects</
td><td>". str_replace("#", " ", "
$application->{"subjectRank_". $counter.
"_taken"}). "</td></tr>";
}

if (isset($academicStanding)) {
$applicationString .= "<tr><td colspan =
'2'>Academic Standing</td><td colspan =
'2'>". anchor($path, "view pdf file",
array("target" => "_blank"))."</td></tr
>";
}
}

$applicationString .= "</table>";
$applicationString .= "<div class = 'text-
center'><p><a class = 'btn btn-success
btn-xs'>available</a>
<a class = 'btn btn-danger btn-xs'>not
available</a></p></div>";
$applicationString .= "<table class = 'table
table-bordered table-responsive text-
center'>";
$applicationString .= "<tr><td>Time</td><td>
Monday</td><td>Tuesday</td><td>Wednesday
</td><td>Thursday</td><td>Friday</td></tr
>";

$schedules = $this->General_Model->
get_user_entry("schedule", $userId);

foreach ($schedules as $schedule) {
for ($row = 0; $row < 5; $row++) {
$applicationString .= "<tr>";
for ($column = 0; $column < 6; $column++) {
if ($column > 0) {
$buttonClass = "";
$applicationString .= "<td>";
if ($schedule->{$dayArray[$column]}."_".
$timeArray_1[$row]} == 0) {
$buttonClass = "btn btn-danger";
}
}
else {

```

```

        $buttonClass = "btn btn-success";
    }
    $applicationString .= "<button class = '
        ".$buttonClass."'></button>";
    $applicationString .= "</td>";
}
else {
    $applicationString .= "<td>".$timeArray_2
        [$row]."</td>";
}
}
$applicationString .= "</tr>";
}
}
$applicationString .= "</table>";

echo json_encode($applicationString);
}

public function get_endorsement() {

    $endorsementId = $_GET["endorsementId"];
    $endorsements = $this->General_Model->
        get_table_entry("endorsement",
            $endorsementId);
    $endorsementString = "<table class = 'table
        table-bordered table-responsive'>";

    $xml = simplexml_load_file(base_url("/
        application/views/apply/endorsement.xml")
        );

    foreach ($endorsements as $endorsement) {
        $endorsementString .= "<tr><td colspan =
            '2'>Endorsement Id</td><td colspan =
            '2'>".$endorsement->endorsementId."</td
            ></tr>";
        $endorsementString .= "<tr><td colspan =
            '2'>Application Id</td><td colspan =
            '2'>".$endorsement->applicationId."</td
            ></tr>";

        $counter = 1;
        foreach ($xml->children() as $question) {
            $endorsementString .= "<tr><td colspan =
                '2'>".$question."</td><td colspan =
                '2'>".$endorsement->{"char_". $counter."
                _score"}."</td></tr>";
            $counter++;
        }

        if (strpos($endorsement->subjects, "#") !==
            false) {

            $subjects = explode("#", $endorsement->
                subjects);
            $codes = explode("#", $endorsement->codes);
            $grades = explode("#", $endorsement->grades
            );
            for ($index = 0; $index < sizeof($subjects)
                ; $index++) {
                $endorsementString .= "<tr><td>Subject
                    Taken</td><td>".$subjects[$index]." ".
                    $codes[$index]."</td>";
                $endorsementString .= "<td>Grade</td><td>".
                    $grades[$index]."</td></tr>";
            }
        }
        else {
            $endorsementString .= "<tr><td>Subject
                Taken</td><td>".$endorsement->subjects.
                " ". $endorsement->codes."</td>";
            $endorsementString .= "<td>Grade</td><td>".
                $endorsement->grades."</td></tr>";
        }
    }

    $attendanceScore = "";
    $recommendationScore = "";

    switch ($endorsement->attendanceScore) {
        case 1: $attendanceScore = "No absences and
            no tardiness"; break;
        case 2: $attendanceScore = "1-2 absences or
            1-2 tardiness"; break;
        case 3: $attendanceScore = "3-4 absences or
            3-4 tardiness"; break;
        case 4: $attendanceScore = "5 and above
            absences or 5 and above tardiness";
            break;
        default: break;
    }

    switch ($endorsement->recommendationScore) {
        case 1: $recommendationScore = "I strongly
            recommend the applicant as peer tutor";
            break;
        case 2: $recommendationScore = "I recommend
            the applicant as peer tutor"; break;
        case 3: $recommendationScore = "I recommend
            the applicant as peer tutor, with
            reservations"; break;
        case 4: $recommendationScore = "I do not
            recommend the applicant as peer tutor";
            break;
        default: break;
    }

    $endorsementString .= "<tr><td>Attendance
        score</td>";
    $endorsementString .= "<td colspan = '3'>".
        $attendanceScore."</td></tr>";
    $endorsementString .= "<tr><td>
        Recommendation score</td>";
    $endorsementString .= "<td colspan = '3'>".
        $recommendationScore."</td></tr>";
    $endorsementString .= "<tr><td>Comments</td
        ><td colspan = '3'>".$endorsement->
        comments."</td></tr>";
}
$endorsementString .= "</table>";
echo json_encode($endorsementString);
}

public function process_approval() {

    $userId = $this->input->post("userId");
    $applicationId = $this->input->post("
        applicationId");
    $users = $this->General_Model->
        get_table_entry("user", $userId);

    $userType = STUDENT;
    $applicationStatus = EVENT_PENDING;
    $approvalCount = EVENT_ACTIVE;
    $hasApplication = EVENT_INACTIVE;
    $notificationType = EVENT_INACTIVE;

    foreach ($users as $user) {
        $approvalCount = $user->approvalCount;
    }

    if (isset($_POST["accept"])) {
        $userType = STUDENT_TUTOR;
        $applicationStatus = EVENT_ACTIVE;

        if ($approvalCount > DEFAULT_INTEGER) {
            $approvalCount++;
        }

        $notificationType = APPLICATION_APPROVED;
        $notificationDetails =
            APPLICATION_APPROVED_MSG;
    }
    if (isset($_POST["reject"])) {
        $applicationStatus = EVENT_INACTIVE;
        $notificationType = APPLICATION_REJECTED;
        $notificationDetails =
            APPLICATION_REJECTED_MSG;
    }
}
$applicationData = array(
    "applicationId" => $applicationId,
    "applicationStatus" => $applicationStatus
);
$userData = array(
    "userId" => $userId,
    "userType" => $userType,
    "hasApplication" => $hasApplication,

```



```

        finish_endorsement");
    }
}

public function process_endorsement() {

    $this->form_validation->set_rules("subjects
    []", "subjects[]", "required");
    $this->form_validation->set_rules("codes[]"
    , "codes[]", "required");

    if ($this->form_validation->run() == false)
    {
        $msg = FORM_INCOMPLETE_INPUT.: subjects
        taken under you";
    }
    else {
        $endorsementId = $this->input->post("
        endorsementId");
        $subjects = implode("#", $this->input->
        post("subjects"));
        $codes = implode("#", $this->input->post(
        "codes"));
        $grades = implode("#", $this->input->post
        ("grades"));
        $hasAccess = EVENT_INACTIVE;
        $status = EVENT_ACTIVE;

        $endorsementData = array(
            "endorsementId" => $endorsementId,
            "subjects" => $subjects,
            "codes" => $codes,
            "grades" => $grades,
            "attendanceScore" => $this->input->post
            ("attendanceScore"),
            "recommendationScore" => $this->input->
            post("recommendationScore"),
            "comments" => $this->input->post("
            comments"),
            "endorsementStatus" => $status,
            "hasAccess" => $hasAccess
        );
        for ($charNumber = 1; $charNumber < 6;
        $charNumber++) {
            $endorsementData = array_merge(
            $endorsementData, array("char_". $charNumber
            . "_score" => $this->input->post("char_".
            $charNumber . "_score")));
        }

        $userIds = $this->input->post("userId");
        $notificationData = $this->
        notification_controller->
        create_notification(FACULTY_ENDORSED,
        $endorsementId, $userIds,
        FACULTY_ENDORSED_MSG);
        $this->General_Model->create_table_entry(
        "notification", $notificationData);
        $this->General_Model->update_table_entry(
        "endorsement", $endorsementData);

        $msg = FORM_SUCCESSFUL;
    }

    $this->process_acceptance($msg);
}
}
?>

```

## Appointment.php

```

<?php if ( ! defined("BASEPATH")) exit("No
    direct script access allowed");

/**
 * @author Quizon, Maria Reena I.
 * filename: Appointment.php
 * description: controller for the schedule page
    in LRC_TMS
 * start date: January 13, 2016

```

```

 * end date:
 * last updated:
 * version number: 1.0
 */

class Appointment extends CI_Controller{
    function __construct(){
        parent::__construct();

        $this->load->helper("url");
        $this->load->library("form_validation");
        $this->load->library("
        Notification_Controller");
        $this->load->library("Validation_Controller
        ");

        $this->load->model("General_Model");
        $this->load->model("Application_Model");
        $this->load->model("Appointment_Model");
        $this->load->model("Calendar_Model");
        $this->load->model("Schedule_Model");
        $this->load->model("User_Model");

        $this->validation_controller->
        check_validation();
    }

    public function index($msg = NULL) {

        $data["msg"] = $msg;
        $this->load->view("templates/navigation");
        $this->load->view("appointment/appointment"
        , $data);
    }

    public function reschedule_appointment() {

        $data["msg"] = FORM_APPOINTMENT_RESCHEDULE;
        $appointmentId = $this->input->post("
        eventId");
        $appointments = $this->General_Model->
        get_table_entry("appointment",
        $appointmentId);

        if ($appointments != DEFAULT_INTEGER) {
            $data["appointmentId"] = $appointmentId;
            $data["notificationType"] = $this->input
            ->post("notificationType");
            foreach ($appointments as $appointment) {
                $data["numberOfStudents"] =
                $appointment->numberOfStudents;
                $data["subject"] = $appointment->
                subject;
                $data["code"] = $appointment->code;
                $data["topic"] = $appointment->topic;
                $data["faculty"] = $appointment->
                faculty;
            }
        }
        $this->load->view("templates/navigation");
        $this->load->view("appointment/appointment"
        , $data);
    }

    public function process_appointment() {

        $userId = $this->session->userdata("userId"
        );
        $appointmentId = $this->input->post("
        appointmentId");

        if ($appointmentId == DEFAULT_INTEGER) {
            $appointmentId = $this->General_Model->
            get_tableId("appointment");
        }

        $feedbackId = $this->General_Model->
        get_tableId("feedback");
        $calendars = $this->General_Model->
        get_table("calendar");
        $userIds = $this->get_tutors($this->input->
        post("date"), "PROCESS_APPOINTMENT");
    }
}

```

```

$tutor_userId = DEFAULT_STRING;
$appointmentStatus = EVENT_PENDING;

$eventId = $appointmentId;

if ($userIds == DEFAULT_INTEGER) {
    echo FORM_APPOINTMENT_NO_TUTORS;
}
else {
    $notificationDetails = array(
        $userId,
        $this->session->userdata("name"),
        $this->input->post("numberOfStudents"),
        $this->input->post("subject"),
        $this->input->post("code"),
        $this->input->post("topic"),
        $this->input->post("date"),
        $this->input->post("time"),
    );
    $notificationDetails = implode("#",
    $notificationDetails);

    $appointmentData = array(
        "appointmentId" => $appointmentId,
        "userId" => $userId,
        "numberOfStudents" => $this->input->
    post("numberOfStudents"),
        "subject" => $this->input->post("
    subject"),
        "code" => $this->input->post("code"),
        "topic" => $this->input->post("topic"),
        "faculty" => $this->input->post("
    faculty"),
        "date" => $this->input->post("date"),
        "time" => $this->input->post("time"),
        "status" => $appointmentStatus,
    );

    $feedbackData = array(
        "feedbackId" => $feedbackId,
        "appointmentId" => $appointmentId
    );

    $notificationData = $this->
    notification_controller->
    create_notification(APPOINTMENT_NEW,
    $appointmentId, $userIds,
    $notificationDetails);
    $this->General_Model->create_table_entry(
    "notification", $notificationData);

    if ($this->input->post("appointmentId")
    == DEFAULT_INTEGER) {
        $notificationData = $this->
        notification_controller->
        create_notification(APPOINTMENT_NEW_ADMIN,
        $appointmentId, ADMIN_ID,
        APPOINTMENT_NEW_ADMIN_MSG);
        $this->General_Model->
        create_table_entry("notification",
        $notificationData);
        $this->General_Model->
        create_table_entry("appointment",
        $appointmentData);
        $this->General_Model->
        create_table_entry("feedback",
        $feedbackData);
    }
    else {
        $notificationType = $this->input->post(
        "notificationType");

        $notificationData = $this->
        notification_controller->
        update_notification($notificationType,
        $appointmentId, EVENT_INACTIVE,
        EVENT_INACTIVE);
        $this->General_Model->
        update_table_entry("appointment",
        $appointmentData);
        $this->General_Model->
        update_table_entry("notification",
        $notificationData);
    }
}

}

public function search_available_time() {
    $adminId = ADMIN_ID;
    $daysToAdd = DEFAULT_INTEGER;
    $timeSearchString = DEFAULT_STRING;
    $subject = $_GET["subject"];

    $currStartDate = date("Y-m-d");
    $currStartDate = strtotime("+2 days",
    strtotime($currStartDate));
    $startDay = date("w", $currStartDate);

    switch ($startDay) {
        case 1: $daysToAdd = 4; break;
        case 2: $daysToAdd = 3; break;
        case 3: $daysToAdd = 2; break;
        case 4: $daysToAdd = 1; break;
        case 5: $daysToAdd = 0; break;
        default: break;
    }

    $time = array("8:30-10:30", "10:30-12:00",
    "1:00-2:30", "2:30-4:00", "4:00-5:00");

    $currEndDate = strtotime("+".$daysToAdd."
    days", $currStartDate);
    $currDate = date("Y-m-d", strtotime(
    $currStartDate));
    $endDay = date("w", $currEndDate);
    $tutorCount = DEFAULT_INTEGER;

    $timeSearchString .= "<table class = 'table
    table-bordered table-hover table-
    responsive small'>";
    for ($index_1 = $startDay; $index_1 <=
    $endDay; $index_1++) {
        $tutorArray = array();
        $index_2 = DEFAULT_INTEGER;
        $currDay = DEFAULT_STRING;

        $tutors = $this->get_tutors($currDate, "
        SEARCH_TIME");
        if ($tutors != DEFAULT_INTEGER) {
            $tutors = explode("#", $tutors);
        }
        switch ($index_1) {
            case 1: $currDay = "mon"; break;
            case 2: $currDay = "tues"; break;
            case 3: $currDay = "wed"; break;
            case 4: $currDay = "thurs"; break;
            case 5: $currDay = "fri"; break;
            default: break;
        }

        $timeSearchString .= "<tr><td>".$currDay."
        </td><td>";
        $schedules = $this->Schedule_Model->
        get_tutor_time($adminId, $currDay);
        foreach ($schedules as $adminKey =>
        $adminValue) {
            if ($adminValue == 1) {
                if ($tutors != DEFAULT_INTEGER) {
                    for ($counter = 0; $counter <
                    sizeof($tutors); $counter++) {
                        $tutorSchedules = $this->
                        Schedule_Model->get_tutor_time($tutors[
                        $counter], $currDay);
                        if ($tutorSchedules[$adminKey] ==
                        1) {
                            $tutorSubject = $this->
                            Application_Model->get_tutor_subject(
                            $tutors[$counter], $subject);
                            if ($tutorSubject == 1) {
                                if (strpos($timeSearchString,
                                $time[$index_2]) === false) {
                                    $tutorName = DEFAULT_STRING
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```



```

        $tutorArray[] = $tutor->userId;
    }
}

if (sizeof($tutorArray) > 0) {
    $userIds = implode("#", $tutorArray);
}
return $userIds;
}
}

public function get_tutor_time($userId, $day)
{

    $time = $this->input->post("time");
    $schedule = $this->Schedule_Model->
get_tutor_time($userId, $day);

    if (!$schedule) {
        return 0;
    }
    else {
        $timeStr = DEFAULT_STRING;
        switch ($time) {
            case "8:30-10:30": $timeStr = "
_0830_1030"; break;
            case "10:30-12:00": $timeStr = "
_1030_1200"; break;
            case "1:00-2:30": $timeStr = "
_0100_0230"; break;
            case "2:30-4:00": $timeStr = "
_0230_0400"; break;
            case "4:00-5:00": $timeStr = "
_0400_0500"; break;
            default: break;
        }
        if ($schedule[$day.$timeStr] == 0) {
            return 0;
        }
        else {
            return 1;
        }
    }
}

public function get_day($date) {
    $timestamp = strtotime($date);
    $day = date("w", $timestamp);

    switch ($day) {
        case "1": $day = "mon"; break;
        case "2": $day = "tues"; break;
        case "3": $day = "wed"; break;
        case "4": $day = "thurs"; break;
        case "5": $day = "fri"; break;
        default: break;
    }
    return $day;
}
}
?>

```

```

$this->load->library("Email_Controller");
$this->load->library("Notification_Controller
");
$this->load->library("Validation_Controller")
;

$this->load->model("General_Model");
$this->load->model("Notification_Model");
$this->load->view("templates/navigation");

$this->validation_controller->
check_validation();
}

public function index($msg = NULL) {

    $appointments = $this->General_Model->
get_table("appointment");

    if ($appointments != DEFAULT_INTEGER) {
        $data["appointments"] = $appointments;
        foreach ($appointments as $appointment) {
            $this->check_tutors($appointment);
        }
    }

    $data["msg"] = $msg;
    $this->load->view("appointment/dashboard",
    $data);
}

public function check_tutors($appointment) {
    $today = date("Y-m-d");
    $appointmentId = DEFAULT_INTEGER;

    if (strtotime($today) >= strtotime(
    $appointment->date) && $appointment->
status == 2) {
        $status = DEFAULT_INTEGER;
        $userId = $appointment->userId;
        $appointmentData = array(
            "appointmentId" => $appointment->
appointmentId,
            "status" => $status
        );

        $notificationData = $this->
notification_controller->
update_notification(APPOINTMENT_NEW,
    $appointmentId, NULL, $status);
        $this->General_Model->update_table_entry("
notification", $notificationData);
        $this->General_Model->update_table_entry("
appointment", $appointmentData);
        $notificationData = $this->
notification_controller->
create_notification(NO_MORE_TUTORS,
    $appointmentId, $userId,
    NO_MORE_TUTORS_MSG);
        $this->General_Model->create_table_entry("
notification", $notificationData);
    }
}
}

```

## Dashboard.php - Appointment

```

<?php if ( ! defined("BASEPATH")) exit("No
direct script access allowed");

```

```

/**
 * @author Quizon, Maria Reena I.
 * filename: Dashboard.php
 * description: Controller for the appointment
    dashboard page in LRC_TMS
 * start date: February 1, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

```

```

class Dashboard extends CI_Controller {
function __construct(){
parent::__construct();

$this->load->helper("url");
$this->load->library("form_validation");

```

```

public function check_attendance() {

    $userId = $this->input->post("userId");
    $appointmentId = $this->input->post("
appointmentId");
    $numberOfStudents = $this->input->post("
numberOfStudents");
    $users = $this->General_Model->
get_table_entry("user", $userId);

    $msg = DEFAULT_STRING;
    $email = DEFAULT_STRING;
    $hasAttended = DEFAULT_INTEGER;
    $isWarned = DEFAULT_INTEGER;
    $isBlocked = DEFAULT_INTEGER;
    $blockCount = DEFAULT_INTEGER;
    $appointmentStatus = EVENT_INACTIVE;

    $userIds = DEFAULT_STRING;
    $notificationType = DEFAULT_INTEGER;

```

```

$notificationDetails = DEFAULT_STRING;
$notificationData = array();

foreach ($users as $user) {
    $hasAttended = $user->hasAttended;
    $blockCount = $user->blockCount;
    $email = $user->email;
}

if (isset($_POST["didAttend"])) {

    $userIds = $userId;
    $notificationType = APPOINTMENT_ATTENDED;
    $notificationDetails =
        APPOINTMENT_ATTENDED_MSG;
    $appointmentStatus = EVENT_FINISHED;
    if (strlen($hasAttended) <= 0) {
        $hasAttended = $appointmentId;
    }
    else {
        $hasAttended .= "#".$appointmentId;
    }
}
elseif (isset($_POST["didNotAttend"])) {

    if ($blockCount >= 1) {
        $isBlocked = 1;
        $blockCount++;
    }
    elseif ($blockCount < 1) {
        $isWarned = 1;
        $blockCount = 1;

        $userIds = $userId;
        $notificationType = WARNING;
        $notificationDetails = WARNING_MESSAGE;
    }

    $appointmentStatus = EVENT_DID_NOT_ATTEND;
}
elseif (isset($_POST["cancel"])) {

    $notificationType =
        APPOINTMENT_CANCELLED_ADMIN;
    $notificationDetails =
        APPOINTMENT_CANCELLED_ADMIN_MSG;

    $userIds = $this->session->userdata("userId"
    )."#".$userId;

    if (strlen($this->input->post("tutor_userId"
    )) > 1) {
        $notificationData = $this->
            notification_controller->
            create_notification(
                APPOINTMENT_CANCELLED_ADMIN_FOR_TUTOR,
                $appointmentId, $this->input->post("
                tutor_userId"),
                APPOINTMENT_CANCELLED_ADMIN_MSG);
        $this->General_Model->create_table_entry("
            notification", $notificationData);
    }
}

$userData = array(
    "userId" => $userId,
    "hasAttended" => $hasAttended,
    "isWarned" => $isWarned,
    "isBlocked" => $isBlocked,
    "blockCount" => $blockCount
);
$appointmentData = array(
    "appointmentId" => $appointmentId,
    "status" => $appointmentStatus
);

if (isset($_POST["didAttend"])) {
    $appointmentData = array_merge(
        $appointmentData, array("
        numberOfStudents" => $numberOfStudents))
    ;
}

```

```

$this->General_Model->update_table_entry("
    appointment", $appointmentData);
$notificationData_1 = $this->
    notification_controller->
    update_notification(APPLICATION_NEW,
        $appointmentId, $userId, EVENT_INACTIVE);
$notificationData_2 = $this->
    notification_controller->
    create_notification($notificationType,
        $appointmentId, $userIds,
        $notificationDetails);

$this->General_Model->update_table_entry("
    notification", $notificationData_1);
$this->General_Model->create_table_entry("
    notification", $notificationData_2);
$this->General_Model->update_table_entry("
    user", $userData);

if ($isBlocked > DEFAULT_INTEGER) {
    $emailType = EMAIL_BLOCK_STUDENT;
    $this->email_controller->create_email(
        $email, $emailType);
}

$this->index($msg);
}
?>

```

## Feedback.php

```

<?php if ( ! defined("BASEPATH")) exit("No
    direct script access allowed");

/**
 * @author Quizon, Maria Reena I.
 * filename: Feedback.php
 * description: Controller for the feedback page
 * in LRC_TMS
 * start date: February 2, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

class Feedback extends CI_Controller {

    function __construct() {
        parent::__construct();

        $this->load->helper("url");
        $this->load->library("form_validation");

        $this->load->model("General_Model");
        $this->load->model("Calendar_Model");
        $this->load->model("Feedback_Model");
        $this->load->model("User_Model");
    }

    public function index($msg = NULL) {

        $counter = DEFAULT_INTEGER;
        $tutor_userId = DEFAULT_INTEGER;
        $appointmentId = DEFAULT_INTEGER;
        $tutor_name = DEFAULT_STRING;

        $appointmentIds = $this->session->userdata("
            hasAttended");
        $appointmentIds = explode("#",
            $appointmentIds);
        for ($index=0; $index < sizeof(
            $appointmentIds); $index++) {
            if (strlen($appointmentIds[$index]) > 1) {
                $appointmentId = $appointmentIds[$index];
                break;
            }
        }
        $appointments = $this->General_Model->
            get_table_entry("appointment",
                $appointmentId);
        foreach ($appointments as $appointment) {

```



```

    $tutor_userId = $appointment->tutor_userId;
}

$users = $this->General_Model->
    get_table_entry("user", $tutor_userId);

foreach ($users as $user) {
    $tutor_name = $user->name;
}

$data["msg"] = $msg;
$data["appointmentId"] = $appointmentId;
$data["appointments"] = $appointments;
$data["tutorName"] = $tutor_name;
$this->load->view("templates/navigation_2");
$this->load->view("feedback/feedback", $data)
;
$counter++;
}

public function process_feedback() {
    $msg = DEFAULT_STRING;
    $userId = $this->session->userdata("userId")
;
    $appointmentId = $this->input->post("
        appointmentId");
    $hasAttended = $this->session->userdata("
        hasAttended");
    $feedbackData = array("appointmentId" =>
        $appointmentId);

    $hasAttended = str_replace($appointmentId, "
        #", $hasAttended);

    if (!preg_match("/\d\w+/", $hasAttended)) {
        $hasAttended = DEFAULT_INTEGER;
    }

    $userData = array(
        "userId" => $userId,
        "hasAttended" => $hasAttended
    );

    for ($questionNumber = 1; $questionNumber <=
        15; $questionNumber++) {
        $feedbackData = array_merge($feedbackData,
            array("question_". $questionNumber. "
                _score" => $this->input->post("question_
                    ". $questionNumber)));
    }
    $feedbackData = array_merge($feedbackData,
        array("goodPoints" => $this->input->post(
            "goodPoints")));
    $feedbackData = array_merge($feedbackData,
        array("improvementAreas" => $this->input
            ->post("improvementAreas")));

    $this->General_Model->update_table_entry("
        user", $userData);
    $this->General_Model->update_table_entry("
        feedback", $feedbackData);
    $this->session->set_userdata(array("
        hasAttended" => $hasAttended));

    $msg = FORM_SUCCESSFUL;
    redirect(base_url("/dashboard"));
}

public function view_feedback() {
    $userId = DEFAULT_INTEGER;
    $isAllowed = DEFAULT_INTEGER;

    $userType = $this->session->userdata("
        userType");
    $startDate = $this->Calendar_Model->get_date(
        "SEM_BEGIN");
    $endDate = $this->Calendar_Model->get_date("
        SEM_END");

    if ($userType == ADMIN) {
        $isAllowed = EVENT_ACTIVE;

        $userId = $this->uri->segment(2);
    }
    else {
        $userId = $this->session->userdata("userId")
;
    }

    if ($isAllowed == DEFAULT_INTEGER) {
        $today = date("Y-m-d");
        if (strtotime($today) < strtotime($endDate))
            {
                $startDate = DEFAULT_INTEGER;
                $endDate = DEFAULT_INTEGER;
            }
    }

    if ($startDate != DEFAULT_INTEGER && $endDate
        != DEFAULT_INTEGER) {
        $feedbacks = $this->Feedback_Model->
            get_feedback($startDate, $endDate,
                $userId);

        if ($feedbacks != DEFAULT_INTEGER) {
            $score = $this->compute_score($feedbacks);
            $comments = $this->get_comments($feedbacks)
;
            $overall = $this->get_overall($score);

            $data["score"] = $score;
            $data["overall"] = $overall;
            $data["comments"] = $comments;
            $data["feedbacks"] = $feedbacks;
        }

        $data["userId"] = $userId;
        $data["userType"] = $this->session->userdata(
            "userType");

        $this->load->view("templates/navigation");
        $this->load->view("feedback/view_feedback",
            $data);
    }

    function compute_score($feedbacks) {
        $score = array();

        for ($counter = 1; $counter < 16; $counter++)
            {
                $sum = 0;
                foreach ($feedbacks as $feedback) {
                    $sum += $feedback->{"question_". $counter. "
                        _score"};
                }
                $average = $sum / sizeof($feedbacks);
                $score[] = $average;
            }

        return $score;
    }

    function get_overall($score) {
        $overall = DEFAULT_INTEGER;
        for ($index = 0; $index < sizeof($score);
            $index++) {
            $overall += $score[$index];
        }
        $overall = $overall/sizeof($score);
        return $overall;
    }

    function get_comments($feedbacks) {
        $comments = array();
        $goodPoints = DEFAULT_STRING;
        $improvementAreas = DEFAULT_STRING;

        foreach ($feedbacks as $feedback) {
            $goodPoints .= $feedback->goodPoints."#";
            $improvementAreas .= $feedback->
                improvementAreas."#";
        }
    }
}

```

```

    $comments[] = $goodPoints;
    $comments[] = $improvementAreas;

    return $comments;
}
}
?>

```

## Dashboard.php - Main

```

<?php if ( ! defined("BASEPATH")) exit("No
direct script access allowed");

/**
 * @author Quizon, Maria Reena I.
 * filename: Dashboard.php
 * description: Controller for the dashboard
page in LRC_TMS
 * start date: January 13, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

class Dashboard extends CI_Controller {
function __construct(){
parent::__construct();

$this->load->helper("url");
$this->load->library("form_validation");
$this->load->library("Notification_Controller
");
$this->load->library("Validation_Controller")
;

$this->load->model("General_Model");
$this->load->model("Notification_Model");

$this->validation_controller->
check_validation();
}

public function index($msg = NULL) {

$isWarned = DEFAULT_INTEGER;
$userId = $this->session->userdata("userId");
$userType = $this->session->userdata("
userType");

$schedules = $this->General_Model->
get_user_entry("schedule", $userId);
$appointments = $this->General_Model->
get_user_entry("appointment", $userId);
$notifications = $this->Notification_Model->
get_notifications($userId);

$mockExams = $this->General_Model->get_table(
"mockexam");
$mockExamStudents = $this->General_Model->
get_table("mockexam_students");

if ($mockExams != DEFAULT_INTEGER) {
$data["mockExams"] = $mockExams;
$data["mockExamStudents"] =
$mockExamStudents;
}
if ($appointments != DEFAULT_INTEGER) {
$data["appointments"] = $appointments;
}

if (null !== $this->session->userdata("
password")) {
$data["password"] = $this->session->userdata
("password");
}

$data["msg"] = $msg;
$data["userId"] = $userId;
$data["userType"] = $userType;
$data["schedules"] = $schedules;
$data["notifications"] = $notifications;
$data["notificationCount"] = sizeof(
$notifications);

```

```

$data["appointmentCount"] = sizeof(
$appointments);
$data["schedule"] = $this->load->view("
templates/schedule", $data, TRUE);

$this->session->set_userdata(array("isWarned"
=> $isWarned));

$this->load->view("templates/navigation");
$this->load->view("main/dashboard", $data);
}

public function cancel_appointment() {

$msg = DEFAULT_STRING;
$appointmentStatus = EVENT_INACTIVE;
$appointmentId = $this->input->post("
appointmentId");
$tutor_userId = $this->input->post("
tutor_userId");

$userIds = ADMIN_ID."#".$this->session->
userdata("userId");

if (strlen($tutor_userId) > 1) {
$userIds .= "#".$tutor_userId;
}

$appointmentData = array(
"appointmentId" => $appointmentId,
"status" => $appointmentStatus,
);

$notificationData = $this->
notification_controller->
update_notification(APPOINTMENT_NEW,
$appointmentId, NULL, EVENT_INACTIVE);
$this->General_Model->update_table_entry("
notification", $notificationData);

$notificationData = $this->
notification_controller->
create_notification(
APPOINTMENT_CANCELLED_STUDENT, $eventId,
$userIds,
APPOINTMENT_CANCELLED_STUDENT_MSG);
$this->General_Model->create_table_entry("
notification", $notificationData);

$this->index($msg);
}

public function process_appointment() {

$msg = DEFAULT_STRING;
$userIds_1 = DEFAULT_STRING;
$userIds_2 = DEFAULT_STRING;
$appointmentStatus = EVENT_PENDING;
$notificationType = DEFAULT_INTEGER;
$notificationDetails = DEFAULT_STRING;

$appointmentId = $this->input->post("eventId"
);
$tutor_userId = $this->session->userdata("
userId");
$notificationId = $this->input->post("
notificationId");

if (isset($_POST["accept"])) {
$appointmentStatus = EVENT_ACTIVE;
$notificationType = TUTOR_ACCEPTED;
$notificationDetails = TUTOR_ACCEPTED_MSG;
$userIds_2 .= $this->session->userdata("
userId")."#".$this->input->post("userId"
);
}
elseif (isset($_POST["reject"])) {
$notificationType = TUTOR_REJECTED;
$notificationDetails = TUTOR_REJECTED_MSG;
$notifications = $this->General_Model->
get_table_entry("notification",
$notificationId);
foreach ($notifications as $notification) {

```

```

    $userIds_1 = $notification->userIds;
}
if (strlen($userIds_1) > 1) {
    $userIds_1 = str_replace($tutor_userId, "#",
        , $userIds_1);
}
$userIds_2 = $this->session->userdata("
    userId");
$tutor_userId = DEFAULT_INTEGER;
}

$appointmentData = array(
    "appointmentId" => $appointmentId,
    "status" => $appointmentStatus,
    "tutor_userId" => $tutor_userId
);

$notificationData_1 = $this->
    notification_controller->
    update_notification(APPOINTMENT_NEW,
        $appointmentId, $userIds_1, EVENT_ACTIVE)
    ;
$notificationData_2 = $this->
    notification_controller->
    create_notification($notificationType,
        $appointmentId, $userIds_2,
        $notificationDetails);

$this->General_Model->update_table_entry("
    appointment", $appointmentData);
$this->General_Model->update_table_entry("
    notification", $notificationData_1);
$this->General_Model->create_table_entry("
    notification", $notificationData_2);

if (!preg_match("/[0-9]/", $userIds_1) &&
    $appointmentStatus != EVENT_ACTIVE) {

    $appointmentStatus = EVENT_INACTIVE;
    $appointmentData = array(
        "appointmentId" => $appointmentId,
        "status" => $appointmentStatus,
    );

    $notificationData = $this->
        notification_controller->
        create_notification(NO_MORE_TUTORS,
            $appointmentId, $this->input->post("
                userId"), NO_MORE_TUTORS_MSG);
    $this->General_Model->update_table_entry("
        appointment", $appointmentData);
    $this->General_Model->create_table_entry("
        notification", $notificationData);
}

$this->index($msg);
}

public function save_schedule() {
    $hasSchedule = EVENT_ACTIVE;
    $dayArray = array("mon", "tues", "wed", "
        thurs", "fri");
    $timeArray = array("0", "0830_1030", "1030
        _1200", "0100_0230", "0230_0400", "0400
        _0500");

    $scheduleData = array("userId" => $this->
        session->userdata("userId"), "hasSchedule"
        => $hasSchedule);
    for ($row = 0; $row < 5; $row++) {
        for ($column = 0; $column < 6; $column++)
        {
            if ($column > 0) {
                $scheduleData = array_merge(
                    $scheduleData, array($dayArray[$row]."_".
                        $timeArray[$column] => $this->input->post(
                            $dayArray[$row]."_".
                                $timeArray[$column]));
            }
        }
    }

    $this->General_Model->update_table_entry("
        schedule", $scheduleData);
    $this->index();
}

```

```

}

public function reset_tables() {
    $users = $this->General_Model->get_table("
        user");
    $this->General_Model->truncate_tables();
    foreach ($users as $user) {
        $this->reset_users($user->userId);
        $this->reset_schedules($user->userId);
    }
    return true;
}

public function reset_users($userId) {
    $userType = STUDENT;
    $userData = array(
        "userId" => $userId,
        "userType" => $userType
    );
    $this->General_Model->update_user_entry("user
        ", $userData);
    return true;
}

public function reset_schedules($userId) {
    $value = DEFAULT_INTEGER;
    $scheduleData = array();
    $day = array("mon", "tues", "wed", "thurs", "
        fri");
    $time = array("_0830_1030", "_1030_1200", "
        _0100_0230", "_0230_0400", "_0400_0500");

    for ($index_1 = 0; $index_1 < sizeof($day);
        $index_1++) {
        for ($index_2 = 0; $index_2 < sizeof($time);
            $index_2++) {
            $scheduleData = array_merge($scheduleData,
                array($day[$index_1].$time[$index_2] =>
                    $value));
        }
    }
    $scheduleData = array_merge($scheduleData,
        array("hasSchedule" => $value));
    $scheduleData = array_merge($scheduleData,
        array("userId", $userId));

    $this->General_Model->update_user_entry("
        schedule", $scheduleData);
    return true;
}

public function logout() {
    $this->session->sess_destroy();
    redirect(base_url("/login"));
}

}
?>

```

## Login.php

```

<?php if ( ! defined("BASEPATH")) exit("No
    direct script access allowed");

/**
 * @author Quizon, Maria Reena I.
 * filename: Login.php
 * description: Controller for the login page in
    LRC_TMS
 * start date: January 13, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

class Login extends CI_Controller {

    function __construct() {
        parent::__construct();

        $this->load->helper("url");
    }
}

```

```

$this->load->library("PHPass");
$this->load->library("Email_Controller");
$this->load->library("form_validation");
$this->load->model("User_Model");
}

public function index($msg = NULL) {
    $data["msg"] = $msg;

    if (null !== $this->session->userdata("userId
    ")) {
        redirect(base_url("/dashboard"));
    }
    else {
        $this->load->view("main/login", $data);
    }
}

public function process_login() {
    $msg = DEFAULT_STRING;

    $userId = $this->input->post("userId");
    $password = $this->input->post("password");
    $result = $this->User_Model->validate_user(
        $userId, $password);
    if ($result == "blocked") {
        $msg = LOGIN_BLOCKED_MSG;
    }
    elseif ($result == "false") {
        $msg = LOGIN_INVALID_MSG;
    }
    elseif ($result == "true") {
        redirect(base_url("/dashboard"));
    }
    }

    $this->index($msg);
}

public function forgot_password() {
    $userId = $this->input->post("userId");
    $email = $this->input->post("email");
    $valid = $this->User_Model->validate_email(
        $userId, $email);
    if (!$valid) {
        $msg = LOGIN_USER_ID_INVALID_MSG;
    }
    else {
        $password = $this->email_controller->
            create_email($email, EMAIL_NEW_PASSWORD)
            ;
        if (!$password) {
            $msg = EMAIL_FAILED;
        }
        else {
            $password = $this->phpass->create_hash(
                $password);
            $userData = array(
                "userId" => $userId,
                "password" => $password
            );
            $this->General_Model->update_table_entry("
                user", $userData);
            $msg = EMAIL_SUCCESSFUL;
        }
    }
    $this->index($msg);
}
}
?>

```

## Profile.php

```

<?php if ( ! defined("BASEPATH")) exit("No
    direct script access allowed");

/**
 * @author Quizon, Maria Reena I.
 * filename: Profile.php
 * description: Controller for the profile page
 * in LRC_TMS
 * start date: March 25, 2016
 * end date:

```

```

* last updated:
* version number: 1.0
*/

class Profile extends CI_Controller {

    function __construct() {
        parent::__construct();

        $this->load->helper("url");

        $this->load->library("PHPass");
        $this->load->library("form_validation");
        $this->load->library("Validation_Controller"
        );

        $this->load->model("General_Model");
        $this->load->model("Application_Model");
        $this->load->model("User_Model");

        $this->load->view("templates/navigation");
        $this->validation_controller->
            check_validation();
    }

    public function index($msg = NULL) {
        $data["msg"] = $msg;
        $this->load->view("main/profile", $data);
    }

    public function process_account() {
        if (isset($_POST["upload"])) {
            $this->upload_picture();
        }
        if (isset($_POST["changeBirthday"])) {
            $this->change_birthday();
        }
        if (isset($_POST["changePassword"])) {
            $this->change_password();
        }
    }

    public function upload_picture() {
        $msg = DEFAULT_STRING;
        $profilePicture = DEFAULT_STRING;

        $isUploaded = false;
        $userId = $this->session->userdata("userId");
        $path = "./uploads/".$userId;

        if(!is_dir($path)) {
            mkdir($path, 0755, TRUE);
        }

        $config = array(
            "upload_path" => $path,
            "allowed_types" => "gif|jpg|jpeg|png",
            "max_size" => "2048000",
            "overwrite" => TRUE
        );

        $this->load->library("upload", $config);

        if (!$this->upload->do_upload("
            profilePicture")) {
            $profilePicture = NULL;
            $msg = FORM_INCOMPLETE_INPUT;
        }
        else {
            $profilePicture = $this->upload->data("
            file_name");
            $isUploaded = true;
        }

        if ($isUploaded) {
            $userData = array(
                "userId" => $userId,
                "profilePicture" => $profilePicture
            );

            $this->General_Model->update_table_entry("
            user", $userData);

```

```

    $msg = FORM_SUCCESSFUL;
}

$this->session->set_userdata(array("
profilePicture" => $profilePicture));
$this->index($msg);
}

public function change_birthday() {

    $userId = $this->session->userdata("userId");
    $this->form_validation->set_rules("birthday",
        "birthday", "required");

    if ($this->form_validation->run() == false) {
        $msg = FORM_INCOMPLETE_INPUT;
    }
    else {
        $birthday = $this->input->post("birthday");

        $userData = array(
            "userId" => $userId,
            "birthday" => $birthday
        );

        $this->General_Model->update_table_entry("
            user", $userData);
        $msg = FORM_SUCCESSFUL;
    }
    $this->session->set_userdata(array("birthday"
        => $birthday));
    $this->index($msg);
}

public function change_password() {

    $msg = DEFAULT_STRING;
    $userPassword = DEFAULT_STRING;
    $userId = $this->session->userdata("userId");
    $users = $this->General_Model->
        get_table_entry("user", $userId);

    foreach ($users as $user) {
        $userPassword = $user->password;
    }

    $this->form_validation->set_rules("
        oldPassword", "oldPassword", "required");
    $this->form_validation->set_rules("
        newPassword", "newPassword", "required");
    $this->form_validation->set_rules("
        retypePassword", "retypePassword", "
        required");

    if ($this->form_validation->run() == false) {
        $msg = FORM_INCOMPLETE_INPUT;
    }
    else {

        $oldPassword = $this->input->post("
            oldPassword");
        $newPassword = $this->input->post("
            newPassword");
        $retypePassword = $this->input->post("
            retypePassword");

        if ($this->phpass->check_password(
            $oldPassword, $userPassword) && (
                $newPassword == $retypePassword)) {
            $retypePassword = $this->phpass->
                create_hash($retypePassword);
            $userData = array(
                "userId" => $userId,
                "password" => $retypePassword
            );
            $this->General_Model->update_table_entry("
                user", $userData);
            $msg = FORM_SUCCESSFUL;
        }
        elseif (!$this->phpass->check_password(
            $oldPassword, $userPassword) && (
                $newPassword == $retypePassword)) {
            $msg = PROFILE_OLD_PASSWORD_MSG;
        }
    }
}

elseif ($this->phpass->check_password(
    $oldPassword, $userPassword) && (
        $newPassword != $retypePassword)) {
    $msg = PROFILE_CONFIRM_PASSWORD_MSG;
}
else {
    $msg = PROFILE_OLD_PASSWORD_MSG;
}
}
$this->index($msg);
}

public function view_profile() {

    $userType = DEFAULT_INTEGER;
    $userId = $this->uri->segment(3);
    $path = $path = base_url("/uploads/".$userId)
        ;

    $users = $this->General_Model->
        get_table_entry("user", $userId);
    $schedules = $this->General_Model->
        get_user_entry("schedule", $userId);

    foreach ($users as $user) {
        $userType = $user->userType;
        $profilePicture = $user->profilePicture;
        $data["profilePicture"] = $profilePicture;
        $data["path"] = $path."/". $profilePicture;
        $data["studentNumber"] = $user->
            studentNumber;
        $data["userType"] = $user->userType;
        $data["name"] = $user->name;
        $data["email"] = $user->email;
        $data["course"] = $user->course;
        $data["college"] = $user->college;
        $data["birthday"] = $user->birthday;
        $data["isBlocked"] = $user->isBlocked;
        $data["blockCount"] = $user->blockCount;
    }

    if ($userType == STUDENT_TUTOR) {
        $applications = $this->Application_Model->
            get_tutor_subjects($userId);
        foreach ($applications as $application) {
            $data["subjectRank_1"] = $application->
                subjectRank_1;
            $data["subjectRank_2"] = $application->
                subjectRank_2;
            $data["subjectRank_3"] = $application->
                subjectRank_3;
            $data["subjectRank_4"] = $application->
                subjectRank_4;
            $data["subjectRank_5"] = $application->
                subjectRank_5;
            $data["subjectRank_6"] = $application->
                subjectRank_6;
        }
    }

    $data["userId"] = $userId;
    $data["schedules"] = $schedules;
    $this->load->view("main/view_profile", $data)
        ;
}
}
?>

Search.php

<?php if ( ! defined("BASEPATH")) exit("No
    direct script access allowed");

/**
 * @author Quizon, Maria Reena I.
 * filename: Search.php
 * description: Controller for the search page
 *           in LRC_TMS
 * start date: March 25, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

```

```

class Search extends CI_Controller {

function __construct() {
parent::__construct();

$this->load->helper("url");
$this->load->library("form_validation");
$this->load->library("Validation_Controller"
);
$this->load->model("General_Model");
$this->load->model("Search_Model");

$this->load->view("templates/navigation");
$this->validation_controller->
check_validation();
}

public function index($results = NULL) {
$data["results"] = $results;
$this->load->view("main/search", $data);
}

public function process_search() {
$results = $this->Search_Model->search($this
->input->post("search"));
$this->index($results);
}
}
?>

```

## Report.php

```

<?php if ( ! defined("BASEPATH")) exit("No
direct script access allowed");

```

```

/**
 * @author Quizon, Maria Reena I.
 * filename: Report.php
 * description: Controller for the reports page
 in LRC_TMS
 * start date: March 31, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

```

```

class Report extends CI_Controller {

function __construct() {
parent::__construct();

$this->load->helper("url");
$this->load->helper("file");
$this->load->library("Pdf");
$this->load->library("form_validation");
$this->load->library("
Notification_Controller");
$this->load->library("Validation_Controller
");

$this->load->model("General_Model");
$this->load->model("Account_Model");
$this->load->model("Report_Model");
$this->validation_controller->
check_validation();
}

public function index ($msg = NULL) {

$breakdown = array();
$batchNumbers = array();
$accounts = $this->General_Model->get_table
("user");
$totalStudents = $this->Report_Model->
get_total_students();
$totalAppointments = $this->Report_Model->
get_total_appointments();
$totalMockExams = $this->Report_Model->
get_total_mockexams();

if ($accounts != DEFAULT_INTEGER) {
foreach ($accounts as $account) {
$batchNumber = $account->batchNumber;

```

```

if ($batchNumber > DEFAULT_INTEGER && !
in_array($batchNumber, $batchNumbers)) {
$batchNumbers[] = $batchNumber;
}
}
for ($index = 0; $index < sizeof(
$batchNumbers); $index++) {
$minUser = $this->Account_Model->
get_user($batchNumbers[$index], "min");
$maxUser = $this->Account_Model->
get_user($batchNumbers[$index], "max");
$breakdown[] = $minUser." - ".$maxUser;
}

$data["batchNumbers"] = $batchNumbers;
$data["breakdown"] = $breakdown;
$data["accounts"] = $accounts;
}

$data["msg"] = $msg;
$data["totalStudents"] = $totalStudents;
$data["totalMockExams"] = $totalMockExams;
$data["totalAppointments"] =
$totalAppointments;

$this->load->view("templates/navigation");
$this->load->view("reports/report_overview",
$data);
}

public function process_reports() {

$period = $_GET["period"];
$reportType = $_GET["reportType"];
$startDate = $_GET["startDate"];
$endDate = $_GET["endDate"];
$breakdownString = "";

if (strlen($startDate) < 1 || strlen(
$endDate) < 1 || $startDate == "undefined"
|| $endDate == "undefined") {
$breakdownString = "<br><div class = '
row'><div class = 'panel text-center lead
'><div class = 'panel-heading'>incomplete
input</div></div></div>";
$breakdownString .= "<div class = 'row'><
div class = 'col col-lg-4'></div><div class
= 'col col-lg-4 text-center'><button class
= 'cancel btn btn-default'>close</button
></div></div>";
}
else {
$records = array();
$breakdownString = "<br><table class = '
table table-bordered table-hover table-
responsive'><tr><td>date range</td><td>
reports</td></tr>";
$breakdown = $this->get_breakdown($period
, $startDate, $endDate);

for ($index_1 = 0; $index_1 < sizeof(
$breakdown); $index_1++) {
$breakdownString .= "<tr>";
$breakdownString .= "<td><input type =
'hidden' name = 'reportType' value = '".
$reportType.'"><input name = 'breakdown[]'
value = ' ".$breakdown[$index_1]."' class =
'form-control' readOnly = true /></td>";
if ($reportType == "studentsServed") {
$numberOfStudents = $this->
Report_Model->get_numberOfStudents(
$breakdown[$index_1]);
$breakdownString .= "<td><input name
= 'result[]' value = ' ".$numberOfStudents.'"
' class = 'form-control' readOnly = true
/></td>";
}
elseif ($reportType == "
frequentStudents") {
$frequentStudents = $this->
Report_Model->get_frequentStudents(
$breakdown[$index_1]);
if (sizeof($frequentStudents) >
DEFAULT_INTEGER) {

```

```

        $breakdownString .= "<td><input
name = 'result[]' value = '';
        for ($index_2 = 0; $index_2 <
sizeof($frequentStudents); $index_2++) {
            $breakdownString .=
$frequentStudents[$index_2]."#";
        }
        $breakdownString .= "' class = '
readonly form-control' readOnly = true /></
td>";
    }
    else {
        $breakdownString .= "<td><input
name = 'result[]' value = 'no reports'
class = 'readonly form-control' readOnly =
true /></td>";
    }
}
elseif ($reportType == "
frequentSubjects") {
    $frequentSubjects = $this->
Report_Model->get_frequentSubjects(
$breakdown[$index_1]);
    if (sizeof($frequentSubjects) >
DEFAULT_INTEGER) {
        $breakdownString .= "<td><input
name = 'result[]' value = '';
        for ($index_2 = 0; $index_2 <
sizeof($frequentSubjects); $index_2++) {
            $breakdownString .=
$frequentSubjects[$index_2]."#";
        }
        $breakdownString .= "' class = '
form-control' readOnly = true /></td>";
    }
    else {
        $breakdownString .= "<td><input
name = 'result[]' value = 'no reports'
class = 'form-control' readOnly = true /></
td>";
    }
}
$breakdownString .= "</tr>";
}
$breakdownString .= "<tr><td colspan =
'2' class = 'text-center'><input type = '
submit' value = 'generate pdf' class = 'btn
btn-success' />&nbsp;&nbsp;<button class = '
cancel btn btn-default'>close</button></td
></tr>";
$breakdownString .= "</table>";
}
echo json_encode($breakdownString);
}

public function process_account_pdf() {
    $today = date("d-m-y");
    $filename = $today."_accounts.pdf";
    $batchNumber = $this->input->post("
batchNumber");
    $accounts = $this->Account_Model->
get_accounts($batchNumber);

    $data["accounts"] = $accounts;
    $data["batchNumber"] = $batchNumber;
    $this->pdf->load_view("templates/account",
$data, true);
    $this->pdf->render();
    $this->pdf->stream($filename);
}

public function process_report_pdf() {
    $today = date("d-m-y");
    $filename = $today."_report.pdf";
    $reportType = $this->input->post("
reportType");
    $breakdown = $this->input->post("breakdown"
);
    $result = $this->input->post("result");

    $data["reportType"] = $reportType;
    $data["breakdown"] = $breakdown;
    $data["result"] = $result;
    $this->pdf->load_view("templates/report",
$data, true);
    $this->pdf->render();
    $this->pdf->stream($filename);
}

public function get_breakdown($period,
$startDate, $endDate) {
    $breakdown = array();
    switch ($period) {
        case "week": $breakdown = $this->
get_weeklyBreakdown($startDate, $endDate);
        break;
        case "month": $breakdown = $this->
get_monthlyBreakdown($startDate, $endDate);
        break;
        case "year": $breakdown = $this->
get_yearlyBreakdown($startDate, $endDate);
        break;
        case "semester": $breakdown = $this->
get_semesterBreakdown($startDate, $endDate)
; break;
        default: break;
    }
    return $breakdown;
}

public function get_weeklyBreakdown(
$startDate, $endDate) {
    $breakdown = array();
    $startDateStamp = strtotime($startDate);
    $endDateStamp = strtotime($endDate);
    $weeks = ceil(ceil(abs($endDateStamp -
$startDateStamp) / 86400) / 8);

    $thirtyArray = array(4, 6, 9, 11);
    $thirtyOneArray = array(1, 3, 5, 7, 8, 10,
12);

    for ($counter = 0; $counter < $weeks;
$counter++) {
        $startDateStr = explode("-", $startDate);
        $currDay = intval($startDateStr[2]);
        $currMonth = intval($startDateStr[1]);
        $currYear = intval($startDateStr[0]);

        $currDay += 7;
        if ($currDay > 31 && in_array($currMonth,
$thirtyOneArray)) {
            $currDay -= 31;
            if ($currMonth >= 12) {
                $currMonth = 1;
                $currYear += 1;
            }
        }
        else {
            $currMonth += 1;
        }
    }
    elseif ($currDay > 30 && in_array(
$currMonth, $thirtyArray)) {
        $currDay -= 30;
        $currMonth += 1;
    }
    elseif ($currDay > 28 && $currMonth == 2)
    {
        if ($currYear % 4 == 0) {
            $currDay -= 28;
        }
        elseif ($currYear % 4 > 0) {
            $currDay -= 29;
        }
        $currMonth += 1;
    }
    if ($currMonth < 10) {
        $currMonth = "0".$currMonth;
    }
    if ($currDay < 10) {
        $currDay = "0".$currDay;
    }
}

```







```

        elseif ($counter == 1) {
            $name = $column->getValue();
        }
        elseif ($counter == 2) {
            $college = $column->getValue();
        }
        elseif ($counter == 3) {
            $course = $column->getValue();
        }
        $counter++;
    }

    $userData = array(
        "userId" => $userId,
        "userType" => $userType,
        "password" => $password,
        "studentNumber" => $studentNumber,
        "name" => $name,
        "email" => $email,
        "course" => $course,
        "college" => $college,
        "batchNumber" => $batchNumber
    );
    $scheduleData = array(
        "userId" => $userId,
        "scheduleId" => $scheduleId,
        "hasSchedule" => $hasSchedule
    );

    $this->General_Model->
    create_table_entry("user", $userData);
    $this->General_Model->
    create_table_entry("schedule",
    $scheduleData);
}

$msg = FORM_SUCCESSFUL;
}

$this->index($msg);
}

public function block_account() {
    $msg = DEFAULT_STRING;
    $userId = $this->input->post("userId");
    $email = $this->input->post("email");
    $isBlocked = $this->input->post("isBlocked");
    $blockCount = intval($this->input->post("blockCount"));

    $userIds = $userId;
    $isWarned = DEFAULT_INTEGER;
    $eventId = DEFAULT_INTEGER;
    $notificationType = DEFAULT_INTEGER;
    $notificationDetails = DEFAULT_STRING;
    $notificationData = array();

    if (isset($_POST["block"])) {
        if ($blockCount >= 1) {
            $isBlocked = 1;
            $blockCount++;
        }
        elseif ($blockCount < 1) {
            $isWarned = 1;
            $blockCount = 1;
            $notificationType = WARNING;
            $notificationDetails = WARNING_MSG;
        }
    }
    if (isset($_POST["unblock"])) {
        $isWarned = DEFAULT_INTEGER;
        $isBlocked = DEFAULT_INTEGER;
        $notificationType = UNBLOCKED;
        $notificationDetails = UNBLOCKED_MSG;
    }

    $userData = array(
        "userId" => $userId,
        "isWarned" => $isWarned,
        "isBlocked" => $isBlocked,
        "blockCount" => $blockCount
    );

    if ($notificationType != DEFAULT_INTEGER) {
        $notificationData = $this->
        notification_controller->
        create_notification($notificationType,
        $eventId, $userIds, $notificationDetails);
    }

    if (sizeof($notificationData) > 1) {
        $this->General_Model->create_table_entry(
        "notification", $notificationData);
    }

    $this->General_Model->update_table_entry("
    user", $userData);

    if ($isBlocked > DEFAULT_INTEGER) {
        $emailType = EMAIL_BLOCK_STUDENT;
        $this->email_controller->create_email(
        $email, $emailType);
    }

    redirect(base_url("/profile/view_profile/" .
    $userId));
}

public function reset_tutors() {
    $resetType = STUDENT;
    $userType = STUDENT_TUTOR;
    $users = $this->General_Model->get_table("
    user");
    foreach ($users as $user) {
        if ($user->userType == $userType) {
            $userData = array(
                "userId" => $user->userId,
                "userType" => $resetType
            );
            $this->General_Model->
            update_table_entry("user", $userData);
        }
    }
    $this->index();
}

public function add_email() {
    $this->form_validation->set_rules("email",
    "email", "required|valid_email");
    if ($this->form_validation->run() == false)
    {
        redirect(base_url("/dashboard"));
    }
    else {
        $userId = $this->session->userdata("
        userId");
        $email = $this->input->post("email");
        $emailType = EMAIL_NEW_EMAIL;

        $result = $this->email_controller->
        create_email($email, $emailType);
        if ($result) {
            $userData = array(
                "userId" => $userId,
                "email" => $email
            );
            $this->General_Model->
            update_table_entry("user", $userData);
        }
        redirect(base_url("/logout"));
    }
}

public function change_password() {
    $userId = $this->session->userdata("userId");
    $this->form_validation->set_rules("
    newPassword", "newPassword", "required");
    $this->form_validation->set_rules("
    retypePassword", "retypePassword", "
    required");

    if ($this->form_validation->run() == false
    || ($this->input->post("newPassword") !=

```

```

$this->input->post("retypePassword"))) {
    $msg = FORM_INCOMPLETE_INPUT;
}
else {
    $retypePassword = $this->input->post("
retypePassword");
    $retypePassword = $this->phpass->
create_hash($retypePassword);
    $userData = array(
        "userId" => $userId,
        "password" => $retypePassword
    );
    $this->session->unset_userdata("password"
);
    $this->General_Model->update_table_entry(
"user", $userData);
}
redirect(base_url("/dashboard"));
}
}

```

## Lesson.php

```

<?php if ( ! defined("BASEPATH")) exit("No
direct script access allowed");

```

```

/**
 * @author Quizon, Maria Reena I.
 * filename: Lesson.php
 * description: Controller for all lesson
uploading functions in LRC_TMS
 * start date: March 2, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

```

```

class Lesson extends CI_Controller {

function __construct() {
    parent::__construct();

    $this->load->helper("form");
    $this->load->helper("url");
    $this->load->library("form_validation");
    $this->load->library("Validation_Controller
");

    $this->load->model("General_Model");
    $this->load->model("Lesson_Model");

    $this->validation_controller->
check_validation();
}

public function index () {

    $this->load->view("templates/navigation");
    $this->load->view("lesson/lesson_index");
}

public function create_lesson($msg = NULL) {

    $action = "Create Lesson";

    $data["msg"] = $msg;
    $data["action"] = $action;
    $data["edit"] = false;
    $this->load->view("templates/navigation");
    $this->load->view("lesson/create_lesson",
$data);
}

public function manage_lesson() {
    $msg = DEFAULT_STRING;
    if (isset($_POST["edit"])) {
        $action = "Edit Lesson";

        $lessonId = $this->input->post("lessonId"
);
        $lessonTitle = $this->input->post("
lessonTitle");

```

```

        $lessonDescription = $this->input->post("
lessonDescription");

        $data["edit"] = true;
        $data["action"] = $action;
        $data["lessonId"] = $lessonId;
        $data["lessonTitle"] = $lessonTitle;
        $data["lessonDescription"] =
$lessonDescription;

        $lessons = $this->General_Model->
get_table_entry("lesson", $lessonId);
        foreach ($lessons as $lesson) {
            $data["lessonLink"] = $lesson->
lessonLink;
            $data["subject"] = $lesson->subject;
            $data["code"] = $lesson->code;
        }

        $this->load->view("templates/navigation"
);
        $this->load->view("lesson/create_lesson",
$data);
    }

    if (isset($_POST["delete"])) {
        $lessonId = $this->input->post("lessonId"
);
        $subjectStr = $this->input->post("subject
");
        $subjectStr = explode(" ", $subjectStr);
        $subject = $subjectStr[0];
        $code = $subjectStr[1];
        $this->General_Model->delete_table_entry(
"lesson", $lessonId);
        redirect("/upload/lesson_controller/
get_lessons/".$subject."_".$code);
    }
}

public function get_lessons($msg = NULL) {

    $subjectStr = $this->uri->segment(3);
    $subjectStr = explode("_", $subjectStr);
    $subject = $subjectStr[0];
    $code = $subjectStr[1];

    if ($subject == "CMSC") {
        $subject = "Computer Science";
    }

    $lessons = $this->Lesson_Model->get_lessons
($subject, $code);
    if ($lessons != DEFAULT_INTEGER) {
        $data["lessons"] = $lessons;
    }

    $msg = $subject." ".$code;

    $data["msg"] = $msg;
    $this->load->view("templates/navigation");
    $this->load->view("lesson/view_lessons",
$data);
}

public function get_lesson_position() {

    $edit = $_GET["edit"];
    $code = $_GET["code"];
    $subject = $_GET["subject"];
    $lessonTitle = $_GET["lessonTitle"];
    $lessonPositionString = DEFAULT_STRING;

    $lessons = $this->Lesson_Model->get_lessons
($subject, $code);
    if ($lessons == DEFAULT_INTEGER) {
        $lessonPositionString = "<div class = '
col col-lg-3'><label>Insert</label></div>";
        $lessonPositionString .= "<div class = '
col col-lg-9'><select name = '
lessonPosition' class = 'form-control'></
div>";
        $lessonPositionString .= "<option value =
'1'>as first entry</option>";
    }
    else {

```

```

    $lessonPositionString = "<div class = '
col col-lg-3'><label>Insert&nbsp;</label></
div>";
    $lessonPositionString .= "<div class = '
col col-lg-4'><label class = 'radio-inline
'><input type = 'radio' name = 'insert'
value = 'before' />before</label>";
    $lessonPositionString .= "<label class =
'radio-inline'><input type = 'radio' name =
'insert' value = 'after' checked = '
checked' />after</label></div>";
    $lessonPositionString .= "<div class = '
col col-lg-5'><select name = '
lessonPosition' class = 'form-control'>";
    if ($edit == 1) {
        $lessonPositionString .= "<option value
= ''>as is</option>";
    }
    foreach ($lessons as $lesson) {
        if (strcmp($lesson->lessonTitle,
$lessonTitle) != DEFAULT_INTEGER) {
            $lessonPositionString .= "<option
value = '". $lesson->lessonPosition. "'>".
$lesson->lessonTitle. "</option>";
        }
    }
    $lessonPositionString .= "</select></div>";
    echo json_encode($lessonPositionString);
}

public function process_lesson() {
    $msg = DEFAULT_STRING;
    $path = "./uploads/lesson";

    if(!is_dir($path)) {
        mkdir($path, 0755, TRUE);
    }

    $config = array(
        "upload_path" => $path,
        "allowed_types" => "pdf",
        "max_size" => "4096000",
        "overwrite" => TRUE
    );

    $this->load->library("upload", $config);

    $lessonLink = DEFAULT_STRING;
    $lessonPosition = intval($this->input->post(
"lessonPosition"));

    if ($lessonPosition != DEFAULT_STRING) {
        if ($this->input->post("insert") == "
before") {
            $lessonPosition--;
        }
        elseif ($this->input->post("insert") == "
after") {
            $lessonPosition++;
        }
        else {
            $lessonPosition = 1;
        }
    }

    $lessonPosition = (string) $lessonPosition;

    $lessonData = array(
        "lessonTitle" => $this->input->post("
lessonTitle"),
        "lessonDescription" => $this->input->post(
"lessonDescription"),
        "lessonPosition" => $lessonPosition,
        "subject" => $this->input->post("subject"
),
        "code" => $this->input->post("code")
    );

    if (isset($_POST["create"])) {
        $lessonId = $this->General_Model->
get_tableId("lesson");

```

```

    if (!$this->upload->do_upload("lessonLink
")) {
        $lessonLink = NULL;
    }
    else {
        $lessonLink = $this->upload->data("
file_name");
    }

    $lessonData = array_merge($lessonData,
array("lessonId" => $lessonId, "lessonLink"
=> $lessonLink));
    $this->General_Model->create_table_entry(
"lesson", $lessonData);
}

if (isset($_POST["update"])) {
    $lessonId = $this->input->post("lessonId"
);
    if ($this->input->post("changeFile") == "
yes") {
        if (!$this->upload->do_upload("
lessonLinkNew")) {
            $lessonLink = NULL;
        }
        else {
            $lessonLink = $this->upload->data("
file_name");
        }
    }
    else {
        $lessonLink = $this->input->post("
lessonLinkHidden");
    }

    $lessonData = array_merge($lessonData,
array("lessonId" => $lessonId, "lessonLink"
=> $lessonLink));
    $this->General_Model->update_table_entry(
"lesson", $lessonData);
}

$msg = FORM_SUCCESSFUL;
redirect("/lessons/get_lessons/" . $this->
input->post("subject") . "_" . $this->input->
post("code"));
}

}

?>

```

## Mock Exam.php

```

<?php if ( ! defined("BASEPATH")) exit("No
direct script access allowed");

/**
 * @author Quizon, Maria Reena I.
 * filename: Mock_Exam.php
 * description: Controller for the mock exam
form in LRC_TMS
 * start date: February 29, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

class Mock_Exam extends CI_Controller {

    function __construct() {
        parent::__construct();

        $this->load->helper("form");
        $this->load->helper("url");
        $this->load->library("form_validation");
        $this->load->library("
Notification_Controller");
        $this->load->library("Validation_Controller
");
    }

    $this->load->model("General_Model");
}

```

```

$this->load->view("templates/navigation");
$this->validation_controller->
check_validation();
}

public function apply_mock_exam() {

    $userId = $this->session->userdata("userId"
);
    $studentName = $this->session->userdata("
name");
    $studentNumber = $this->session->userdata("
studentNumber");
    $mockExamId = $this->input->post("
mockExamId");
    $slots = $this->input->post("availableSlots
");

    if (isset($_POST["apply"])) {
        $studentData = array(
            "mockExamId" => $mockExamId,
            "userId" => $userId,
            "studentNumber" => $studentNumber,
            "studentName" => $studentName
        );

        $slots--;
        $mockExamData = array(
            "mockExamId" => $mockExamId,
            "availableSlots" => $slots
        );

        $this->General_Model->create_table_entry(
"mockexam_students", $studentData);
        $this->General_Model->update_table_entry(
"mockexam", $mockExamData);
    }
    if (isset($_POST["cancel"])) {
        $slots++;
        $this->General_Model->delete_user_entry(
"mockexam_students", $mockExamId, $userId);

        $mockExamData = array(
            "mockExamId" => $mockExamId,
            "availableSlots" => $slots
        );
        $this->General_Model->update_table_entry(
"mockexam", $mockExamData);
        $this->session->set_flashdata("
userId_remove", $userId);
    }
    echo FORM_SUCCESSFUL;
    redirect(base_url("/events/get_students/"
.$mockExamId));
}
}

```

## Upload.php

```

<?php if ( ! defined("BASEPATH")) exit("No
direct script access allowed");

/**
 * @author Quizon, Maria Reena I.
 * filename: Upload.php
 * description: Controller for all uploading
functions in LRC_TMS
 * start date: February 19, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

class Upload extends CI_Controller {

    function __construct() {
        parent::__construct();

        $this->load->helper("form");
        $this->load->helper("url");
        $this->load->library("form_validation");

```

```

$this->load->library("
Notification_Controller");
$this->load->library("Validation_Controller
");

$this->load->model("General_Model");
$this->load->view("templates/navigation");
$this->validation_controller->
check_validation();
}

public function index($msg = NULL, $event =
NULL) {
    $tab = "#overview";
    $config = array(
        "show_next_prev" => TRUE,
        "next_prev_url" => base_url("/events")
    );
    $config["template"] = "{table_open}<table
class='calendar'>{/table_open}
        {week_day_cell}<th class='
day_header'>{week_day}</th>{/
week_day_cell}
        {cal_cell_content}<span class='
day_listing'></span>&nbsp;{
&bull; {content}&nbsp;{/
cal_cell_content}
        {cal_cell_content_today}<div class
='today'><span class='
day_listing'>{day}</span>&bull;
        {content}</div>{/
cal_cell_content_today}
        {cal_cell_no_content}<span class='
day_listing'>{day}</span>&nbsp;{
}/cal_cell_no_content}
        {cal_cell_no_content_today}<div
class='today'><span class='
day_listing'>{day}</span></div
>{/cal_cell_no_content_today}";

    $entries = $this->General_Model->get_table(
"calendar");
    $this->load->library("calendar", $config);

    if ($entries != DEFAULT_INTEGER) {
        $data["entries"] = $entries;
    }
    if (isset($event)) {
        $tab = "#create-event";
    }

    $data["msg"] = $msg;
    $data["event"] = $event;
    $this->load->view("upload/upload_events",
$data);
}

public function manage_event() {

    $flag = 5;
    $msg = DEFAULT_STRING;
    $eventType = $this->input->post("eventType"
);
    $calendarId = $this->input->post("
calendarId");

    $event = array();
    $calendarData = array("calendarId" =>
$calendarId, "flag" => $flag);

    if (isset($_POST["edit"])) {
        $event["calendarId"] = $calendarId;
        $event["date"] = $this->input->post("date
");
        $event["eventId"] = $this->input->post("
eventId");
        $event["eventName"] = $this->input->post(
"eventName");
        $event["eventType"] = $this->input->post(
"eventType");
        $event["additionalDetails"] = $this->
input->post("additionalDetails");

        if ($eventType == "MOCK_EXAM") {

```

```

        $mockExamId = $this->input->post("
eventid");
        $mockExams = $this->General_Model->
get_table_entry("mockexam", $mockExamId);
        foreach ($mockExams as $mockExam) {
            $event["subject"] = $mockExam->
subject;
            $event["code"] = $mockExam->code;
            $event["time"] = $mockExam->time;
            $event["slots"] = $mockExam->
originalSlots;
            $event["discussionDate"] = $mockExam
->discussionDate;
            $event["discussionTime"] = $mockExam
->discussionTime;
        }
        $event["mockExamId"] = $mockExamId;
    }
}
if (isset($_POST["delete"])) {
    $this->General_Model->update_table_entry(
"calendar", $calendarData);
}
if (isset($_POST["finish"])) {
    $mockExamId = $this->input->post("eventId
");
    $this->General_Model->update_table_entry(
"calendar", $calendarData);
    $this->cancel_mockexam("finish");
}
if (isset($_POST["cancel"])) {
    $mockExamId = $this->input->post("eventId
");
    $this->General_Model->update_table_entry(
"calendar", $calendarData);
    $this->cancel_mockexam("cancel");
}
$this->index($msg, $event);
}

public function get_students() {
    $mockExamId = $this->uri->segment(3);
    $mockExams = $this->General_Model->
get_table_entry("mockexam", $mockExamId);
    $students = $this->General_Model->
get_table_entry("mockexam_students",
$mockExamId);
    if ($mockExams != DEFAULT_INTEGER) {
        $data["mockExams"] = $mockExams;
    }
    if ($students != DEFAULT_INTEGER) {
        $studentString = DEFAULT_STRING;
        foreach ($students as $student) {
            $studentString .= $student->userId."#";
        }
        if (null !== $this->session->flashdata("
userId_remove")) {
            $userId_remove = $this->session->
flashdata("userId_remove");
            $studentString = str_replace(
$userId_remove, "", $studentString);
        }
        $data["students"] = $students;
        $data["studentString"] = $studentString;
    }
    $this->load->view("upload/apply_mock_exam",
$data);
}

public function cancel_mockexam($action) {
    $detail = "was cancelled.";
    $status = EVENT_INACTIVE;
    $notificationType = DEFAULT_INTEGER;

    if ($action == "finish") {
        $status = EVENT_FINISHED;
        $detail = "is finished.";
    }

    $mockExamId = $this->input->post("eventId")
;
    $mockExams = $this->General_Model->
get_table_entry("mockexam", $mockExamId);

    $mockExamData = array(
        "mockExamId" => $mockExamId,
        "status" => $status
    );

    $notificationDetails = "The mock exam ".
$detail;
    $notificationData = $this->
notification_controller->
update_notification(LRC MOCK_EXAM_NEW,
$mockExamId, ALL, $status);
    $this->General_Model->update_table_entry("
mockexam", $mockExamData);
    $this->General_Model->update_table_entry("
notification", $notificationData);

    $notificationData = $this->
notification_controller->
create_notification(LRC MOCK_EXAM_CANCELLED
, $mockExamId, ALL, $notificationDetails);
    $this->General_Model->create_table_entry("
notification", $notificationData);
    redirect(base_url("/events"));
}

public function process_event() {
    $msg = DEFAULT_STRING;

    $mockExamData = array();
    $notificationData = array();

    $mockExamId = $this->General_Model->
get_tableId("mockexam");
    $calendarId = $this->General_Model->
get_tableId("calendar");
    $additionalDetails = $this->input->post("
additionalDetails");
    $eventName = $this->input->post("eventName"
);
    $eventType = $this->input->post("eventType"
);
    $date = $this->input->post("date");

    $flag = $this->set_flag($eventType);
    $isAllowed = $this->set_isAllowed(
$eventType);

    if (isset($_POST["update"])) {
        $calendarId = $this->input->post("
calendarId");
        $mockExamId = $this->input->post("
mockExamId");
    }

    $calendarData = array(
        "calendarId" => $calendarId,
        "eventName" => $eventName,
        "eventType" => $eventType,
        "date" => $date,
        "flag" => $flag,
        "isAllowed" => $isAllowed,
        "additionalDetails" => $additionalDetails
    );

    if ($eventType == "MOCK_EXAM") {
        $mockExamData = array(
            "mockExamId" => $mockExamId,
            "subject" => $this->input->post("
subject"),
            "code" => $this->input->post("code"),
            "date" => $this->input->post("date"),
            "time" => $this->input->post("time"),
            "originalSlots" => $this->input->post("
slots"),
            "availableSlots" => $this->input->post("
slots"),
            "discussionDate" => $this->input->post("
discussionDate"),
            "discussionTime" => $this->input->post("
discussionTime"),
            "additionalDetails" => $this->input->
post("additionalDetails")
        );
    }
}

```

```

);

$notificationDetails = array(
    $this->input->post("subject"),
    $this->input->post("code"),
    $this->input->post("date"),
    $this->input->post("time"),
    $this->input->post("discussionDate"),
    $this->input->post("discussionTime"),
    $this->input->post("additionalDetails")
);

$notificationDetails = implode("#",
notificationDetails);
$notificationData = $this->
notification_controller->
create_notification(LRC MOCK_EXAM_NEW,
$mockExamId, ALL, $notificationDetails);
$calendarData = array_merge($calendarData
, array("eventId" => $mockExamId));
}
elseif ($eventType == "LRC_EVENT") {
    $notificationData = $this->
notification_controller->
create_notification(LRC_ANNOUNCEMENT,
$calendarId, ALL, $additionalDetails);
}

if (isset($_POST["create"])) {
    $this->General_Model->create_table_entry(
"calendar", $calendarData);
    if ($eventType == "MOCK_EXAM") {
        $this->General_Model->
create_table_entry("notification",
$notificationData);
        $this->General_Model->
create_table_entry("mockexam",
$mockExamData);
    }
    elseif ($eventType == "lrcEvent") {
        $this->General_Model->
create_table_entry("notification",
$notificationData);
    }
}

if (isset($_POST["update"])) {
    $this->General_Model->update_table_entry(
"calendar", $calendarData);
    if ($eventType == "MOCK_EXAM") {
        $this->General_Model->
update_table_entry("mockexam",
$mockExamData);
    }
}

echo FORM_SUCCESSFUL;
redirect(base_url("/events"));
}

public function set_isAllowed($eventType) {
    $isAllowed = 1;
    switch ($eventType) {
        case "semBegin": $isAllowed = 0; break;
        case "semEnd": $isAllowed = 0; break;
        case "holiday": $isAllowed = 0; break;
        case "lrcEvent": $isAllowed = 1; break;
        case "mockExam": $isAllowed = 1; break;
        default: $isAllowed = 1; break;
    }
    return $isAllowed;
}

public function set_flag($eventType) {
    $flag = 0;
    switch ($eventType) {
        case "SEM_BEGIN": $flag = 1; break;
        case "SEM_END": $flag = 2; break;
        default: $flag = 0; break;
    }
    return $flag;
}
}
}

```

```
?>
```

## Account\_Model.php

```

<?php if ( ! defined("BASEPATH")) exit("No
direct script access allowed");

/**
 * @author Quizon, Maria Reena I.
 * filename: Account_Model.php
 * description: Model for the account table in
LRC_TMS
 * start date: April 15, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

class Account_Model extends CI_Model {

    function __construct(){
        parent::__construct();
    }

    function get_batchNumber() {

        $batchNumber = 10000;
        $this->db->select_max("batchNumber");
        $query = $this->db->get("user");
        $row = $query->row();
        if ($row->batchNumber > DEFAULT_INTEGER) {
            $batchNumber = $row->batchNumber + 1;
        }
        return $batchNumber;
    }

    function get_user($batchNumber, $condition) {
        $this->db->where("batchNumber", $batchNumber)
;
        if ($condition == "min") {
            $this->db->select_min("name");
            $query = $this->db->get("user");
            $row = $query->row();
            $minUser = $row->name;
            return $minUser;
        }
        elseif ($condition == "max") {
            $this->db->select_max("name");
            $query = $this->db->get("user");
            $row = $query->row();
            $maxUser = $row->name;
            return $maxUser;
        }
    }

    function get_accounts($batchNumber) {
        if ($batchNumber != DEFAULT_INTEGER) {
            $this->db->where("batchNumber", $batchNumber
);
        }
        $query = $this->db->get("user");
        return $query->result();
    }
}

```

## Application\_Model.php

```

<?php if ( ! defined("BASEPATH")) exit("No
direct script access allowed");

/**
 * @author Quizon, Maria Reena I.
 * filename: Application_Model.php
 * description: Model for the application table
in LRC_TMS
 * start date: Feburary 6, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

class Application_Model extends CI_Model {

```

```

function __construct(){
    parent::__construct();
}

function get_applications() {
    $this->db->select("*");
    $this->db->from("application");
    $this->db->join("endorsement", "endorsement.
        endorsementId = application.applicationId
        ");
    $this->db->join("user", "user.userId =
        application.userId");
    $query = $this->db->get();
    if ($query->num_rows() > 0) {
        return $query->result();
    }
    else {
        return 0;
    }
}

function get_application($endorsementId) {
    $this->db->where("endorsementId",
        $endorsementId);
    $query = $this->db->get("endorsement");
    if ($query->num_rows() > 0) {
        return $query->result();
    }
    else {
        return 0;
    }
}

function get_tutor_subject($userId, $subject)
    {
    $this->db->where("userId", $userId);
    $this->db->where("subjectRank_1", $subject);
    $this->db->order_by("applicationId", "desc");
    $query = $this->db->get("application");
    if ($query->num_rows() > 0) {
        return 1;
    }
    else {
        return 0;
    }
}

function get_tutor_subjects($userId) {
    $this->db->where("userId", $userId);
    $this->db->order_by("applicationId", "desc");
    $query = $this->db->get("application");
    if ($query->num_rows() > 0) {
        return $query->result();
    }
    else {
        return 0;
    }
}
}

```

## Appointment\_Model.php

```

<?php if ( ! defined("BASEPATH")) exit("No
    direct script access allowed");

/**
 * @author Quizon, Maria Reena I.
 * filename: Appointment_Model.php
 * description: Model for the appointment table
 * in LRC_TMS
 * start date: April 9, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

class Appointment_Model extends CI_Model {

    function __construct(){
        parent::__construct();
    }
}

```

```

function get_appointment_count($date, $time) {

    $appointmentCount = DEFAULT_INTEGER;
    $this->db->where("date", $date);
    $this->db->where("time", $time);

    $query = $this->db->get("appointment");
    if ($query->num_rows() > 0) {
        foreach ($query->result() as $row) {
            if ($row->status == EVENT_ACTIVE || $row->
                status == EVENT_PENDING) {
                $appointmentCount++;
            }
        }
    }

    return $appointmentCount;
}
}

```

## Calendar\_Model.php

```

<?php if ( ! defined("BASEPATH")) exit("No
    direct script access allowed");

/**
 * @author Quizon, Maria Reena I.
 * filename: Calendar_Model.php
 * description: Model for the calendar table in
 * LRC_TMS
 * start date: April 9, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

class Calendar_Model extends CI_Model {

    function __construct(){
        parent::__construct();
    }

    function get_isAllowed($date) {

        $isAllowed = 1;
        $this->db->where("date", $date);

        $query = $this->db->get("calendar");

        if ($query->num_rows() > 0) {
            $row = $query->row();
            $isAllowed = $row->isAllowed;
        }

        return $isAllowed;
    }

    function get_date($condition){

        $flag = DEFAULT_INTEGER;
        if ($condition == "SEM_BEGIN") {
            $flag = 1;
        }
        elseif ($condition == "SEM_END") {
            $flag = 2;
        }

        $this->db->order_by("date", "desc");
        $this->db->where("flag", $flag);
        $query = $this->db->get("calendar");
        if ($query->num_rows() > 0) {
            $row = $query->row();
            return $row->date;
        }
        else {
            return 0;
        }
    }
}
}

```



## Feedback\_Model.php

```
<?php if ( ! defined("BASEPATH")) exit("No
direct script access allowed");

/**
 * @author Quizon, Maria Reena I.
 * filename: Feedback_Model.php
 * description: Model for the feedback table in
LRC_TMS
 * start date: March 5, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

class Feedback_Model extends CI_Model {

function __construct(){
parent::__construct();
}

function get_feedback($startDate, $endDate,
$tutor_userId) {
$this->db->select("*");
$this->db->where("tutor_userId",
$tutor_userId);
$this->db->where("date >=", $startDate);
$this->db->where("date <=", $endDate);
$this->db->where("status", EVENT_FINISHED);
$this->db->from("appointment");
$this->db->join("feedback", "feedback.
feedbackId = appointment.appointmentId");
$query = $this->db->get();
if ($query->num_rows() > 0) {
return $query->result();
}
else {
return 0;
}
}

}
```

## General\_Model.php

```
<?php if ( ! defined("BASEPATH")) exit("No
direct script access allowed");

/**
 * @author Quizon, Maria Reena I.
 * filename: General_Model.php
 * description: Model for all common SQL
functionalities in LRC_TMS
 * start date: March 5, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

class General_Model extends CI_Model {

function __construct(){
parent::__construct();
}

function count_entries($table) {
$entryCount = $this->db->count_all($table);
return $entryCount;
}

function get_tableId($table) {

$tableId = 10000;
$this->db->select_max($table."Id");
$query = $this->db->get($table);
$row = $query->row();
if ($row->{$table."Id"} > 0) {
$tableId = $row->{$table."Id"} + 1;
}
return $tableId;
}

}
```

```
function get_table($table) {
$query = $this->db->get($table);
if ($query->num_rows() > 0) {
return $query->result();
} else {
return 0;
}
}

function get_user_entry($table, $userId) {
$this->db->where("userId", $userId);
$query = $this->db->get($table);
if ($query->num_rows() > 0) {
return $query->result();
} else {
return 0;
}
}

function get_table_entry($table, $tableId) {
$temp = $table;
if (strlen($tableId) > 5 && strpos($tableId,
"#") === false) {
$temp = "endorsementLink";
}
elseif ($table == "mockexam") {
$temp = "mockExamId";
}
elseif ($table == "mockexam_students") {
$temp = "mockExamId";
}
else {
$temp = $table."Id";
}
$this->db->where($temp, $tableId);
$query = $this->db->get($table);
if ($query->num_rows() > 0) {
return $query->result();
} else {
return 0;
}
}

function create_table_entry($table, $data) {
$this->db->insert($table, $data);
}

function update_table_entry($table, $data) {
extract($data);
if ($table == "notification") {
$this->db->where("notificationType",
$notificationType);
$this->db->where("eventId", $eventId);
}
elseif ($table == "feedback") {
$this->db->where("appointmentId",
$appointmentId);
}
elseif ($table == "mockexam") {
$this->db->where("mockExamId", $mockExamId);
}
elseif ($table == "schedule") {
$this->db->where("userId", $userId);
}
else {
$this->db->where($table."Id", ${$table."Id"}
);
}
$this->db->update($table, $data);
return true;
}

function delete_user_entry($table, $tableId,
$userId) {
if ($table == "mockexam_students") {
$this->db->where("mockExamId", $tableId);
}
else {
$this->db->where($table."Id", $tableId);
}
$this->db->where("userId", $userId);
$this->db->delete($table);
return true;
}
```

```

}

function delete_table_entry($table, $tableId)
{
    $this->db->where($table."Id", $tableId);
    $this->db->delete($table);
    return true;
}

function truncate_tables() {
    $this->db->truncate("notification");
    $this->db->truncate("calendar");
    return true;
}
}
}

```

## Lesson\_Model.php

```

<?php if ( ! defined("BASEPATH")) exit("No
direct script access allowed");

/**
 * @author Quizon, Maria Reena I.
 * filename: Lesson_Model.php
 * description: Model for the lesson table in
LRC_TMS
 * start date: March 4, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

class Lesson_Model extends CI_Model {

    function __construct(){
        parent::__construct();
    }

    function get_lessons($subject, $code) {
        if ($subject != NULL && $code != NULL) {
            $this->db->where("subject", $subject);
            $this->db->where("code", $code);
            $this->db->order_by("lessonPosition", "asc")
                ;
        }
        $query = $this->db->get("lesson");
        if ($query->num_rows() > 0) {
            return $query->result();
        }
        else {
            return 0;
        }
    }

}
}

```

## Notification\_Model.php

```

<?php if ( ! defined("BASEPATH")) exit("No
direct script access allowed");

/**
 * @author Quizon, Maria Reena I.
 * filename: Notifications_Model.php
 * description: Model for the notifications
table in LRC_TMS
 * start date: January 25, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

class Notification_Model extends CI_Model {

    function __construct(){
        parent::__construct();
    }

    function get_notifications($userId) {
        $notifications = array();
        $this->db->order_by("notificationId", "desc")
            ;
    }

```

```

$query = $this->db->get("notification");
if ($query->num_rows() > 0) {
    foreach ($query->result() as $row) {
        if (strpos($row->userIds, $userId) !==
            false || ($row->userIds == ALL && $this
->session->userdata("userType") > ADMIN
            )) {
                $notifications[] = $row;
            }
        }
    }
}
return $notifications;
}
}
}
}

```

## Report\_Model.php

```

<?php if ( ! defined("BASEPATH")) exit("No
direct script access allowed");

/**
 * @author Quizon, Maria Reena I.
 * filename: Report_Model.php
 * description: Model for all generating reports
in LRC_TMS
 * start date: April 3, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

class Report_Model extends CI_Model {

    function __construct(){
        parent::__construct();
        $this->load->model("General_Model");
    }

    function get_total_appointments() {
        $status = EVENT_FINISHED;
        $this->db->where("status", $status);
        return $this->db->count_all_results("
            appointment");
    }

    function get_total_mockexams() {
        $status = EVENT_FINISHED;
        $this->db->where("status", $status);
        return $this->db->count_all_results("mockexam
            ");
    }

    function get_total_students() {
        $status = EVENT_FINISHED;
        $numberOfStudents = DEFAULT_INTEGER;
        $this->db->where("status", $status);
        $query = $this->db->get("appointment");
        if ($query->num_rows() > 0) {
            foreach ($query->result() as $row) {
                $numberOfStudents += $row->numberOfStudents
                    ;
            }
        }
        $this->db->where("status", $status);
        $query = $this->db->get("mockexam");
        if ($query->num_rows() > 0) {
            foreach ($query->result() as $row) {
                $numberOfStudents += ($row->originalSlots -
                    $row->availableSlots);
            }
        }
        return $numberOfStudents;
    }
}

```

```

function get_numberOfStudents($breakdownDate)
{
    $status = EVENT_FINISHED;
    $numberOfStudents = DEFAULT_INTEGER;
    $breakdownDateStr = explode(":",
        $breakdownDate);
    $startDate = $breakdownDateStr[0];

```

```

$endDate = $breakdownDateStr[1];
$this->db->select("*");
$this->db->from("appointment");
$this->db->where("date >=", $startDate);
$this->db->where("date <=", $endDate);
$this->db->where("status", $status);

$query = $this->db->get();

if ($query->num_rows() > 0) {
    foreach ($query->result() as $row) {
        $numberOfStudents += $row->numberOfStudents
    }
}

$this->db->select("*");
$this->db->from("mockexam");
$this->db->where("date >=", $startDate);
$this->db->where("date <=", $endDate);
$this->db->where("status", $status);

$query = $this->db->get();

if ($query->num_rows() > 0) {
    foreach ($query->result() as $row) {
        $numberOfStudents += ($row->originalSlots -
            $row->availableSlots);
    }
}

return $numberOfStudents;
}

function get_frequentStudents($breakdownDate)
{
    $status = EVENT_FINISHED;
    $currName = DEFAULT_STRING;
    $students = array();
    $studentCount = array();
    $frequentStudents = array();
    $breakdownDateStr = explode(":",
        $breakdownDate);
    $startDate = $breakdownDateStr[0];
    $endDate = $breakdownDateStr[1];
    $this->db->where("date >=", $startDate);
    $this->db->where("date <=", $endDate);
    $this->db->where("status", $status);
    $query = $this->db->get("appointment");

    if ($query->num_rows() > 0) {
        foreach ($query->result() as $row) {
            $users = $this->General_Model->
                get_table_entry("user", $row->userId);
            foreach ($users as $user) {
                $currName = $user->name;
            }
            if (!in_array($currName, $students)) {
                $students[] = $currName;
            }
        }
        for ($index = 0; $index < sizeof($students);
            $index++) {
            $count = 0;
            $users = $this->General_Model->
                get_table_entry("user", $row->userId);
            foreach ($users as $user) {
                $currName = $user->name;
            }
            foreach ($query->result() as $row) {
                if ($students[$index] == $currName) {
                    $count++;
                }
            }
            $studentCount[] = $count;
        }
    }

    if (sizeof($students) > DEFAULT_INTEGER) {
        $frequentStudents = $this->sort_list(
            $students, $studentCount);
    }
}

}
return $frequentStudents;
}

function get_frequentSubjects($breakdownDate)
{
    $status = EVENT_FINISHED;
    $subjects = array();
    $subjectCount = array();
    $frequentSubjects = array();
    $breakdownDateStr = explode(":",
        $breakdownDate);
    $startDate = $breakdownDateStr[0];
    $endDate = $breakdownDateStr[1];
    $this->db->where("date >=", $startDate);
    $this->db->where("date <=", $endDate);
    $this->db->where("status", $status);
    $query = $this->db->get("appointment");

    if ($query->num_rows() > 0) {
        foreach ($query->result() as $row) {
            if (!in_array($row->subject." ".$row->code,
                $subjects)) {
                $subjects[] = $row->subject." ".$row->
                    code;
            }
        }
        for ($index = 0; $index < sizeof($subjects);
            $index++) {
            $count = 0;
            foreach ($query->result() as $row) {
                if ($subjects[$index] == $row->subject." "
                    . $row->code) {
                    $count++;
                }
            }
            $subjectCount[] = $count;
        }
    }

    if (sizeof($subjects) > DEFAULT_INTEGER) {
        $frequentSubjects = $this->sort_list(
            $subjects, $subjectCount);
    }
    return $frequentSubjects;
}

function sort_list($subjectArray, $countArray)
{
    $maxIndex = 0;
    $frequentArray = array();
    for ($index_1 = sizeof($subjectArray)-1;
        $index_1 > 0; $index_1--) {
        for ($index_2 = 1; $index_2 < $index_1;
            $index_2++) {
            if ($countArray[$maxIndex] < $countArray[
                $index_2]) {
                $maxIndex = $index_2;
            }
        }
        $temp_1 = $subjectArray[$index_1];
        $temp_2 = $countArray[$index_1];
        $subjectArray[$index_1] = $subjectArray[
            $maxIndex];
        $countArray[$index_1] = $countArray[
            $maxIndex];
        $subjectArray[$maxIndex] = $temp_1;
        $countArray[$maxIndex] = $temp_2;
    }

    for ($index = 0; $index < sizeof(
        $subjectArray); $index++) {
        $frequentArray[] = $subjectArray[$index]."-".
            $countArray[$index];
    }

    return $frequentArray;
}
}
}

```

## Schedule\_Model.php

```
<?php if ( ! defined("BASEPATH")) exit("No
direct script access allowed");

/**
 * @author Quizon, Maria Reena I.
 * filename: Schedule_Model.php
 * description: Model for the schedule table in
   LRC_TMS
 * start date: January 14, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

class Schedule_Model extends CI_Model {

function __construct(){
parent::__construct();
}

public function get_tutor_time($userId, $day)
{
$this->db->where("userId", $userId);
$query = $this->db->get("schedule");

if ($query->num_rows() == 1) {
$row = $query->row();
$scheduleData = array(
$day."_0830_1030" => $row->{$day."_
_0830_1030"},
$day."_1030_1200" => $row->{$day."_
_1030_1200"},
$day."_0100_0230" => $row->{$day."_
_0100_0230"},
$day."_0230_0400" => $row->{$day."_
_0230_0400"},
$day."_0400_0500" => $row->{$day."_
_0400_0500"}
);
return $scheduleData;
}
return 0;
}
}
?>
```

## Search\_Model.php

```
<?php if ( ! defined("BASEPATH")) exit("No
direct script access allowed");

/**
 * @author Quizon, Maria Reena I.
 * filename: Search_Model.php
 * description: Model for searching functions in
   LRC_TMS
 * start date: March 25, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

class Search_Model extends CI_Model {

function __construct(){
parent::__construct();
}

public function search($search) {

$adminId = 10001;
$minPercent = 75;
$student_userType = STUDENT;
$tutor_userType = STUDENT_TUTOR;
$studentString = "student";
$tutorString = "tutor";

$this->db->select("*");
$this->db->from("user");
$this->db->or_like("studentNumber", $search);
$this->db->or_like("name", $search);
$this->db->or_like("course", $search);
```

```

$this->db->or_like("college", $search);

similar_text($search, $tutorString,
$percent_1);
similar_text($search, $studentString,
$percent_2);

if ($search != 10001) {
$this->db->or_like("userId", $search);
}
if ($percent_1 >= $minPercent) {
$this->db->or_like("userType",
$tutor_userType);
}
if ($percent_2 >= $minPercent) {
$this->db->or_like("userType",
$student_userType);
$this->db->or_like("userType",
$tutor_userType);
}

$query = $this->db->get();
return $query->result();
}
}
?>
```

## User\_Model.php

```
<?php if ( ! defined("BASEPATH")) exit("No
direct script access allowed");

/**
 * @author Quizon, Maria Reena I.
 * filename: User_Model.php
 * description: Model for the users table in
   LRC_TMS
 * start date: January 13, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

class User_Model extends CI_Model {

function __construct(){
parent::__construct();
$this->load->library("PHPass");
}

public function get_tutors($userType) {
$this->db->where("userType", $userType);
$query = $this->db->get("user");
if ($query->num_rows() > 0) {
return $query->result();
} else {
return 0;
}
}

public function validate_email($userId, $email
) {
$this->db->where("userId", $userId);
$this->db->where("email", $email);
$query = $this->db->get("user");

if ($query->num_rows() == 1) {
return true;
}
return false;
}

public function validate_user($userId,
$password) {

$this->db->where("userId", $userId);
$query = $this->db->get("user");

if ($query->num_rows() == 1) {
$row = $query->row();

if ($row->isBlocked == 1) {
```

```

    return "blocked";
}
else {
    $check = false;
    if (strlen($row->password) < 8) {
        if ($password == $row->password) {
            $check = true;
        }
    }
    else {
        $check = $this->phpass->check_password(
            $password, $row->password);
    }

    if ($check) {
        $data = array(
            "userId" => $row->userId,
            "userType" => $row->userType,
            "studentNumber" => $row->studentNumber,
            "name" => $row->name,
            "email" => $row->email,
            "course" => $row->course,
            "college" => $row->college,
            "birthday" => $row->birthday,
            "profilePicture" => $row->profilePicture,
            "hasAttended" => $row->hasAttended,
            "hasApplication" => $row->hasApplication,
            "approvalCount" => $row->approvalCount,
            "loginCount" => $row->loginCount,
            "validated" => true
        );

        if (strlen($row->password) < 8) {
            $data = array_merge($data, array("
                password" => $row->password));
        }

        $this->session->set_userdata($data);
        return "true";
    }
    else {
        return "false";
    }
}
return "false";
}
}
?>

```

## accept\_endorsement.php

```

<?php
/**
 * @author Quizon, Maria Reena I.
 * filename: accept_endorsement.php
 * description: view for the accept endorsement
 * page in LRC_TMS
 * start date: February 21, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

$cancel = array("class" => "btn btn-danger btn-
    block");
$submit = array("class" => "btn btn-success btn
    -block");

?>
<html>
<head>
</head>
<body>
    <div class = "container-fluid">
        <div class = "row">
            <div class="col-lg-3">
                <!--Affix start-->
                <ul id = "sidebar" class = "nav nav-pills
                    nav-stacked text-justified">
                    <li class = "active"><a data-toggle = "tab
                        " href="#">Accept Endorsement Request
                    </a></li>
                    <li><a href="#">Answer Endorsement </a></li>
                </ul>
            </div>

```

```

                <li><a href="#">Finish Endorsement </a></li>
            </ul>
            <!--Affix end-->
        </div>
        <div class="col-lg-9">
            <!--Content start-->
            <div class="wrapper">
                <div class = "tab-content">
                    <div id = "accept-endorsement">
                        <div class = "col col-lg-12">
                            <div class = "panel panel-heading lead"
                                >Accept Endorsement Request</div>
                            <div class = "panel-body">
                                <div class = "row">
                                    <div class = "col col-lg-3"></div>
                                    <div class = "col col-lg-6 text-
                                        center">
                                        <img src = "<?php echo $path;?>"
                                            class = "img-thumbnail"><br/>
                                        This is the student you are going to
                                        endorse for.<br/>Are you
                                        willing to accept this student's
                                        endorsement request?<br/>
                                    </div>
                                    <div class = "col col-lg-3"></div>
                                </div>
                                <div class = "row"><br/>
                                <div class = "col col-lg-3"></div>
                                <div class = "col col-lg-3 text-
                                    center">
                                    <?php
                                        echo form_open("endorsement/
                                            process_acceptance/" .
                                                $endorsementLink);
                                        echo form_submit("accept", "yes",
                                            $submit);
                                    ?>
                                </div>
                                <div class = "col col-lg-3 text-
                                    center">
                                    <?php
                                        echo form_submit("reject", "no",
                                            $cancel);
                                        echo form_close();
                                    ?>
                                </div>
                                <div class = "col col-lg-3"></div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</body>

```

## apply\_1.php

```

<?php
/**
 * @author Quizon, Maria Reena I.
 * filename: apply_1.php
 * description: first view for the apply page in
 * LRC_TMS
 * start date: February 5, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

$submit = array("class" => "btn btn-success");
$input = array("class" => "form-control", "
    required" => true);
$readonly = array("class" => "readonly form-
    control", "readOnly" => true);
$birthday = $this->session->userdata("birthday"
    );

if (!isset($birthday)) { $birthday =
    DEFAULT_STRING; }
if (isset($msg)) { echo $msg; }

?>

```

```

<html>
<head>
<link href="<?php echo base_url('application/
css/login.css'); ?>" rel="stylesheet">
<link href="<?php echo base_url('application/
css/dashboard.css'); ?>" rel="stylesheet">
<link href="<?php echo base_url('application/
css/application.dashboard.css'); ?>" rel="
stylesheet">
<script src="<?php echo base_url('application/
js/jquery.application.js'); ?>"></script>
<script charset="utf-8">
$(function() {
$("#datetimepicker").datetimepicker({
pickTime: false
});
$("#done").click(function() {
$(".modal").hide();
$(".modal-backdrop").hide();
});
});
</script>
</head>
<body>
<div class = "container-fluid">
<div class = "row">
<div class="col-lg-3">
<!--Affix start-->
<ul id = "sidebar" class = "nav nav-pills
nav-stacked text-justified">
<li class = "active"><a data-toggle = "tab
" href="#">Application - Step 1 of 3</
a></li>
<li><a href="#">Application - Step 2 of
3</a></li>
<li><a href="#">Application - Step 3 of
3</a></li>
</ul>
<!--Affix end-->
</div>
<div class="col-lg-9">
<!--Content start-->
<div class="wrapper">
<div class = "tab-content">
<div id = "apply-1">
<div class = "panel panel-heading lead">
Apply to become a tutor - Step 1 of
3</div>
<div class = "panel-body">
<?php
echo form_open("apply/process_1");
?>
<div class = "row">
<div class = "col col-lg-4"><label>
Your Name</label></div>
<div class = "col col-lg-8">
<?php
echo form_input("name", $this->
session->userdata("name"),
$readonly);
?>
</div>
</div>
<div class = "row">
<div class = "col col-lg-4"><label>
Your Student Number</label></div>
<div class = "col col-lg-8">
<?php
echo form_input("studentNumber",
substr_replace($this->session->
userdata("studentNumber"), " - "
, 4, 0), $readonly);
?>
</div>
</div>
<div class = "row">
<div class = "col col-lg-4"><label>
Your Email Address</label></div>
<div class = "col col-lg-8">
<?php
echo form_input("email", $this->
session->userdata("email"),
$readonly);
?>
</div>
</div>
<div class = "row">
<div class = "col col-lg-4"><label>

```

```

Your Birthday</label></div>
<div class = "col col-lg-8">
<input type = "date" name = "birthday
" value = "<?php echo $birthday;
?" id = "birthday" class = "form
-control" required/>
</div>
</div>
<div class = "row">
<div class = "col col-lg-4"></div>
<div class = "col col-lg-8 text-center
"><br/>
<button id = "add-organization" class
= "btn btn-default btn-sm">add
another organization</button>
<button id = "no-organization" class
= "btn btn-default btn-sm">no
organization</button>
</div>
</div>
<div id = "organization" class = "col
col-lg-12"></div>
<div class = "col col-lg-12 text-center
"><br/>
<?php
echo form_submit("proceed", "proceed
to step 2 of 3", $submit);
echo form_close();
?>
<p></p>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<?php
require_once("reminder_modal.php");
?>
</body>

```

## apply\_2.php

```

<?php
/**
 * @author Quizon, Maria Reena I.
 * filename: apply_2.php
 * description: second view for the apply page
 in LRC_TMS
 * start date: February 6, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

$submit = array("class" => "btn btn-success");
$input = array("class" => "form-control", "
required" => true);
$subjects = array("" => "no subject", "
Chemistry" => "Chemistry", "Mathematics" =>
"Mathematics", "Physics" => "Physics",
"Biology" => "Biology", "Computer-Science"
=> "Computer-Science", "
Communications" => "Communications");

?>
<html>
<head>
<script src="<?php echo base_url('application
/js/jquery.application.js'); ?>"></script>
</head>
<body>
<div class = "container-fluid">
<div class = "row">
<div class="col-lg-3">
<!--Affix start-->
<ul id = "sidebar" class = "nav nav-pills
nav-stacked text-justified">
<li><a href="#">Application - Step 1 of
3</a></li>
<li class = "active"><a data-toggle = "
tab" href="#">Application - Step 2 of
3</a></li>

```

```

    <li><a href="#">Application - Step 3 of
      3</a></li>
  </ul>
</ul>
<!--Affix end-->
</div>
<div class="col-lg-9">
  <!--Content start-->
  <div class="wrapper">
    <div class="tab-content">
      <div id="apply-2">
        <div class="panel panel-heading lead">
          Apply to become a tutor - Step 2 of
          3</div>
        <div class="panel-body">
          <?php
            echo form_open_multipart("apply/
              process_2");
          ?>
          <div id="subject-rank">
            <?php
              for ($counter = 1; $counter < 7;
                $counter++) {
                echo "<div id='subject-rank-".
                  $counter."-parent-div' class='
                    row form-group'>";
                echo "<div class='col col-lg-4'><
                  label>".form_label("rank ".
                    $counter)."</label></div>";
                echo "<div class='col col-lg-8'>"
                  ;
                echo "<select index='".
                  $counter."
                  ' name='subjectRank_".
                  $counter."' id='subject-rank-
                  ".$counter."' class='subject-
                  rank form-control' required>";
                foreach ($subjects as $key =>
                  $value) {
                  echo "<option value='".
                    $key."'>"
                    . $value."</option>";
                }
                echo "</select></div>";
                echo "<div id='subject-rank-".
                  $counter."-div'></div>";
                echo "</div>";
              }
            ?>
          </div>
          <div class="row">
            <div class="col col-lg-4"><label>
              Academic Standing (all semesters)
            </label></div>
            <div class="col col-lg-8 form-group"
              >
              <?php
                echo form_upload("academicStanding",
                  "", $input);
              ?>
            </div>
          </div>
          <div class="col col-lg-12 text-center"
            ><br/>
            <?php
              echo form_submit("proceed", "proceed
                to step 3 of 3", $submit);
              echo form_close();
            ?>
            <p></p>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
</body>
</html>

```

## apply\_3.php

```

<?php
/**
 * @author Quizon, Maria Reena I.
 * filename: apply_3.php

```

```

 * description: third view for the apply page in
  LRC_TMS
 * start date: February 9, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

$submit = array("class" => "btn btn-success");
$input = array("class" => "form-control", "
  required" => true);
$xml = simplexml_load_file(base_url("/
  application/views/apply/faculty.xml"));
$profilePicture = $this->session->userdata("
  profilePicture");

?>
<html>
<head>
  <link href="<?php echo base_url('application/
    css/application.css')?>" rel="stylesheet">
  <script src="<?php echo base_url('application
    /js/jquery.application.js'); ?>"></script>
</head>
<body>
  <div class="container-fluid">
    <div class="row">
      <div class="col-lg-3">
        <!--Affix start-->
        <ul id="sidebar" class="nav nav-pills
          nav-stacked text-justified">
          <li><a href="#">Application - Step 1 of
            3</a></li>
          <li><a href="#">Application - Step 2 of
            3</a></li>
          <li class="active"><a data-toggle="tab
            " href="#">Application - Step 3 of 3</
            a></li>
        </ul>
        <!--Affix end-->
      </div>
      <div class="col-lg-9">
        <!--Content start-->
        <div class="wrapper">
          <div class="tab-content">
            <div id="apply-2">
              <div class="panel panel-heading lead">
                Apply to become a tutor - Step 3 of
                3</div>
              <div class="panel-body">
                <?php
                  $facultyOptions = array();
                  foreach ($xml->faculty as $faculty) {
                    $facultyOptions = array_merge(
                      $facultyOptions, array("$faculty
                        ->email" => $faculty->name));
                  }

                  echo form_open_multipart("apply/
                    process_3");
                ?>
                <div class="row">
                  <div class="col col-lg-4"><label>
                    Current Profile Image</label></div>
                  <div class="col col-lg-8 text-center"
                    >
                    <?php
                      if (isset($profilePicture)) {
                        echo "<img src='".
                          $path."' class
                          = 'application-image img-
                          thumbnail'><p></p>";
                        echo "This is already your profile
                          picture. Please make sure this
                          is really you.<br/>";
                        echo form_hidden("profilePicture",
                          $profilePicture);
                      }
                      else {
                        echo "<div class='col col-lg
                          -2'></div>";
                        echo "<div class='col col-lg-8'>"
                          ;
                        echo form_upload("profilePicture",
                          "", $input);
                        echo "</div>";
                        echo "<div class='col col-lg
                          -2'></div>";
                      }
                }
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

    }
    echo form_hidden("hasPicture",
        $hasPicture);
    ?>
</div>
</div>
<div class = "row">
<div class = "col col-lg-4"><label>
    Faculty Name</label></div>
<div class = "col col-lg-8">
<div class = "col col-lg-2"></div>
<div class = "col col-lg-8"><br/>
    <?php
        echo form_dropdown("faculty",
            $facultyOptions, "", $input);
    ?>
</div>
<div class = "col col-lg-2"></div>
</div>
</div>
<div class = "row">
<div class = "col col-lg-4">
<label>Free Time Schedule</label>
</div>
<div class = "col col-lg-8 text-center">
    <br/>
<div class = "col col-lg-3"></div>
<div class = "col col-lg-6 panel
    panel-body">
    <?php
        echo $schedule;
    ?>
<br/><div class = "panel panel-xs">
    <p class = "small">
        This is where you input your free
        time schedule for the
        current semester.
        Click the red buttons above to
        switch from <br/><br/>
    <a class = "btn btn-success btn-xs"
        ">free during this time slot</a>
    <br/><br/>
    <a class = "btn btn-danger btn-xs"
        ">busy during this time slot</a>
    </p>
</div>
</div>
<div class = "col col-lg-3"></div>
</div>
</div>
<div class = "text-center"><br/>
    <?php
        echo form_submit("submit", "submit
            application form", $submit);
        echo form_close();
    ?>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</body>

```

## dashboard.php - application

```

<?php if ( ! defined("BASEPATH")) exit("No
    direct script access allowed");

```

```

/**
 * @author Quizon, Maria Reena I.
 * filename: dashboard.php
 * description: view for the application
    dashboard page in LRC_TMS
 * start date: February 25, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

```

```

$cancel = array("class" => "btn btn-danger");

```

```

$submit = array("class" => "btn btn-success");

```

```

function print_application($application) {
    $path = base_url("./uploads/" . $application->
        userId . "/" . $application->profilePicture);
    if (!isset($application->profilePicture)) {
        $path = base_url("./uploads/default.png");
    }
}

```

```

$readonly = array("class" => "readonly form-
    control", "readOnly" => true);

```

```

echo "<div class = 'row panel'>";
echo "<div class = 'panel panel-heading lead
    text-right'>.date("F j Y, g:m a",
    strtotime($application->lastModified))."
    </div>";

```

```

echo "<div class = 'panel-body'>";
echo "<input type = 'hidden' name = '
    applicationId' value = '". $application->
    applicationId.' />";

```

```

echo "<input type = 'hidden' name = '
    endorsementId' value = '". $application->
    endorsementId.' />";

```

```

echo "<div class = 'col col-lg-4 text-center'
    '>";

```

```

echo "<img src = '". $path.'" class = 'img-
    thumbnail profile-image'>";

```

```

echo "</div>";
echo "<div class = 'col col-lg-8'>";
echo "<table class = 'table table-condensed
    table-responsive'>";

```

```

echo "<tr><td>.form_label("Application Id"
    )."</td><td>.anchor("#", $application
    ->applicationId, array("data-toggle" =>
    "modal", "data-target" => "#
    application-modal", "class" => "
    application-id"))."</td></tr>";

```

```

echo "<tr><td>.form_label("Endorsement Id"
    )."</td><td>.anchor("#", $application
    ->endorsementId, array("data-toggle" =>
    "modal", "data-target" => "#
    endorsement-modal", "class" => "
    endorsement-id"))."</td></tr>";

```

```

echo "<tr><td>.form_label("Applicant's
    User Id")."</td><td>.form_input("
    userId", $application->userId,
    $readonly)."</td></tr>";

```

```

echo "<tr><td>.form_label("Applicant's
    Name")."</td><td>.form_input("name",
    $application->name, $readonly)."</td></tr>";

```

```

echo "<tr><td>.form_label("Student Number"
    )."</td><td>.form_input("studentNumber
    ", substr_replace($application->
    studentNumber, " - ", 4, 0), $readonly)
    ."</td></tr>";

```

```

echo "</table>";
echo "</div>";
echo "</div>";
echo "</div>";
}

```

```

?>
<html>
<head>

```

```

<link href= "<?php echo base_url('application/
    css/profile.css');?>" rel="stylesheet"

```

```

<link href= "<?php echo base_url('application
    /css/application.dashboard.css'); ?>" rel=
    "stylesheet"

```

```

<script src= "<?php echo base_url('application
    /js/jquery.application.dashboard.js');?>"
    ></script>

```

```

</head>
<body>

```

```

<div class = "container-fluid">

```

```

<div class = "row">

```

```

<div class="col-lg-3">

```

```

<!--Affix start-->

```

```

<ul id = "sidebar" class = "nav nav-pills
    nav-stacked text-justified">

```

```

<li class = "active"><a data-toggle = "tab
    " href="#pending">Pending</a></li>

```

```

<li><a data-toggle = "tab" href="#approved
    ">Approved</a></li>

```



```

</li><a data-toggle = "tab" href="#rejected
">Rejected</a></li>
</ul>
<!--Affix end-->
</div>
<div class="col-lg-9">
<!--Content start-->
<div class="wrapper">
<div class = "tab-content">
<div id = "pending" class = "tab-pane
fade in active">
<div class = "col col-lg-1"></div>
<div class = "col col-lg-10">
<?php
$counter = 0;
if (isset($applications)) {
foreach ($applications as $application
) {
if ($application->applicationStatus
== EVENT_PENDING) {
echo form_open("apply/dashboard/
process_approval");
print_application($application);
echo "<div class = 'text-center'>";
echo form_submit("accept", "accept
application", $submit);
echo form_submit("reject", "reject
application", $cancel);
echo "</div>";
echo form_close();
$counter++;
}
}
}
if ($counter == 0) {
echo "<div class = 'panel text-center
lead'><div class = 'panel-heading
'>no pending applications yet</div
></div>";
}
?>
</div>
</div>
<div id = "approved" class = "tab-pane
fade">
<div class = "col col-lg-1"></div>
<div class = "col col-lg-10">
<?php
$counter = 0;
if (isset($applications)) {
foreach ($applications as $application
) {
if ($application->applicationStatus
== EVENT_ACTIVE) {
print_application($application);
$counter++;
}
}
}
if ($counter == 0) {
echo "<div class = 'panel text-center
lead'><div class = 'panel-heading
'>no approved applications yet</
div></div>";
}
?>
</div>
</div>
<div id = "rejected" class = "tab-pane
fade">
<div class = "col col-lg-1"></div>
<div class = "col col-lg-10">
<?php
$counter = 0;
if (isset($applications)) {
foreach ($applications as $application
) {
if ($application->applicationStatus
== EVENT_INACTIVE) {
print_application($application);
$counter++;
}
}
}
}
if ($counter == 0) {

```

```

echo "<div class = 'panel text-center
lead'><div class = 'panel-heading
'>no rejected applications yet</
div></div>";
}
?>
</div>
</div>
<div id = "application-modal" class = "
modal fade" role = "dialog">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header"><button
type="button" class="close" data-
dismiss="modal">&times;</button>
<h4 class="modal-title">Application
Details</h4></div>
<div id = "application-form" class="
modal-body"></div>
</div>
</div>
<div id = "endorsement-modal" class = "
modal fade" role = "dialog">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header"><button
type="button" class="close" data-
dismiss="modal">&times;</button>
<h4 class="modal-title">Endorsement
Details</h4></div>
<div id = "endorsement-form" class="
modal-body"></div>
</div>
</div>
</div>
</div>
</div>
</div>
</body>

```

## endorse.php

```

<?php
/**
 * @author Quizon, Maria Reena I.
 * filename: endorse.php
 * description: view for the endorsement page in
LRC_TMS
 * start date: February 10, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

$radio = array("required" => true);
$input = array("class" => "form-control");
$submit = array("class" => "btn btn-success");
$readonly = array("class" => "readonly form-
control", "readOnly" => true);

$xml = simplexml_load_file(base_url("/
application/views/apply/endorsement.xml"));

$attendanceOptions = array(
"1" => "no absences and no tardiness",
"2" => "1-2 absences or 1-2 tardiness",
"3" => "3-4 absences or 3-4 tardiness",
"4" => "5 and above absences or 5 and above
tardiness"
);
$recommendationOptions = array(
"1" => "I strongly recommend the applicant as
peer tutor",
"2" => "I recommend the applicant as peer
tutor",
"3" => "I recommend the applicant as peer
tutor, with reservations",
"4" => "I do not recommend the applicant as
peer tutor"
);
if (isset($msg)) { echo $msg; }
?>

```









```

        nav-stacked text-justified">
<li class = "active"><a data-toggle = "tab
    " href="#pending">Pending</a></li>
<li><a data-toggle = "tab" href="#finished
    ">Finished</a></li>
<li><a data-toggle = "tab" href="#
    cancelled">Cancelled</a></li>
</ul>
<!--Affix end-->
</div>
<div class="col-lg-9">
<!--Content start-->
<div class="wrapper">
<div class = "tab-content">
<div id = "pending" class = "tab-pane
    fade in active">
<div class = "col col-lg-1"></div>
<div class = "col col-lg-10">
<?php
$counter = 0;
if (isset($appointments)) {
foreach ($appointments as $appointment
    ) {
if ($appointment->status ==
    EVENT_ACTIVE || $appointment->
    status == EVENT_PENDING) {
echo form_open("appointment/
    dashboard/check_attendance");
print_appointment($appointment);
echo "<div class = 'row text-center
    '>";
if (strtotime($today) >= strtotime(
    $appointment->date) &&
    $appointment->status == 1) {
echo "<div class = 'row form-inline
    '>";
echo "<label>Number of students who
    attended: </label>&nbsp;<input
    type = 'number' name =
    'numberOfStudents' value = '".
    $appointment->numberOfStudents.
    "' min = '1' max = '".
    $appointment->numberOfStudents.
    "' class = 'form-control' />&
    &nbsp; ";
echo form_submit("didAttend", "
    tutee attended", $submit);
echo form_submit("didNotAttend", "
    tutee did not attend", $cancel)
    ;
echo "</div><br/>";
}
echo form_submit("cancel", "cancel
    appointment", $cancel);
echo "</div>";
echo form_close();
$counter++;
}
}
}
if ($counter == 0) {
echo "<div class = 'panel text-center
    lead'><div class = 'panel-heading
    '>no pending appointments</div></
    div>";
}
?>
</div>
<div class = "col col-lg-1"></div>
</div>
<div id = "finished" class = "tab-pane
    fade">
<div class = "col col-lg-1"></div>
<div class = "col col-lg-10">
<?php
$counter = 0;
if (isset($appointments)) {
foreach ($appointments as $appointment
    ) {
if ($appointment->status ==
    EVENT_FINISHED) {
print_appointment($appointment);
$counter++;
}
}
}
}
}

```

```

if ($counter == 0) {
echo "<div class = 'panel text-center
    lead'><div class = 'panel-heading
    '>no finished appointments yet</
    div></div>";
}
?>
</div>
<div class = "col col-lg-1"></div>
</div>
<div id = "cancelled" class = "tab-pane
    fade">
<div class = "col col-lg-1"></div>
<div class = "col col-lg-10">
<?php
$counter = 0;
if (isset($appointments)) {
foreach ($appointments as $appointment
    ) {
if ($appointment->status ==
    EVENT_INACTIVE) {
print_appointment($appointment);
$counter++;
}
}
}
if ($counter == 0) {
echo "<div class = 'panel text-center
    lead'><div class = 'panel-heading
    '>no cancelled appointments</div
    ></div>";
}
?>
</div>
<div class = "col col-lg-1"></div>
</div>
</div>
</div>
</div>
</div>
</div>
</body>

```

## feedback.php

```

<?php
/**
 * @author Quizon, Maria Reena I.
 * filename: feedback.php
 * description: view for the feedback page in
    LRC_TMS
 * start date: February 2, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

$xml = simplexml_load_file(base_url("/
    application/views/feedback/questions.xml"))
    ;

$input = array("class" => "form-control");
$readonly = array("class" => "readonly form-
    control", "readOnly" => true);
$submit = array("class" => "btn btn-success");

?>
<html>
<head></head>
<body>
<div class = "container-fluid">
<div class = "row">
<div class="col-lg-3">
<!--Affix start-->
<ul id = "sidebar" class = "nav nav-pills
    nav-stacked text-justified">
<li class = "active"><a data-toggle = "tab
    " href="#">Answer Feedback</a></li>
</ul>
<!--Affix end-->
</div>
<div class="col-lg-9">
<!--Content start-->
<div class="wrapper">
<div class = "tab-content">
<div id = "feedback">

```







```

<li class = "active"><a data-toggle = "tab
" href="#create-lesson"><?php echo
$action; ?></a></li>
</ul>
<!--Affix end-->
</div>
<div class="col-lg-9">
<!--Content start-->
<div class="wrapper">
<div class = "tab-content">
<div id = "create-lesson">
<div class = "panel panel-heading lead"
><?php echo $action; ?></div>
<div class = "panel-body">
<div class = "col col-lg-9 form-group">
<?php
echo form_open_multipart("lessons/
process_lesson");
echo form_hidden("lessonId",
$lessonId);
echo "<div class = 'col col-lg-3'>";
echo form_label("Lesson title");
echo "</div>";
echo "<div class = 'col col-lg-9'>";
echo form_input("lessonTitle",
$lessonTitle, $input);
echo "</div>";
echo "<div class = 'col col-lg-3'>";
echo form_label("Lesson description")
;
echo "</div>";
echo "<div class = 'col col-lg-9'>";
echo form_textarea("lessonDescription
", $lessonDescription, $input);
echo "</div>";
echo "<div class = 'col col-lg-3'>";
echo form_label("Subject");
echo "</div>";
echo "<div class = 'col col-lg-9'>";
echo form_dropdown(array("name" => "
subject", "id" => "subject", "
class" => "form-control", "
required" => true));
echo form_dropdown(array("name" => "
code", "id" => "code", "class" =>
"form-control", "required" =>
true));
echo "</div>";
?>
<input type = "hidden" name = "
lessonTitleHidden" id = "lesson-
title" value = "<?php echo
$lessonTitle;?>" />
<input type = "hidden" name = "edit"
id = "edit" value = "<?php echo
$edit;?>" />
<?php
if (!$edit) {
echo "<div class = 'col col-lg-3'>";
echo form_label("PDF file");
echo "</div>";
echo "<div class = 'col col-lg-9'>";
echo form_upload("lessonLink", "",
$input);
echo "</div>";
}
else {
echo form_hidden("lessonLinkHidden",
$lessonLink);
echo "<div class = 'col col-lg-3'>";
echo form_label("Change file?");
echo "</div>";
echo "<div class = 'col col-lg-9'>";
echo "<label class = 'radio-inline'>
.form_radio("changeFile", "yes"
)." yes</label>";
echo "<label class = 'radio-inline'>
.form_radio("changeFile", "no",
true)." no</label>";
echo "</div>";
}
?>
</div>
<div id = "lesson-link" class = "col
col-lg-9" style = "display: none;">
<?php
echo "<div class = 'col col-lg-3'><
label>Change PDF file: </label></
div>";
echo "<div class = 'col col-lg-9'><
input type = 'file' name = '
lessonLinkNew' id = 'lesson-link-
new' class = 'form-control'><br
/></div>";
?>
</div>
<div id = "lesson-position" class = "
col col-lg-9"></div>
<div class = "col col-lg-9 form-group
text-center"><br/>
<?php
echo anchor("/lessons", "back", $back)
;
if (!$edit) {
echo form_submit("create", "create
lesson", $submit);
}
else {
echo form_submit("update", "update
lesson", $submit);
}
echo form_close();
?>
</div>
</div>
</div>
</div>
</div>
</div>
</body>

```

## lesson\_index.php

```

<?php
/**
 * @author Quizon, Maria Reena I.
 * filename: lesson_index.php
 * description: view for accessing all available
online lessons in LRC_TMS
 * start date: March 4, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

$chemSubjects = array("Chemistry 14", "
Chemistry 18", "Chemistry 27", "Chemistry
31", "Chemistry 40", "Chemistry 156", "
Chemistry 157", "NSI part");
$mathSubjects = array("Mathematics 1", "
Mathematics 11", "Mathematics 14", "
Mathematics 17", "Mathematics 101", "
Mathematics 73", "Mathematics 74", "
Mathematics 75");
$physicsSubjects = array("Physics 21", "Physics
51", "Physics 52", "Physics 71", "Physics
72", "Physics 73", "NSI part");
$bioSubjects = array("Biology 10", "Biology 20"
, "Biology 21", "Biology 22", "Biology 25",
"Biology 30", "Biology 100", "Biology 101"
);
$commSubjects = array("Communications 1", "
Communications 2", "Communications 3");
$comsciSubjects = array("CMSC 11", "CMSC 22", "
CMSC 55");

$anchor = array("class" => "list-group-item");
?>
<html>
<head>
<script src="<?php echo base_url('application/
js/jquery.lesson.js'); ?>"></script>
</head>
<body>
<div class = "container-fluid">
<div class = "row">
<div class="col-lg-3">
<!--Affix start-->
<ul id = "sidebar" class = "nav nav-pills
nav-stacked text-justified">

```

```

<li class = "active"><a data-toggle = "tab
" href="#chemistry">Chemistry</a></li>
<li><a data-toggle = "tab" href="#
mathematics">Mathematics</a></li>
<li><a data-toggle = "tab" href="#physics"
>Physics</a></li>
<li><a data-toggle = "tab" href="#biology"
>Biology</a></li>
<li><a data-toggle = "tab" href="#computer
-science">Computer Science</a></li>
<li><a data-toggle = "tab" href="#
communications">Communications</a></li
>
<?php
if ($this->session->userdata("userType")
== ADMIN) {
echo "<li>".anchor("lessons/
create_lesson", "Create Lesson")."</
li>";
}
?>
</ul>
<!--Affix end-->
</div>
<div class="col-lg-9">
<!--Content start-->
<div class="wrapper">
<div class = "tab-content">
<div id = "chemistry" class="tab-pane
fade in active">
<div class = "panel panel-heading lead">
Chemistry subjects</div>
<div class = "panel panel-body">
<?php
for ($counter=0; $counter < sizeof(
$schemSubjects); $counter++) {
echo anchor("lessons/get_lessons/" .
str_replace(" ", "-",
$schemSubjects[$counter]),
$schemSubjects[$counter], $anchor)
;
}
?>
</div>
</div>
</div>
<div id = "mathematics" class="tab-pane
fade">
<div class = "panel panel-heading lead">
Mathematics subjects</div>
<div class = "panel panel-body">
<?php
for ($counter=0; $counter < sizeof(
$mathSubjects); $counter++) {
echo anchor("lessons/get_lessons/" .
str_replace(" ", "-",
$mathSubjects[$counter]),
$mathSubjects[$counter], $anchor)
;
}
?>
</div>
</div>
<div id = "physics" class="tab-pane fade"
>
<div class = "panel panel-heading lead">
Physics subjects</div>
<div class = "panel panel-body">
<?php
for ($counter=0; $counter < sizeof(
$physicsSubjects); $counter++) {
echo anchor("lessons/get_lessons/" .
str_replace(" ", "-",
$physicsSubjects[$counter]),
$physicsSubjects[$counter],
$anchor);
}
?>
</div>
</div>
<div id = "biology" class="tab-pane fade"
>
<div class = "panel panel-heading lead">
Biology subjects</div>
<div class = "panel panel-body">
<?php
for ($counter=0; $counter < sizeof(
$bioSubjects); $counter++) {
echo anchor("lessons/get_lessons/" .
str_replace(" ", "-",
$bioSubjects[$counter]),
$bioSubjects[$counter], $anchor);
}
?>
</div>
</div>
<div id = "communications" class="tab-
pane fade">
<div class = "panel panel-heading lead">
Communications Subjects</div>
<div class = "panel panel-body">
<?php
for ($counter=0; $counter < sizeof(
$commSubjects); $counter++) {
echo anchor("lessons/get_lessons/" .
str_replace(" ", "-",
$commSubjects[$counter]),
$commSubjects[$counter], $anchor)
;
}
?>
</div>
</div>
</div>
</div>
</div>
</div>
</body>

```

## view\_lessons.php

```

<?php
/**
 * @author Quizon, Maria Reena I.
 * filename: view_lesson.php
 * description: view for the view lessons page
 in LRC_TMS
 * start date: March 5, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

$userType = $this->session->userdata("userType"
);

function print_lesson($lesson, $userType) {
$readonly = array("class" => "readonly form-
control", "readOnly" => true);
$cancel = array("class" => "btn btn-danger");
$submit = array("class" => "btn btn-success");

echo form_open("lessons/manage_lesson");
echo form_hidden("lessonId", $lesson->lessonId
);
echo form_hidden("lessonTitle", $lesson->
lessonTitle);
echo "<div class = 'row'>";
echo "<div class = 'panel panel-heading lead
text-right'>".date("F j Y, g:m a",
strtotime($lesson->lastModified))."</div>
";
echo "<div class = 'panel-body'>";
echo "<div class = 'row'>";

```







```

        reschedule appointment", $button);
    }
    break;
case FACULTY_REJECTED:
    $action = form_button("change", "change
    faculty", $button);
    $action .= form_submit("cancel", "cancel
    application", $cancel);
    $action .= "<div class = 'faculty-select
    text-center'>";
    $action .= "</div>";
    break;
case LRC MOCK_EXAM_NEW:
    $mockExamId = $eventId;
    $mockExamString = DEFAULT_STRING;
    $mockExamAction = DEFAULT_STRING;
    if ($mockExamStudents != DEFAULT_INTEGER) {
        foreach ($mockExamStudents as
            $mockExamStudent) {
            if ($mockExamStudent->mockExamId ==
                $mockExamId) {
                $mockExamString .= $mockExamStudent->
                userId."#";
            }
        }
    }
    if (strpos($mockExamString, $userId) !=
        false) {
        $mockExamAction = "view students/cancel
        application";
    }
    else {
        $mockExamAction = "sign up for mock exam";
    }
    $action .= anchor("events/get_students/" .
        $eventId, $mockExamAction, $submit)."<br
    />";
    break;
default: break;
}
return $action;
}
?>

```

## email\_modal.php

```

<?php
/**
 * @author Quizon, Maria Reena I.
 * filename: email_modal.php
 * description: view for the email modal in
    LRC_TMS
 * start date: May 8, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

$input = array("class" => "form-control");
$submit = array("class" => "btn btn-success");
?>
<div class="modal in" id="add-email" role="
    dialog">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h4 class="modal-title">Add Email</h4>
            </div>
            <div class="modal-body">
                <div class = "row text-center">
                    <div class = "col col-lg-1"></div>
                    <div class = "col col-lg-10">
                        <p>Welcome to LRC's Tutorial Management
                        System.
                            In order to make full use of the
                            system, you are required to
                            enter a valid email address.</p>
                    </div>
                    <div class = "col col-lg-1"></div>
                </div>
                <div class = "row">
                    <div class = "col col-lg-2"></div>
                    <div class = "col col-lg-8">

```

```

        <?php
            echo form_open("accounts/
            add_email");
            echo form_input("email", "", $input);
            ?>
        </div>
        <div class = "col col-lg-2"></div>
    </div>
    <div class="modal-footer">
        <div class = "text-center">
            <?php
                echo form_submit("submit", "add email",
                $submit);
                echo form_close();
                ?>
            </div>
        </div>
    </div>
    <div class="modal-backdrop fade in"></div>

```

## login.php

```

<?php
/**
 * @author Quizon, Maria Reena I.
 * filename: login.php
 * description: view for the login page in
    LRC_TMS
 * start date: January 13, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

$input = array ("class" => "form-control", "
    required" => true);
$submit = array("class" => "btn btn-success btn
    -sm");

if (isset($msg)) { echo $msg; }
?>
<html>
<head>
    <title>LRC Tutorial Management System</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-
    width, initial-scale=1">
    <link href="<?php echo base_url('assets/
    bootstrap/css/bootstrap.css');?>" rel="
    stylesheet">
    <link href="<?php echo base_url('assets/
    bootstrap/css/bootstrap.min.css');?>" rel=
    "stylesheet">
    <link href="<?php echo base_url('application/
    css/login.css'); ?>" rel="stylesheet">

    <script src="<?php echo base_url('assets/
    jquery-1.12.0.min.js'); ?>"></script>
    <script src="<?php echo base_url('assets/
    jquery-ui/jquery-ui.min.js'); ?>"></script
    >
    <script src="<?php echo base_url('assets/
    bootstrap-validator/js/validator.js'); ?>"
    ></script>
    <script src="<?php echo base_url('assets/
    bootstrap/js/bootstrap.min.js'); ?>"></
    script>
</head>
<body data-spy = "scroll" data-target = ".
    navbar" data-offset = "50">
    <nav class = "navbar navbar-inverse navbar-
    fixed-top">
        <div class = "container-fluid">
            <div class = "navbar-header">
                <button type="button" class="navbar-toggle"
                data-toggle="collapse" data-target="#
                navigation">
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>

```







```

        echo "<div class = 'col col-lg-5'><
        label>New Password</label></div>";
        echo "<div class = 'col col-lg-7'>";
        echo form_password("newPassword", "",
            $input);
        echo "</div>";
        echo "</div>";
        echo "<div class = 'row'>";
        echo "<div class = 'col col-lg-5'><
        label>Confirm New Password</label></div
        >";
        echo "<div class = 'col col-lg-7'>";
        echo form_password("retypePassword", ""
            , $input);
        echo "</div>";
        echo "</div>";
        ?>
    </div>
    <div class = "col col-lg-2"></div>
</div>
<div class="modal-footer">
    <div class = "text-center">
        <?php
            echo form_submit("submit", "change
                password", $submit);
            echo form_close();
            ?>
        </div>
    </div>
</div>
</div>
</div>
</div>
</div>
<div class="modal-backdrop fade in"></div>

```

## profile.php

```

<?php
/**
 * @author Quizon, Maria Reena I.
 * filename: profile.php
 * description: view for the profile page in
   LRC TMS
 * start date: March 25, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

$input = array("class" => "form-control", "
    required" => true);
$readonly = array("class" => "readonly form-
    control", "readOnly" => true);
$submit = array("class" => "btn btn-success");

if (isset($msg)) { echo $msg; }

$userId = $this->session->userdata("userId");
$userType = $this->session->userdata("userType"
    );
$name = $this->session->userdata("name");
$email = $this->session->userdata("email");
$course = $this->session->userdata("course");
$college = $this->session->userdata("college");
$birthday = $this->session->userdata("birthday"
    );
$studentNumber = $this->session->userdata("
    studentNumber");
$profilePicture = $this->session->userdata("
    profilePicture");

$path = base_url("/uploads/" . $userId . "/" .
    $profilePicture);

?>
<html>
<head>
<link href = "<?php echo base_url('application/
    css/profile.css');?>" rel="stylesheet">
<script type="text/javascript">
    $(function() {
        $("#datetimepicker").datetimepicker({
            pickTime: false
        });
    });

```

```

    });
</script>
</head>
<body>
    <div class = "container-fluid">
        <div class = "row">
            <div class="col-lg-3">
                <!--Affix start-->
                <ul id = "sidebar" class = "nav nav-pills
                    nav-stacked text-justify">
                    <li class = "active"><a data-toggle = "tab
                        " href="#profile">Profile</a></li>
                    <li><a data-toggle = "tab" href="#settings
                        ">Change Details</a></li>
                </ul>
                <!--Affix end-->
            </div>

            <!--Content start-->
            <div class="col-lg-9">
                <div class="wrapper">
                    <div class = "tab-content">
                        <div id = "profile" class="tab-pane fade
                            in active">
                            <div class = "col col-lg-12">
                                <div class = "panel">
                                    <div class = "panel panel-heading lead
                                        ">User Details</div>
                                    <div class = "panel-body">
                                        <div class = "col col-lg-4 text-
                                            center">
                                            <?php
                                                if (!isset($profilePicture)) {
                                                    $path = base_url("./uploads/
                                                        default.png");
                                                }
                                                ?>
                                            <img src = "<?php echo $path; ?>"
                                                class = "profile-image img-
                                                    thumbnail">
                                        </div>
                                    <div class = "col col-lg-8">
                                        <?php
                                            echo "<div class = 'col col-lg-4'>"
                                                ;
                                            echo form_label("User Id");
                                            echo "</div>";

                                            echo "<div class = 'col col-lg-8'>"
                                                ;
                                            echo form_input("userId", $userId,
                                                $readonly);
                                            echo "</div>";

                                            echo "<div class = 'col col-lg-4'>"
                                                ;
                                            echo form_label("Full Name");
                                            echo "</div>";

                                            echo "<div class = 'col col-lg-8'>"
                                                ;
                                            echo form_input("name", $name,
                                                $readonly);
                                            echo "</div>";

                                            if (isset($birthday)) {
                                                echo "<div class = 'col col-lg-4'>"
                                                    ;
                                                echo form_label("Birthday");
                                                echo "</div>";
                                                echo "<div class = 'col col-lg-8'>"
                                                    ;
                                                echo form_input("currBirthday",
                                                    date("F j, Y", strtotime(
                                                        $birthday)), $readonly);
                                                echo "</div>";
                                            }

                                            if (isset($studentNumber)) {
                                                echo "<div class = 'col col-lg-4'>"
                                                    ;
                                                echo form_label("Student Number");
                                                echo "</div>";
                                                echo "<div class = 'col col-lg-8'>"
                                                    ;
                                                echo form_input("studentNumber",
                                                    substr_replace($studentNumber,
                                                        " - ", 4, 0), $readonly);
                                                echo "</div>";
                                            }
                                        </div>
                                    </div>
                                </div>
                            </div>
                    </div>
                </div>
            </div>
        </div>
    </body>

```

```

}
if (isset($course)) {
    echo "<div class = 'col col-lg-4'>";
    echo form_label("Course");
    echo "</div>";
    echo "<div class = 'col col-lg-8'>";
    echo form_input("course", $course, $readonly);
    echo "</div>";
}
if (isset($college)) {
    echo "<div class = 'col col-lg-4'>";
    echo form_label("College");
    echo "</div>";
    echo "<div class = 'col col-lg-8'>";
    echo form_input("college", $college, $readonly);
    echo "</div>";
}
?>
</div>
</div>
</div>
</div>
<div id = "settings" class="tab-pane fade">
<div class = "panel">
<div class = "panel panel-heading lead">
    >Change Profile Picture</div>
<div class = "panel-body">
<div class = "col col-lg-9 form-group">
    <?php
    echo form_open_multipart("profile/
        process_account");
    echo "<div class = 'col col-lg-3'>";
    echo form_label("Choose from file ");
    echo "</div>";
    echo "<div class = 'col col-lg-9'>";
    echo form_upload("profilePicture", "
        ", $input);
    echo "<p class = 'small'><br/>
        Supported formats are: GIF, JPG,
        or PNG.</p>";
    echo "</div>";
    ?>
</div>
<div class = "col col-lg-3 text-center">
    <?php
    echo form_submit("upload", "upload
        new image", $submit);
    echo form_close();
    ?>
</div>
</div>
</div>
<!--
<div class = "panel">
<div class = "panel panel-heading lead">
    >Change Birthday</div>
<div class = "panel-body">
<div class = "col col-lg-9 form-group">
    <?php
    echo form_open("profile/
        process_account");
    ?>
<div class = "col col-lg-3">
    <label for = "birthday">Select date
    </label>
</div>
<div class = "col col-lg-9">
    <input type = "date" name = "
        birthday" value = "<?php echo
        $birthday; ?>" id = "birthday"
        class = "form-control" required
    />
</div>
</div>
</div>
<div class = "col col-lg-3 text-center
    >
    <?php
    echo form_submit("changeBirthday", "
        change birthday", $submit);
    echo form_close();
    ?>
</div>
</div>
</div>
</div>
<div id = "settings" class="tab-pane fade">
<div class = "panel">
<div class = "panel panel-heading lead">
    >Change Password</div>
<div class = "panel-body">
<div class = "col col-lg-9">
    <?php
    echo form_open_multipart("profile/
        process_account");
    echo "<div class = 'row'>";
    echo "<div class = 'col-lg-4'>";
    echo form_label("Type old password"
        );
    echo "</div>";
    echo "<div class = 'col col-lg-8'>";
    ?>
    <input type = "password" name = "
        oldPassword" placeholder = "Type
        old password" class = "form-
        control" required />
    <?php
    echo "</div>";
    echo "</div>";
    echo "<div class = 'row'>";
    echo "<div class = 'col-lg-4'>";
    echo form_label("Type new password"
        );
    echo "</div>";
    echo "<div class = 'col col-lg-8'>";
    ?>
    <input type = "password" name = "
        newPassword" placeholder = "Type
        new password" id = "new-
        password" class = "form-control"
        required />
    <?php
    echo "</div>";
    echo "</div>";
    echo "<div class = 'row'>";
    echo "<div class = 'col-lg-4'>";
    echo form_label("Confirm new
        password");
    echo "</div>";
    echo "<div class = 'col col-lg-8'>";
    ?>
    <input type = "password" name = "
        retypePassword" placeholder = "
        Confirm new password" id = "
        confirm-password" class = "form-
        control" data-match = "#new-
        password" data-match-error = "
        These do not match." required />
    <?php
    echo "</div>";
    echo "</div>";
    ?>
</div>
<div class = "col col-lg-3 text-center">
    ><br/><br/><br/>
    <?php
    $submit = array("class" => "btn btn-
        success", "onClick" => "confirm
        ('Are you sure you want to
        change your password?')");
    echo form_submit("changePassword", "
        change password", $submit);
    echo form_close();
    ?>
</div>
</div>
</div>
</div>
</div>
</div>
</div>

```



```

* last updated:
* version number: 1.0
*/

$readonly = array("class" => "readonly form-
control", "readOnly" => true);
$cancel = array("class" => "btn btn-danger");
$submit = array("class" => "btn btn-success");

?>
<html>
<head>
</head>
<body>
<div class = "container-fluid">
<div class = "row">
<div class="col-lg-3">
<!--Affix start-->
<ul id = "sidebar" class = "nav nav-pills
nav-stacked text-justify">
<li class = "active"><a data-toggle = "tab
" href="#profile">View Profile</a></li
>
<?php
if ($userType == STUDENT_TUTOR) {
echo "<li><a data-toggle = 'tab' href='#
tutor-details'>Tutor Details</a></li
>";
}
?>
</ul>
<!--Affix end-->
</div>
<div class="col-lg-9">
<!--Content start-->
<div class="wrapper">
<div class = "tab-content">
<div id = "profile" class="tab-pane fade
in active">
<div class = "col col-lg-12">
<div class = "panel">
<div class = "panel panel-heading lead
">User Details</div>
<div class = "panel-body">
<div class = "col col-lg-4 text-
center">
<?php
if (!isset($profilePicture)) {
$path = base_url("./uploads/
default.png");
}
echo form_open("accounts/
block_account");
echo form_hidden("isBlocked",
$isBlocked);
echo form_hidden("email", $email);
echo "<img src = '". $path.'" class
= 'profile-image img-thumbnail
'>";
if ($userType == STUDENT_TUTOR) {
echo "<div class = 'row'>This
student is an approved student
-tutor this semester.<br/></
div>";
}
?>
</div>
<div class = "col col-lg-8">
<?php
echo "<div class = 'col col-lg-4'>"
;
echo form_label("User Id");
echo "</div>";

echo "<div class = 'col col-lg-8'>"
;
echo form_input("userId", $userId,
$readonly);
echo "</div>";

echo "<div class = 'col col-lg-4'>"
;
echo form_label("Full Name");
echo "</div>";

echo "<div class = 'col col-lg-8'>"
;
echo form_input("name", $name,
$readonly);

echo "</div>";
}
if (isset($birthday)) {
echo "<div class = 'col col-lg-4'>"
;
echo form_label("Birthday");
echo "</div>";
echo "<div class = 'col col-lg-8'>"
;
echo form_input("currBirthday",
date("F j, Y", strtotime(
$birthday)), $readonly);
echo "<br/></div>";
}

if (isset($studentNumber)) {
echo "<div class = 'col col-lg-4'>"
;
echo form_label("Student Number");
echo "</div>";
echo "<div class = 'col col-lg-8'>"
;
echo form_input("studentNumber",
substr_replace($studentNumber,
" - ", 4, 0), $readonly);
echo "</div>";
}

if (isset($course)) {
echo "<div class = 'col col-lg-4'>"
;
echo form_label("Course");
echo "</div>";
echo "<div class = 'col col-lg-8'>"
;
echo form_input("course", $course,
$readonly);
echo "</div>";
}

if (isset($college)) {
echo "<div class = 'col col-lg-4'>"
;
echo form_label("College");
echo "</div>";
echo "<div class = 'col col-lg-8'>"
;
echo form_input("college",
$college, $readonly);
echo "<br/></div>";
}

if ($this->session->userdata("
userType") == ADMIN && $userId
!= ADMIN_ID) {
echo "<div class = 'col col-lg-4'>"
;
echo form_label("Block Count");
echo "</div>";
echo "<div class = 'col col-lg-8'>"
;
echo form_input("blockCount",
$blockCount, $readonly);
echo "<br/></div>";

echo "<div class = 'text-center'>"
;
if ($isBlocked == 0 && $blockCount
== EVENT_INACTIVE) {
echo form_submit("block", "warn
student", $cancel)."<br/>";
}
elseif ($isBlocked == 0 &&
$blockCount >= EVENT_ACTIVE) {
echo form_submit("block", "block
student", $cancel)."<br/>";
}
elseif ($isBlocked == 1) {
echo form_submit("unblock", "
unblock student", $submit)."<
br/>";
}
}
echo "</div>";
}
echo form_close();
?>
</div>

```





```

/**
 * @author Quizon, Maria Reena I.
 * filename: report.php
 * description: template for all reported
 * accounts in LRC_TMS
 * start date: April 16, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

ini_set('memory_limit', '256M');

?>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-
width, initial-scale=1">
<link href="<?php echo base_url('assets/
bootstrap/css/bootstrap.css');?>" rel="
stylesheet">
<link href="<?php echo base_url('assets/
bootstrap/css/bootstrap.min.css');?>" rel
="stylesheet">
<style>
thead:before, thead:after { display: none; }
tbody:before, tbody:after { display: none; }
.header {
padding-left: 11%;
}
.header-image {
width: 15%;
height: 15%;
}
.header-table td {
padding-left: 5%;
padding-right: 5%;
}
.body-table td {
padding-left: 5%;
padding-right: 5%;
}
</style>
</head>
<body>
<div class = "container-fluid">
<div class = "row text-center header">
<table class = "text-center header-table">
<tr>
<td><img src = "../uploads/lrc_logo.png"
class = "header-image"></td>
<td>
<h3><strong>University of the Philippines
Manila</strong></h3>
<h4><strong>Learning Resource Center</
strong></h4>
<h5>2F Gusaling Joaquin Gonzales, Padre
Faura St.<br/>Ermita, Manila 1000<br
/>
Landline: 336 5937 | Mobile No:
09279152834<br/>
Email Address: http://www.upm.edu.ph/lrc
<br/>
</h5>
</td>
<td><img src = "../uploads/upm_logo.png"
class = "header-image"></td>
</tr>
</table>
</div>
<div class = "row body text-center">
<hr>
<p><strong>List of all generated accounts by
the Learning Resource Center Tutorial
Management System</strong></p>
<table class = "table-bordered text-
center body-table">
<tr><th>Student Number</th><th>Name</th><th>
Course</th><th>College</th><th>User Id
</th>
<?php
if ($batchNumber != DEFAULT_INTEGER) {
echo "<th>Password</th>";
}
echo "</tr>";

```

```

foreach ($accounts as $account) {
if ($account->userType > ADMIN) {
echo "<tr>";
echo "<td>".$account->studentNumber."</
td>";
echo "<td>".$account->name."</td>";
echo "<td>".$account->course."</td>";
echo "<td>".$account->college."</td>";
echo "<td>".$account->userId."</td>"
;
if ($batchNumber != DEFAULT_INTEGER)
{
if (strlen($account->password) < 8)
{
echo "<td>".$account->password."</
td>";
}
else {
echo "<td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&";
}
}
echo "</tr>";
}
}
?>
</table>
</div>
</div>
</body>
</html>

```

## navigation.php

```

<?php if ( ! defined("BASEPATH")) exit("No
direct script access allowed");

/**
 * @author Quizon, Maria Reena I.
 * filename: navigation.php
 * description: view for the navigation template
 * in LRC_TMS
 * start date: January 13, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

?>
<html>
<head>
<title>LRC Tutorial Management System</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-
width, initial-scale=1">

<link href="<?php echo base_url('assets/
bootstrap/css/bootstrap.css');?>" rel="
stylesheet">
<link href="<?php echo base_url('assets/
bootstrap/css/bootstrap.min.css');?>" rel
="stylesheet">
<link href="<?php echo base_url('assets/
bootstrap/css/bootstrap-datetimepicker.min
.css');?>" rel="stylesheet">
<link href="<?php echo base_url('application/
css/main.css');?>" rel="stylesheet">
<link href="<?php echo base_url('application/
css/navigation.css');?>" rel="stylesheet">

<!--
<script src="https://ajax.googleapis.com/ajax
/libs/jquery/1.12.0/jquery.min.js"></script
>
<script src="http://ajax.googleapis.com/ajax/
libs/jqueryui/1.11.4/jquery-ui.min.js"></
script>
<script src="https://cdnjs.cloudflare.com/ajax
/libs/moment.js/2.12.0/moment.js"></script
>-->

<script src="<?php echo base_url('assets/
jquery-1.12.0.min.js');?>"></script>
<script src="<?php echo base_url('assets/

```

```

        jquery-ui/jquery-ui.min.js'); ?>"></script>
<script src="<?php echo base_url('assets/
moment/moment.js'); ?>"></script>

<script src="<?php echo base_url('application/
js/jquery.affix.js'); ?>"></script>
<script src="<?php echo base_url('assets/
bootstrap/js/bootstrap.min.js'); ?>"></
script>
<script src="<?php echo base_url('assets/
bootstrap-validator/js/validator.js'); ?>"
></script>
<script src="<?php echo base_url('assets/
bootstrap/js/bootstrap-datetimepicker.min.
js'); ?>"></script>

</head>
<body>
<nav class = "navbar navbar-inverse navbar-
fixed-top">
<div class = "container-fluid">
<div class = "navbar-header">
<button type="button" class="navbar-toggle"
data-toggle="collapse" data-target="#
navigation">
<span class="icon-bar"></span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
</button>
</div>
<div class="collapse navbar-collapse" id="
navigation">
<ul class = "nav navbar-nav">
<?php
$userType = $this->session->userdata("
userType");
$userTypeStr = DEFAULT_INTEGER;
echo "<li>".anchor("/dashboard", "<span
class='glyphicon glyphicon-home'></
span>Home")."</li>";
echo "<li>".anchor("/search", "<span class
='glyphicon glyphicon-search'></span>
Search")."</li>";

switch ($userType) {
case ADMIN:
$userTypeStr = "Administrator ";
echo "<li>".anchor("apply/dashboard", "<
span class='glyphicon glyphicon-th-
list'></span>Applications")."</li>";
echo "<li>".anchor("appointment/
dashboard", "<span class='glyphicon
glyphicon-time'></span>Tutorial
Sessions")."</li>";
echo "<li>".anchor("/lessons", "<span
class='glyphicon glyphicon-book'></
span>Online Lessons")."</li>";
echo "<li>".anchor("/events", "<span
class='glyphicon glyphicon-calendar
'></span>Calendar")."</li>";
echo "<li>".anchor("/reports", "<span
class='glyphicon glyphicon-stats'></
span>Reports")."</li>";
echo "<li>".anchor("/accounts", "<span
class='glyphicon glyphicon-list-alt
'></span>Accounts")."</li>";
break;
case STUDENT:
$userTypeStr = "Student ";
if ($this->session->userdata("
hasApplication") == DEFAULT_INTEGER)
{
echo "<li>".anchor("/apply", "<span
class='glyphicon glyphicon-th-list
'></span>Apply")."</li>";
}
elseif ($this->session->userdata("
approvalCount") > DEFAULT_INTEGER) {
echo "<li>".anchor("/apply", "<span
class='glyphicon glyphicon-th-list
'></span>Renew")."</li>";
}
echo "<li>".anchor("/appointment", "<
span class='glyphicon glyphicon-time
'></span>Tutorial Sessions")."</li>"
;
echo "<li>".anchor("/lessons", "<span
class='glyphicon glyphicon-book'></
span>Online Lessons")."</li>";
break;
case STUDENT_TUTOR:
$userTypeStr = "Student-tutor ";
echo "<li>".anchor("/appointment", "<
span class='glyphicon glyphicon-time
'></span>Tutorial Sessions")."</li>"
;
echo "<li>".anchor("/view_feedback/" .
$this->session->userdata("userId"),
"<span class='glyphicon glyphicon-con-
stats'></span>View Feedback")."</li>"
;
echo "<li>".anchor("/lessons", "<span
class='glyphicon glyphicon-book'></
span>Online Lessons")."</li>";
break;
default: break;
}
?>
</ul>
<ul class = "nav navbar-nav navbar-right">
<?php
echo "<li>".anchor("/profile", "<span
class='glyphicon glyphicon-user'></
span>". $userTypeStr. $this->session->
userdata("name")."</li>";
echo "<li>".anchor("/logout", "<span
class='glyphicon glyphicon-log-out
'></span>Logout")."</li>";
?>
</ul>
</div>
</nav>
</body>
</html>

```

## navigation\_2.php

```

<?php if ( ! defined("BASEPATH")) exit("No
direct script access allowed");

/**
 * @author Quizon, Maria Reena I.
 * filename: navigation_2.php
 * description: view for the second navigation
 * template in LRC_TMS
 * start date: May 7, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */
?>
<html>
<head>
<title>LRC Tutorial Management System</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-
width, initial-scale=1">

<link href="<?php echo base_url('assets/
bootstrap/css/bootstrap.css'); ?>" rel="
stylesheet">
<link href="<?php echo base_url('assets/
bootstrap/css/bootstrap.min.css'); ?>" rel
="stylesheet">
<link href="<?php echo base_url('assets/
bootstrap/css/bootstrap-datetimepicker.min.
css'); ?>" rel="stylesheet">
<link href="<?php echo base_url('application/
css/main.css'); ?>" rel="stylesheet">
<link href="<?php echo base_url('application/
css/navigation.css'); ?>" rel="stylesheet">

<script src="<?php echo base_url('assets/
jquery-1.12.0.min.js'); ?>"></script>

```



```

<script src="<?php echo base_url('assets/
jquery-ui/jquery-ui.min.js'); ?>"></script>
<script src="<?php echo base_url('assets/
moment/moment.js'); ?>"></script>
<script src="<?php echo base_url('application
/js/jquery.affix.js'); ?>"></script>
<script src="<?php echo base_url('assets/
bootstrap/js/bootstrap.min.js'); ?>"></
script>
<script src="<?php echo base_url('assets/
bootstrap-validator/js/validator.js'); ?>"
></script>
<script src="<?php echo base_url('assets/
bootstrap/js/bootstrap-datetimepicker.min.
js'); ?>"></script>
</head>
<body>
<nav class = "navbar navbar-inverse navbar-
fixed-top">
<div class = "container-fluid">
<div class = "navbar-header">
<button type="button" class="navbar-toggle"
data-toggle="collapse" data-target="#
navigation">
<span class="icon-bar"></span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
</button>
<a class="navbar-brand" href= "<?php echo
base_url('/login'); ?>" >LRC</a>
</div>
<div class="collapse navbar-collapse" id="
navigation">
<ul class = "nav navbar-nav">
<?php
echo "<li>".anchor("/login", "<span class
='glyphicon glyphicon-home'></span>
Home")."</li>";
?>
</ul>
</div>
</div>
</nav>
</body>
</html>
padding-right: 5%;
}
.body-table td {
padding-left: 5%;
padding-right: 5%;
}
</style>
</head>
<body>
<div class = "container-fluid">
<div class = "row text-center header">
<table class = "text-center header-table">
<tr>
<td><img src = "./uploads/lrc_logo.png"
class = "header-image"></td>
<td>
<h3><strong>University of the Philippines
Manila</strong></h3>
<h4><strong>Learning Resource Center</
strong></h4>
<h5>2F Gusaling Joaquin Gonzales, Padre
Faura St.<br/>Ermita, Manila 1000<br
/>
Landline: 336 5937 | Mobile No:
09279152834<br/>
Email Address: http://www.upm.edu.ph/lrc
<br/>
</h5>
</td>
<td><img src = "./uploads/upm_logo.png"
class = "header-image"></td>
</tr>
</table>
</div>
<div class = "row body text-center">
<hr>
<?php
$reportTitle = "";
switch ($reportType) {
case "studentsServed": $reportTitle = "
Number of Students Served"; break;
case "frequentStudents": $reportTitle = "
Most Frequent Students"; break;
case "frequentSubjects": $reportTitle = "
Most Frequent Subjects"; break;
default: break;
}
?>
<h4><p><strong><?php echo $reportTitle?></
strong></p></h4>
<table class = "table table-bordered body-
table">
<tr><th>Breakdown</th><th>Results</th></tr>
<?php
$total = 0;
for ($index=0; $index < sizeof($breakdown)
; $index++) {
echo "<tr>";
echo "<td>".str_replace(":", " to ",
$breakdown[$index])."</td>";

if ($reportType == "studentsServed")
{
echo "<td>". $result[$index]."</td>";
$total += $result[$index];
}
else {
$token = strtok($result[$index], "#"
);
echo "<td>";
while ($token !== false) {
echo $token."<br/>";
$token = strtok("#");
}
echo "</td>";
}
echo "</tr>";
}
if ($total > 0) {
echo "<tr><td><strong>Total</strong></td
><td>". $total."</td></tr>";
}
?>
</table>
</div>

```

## report.php

```

<?php
/**
 * @author Quizon, Maria Reena I.
 * filename: report.php
 * description: template for all reports in
LRC_TMS
 * start date: April 11, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */
?>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-
width, initial-scale=1">
<link href="<?php echo base_url('assets/
bootstrap/css/bootstrap.css'); ?>" rel="
stylesheet">
<link href="<?php echo base_url('assets/
bootstrap/css/bootstrap.min.css'); ?>" rel
="stylesheet">
<style>
thead:before, thead:after { display: none; }
tbody:before, tbody:after { display: none; }
.header {
padding-left: 11%;
}
.header-image {
width: 15%;
height: 15%;
}
.header-table td {
padding-left: 5%;
padding-right: 5%;
}

```

```

<div class = "row footer">
  <p></p>
  Submitted by: <br/><br/>
  <strong>Sarah C. Johnson, M.Sc.</strong><br/>
  </div>
  <em>Director</em>
</div>
</body>
</html>

```

## schedule.php

```

<?php if ( ! defined("BASEPATH")) exit("No
direct script access allowed");

/**
 * @author Quizon, Maria Reena I.
 * filename: schedule.php
 * description: view for the schedule template
   in LRC_TMS
 * start date: January 14, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */
?>

<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-
width, initial-scale=1">
  <link href= "<?php echo base_url('application
/css/main.css')?>" rel="stylesheet">
  <script src="https://ajax.googleapis.com/ajax/
libs/jquery/1.12.0/jquery.min.js"></script
>
  <script src="http://ajax.googleapis.com/ajax/
libs/jqueryui/1.11.4/jquery-ui.min.js"></
script>
  <script src="https://cdnjs.cloudflare.com/ajax
/libs/moment.js/2.12.0/moment.js"></script
>
</head>
<body>
  <div class = "container-fluid">
    <div class = "row">
      <div class = "col col-sm-6">
        <table class = "table table-centered table-
bordered table-condensed text-center
small">
          <tr><td style = "padding-right: 20px;">Time
          </td><td>M</td><td>T</td><td>W</td><td>
          T</td><td>F</td></tr>
        </table>
        <?php
        $dayArray = array("0", "mon", "tues", "wed
", "thurs", "fri");
        $timeArray_1 = array("0830_1030", "1030
_1200", "0100_0230", "0230_0400", "
0400_0500");
        $timeArray_2 = array("8:30-10:30", "
10:30-12:00", "1:00-2:30", "2:30-4:00"
, "4:00-5:00");

        foreach ($schedules as $schedule) {
          for ($row = 0; $row < 5; $row++) {
            echo "<tr>";
            for ($column = 0; $column < 6; $column
            ++> {
              if ($column > 0) {
                if ($schedule->{$dayArray[$column]."_
.$timeArray_1[$row]} == 0) {
                  $buttonClass = "btn btn-danger";
                }
                else {
                  $buttonClass = "btn btn-success";
                }
                echo "<td>";
                echo form_button(array("name" =>
                $dayArray[$column]."_
                $timeArray_1[$row], "class" => "
                free-time ".$buttonClass));
                echo form_hidden($dayArray[$column]."_

```

```

        ".$timeArray_1[$row], $schedule->{
        $dayArray[$column]."_
        $timeArray_1[$row]}, array("name"
=> $dayArray[$column]."_
        $timeArray_1[$row]));
        echo "</td>";
      }
    }
    else {
      echo "<td style = 'font-size: 8px;'>";
      echo $timeArray_2[$row];
      echo "</td>";
    }
  }
  echo "</tr>";
}
?>
</table>
</div>
</div>
</body>
</html>

```

## apply\_mock\_exam.php

```

<?php
/**
 * @author Quizon, Maria Reena I.
 * filename: apply_mock_exam.php
 * description: View for applying for a mock
   exam event in LRC_TMS
 * start date: March 23, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

$readonly = array("class" => "readonly form-
control", "readOnly" => true);
$cancel = array("class" => "btn btn-danger");
$submit = array("class" => "btn btn-success");
$back = array("class" => "btn btn-default");

if (!isset($studentString)) { $studentString =
DEFAULT_STRING; }
if (isset($msg)) { echo $msg; }

?>
<html>
<head></head>
<body>
  <div class = "container-fluid">
    <div class = "row">
      <div class="col-lg-3">
        <!--Affix start-->
        <ul id = "sidebar" class = "nav nav-pills
nav-stacked text-justified">
          <li class = "active"><a data-toggle = "tab
" href="#mock-exam">View Mock Exam
          Details</a></li>
          <li><a data-toggle = "tab" href="#view-
students">View Students</a></li>
        </ul>
        <!--Affix end-->
      </div>
      <div class="col-lg-9">
        <!--Content start-->
        <div class="wrapper">
          <div class = "tab-content">
            <div id = "mock-exam" class = "tab-pane
fade in active">
              <div class = "panel panel-heading lead">
                About this Mock Exam</div>
              <div class = "panel-body">
                <?php
                if (isset($mockExams)) {
                  foreach ($mockExams as $mockExam) {
                    echo form_open("events/
                    apply_mock_exam");
                    echo form_hidden("mockExamId",
                    $mockExam->mockExamId);

                    echo "<div class = 'row'>";
                    echo "<div class = 'col col-lg-3'>
                    .form_label("Subject")."</div>"

```



```

<div class="modal-dialog">
  <div class="modal-content">
    <div class="modal-header">
      <button type="button" class="close" data-
dismiss="modal">&times;</button>
      <h4 class="modal-title">Excel File
Format</h4>
    </div>
    <div class="modal-body">
      <div class = "row text-justify">
        <div class = "col col-lg-1"></div>
        <div class = "col col-lg-10">
          <p>Generating accounts is a one-way
process. You cannot delete accounts!
Therefore it is imperative that the
following excel format must be
followed:</p>
          </div>
          <div class = "col col-lg-1"></div>
        </div>
        <div class ="row">
          <div class = "col col-lg-2"></div>
          <div class = "col col-lg-8 img-hover text-
center">
            
            <br/><span class = "small">(hover to zoom
in)</span>
          </div>
          <div class = "col col-lg-2"></div>
        </div>
        <div class = "row text-justified">
          <div class = "col col-lg-1"></div>
          <div class = "col col-lg-10"><br/>
            Be sure to:
            <ul>
              <li>Remove all headers</li>
              <li>Remove all unnecessary spacing in
the document, especially in between the
required columns</li>
              <li>Have all the information needed in
the prescribed format</li>
            </ul>
          </div>
          <div class = "col col-lg-1"></div>
        </div>
      <div class="modal-footer">
        <div class = "text-center">
          <button type="button" class="btn btn-
success" data-dismiss="modal">OK, got
it.</button>
        </div>
      </div>
    </div>
  </div>
</div>

```

## manage\_accounts.php

```

<?php
/**
 * @author Quizon, Maria Reena I.
 * filename: manage_accounts.php
 * description: View for managing accounts in
LRC_TMS
 * start date: March 21, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

require_once("excel_modal.php");

$input = array("class" => "form-control", "
required" => true);
$cancel = array("class" => "btn btn-danger");
$submit = array("class" => "btn btn-success");

if (isset($msg)) { echo $msg; }

function print_accounts($account) {
  $path = "/uploads";
  $default = "default.png";

```

```

$anchor = array("class" => "btn btn-default
btn-sm");
$readonly = array("class" => "readonly form-
control", "readOnly" => true);

echo "<div class = 'row well well-lg'>";
echo "<div class = 'col col-lg-3 text-center
'>";
  if (strlen($account->profilePicture) >
DEFAULT_INTEGER) {
    $path = base_url($path."/". $account->userId
."/". $account->profilePicture);
  }
  else {
    $path = base_url($path."/". $default);
  }
  echo "<img src = '". $path.'" class = 'img-
thumbnail' >";
echo "</div>";
echo "<div class = 'col col-lg-9'>";
echo "<p class = 'lead'>".anchor("profile/
view_profile/" . $account->userId,
$account->name, array("target" => "
_blank"))."</p>";
echo "<div class = 'row'>";
echo "<div class = 'col col-lg-4'>";
  echo form_label("Student Number");
echo "</div>";
echo "<div class = 'col col-lg-8'>";
  echo form_input("studentNumber",
substr_replace($account->studentNumber,
" - ", 4, 0), $readonly);
echo "</div>";
echo "</div>";
echo "<div class = 'row'>";
echo "<div class = 'col col-lg-4'>";
  echo form_label("User Id");
echo "</div>";
echo "<div class = 'col col-lg-8'>";
  echo form_input("userId", $account->userId,
$readonly);
echo "</div>";
echo "</div>";

if ($account->userType == 3) {
  echo "<div class = 'text-center'><br/>";
  echo anchor("view_feedback/" . $account->
userId, "view feedback", $anchor);
  echo "</div>";
}
echo "</div>";
echo "</div>";
}

?>
<html>
<head>
  <link href = "<?php echo base_url('application
/css/application_dashboard.css'); ?>" rel =
"stylesheet">
</head>
<body>
  <div class = "container-fluid">
    <div class = "row">
      <div class="col-lg-3">
        <!--Affix start-->
        <ul id = "sidebar" class = "nav nav-pills
nav-stacked text-justified">
          <li class = "active"><a data-toggle = "tab
" href="#overview">Overview</a></li>
          <li><a data-toggle = "tab" href="#view-
tutors">View Tutors</a></li>
          <li><a data-toggle = "tab" href="#view-
blocked">View Blocked Students</a></li>
          <li><a data-toggle = "tab" href="#create-
accounts">Generate Accounts</a></li>
        </ul>
        <!--Affix end-->
      </div>
      <div class="col-lg-9">
        <!--Content start-->
        <div class="wrapper">
          <div class = "tab-content">
            <div id = "overview" class="tab-pane fade
in active">

```

```

<div class = "col col-lg-2"></div>
<div class = "col col-lg-8">
<?php
    $counter = 0;
    $batchNumber = DEFAULT_INTEGER;
    if (isset($accounts)) {
        echo form_open("reports/
            process_account_pdf");
        echo form_hidden("batchNumber",
            $batchNumber);
        if (sizeof($accounts) > 1) {
            echo "<div class = 'text-center'>";
            echo form_submit("generate", "
                generate pdf for all accounts",
                $submit);
            echo "</div><br/>";
        }
        foreach ($accounts as $account) {
            if ($account->userType > ADMIN) {
                print_accounts($account);
                $counter++;
            }
        }
        echo form_close();
    }
    if ($counter == 0) {
        echo "<div class = 'panel text-center
            lead'><div class = 'panel-
            heading'>no generated accounts
            yet</div></div>";
    }
    ?>
</div>
</div>
<div id = "view-tutors" class = "tab-pane
    fade">
<div class = "col col-lg-2"></div>
<div class = "col col-lg-8">
<?php
    $counter = 0;
    $tutorCount = 0;
    if (isset($accounts)) {
        echo form_open("accounts/reset_tutors
            ");
        foreach ($accounts as $account) {
            if ($account->userType ==
                STUDENT_TUTOR) {
                $tutorCount++;
            }
        }
        if ($tutorCount > 0) {
            echo "<div class = 'text-center'>";
            echo form_submit("reset", "revert
                tutor accounts to student
                accounts", $cancel);
            echo "</div><br/>";
        }
        foreach ($accounts as $account) {
            if ($account->userType ==
                STUDENT_TUTOR) {
                print_accounts($account);
                $counter++;
            }
        }
        echo form_close();
    }
    if ($counter == 0) {
        echo "<div class = 'panel text-center
            lead'><div class = 'panel-
            heading'>no student-tutors yet</
            div></div>";
    }
    ?>
</div>
</div>
<div id = "view-blocked" class = "tab-
    pane fade">
<div class = "col col-lg-2"></div>
<div class = "col col-lg-8">
<?php
    $counter = 0;
    if (isset($accounts)) {
        foreach ($accounts as $account) {
            if ($account->isBlocked == 1) {
                print_accounts($account);
                $counter++;
            }
        }
    }
    if ($counter == 0) {
        echo "<div class = 'panel text-center
            lead'><div class = 'panel-
            heading'>no blocked students</div
            ></div>";
    }
    ?>
</div>
</div>
<div id = "create-accounts" class="tab-
    pane fade">
<div class = "panel panel-heading lead">
    Generate Multiple Accounts</div>
<div class = "panel-body">
<div class = "col col-lg-9 form-group">
<?php
    echo form_open_multipart("accounts/
        process_accounts");
    echo "<div class = 'col col-lg-4'>";
    echo form_label("Choose an excel file
        ");
    echo "<br/><a href = '#' data-toggle
        = 'modal' data-target = '#excel-
        modal'>important!</a>";
    echo "</div>";
    echo "<div class = 'col col-lg-8'>";
    echo form_upload("account", "",
        $input);
    echo "</div>";
    ?>
</div>
<div class = "col col-lg-3 text-center"
    >
<?php
    echo form_submit("create", "create
        accounts", $submit);
    echo form_close();
    ?>
</div>
</div>
<div class = "panel panel-heading lead">
    Generate One Account</div>
<div class = "panel-body">
<div class = "col col-lg-9 form-group">
<?php
    echo form_open("accounts/
        process_account");
    echo "<div class = 'col col-lg-4'>";
    echo form_label("Student number");
    echo "</div>";
    echo "<div class = 'col col-lg-8'>";
    echo "<div class = 'col-xs-5'><input
        name = 'studentNumber_1'
        placeholder = '20XX' class = '
        form-control' required /></div>"
        ;
    echo "<div class = 'col-xs-2 text-
        center'>.form_label("-").</div
        >";
    echo "<div class = 'col-xs-5'><input
        name = 'studentNumber_1'
        placeholder = 'XXXXX' class = '
        form-control' required /></div>"
        ;
    echo "</div>";
    echo "<div class = 'col col-lg-4'>";
    echo form_label("Student name");
    echo "</div>";
    echo "<div class = 'col col-lg-8'>";
    echo "<input name = 'name'
        placeholder = 'format: LAST NAME
        FIRST NAME MIDDLE' class = '
        form-control' required />";
    echo "</div>";
    echo "<div class = 'col col-lg-4'>";
    echo form_label("Course");
    echo "</div>";
    echo "<div class = 'col col-lg-8'>";
    echo "<input name = 'course'
        placeholder = 'e.g PA' class = '
        form-control' required />";
    echo "</div>";
    echo "<div class = 'col col-lg-4'>";

```





```

}
else {
    $calendarId = DEFAULT_INTEGER;
    $additionalDetails = DEFAULT_STRING;
    $eventType = DEFAULT_STRING;
    $eventName = DEFAULT_STRING;
    $date = DEFAULT_STRING;
}

if ($eventType == "MOCK_EXAM") {
    $mockExamId = $event["mockExamId"];
    $time = $event["time"];
    $code = $event["code"];
    $slots = $event["slots"];
    $subject = $event["subject"];
    $discussionDate = $event["discussionDate"];
    $discussionTime = $event["discussionTime"];
}
else {
    $mockExamId = DEFAULT_INTEGER;
    $time = DEFAULT_STRING;
    $code = DEFAULT_STRING;
    $slots = DEFAULT_STRING;
    $subject = DEFAULT_STRING;
    $discussionDate = DEFAULT_STRING;
    $discussionTime = DEFAULT_STRING;
}
}
?>

```

## Email\_Controller.php

```

<?php if ( ! defined("BASEPATH")) exit("No
direct script access allowed");

/**
 * @author Quizon, Maria Reena I.
 * filename: Email_Controller.php
 * description: Library for all email functions
 in LRC_TMS
 * start date: February 20, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

class Email_Controller {

    public function __construct() {
        $this->CI =& get_instance();
    }

    public function create_email ($to, $emailType)
    {

        $this->CI->load->helper("url");
        $this->CI->load->library("email");
        $this->CI->load->library("session");
        $this->CI->load->model("General_Model");

        $subject = DEFAULT_STRING;
        $message = DEFAULT_STRING;
        $adminEmail = DEFAULT_STRING;
        $adminId = ADMIN_ID;
        $users = $this->CI->General_Model->
            get_table_entry("user", $adminId);

        foreach ($users as $user) {
            $adminEmail = $user->email;
        }

        $from = $adminEmail;
        $name = EMAIL_LRC_ADMIN_NAME;
        $endorsementLink = substr(sha1(rand()), 0,
            10);

        switch ($emailType) {
            case EMAIL_BLOCK_STUDENT:
                $subject = EMAIL_SUBJECT_BLOCK_STUDENT;
                $message = "You have been blocked from
                    using the system.\nPlease contact the
                    LRC Administrator regarding this matter
                    .";
                break;

```

```

case EMAIL_ENDORSEMENT:
    $subject = EMAIL_SUBJECT_ENDORSEMENT;
    $message = "A student has requested an
        endorsement from you. The link to the
        online endorsement form is below. \n";
    $message .= base_url("/endorsement/".
        $endorsementLink);
    break;
case EMAIL_NEW_EMAIL:
    $subject = EMAIL_SUBJECT_NEW_EMAIL;
    $message = "If you have received this email
        , it means you have successfully
        activated your account in the Learning
        Resource Center Tutorial Management
        System. \n";
    $message .= "You can now go back to the
        website. Thank you. ";
    $message .= base_url("/login");
    break;
case EMAIL_NEW_PASSWORD:
    $subject = EMAIL_SUBJECT_NEW_PASSWORD;
    $message = "Here is your requested new
        password: ";
    $message .= $endorsementLink;
    $message .= "\nYou may change your password
        anytime at your profile in the website
        .";
    break;
default: break;
}

$this->CI->email->to($to);
$this->CI->email->from($from, $name);
$this->CI->email->subject($subject);
$this->CI->email->message($message);

if (!$this->CI->email->send()) {
    return false;
}
else {
    if ($emailType == EMAIL_ENDORSEMENT ||
        $emailType == EMAIL_NEW_PASSWORD) {
        return $endorsementLink;
    }
    else {
        return true;
    }
}
}
?>

```

## Notification\_Controller.php

```

<?php if ( ! defined("BASEPATH")) exit("No
direct script access allowed");

/**
 * @author Quizon, Maria Reena I.
 * filename: Notification_Controller.php
 * description: Library for all notification
 functions in LRC_TMS
 * start date: February 13, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

class Notification_Controller {

    public function __construct() {
        $this->CI =& get_instance();
    }

    public function create_notification (
        $notificationType, $eventId, $userIds,
        $notificationDetails) {

        $this->CI->load->model("General_Model");

        $notificationId = $this->CI->General_Model->
            get_tableId("notification");
        $notificationData = array(
            "notificationId" => $notificationId,

```



```

        "eventId" => $eventId,
        "userIds" => $userIds,
        "notificationType" => $notificationType,
        "notificationDetails" =>
            $notificationDetails
    );

    return $notificationData;
}

public function update_notification(
    $notificationType, $eventId, $userIds,
    $notificationStatus) {

    $notificationData = array(
        "notificationType" => $notificationType,
        "eventId" => $eventId,
        "userIds" => $userIds,
        "status" => $notificationStatus
    );

    return $notificationData;
}
?>

```

## Pdf.php

```

<?php defined('BASEPATH') OR exit('No direct
script access allowed');

/**
 * CodeIgniter PDF Library
 *
 * Generate PDF's in your CodeIgniter
 * applications.
 *
 * @package CodeIgniter
 * @subpackage Libraries
 * @category Libraries
 * @author Chris Harvey
 * @license MIT License
 * @link https://github.com/chrisnharvey/
 * CodeIgniter-PDF-Generator-Library
 */

require_once(dirname(__FILE__) . '/dompdf/
dompdf_config.inc.php');

class Pdf extends DOMPDF
{
    /**
     * Get an instance of CodeIgniter
     *
     * @access protected
     * @return void
     */
    protected function ci()
    {
        return get_instance();
    }

    /**
     * Load a CodeIgniter view into domPDF
     *
     * @access public
     * @param string $view The view to load
     * @param array $data The view data
     * @return void
     */
    public function load_view($view, $data = array
        ())
    {
        $html = $this->ci()->load->view($view, $data,
            TRUE);

        $this->load_html($html);
    }
}

```

## PHPass.php

```

<?php if (!defined('BASEPATH')) exit('No direct
script access allowed');

//http://forum.codeigniter.com/archive/index.
php?thread-50050.html

require_once('phpass/PasswordHash.php');

define('PHPASS_HASH_STRENGTH', 8);
define('PHPASS_HASH_PORTABLE', false);

class PHPass {

    function create_hash($raw_password) {
        // Passwords should never be longer than 72
        // characters to prevent DoS attacks
        if (strlen($raw_password) > 72) { die("
Password must be 72 characters or less"); }

        // Hash raw_password using phpass
        $hasher = new PasswordHash(
            PHPASS_HASH_STRENGTH, PHPASS_HASH_PORTABLE)
        ;
        $hashed_password = $hasher->HashPassword(
            $raw_password);

        // Return the hashed password
        return $hashed_password;
    }

    function check_password($raw_password,
        $hashed_password) {
        //Check if raw_password matches
        //hashed_password when it is hashed using
        //phpass
        $hasher = new PasswordHash(
            PHPASS_HASH_STRENGTH, PHPASS_HASH_PORTABLE)
        ;
        if ($hasher->CheckPassword($raw_password,
            $hashed_password)) {return true;}
        else {return false;}
    }
}
?>

```

## PHPExcel\_Controller.php

```

<?php if (!defined('BASEPATH')) exit('No direct
script access allowed');

require_once APPPATH."/libraries/PHPExcel.php";

class PHPExcel_Controller extends PHPExcel {
    public function __construct() {
        parent::__construct();
    }
}

```

## Validation\_Controller.php

```

<?php if ( ! defined("BASEPATH")) exit("No
direct script access allowed");

/**
 * @author Quizon, Maria Reena I.
 * filename: Validation_Controller.php
 * description: Library for all validation
 * functions in LRC_TMS
 * start date: March 1, 2016
 * end date:
 * last updated:
 * version number: 1.0
 */

class Validation_Controller {

    public function __construct() {
        $this->CI =& get_instance();
        $this->CI->load->helper("form");
        $this->CI->load->helper("url");
        $this->CI->load->library("session");
        $this->CI->load->library("form_validation");
    }
}

```

```

public function check_validation() {
    if (strlen($this->CI->session->userdata("
hasAttended")) > 1) {
        redirect(base_url("/feedback"));
    }
    if (!$this->CI->session->userdata("validated
")) {
        redirect(base_url("/logout"));
    }
}
}
?>

```

## constants.php, lines 87-207

```

define('DEFAULT_INTEGER', 0);
define('DEFAULT_STRING', "");

define('ADMIN', 1);
define('STUDENT', 2);
define('STUDENT_TUTOR', 3);
define('ADMIN_ID', 10000);
define('ALL', "all");

define('APPLICATION_FIRST', 0);
define('APPLICATION_RENEWAL', 1);

define('EMAIL_LRC_ADMIN_NAME', "LRC
Administrator");
define('EMAIL_BLOCK_STUDENT', 1);
define('EMAIL_ENDORSEMENT', 2);
define('EMAIL_NEW_EMAIL', 3);
define('EMAIL_NEW_PASSWORD', 4);

define('EMAIL_SUBJECT_BLOCK_STUDENT', "[LRC]
Blocked User");
define('EMAIL_SUBJECT_ENDORSEMENT', "[LRC]
Endorsement");
define('EMAIL_SUBJECT_NEW_EMAIL', "[LRC]
Changed Email");
define('EMAIL_SUBJECT_NEW_PASSWORD', "[LRC]
Reset Password");

define('EVENT_INACTIVE', 0);
define('EVENT_ACTIVE', 1);
define('EVENT_PENDING', 2);
define('EVENT_FINISHED', 3);
define('EVENT_DID_NOT_ATTEND', 4);

define("NO_EVENT_TYPE", "select event type");
define("SEM_BEGIN", "Beginning of semester");
define("SEM_END", "End of semester");
define("LRC_EVENT", "LRC-related event");
define("MOCK_EXAM", "Simulated exam event");
define("HOLIDAY", "National Holiday");
define("OTHER", "Other events");

define('EMAIL_FAILED', "<script>alert('there
was an error sending the email');</script>"
);
define('EMAIL_SUCCESSFUL', "<script>alert('an
email was successfully sent');</script>");
define('FORM_INCOMPLETE_INPUT', "<script>alert
('incomplete input');</script>");
define('FORM_SUCCESSFUL', "<script>alert('
successful');</script>");
define('FORM_APPOINTMENT_RESCHEDULE', "<div
class = 'alert alert-warning text-center'>
reschedule appointment</div>");
define('FORM_APPOINTMENT_NO_TUTORS', "<div
class = 'alert alert-warning text-center'>
There are no more tutors in your chosen
time frame. Either there are no more tutors
or you should select another date.</div>"
);
define('APPLICATION_PROCEED_1', "proceed_1");
define('APPLICATION_PROCEED_2', "proceed_2");
define('APPLICATION_PROCEED_3', "proceed_3");
define('APPLICATION_INCOMPLETE_1', "
incomplete_1");

```

```

define('APPLICATION_INCOMPLETE_2', "
incomplete_2");
define('APPLICATION_INCOMPLETE_3', "
incomplete_3");

define('WARNING', 100);
define('BLOCKED', 101);
define('UNBLOCKED', 102);

define('APPLICATION_NEW', 200);
define('APPLICATION_RENEWED', 201);
define('APPLICATION_APPROVED', 202);
define('APPLICATION_REJECTED', 203);
define('APPLICATION_CHANGED_STUDENT', 204);
define('APPLICATION_CANCELLED_STUDENT', 205);
define('FACULTY_ENDORSED', 206);
define('FACULTY_REJECTED', 207);

define('APPOINTMENT_NEW', 300);
define('APPOINTMENT_NEW_ADMIN', 301);
define('APPOINTMENT_CANCELLED_ADMIN', 302);
define('APPOINTMENT_CANCELLED_ADMIN_FOR_TUTOR',
312);
define('APPOINTMENT_CANCELLED_STUDENT', 303);
define('APPOINTMENT_TODAY', 304);
define('APPOINTMENT_ATTENDED', 305);
define('APPOINTMENT_RESCHEDULED', 306);
define('TUTOR_ACCEPTED', 307);
define('TUTOR_REJECTED', 308);
define('NO_MORE_TUTORS', 309);

define('CALENDAR_SET', 400);
define('CALENDAR_UPDATED', 401);
define('LRC_ANNOUNCEMENT', 402);
define('LRC MOCK_EXAM_NEW', 403);
define('LRC MOCK_EXAM_CANCELLED', 404);
define('LRC MOCK_EXAM_FULL_SLOTS', 405);
define('LRC MOCK_EXAM_TODAY', 406);

define('WARNING_MSG', "You have been issued a
warning against using this system.");
define('BLOCKED_MSG', "You have been blocked
from using the system.");
define('UNBLOCKED_MSG', "You have been
unblocked from using the system.");

define('LOGIN_BLOCKED_MSG', "<script>alert('You
have been blocked from using the system.');"
);</script>");
define('LOGIN_INVALID_MSG', "<script>alert('You
have entered invalid credentials.');"
);</script>");
define('LOGIN_USER_ID_INVALID_MSG', "<script>
alert('You have entered an invalid User Id
.');"
);</script>");

define('PROFILE_OLD_PASSWORD_MSG', "<script>
alert('You incorrectly typed your old
password.');"
);</script>");
define('PROFILE_CONFIRM_PASSWORD_MSG', "<script>
>alert('Please reconfirm your new password
');"
);</script>");

define('APPLICATION_NEW_MSG', "A student has
filed an application.");
define('APPLICATION_RENEWED_MSG', "A student
has renewed an application.");
define('APPLICATION_APPROVED_MSG', "Your
application has been approved by the
administrator.");
define('APPLICATION_REJECTED_MSG', "Your
application has been rejected by the
administrator.");
define('APPLICATION_CHANGED_STUDENT_MSG', "You
requested an endorsement from another
faculty member.");
define('APPLICATION_CANCELLED_STUDENT_MSG', "A
student cancelled the application");
define('FACULTY_ENDORSED_MSG', "Your
endorsement request has been answered.");
define('FACULTY_REJECTED_MSG', "Your
endorsement request has been rejected. Do
you wish to change your choice of faculty
member?");

```

```

define('APPOINTMENT_NEW_ADMIN_MSG', "A student
has requested a new tutorial session.");
define('APPOINTMENT_CANCELLED_ADMIN_MSG', "The
administrator has cancelled the tutorial
session.");
define('APPOINTMENT_CANCELLED_STUDENT_MSG', "A
student has cancelled the tutorial session.
");
define('APPOINTMENT_TODAY_MSG', "There is an
appointment scheduled today.");
define('APPOINTMENT_ATTENDED_MSG', "You
attended an appointment.");
define('APPOINTMENT_RESCHEDULED_MSG', "A
student rescheduled an appointment.");
define('TUTOR_ACCEPTED_MSG', "A tutorial
request was accepted.");
define('TUTOR_REJECTED_MSG', "A tutorial
request was rejected.");
define('NO_MORE_TUTORS_MSG', "There are no more
tutors available to accept the tutorial
request.");

define('CALENDAR_SET_MSG', "The academic
calendar for the system must be set.");
define('CALENDAR_UPDATED_MSG', "The academic
calendar has been updated.");
define('LRC_ANNOUNCEMENT_MSG', "Announcement");
define('LRC MOCK_EXAM_NEW_MSG', "Mock exam
schedule");
define('LRC MOCK_EXAM_CANCELLED_MSG', "The mock
exam has been cancelled.");
define('LRC MOCK_EXAM_FULL_SLOTS_MSG', "There
mock exam already has full slots.");
define('LRC MOCK_EXAM_TODAY_MSG', "There is a
mock exam scheduled today.");

```

## routes.php, lines 57-119

```

$route['login'] = 'main/Login/index';
$route['login/process_login'] = 'main/Login/
process_login';
$route['login/forgot_password'] = 'main/Login/
forgot_password';
$route['dashboard'] = 'main/Dashboard/index';
$route['dashboard/cancel_appointment'] = 'main/
Dashboard/cancel_appointment';
$route['dashboard/process_appointment'] = 'main
/Dashboard/process_appointment';
$route['dashboard/save_schedule'] = 'main/
Dashboard/save_schedule';
$route['logout'] = 'main/Dashboard/logout';
$route['profile'] = 'main/Profile/index';
$route['profile/process_account'] = 'main/
Profile/process_account';
$route['profile/view_profile/:num'] = 'main/
Profile/view_profile';
$route['search'] = 'main/Search/index';
$route['search/process_search'] = 'main/Search/
process_search';

$route['apply'] = 'apply/Application/index';
$route['apply/process_1'] = 'apply/Application/
process_application_1';
$route['apply/process_2'] = 'apply/Application/
process_application_2';
$route['apply/process_3'] = 'apply/Application/
process_application_3';
$route['apply/change_application/:num'] = '
apply/Application/change_application';
$route['apply/dashboard'] = 'apply/Dashboard/
index';
$route['apply/dashboard/process_approval'] = '
apply/Dashboard/process_approval';

$route['endorsement/:any'] = 'apply/Endorsement
/index';
$route['endorsement/process_acceptance/:any'] =
'apply/Endorsement/process_acceptance';
$route['endorsement/process_endorsement/:any']
= 'apply/Endorsement/process_endorsement';

```

```

$route['appointment'] = 'appointment/
Appointment/index';
$route['appointment/reschedule_appointment'] =
'appointment/Appointment/
reschedule_appointment';
$route['appointment/process_appointment'] = '
appointment/Appointment/process_appointment
';

$route['appointment/dashboard'] = 'appointment/
Dashboard/index';
$route['appointment/dashboard/check_attendance']
= 'appointment/Dashboard/check_attendance
';

$route['feedback'] = 'feedback/Feedback/index';
$route['feedback/process_feedback'] = 'feedback
/Feedback/process_feedback';
$route['view_feedback/:num'] = 'feedback/
Feedback/view_feedback';

$route['lessons'] = 'upload/Lesson/index';
$route['lessons/create_lesson'] = 'upload/
Lesson/create_lesson';
$route['lessons/manage_lesson'] = 'upload/
Lesson/manage_lesson';
$route['lessons/get_lessons/:any'] = 'upload/
Lesson/get_lessons';
$route['lessons/process_lesson'] = 'upload/
Lesson/process_lesson';

$route['events'] = 'upload/Upload/index';
$route['events/:num/:num'] = 'upload/Upload/
index/$1/$2';
$route['events/manage_event'] = 'upload/Upload/
manage_event';
$route['events/process_event'] = 'upload/Upload
/process_event';
$route['events/get_students/:num'] = 'upload/
Upload/get_students';
$route['events/cancel_mockexam'] = 'upload/
Upload/cancel_mockexam';
$route['events/apply_mock_exam'] = 'upload/
Mock_Exam/apply_mock_exam';

$route['reports'] = 'report/Report/index';
$route['reports/process_report_pdf'] = 'report/
Report/process_report_pdf';
$route['reports/process_account_pdf'] = 'report
/Report/process_account_pdf';

$route['accounts'] = 'upload/Account/index';
$route['accounts/process_account'] = 'upload/
Account/process_account';
$route['accounts/process_accounts'] = 'upload/
Account/process_accounts';
$route['accounts/block_account'] = 'upload/
Account/block_account';
$route['accounts/reset_tutors'] = 'upload/
Account/reset_tutors';
$route['accounts/add_email'] = 'upload/Account/
add_email';
$route['accounts/change_password'] = 'upload/
Account/change_password';

```

## application.css

```

.application-image {
width: 50%;
}

```

## application.dashboard.css

```

.img-hover img {
-webkit-transition: all .3s ease; /* Safari
and Chrome */
-moz-transition: all .3s ease; /* Firefox */
-o-transition: all .3s ease; /* IE 9 */
-ms-transition: all .3s ease; /* Opera */
transition: all .3s ease;
}

```

```

.img-hover img:hover {
  -webkit-backface-visibility: hidden;
  backface-visibility: hidden;
  -webkit-transform:translateZ(0) scale(1.5);
  /* Safari and Chrome */
  -moz-transform:scale(1.5); /* Firefox */
  -ms-transform:scale(1.5); /* IE 9 */
  -o-transform:translatZ(0) scale(1.5); /*
  Opera */
  transform:translatZ(0) scale(1.5);
}

.grayscale {
  -webkit-filter: brightness(1.10) grayscale
    (100%) contrast(90%);
  -moz-filter: brightness(1.10) grayscale(100%)
    contrast(90%);
  filter: brightness(1.10) grayscale(100%);
}

.modal-dialog{
  overflow-y: initial
}
.modal-body{
  height: 80%;
  overflow-y: auto;
}

```

## calendar.css

```

.calendar {
  width: 100%;
  border-collapse: collapse;
}

.calendar tbody tr:first-child th {
  color: #505050;
  margin: 0 0 10px 0;
  text-align: center;
}

.day_header {
  font-weight: normal;
  text-align: center;
  color: #757575;
  font-size: 10px;
}

.calendar td {
  width: 14%; /* Force all cells to be about the
  same width regardless of content */
  border:1px solid #CCC;
  height: 100%;
  vertical-align: top;
  font-size: 10px;
  padding: 0;
}

.calendar td:hover {
  background: #F3F3F3;
}

.day_listing {
  display: block;
  text-align: right;
  font-size: 12px;
  color: #2C2C2C;
  padding: 5px 5px 0 0;
}

div.today {
  background: #E9EFF7;
  height: 100%;
}

```

## dashboard.css

```

.modal.in {
  display: block;
}

```

## login.css

```

* {
  -webkit-border-radius: 0 !important;
  -moz-border-radius: 0 !important;
  border-radius: 0 !important;
}

.glyphicon {
  padding-right: 10px;
}

.modal {
  text-align: center;
  padding: 0;
}

.modal:before {
  content: '';
  display: inline-block;
  height: 100%;
  vertical-align: middle;
  margin-right: -4px;
}

.modal-dialog {
  display: inline-block;
  text-align: left;
  vertical-align: middle;
}

.carousel-inner {
  height: 100%;
  width: 100%;
  padding: 0%;
  margin: 0%;
}

.carousel-control.left, .carousel-control.right
{
  background-image: none
}

.overlay {
  position: absolute;
  top: 5%;
  left: 80%;
  width: 100%;
  height: 100%;
  z-index: 10;
  pointer-events: none;
}

.overlay img {
  height: 30%;
}

@media (min-width: 768px) and (max-width: 979px
) {
  .overlay {
    width:50%;
    height:50%;
  }
}

@media (max-width: 767px) {
  .overlay {
    width:50%;
    height:50%;
  }
}

@media (max-width: 480px) {
  .overlay{
    width:50%;
    height:50%;
  }
}

.section {
  padding-top: 50px;
  height: 100%;
}

.user-ability {
  height: 70%;
}

#home {
  background-color: #FFFFFF;
}

```

```

}
#about {
background-color: #FFFFFF;
overflow-y: auto;
}
#contact {
background-color: #FFFFFF;
}

```

## main.css

```

* {
-webkit-border-radius: 0 !important;
-moz-border-radius: 0 !important;
border-radius: 0 !important;
}

body {
padding-top: 70px;
}

label {
padding-top: 5px;
}

.panel-footer {
background-color: #FFFFFF;
border: none;
padding-top: 0px;
}

.nav-pills > li.active > a, .nav-pills > li.
active > a:hover, .nav-pills > li.active >
a:focus {
background-color: #2ECC71;
}

.table-centered {
margin: 0 auto !important;
float: none !important;
}

.btn-success {
background-color: #2ECC71;
}

.btn-danger {
background-color: #E74C3C;
}

#sidebar li.active a,
#sidebar li.active a:hover {
background-color: #2ECC71;
}

```

## navigation.css

```

.glyphicon {
padding-right: 10px;
}

```

## profile.css

```

input[type="password"] {
margin-right: 5px;
}

.profile-image {
margin-bottom: 25px;
}

```

## jquery.affix.js

```

/*
* http://fellowtuts.com/twitter-bootstrap/
bootstrap-3-sidebar-nav-fixed-responsive-
affix-full-example/
*
*/

```

```

$(document).ready(function() {
var offset = 70;
// Function to handle affix width and classes
in affix menu on page loading, scrolling
or resizing
function affix() {
// Fit affix width to its parent's width
var $affixElement = $("div[data-spy='affix
']");
$affixElement.width($affixElement.parent().
width());

// Position of vertical scrollbar
var position = $(window).scrollTop();
if (position >= offset) {
$(".wrapper .section").each(function(i) {
// Current content block's position
less body padding
var current = $(this).offset().top -
offset - 1;

// Add active class to corresponding
affix menu while removing the same from
siblings as per position) of current block
if (current <= position) {
$("a", $affixElement).eq(i).addClass(
"active").siblings().removeClass("active");
}
} else {
$("a", $affixElement).find(".active").
removeClass("active").end().find(":first").
addClass("active");
}
}
});

// Call to function on DOM ready
affix();

// Call on scroll or resize
$(window).on("scroll resize", function() {
affix();
});
});

```

## jquery.application.dashboard.js

```

$(document).ready(function() {

$(".application-id").click(function() {
$("#application-form").empty();
$.ajax({
url: "/tms_agila/apply/Dashboard/
get_application?applicationId=" + $(this
).text(),
type: "GET",
dataType: "json",
success: function (applicationString) {
$("#application-form").append(
applicationString);
}
});
});

$(".endorsement-id").click(function() {
$("#endorsement-form").empty();
$.ajax({
url: "/tms_agila/apply/Dashboard/
get_endorsement?endorsementId=" + $(this
).text(),
type: "GET",
dataType: "json",
success: function (endorsementString) {
$("#endorsement-form").append(
endorsementString);
}
});
});
});
});

```

## jquery.application.js

```
$(document).ready(function() {  
    var chemSubjects = ["none", "14", "18", "27",  
        "31", "40", "156", "NSI part"];  
    var mathSubjects = ["none", "1", "11", "14",  
        "17", "101", "73", "74", "75"];  
    var physicsSubjects = ["none", "21", "51", "  
        52", "71", "72", "73", "NSI part"];  
    var bioSubjects = ["none", "10", "20", "21",  
        "22", "25", "30", "100", "101"];  
    var comsciSubjects = ["none", "11", "21", "55  
        "];  
    var commSubjects = ["none", "1", "2", "3"];  
  
    var minFields = 0;  
    var maxFields = 5;  
  
    var stack = [];  
    var oldSubject;  
    var subjectSelect;  
    var organizationSelect;  
  
    $("#add-organization").click(function(e) {  
        e.preventDefault();  
        if(minFields < maxFields){  
            add_organization(minFields);  
            $("#no-organization").attr("disabled",  
                true);  
            minFields++;  
        }  
    });  
    $("#no-organization").click(function(e) {  
        e.preventDefault();  
        var organizationSelect = "<div class = 'row  
>";  
        organizationSelect += "<div class = 'col  
col-lg-12'><br/>";  
        organizationSelect += "<div class = 'col  
col-lg-4'><label>Organization Name</label  
></div>";  
        organizationSelect += "<div class = 'col  
col-lg-8 text-center'><input name = '  
organizations[]' value = 'no organization'  
class = 'form-control' readOnly = 'true' />  
";  
        organizationSelect += "<input name = '  
positions[]' value = 'no position' class =  
'form-control' readOnly = 'true' /><a href=  
'#' class= 'remove-org' >remove</a></div>"  
        ;  
        organizationSelect += "</div></div>";  
        $("#organization").empty();  
        $("#organization").append(  
            organizationSelect);  
        $("#add-organization").attr("disabled",  
            true);  
    });  
    $("#organization").on("click", ".remove-org",  
        function(e){  
            minFields--;  
            e.preventDefault();  
            $(this).parent("div").parent("div").remove  
            ();  
            $("#add-organization").attr("disabled",  
                false);  
            if (minFields <= 0) {  
                $("#no-organization").attr("disabled",  
                    false);  
            }  
        });  
  
    $(".subject-rank").on("focus", function(e) {  
        oldSubject = $(this).val();  
    }).change(function(e) {  
        var currSubject = this.options[e.target.  
            selectedIndex].text;  
        var currSubjectSelect = $(this).attr("index  
            ");  
        if (stack.length > 0 &&& oldSubject !== "")  
            {  
                stack = jQuery.grep(stack, function(value  
            ) {  
                    return value != oldSubject;  
                });  
            }  
        add_subject(oldSubject, currSubjectSelect);  
        remove_subject(currSubject,  
            currSubjectSelect);  
        create_subject_div(currSubject,  
            currSubjectSelect);  
        stack.push(currSubject);  
    });  
    $("#subject-rank").on("click", ".remove-  
subject", function(e){  
        e.preventDefault();  
        subjectSelect = $(this).parent("div").  
            parent("div").parent("div").find("select").  
            attr("id");  
  
        var removedSubject = $("#" + subjectSelect)  
            .val();  
        var currSubjectSelect = $("#" +  
            subjectSelect).attr("index");  
        stack = jQuery.grep(stack, function(value)  
            {  
                return value != removedSubject;  
            });  
        add_subject(removedSubject,  
            currSubjectSelect);  
        $("#" + subjectSelect).val("");  
        $(this).parent("div").remove();  
    });  
  
    $(".free-time").click(function() {  
        var freeTimeId = $(this).attr("name");  
        if ($("#input[name= "+freeTimeId+"]").val()  
            === "0") {  
            $("#input[name= "+freeTimeId+"]").val("1")  
            ;  
            $(this).attr("class", "btn btn-success");  
        }  
        else {  
            $("#input[name= "+freeTimeId+"]").val("0")  
            ;  
            $(this).attr("class", "btn btn-danger");  
        }  
    });  
  
    function add_organization(minFields) {  
        var organization = "<div class = 'row'>";  
        organization += "<div class = 'col col-lg  
-12'><br/>";  
        organization += "<div class = 'col col-lg  
-4'><label>Organization Name</label></div>"  
        ;  
        organization += "<div class = 'col col-lg  
-8 text-center'><select name = '  
organizations[]' id = 'organization-select-  
"+minFields+"' class = 'form-control'>";  
  
        $.ajax({  
            url: "/tms_agila/application/js/  
            organizations.xml",  
            type: "GET",  
            dataType: "xml",  
            error: function(){  
                alert("Error loading XML document");  
            },  
            success: function(xml){  
                $(xml).find("organization").each(  
                    function() {  
                        $("#organization-select-"  
                            +minFields).  
                            append("<option value = '" +  
                                $(this).text().  
                                toString() + "' >" +  
                                $(this).text().toStrin  
                                g() + "</option>");  
                    }  
                );  
                $("#organization-select-"  
                    +minFields).  
                    append("</select>");  
            }  
        });  
  
        organization += "<input name= 'positions[]'  
            class = 'form-control' placeholder = '  
            Enter position' required/><a href= '#'  
            class= 'remove-org' >remove</a></div>";  
        organization += "</div></div>";  
    }  
});
```

```

    $("#organization").append(organization);
}

function add_subject(replacedSubject,
currSubjectSelect) {
    for (var index = 1; index < 7; index++) {
        if (index != currSubjectSelect) {
            $("#subject-rank-"+ index).append( $("#<
option>", { value: replacedSubject, text:
replacedSubject} ));
        }
    }
}

function remove_subject(currSubject,
currSubjectSelect) {
    for (var index = 1; index < 7; index++) {
        if (index != currSubjectSelect) {
            $("#subject-rank-"+ index + " option[
value = "+ currSubject +"]").remove();
        }
    };
}

function create_subject_div(currSubject,
currSubjectSelect) {
    var subjectToAppend;
    var subjectAppend = "<div id = 'subject-
rank-" + currSubjectSelect + "-taken-div'
class = 'text-center'><br/>";

    switch (currSubject) {
        case "Chemistry": subjectToAppend =
chemSubjects; break;
        case "Mathematics": subjectToAppend =
mathSubjects; break;
        case "Physics": subjectToAppend =
physicsSubjects; break;
        case "Biology": subjectToAppend =
bioSubjects; break;
        case "Computer-Science": subjectToAppend
= comsciSubjects; break;
        case "Communications": subjectToAppend =
commSubjects; break;
        default: break;
    }

    $.each(subjectToAppend, function (index,
value) {
        subjectAppend += "<label class = '
checkbox-inline'><input type = 'checkbox'
name = 'subjectRank_" + currSubjectSelect + "
_taken[]" value = '" + value + "'>" + value
+ "</label>";
    });

    subjectAppend += "&nbsp;&nbsp;&nbsp;<a href = '#'
class = 'remove-subject'>remove</a></label
></div>";
    if ($("#subject-rank-"+ currSubjectSelect +
"-div").html().length > 0) {
        $("#subject-rank-"+ currSubjectSelect +"-
div").empty();
    }
    $("#subject-rank-"+ currSubjectSelect +"-
div").append(subjectAppend);
}
});

```

## jquery.appointment.js

```

$(document).ready(function() {
    $("#datetimepicker").on("changeDate", function
() {
        if ($("#date-picker").val() != "") {
            $.ajax({
                url: "/tms_agila/appointment/Appointment/
get_available_time?date=" + $("#date-
picker").val(),
                type: "GET",
                dataType: "json",

```

```

                success: function(timeString) {
                    $("#time").empty();
                    $("#time").append(timeString);
                    if (timeString.indexOf("invalid date
                    selected") != -1) {
                        $("#input:submit").attr("disabled", true);
                    }
                    else {
                        $("#input:submit").attr("disabled", false)
                    }
                }
            });
        }
    }
    else {
        $("#time").empty();
    }
});
$("#subject-search").on("change", function() {
    $("#time-search").empty();
    if ($("#subject-search").val().indexOf("no
subject selected") == -1) {
        $.ajax({
            url: "/tms_agila/appointment/Appointment/
search_available_time?subject=" + $("#
subject-search").val(),
            type: "GET",
            dataType: "json",
            success: function(timeSearchString) {
                $("#time-search").append(timeSearchString)
            }
        });
    }
});
$("#time").on("change", "#time-select",
function() {
    $.ajax({
        url: "/tms_agila/appointment/Appointment/
get_slots?date=" + $("#date-picker").val
() + "&time=" + $("#time-select").val(),
        type: "GET",
        dataType: "json",
        success: function(availabilityString) {
            $("#availability").empty();
            $("#availability").append(
availabilityString);
            if (availabilityString.indexOf("there are
no more slots") != -1) {
                $("#input:submit").attr("disabled", true);
            }
            else {
                $("#input:submit").attr("disabled", false);
            }
        }
    });
});
});

```

## jquery.endorsement.js

```

$(document).ready(function() {
    var minimumFields = 1;
    var maxFields = 10;

    var subjectArray = ["Chemistry", "Mathematics"
, "Physics", "Biology", "Computer Science"
, "Communications"];
    var gradeArray = ["1.00", "1.25", "1.50", "
1.75", "2.00", "2.25", "2.50", "2.75", "
3.00", "4.00", "5.00"];

    var subjectAppend = "<div class = 'row'>";
    subjectAppend += "<div class = 'col col-lg
-12'>";
    subjectAppend += "<div class = 'col col-lg
-4'><label>Subject</label></div>";
    subjectAppend += "<div class = 'col col-lg
-8 form-group text-center'><select name = '
subjects[]" class = 'form-control'>";

```

```

for (var index = 0; index < subjectArray.
length; index++) {
    subjectAppend += "<option value = '" +
subjectArray[index] + "'>" + subjectArray[
index] + "</option>";
}
subjectAppend += "</select>";
subjectAppend += "<input name = 'codes[]',
class = 'form-control' required/><select
name = 'grades[]' class = 'form-control' >"
;

for (var index = 0; index < gradeArray.
length; index++) {
    subjectAppend += "<option value = '" +
gradeArray[index] + "'>" + gradeArray[index
] + "</option>";
}

subjectAppend += "</select>";
subjectAppend += "<a href= '#' class= '
remove-subject'>remove</a></div><br/>";
subjectAppend += "</div></div>";

$("#add-subject").click(function(e) {
    e.preventDefault();
    if(minimumFields < maxFields){
        minimumFields++;
        $("#subject").append(subjectAppend);
    }
});
$("#subject").on("click", ".remove-subject",
function(e){
    e.preventDefault();
    $(this).parent("div").parent("div").remove
();
    minimumFields--;
});
});

```

## jquery.lesson.js

```

$(document).ready(function() {

$("#input:radio[name~= 'changeFile']").click(
function() {
    if ($(this).val() === "yes") {
        $("#lesson-link").show();
        $("#lesson-link-new").attr("required", true)
    }
    else {
        $("#lesson-link").hide();
        $("#lesson-link-new").attr("required", false
);
    }
});

$("#code").on("change", function() {
$.ajax({
    url: "/tms_agila/upload/Lesson/
get_lesson_position?subject=" + $("#
subject").val() + "&code=" + $("#code").
val() + "&lessonTitle=" + $("#lesson-
title").val() + "&edit=" + $("#edit").
val(),
    type: "GET",
    dataType: "json",
    success: function (lessonString) {
        $("#lesson-position").empty();
        $("#lesson-position").append(lessonString);
    }
});
});
});

```

## jquery.main.dashboard.js

```

$(document).ready(function() {

```

```

$(".notification").on("click", ".change-
faculty", function() {
$.ajax({
    url: "/tms_agila/apply/Application/
get_faculty",
    type: "GET",
    dataType: "json",
    success: function(facultyString) {
        $(".change-faculty").attr("disabled", true)
        ;
        $(".faculty-select").empty();
        $(".faculty-select").append(facultyString);
    }
});
});
$(".notification").on("click", ".cancel",
function() {
    $(".change-faculty").attr("disabled", false);
    $(".faculty-select").empty();
});
});

```

## jquery.reports.js

```

$(document).ready(function(e) {

$(".submit").on("click", function() {
    var reportType = $(".ul.nav > li.active").
attr("type");
    var period = $("#period").val();
    var startDate = $("#start-date").val();
    var endDate = $("#end-date").val();

$.ajax({
    url: "/tms_agila/report/Report/
process_reports?reportType=" + reportType +
"&period=" + period + "&startDate=" +
startDate + "&endDate=" + endDate,
    type: "GET",
    dataType: "json",
    success: function (breakdownString) {
        $("#chart").empty();
        $("#chart").append(breakdownString);
    }
});

});

$("#chart").on("click", ".cancel", function()
{
    $("#chart").empty();
});
});

```

## jquery.scroll.js

```

$(document).ready(function() {
$(".navbar ul li a[href^=#]").on('click',
function(event) {
    var target;
    target = this.hash;

    event.preventDefault();

    var navOffset;
    navOffset = $(' .navbar ').height();

    return $('html, body').animate({
        scrollTop: $(this.hash).offset().top -
navOffset
    }, 300, function() {
        return window.history.pushState(null, null
, target);
    });
});
});

```

## jquery.subject.js

```

$(document).ready(function() {

```



```

var subjects = {
  "no subject selected" : [""],
  "Chemistry": ["14", "18", "27", "31", "40", "156", "NSI part"],
  "Mathematics": ["1", "11", "14", "17", "101", "73", "74", "75"],
  "Physics": ["21", "51", "52", "71", "72", "73", "NSI part"],
  "Biology": ["10", "20", "21", "22", "25", "30", "100", "101"],
  "Computer-Science": ["11", "21", "55"],
  "Communications": ["1", "2", "3"]
};

for (var prop in subjects) {
  var first = prop;
  if (currSubject.length > 0 && currSubject === first) {
    $("#subject").append(
      $("#<option selected>").
      attr("value", first).
      text(first)
    );
  }
  else {
    $("#subject").append(
      $("#<option>").
      attr("value", first).
      text(first)
    );
  }
}

$("#subject").change(
function() {
  var firstKey = $("#subject").val();
  $("#code").html("");
  for (var index = 0; index < subjects[
    firstKey].length; index++) {
    var second = subjects[firstKey][index];
    if (currCode.length > 0 && currCode === second) {
      $("#code").append(
        $("#<option selected>").
        attr("value", second).
        text(second)
      );
    }
    else {
      $("#code").append(
        $("#<option>").
        attr("value", second).
        text(second)
      );
    }
  }
  $("#code").change();
}).change();
});

```

## jquery.subject.search.js

```

$(document).ready(function() {
  var subjects = {
    "no subject selected" : [""],
    "Chemistry": ["14", "18", "27", "31", "40", "156", "NSI part"],
    "Mathematics": ["1", "11", "14", "17", "101", "73", "74", "75"],
    "Physics": ["21", "51", "52", "71", "72", "73", "NSI part"],
    "Biology": ["10", "20", "21", "22", "25", "30", "100", "101"],
    "Computer-Science": ["11", "21", "55"],
    "Communications": ["1", "2", "3"]
  };
  for (var prop in subjects) {

```

```

    var first = prop;
    $("#subject-search").append(
      $("#<option>").
      attr("value", first).
      text(first)
    );
  }
  $("#subject-search").change(
function() {
  var firstKey = $("#subject-search").val();
  $("#code-search").html("");
  for (var index = 0; index < subjects[
    firstKey].length; index++) {
    var second = subjects[firstKey][index];
    $("#code-search").append(
      $("#<option>").
      attr("value", second).
      text(second)
    );
  }
  $("#code-search").change();
}).change();
});

```

## jquery.upload.js

```

$(document).ready(function() {
  var subjectArray = ["no subject selected", "Chemistry", "Mathematics", "Physics", "Biology", "Computer Science", "Communications"];
  var chemSubjects = ["14", "18", "27", "31", "40", "156", "NSI part"];
  var mathSubjects = ["1", "11", "14", "17", "101", "73", "74", "75"];
  var physicsSubjects = ["21", "51", "52", "71", "72", "73", "NSI part"];
  var bioSubjects = ["10", "20", "21", "22", "25", "30", "100", "101"];
  var comsciSubjects = ["11", "21", "55"];
  var commSubjects = ["1", "2", "3"];

  var cancelButton = "<div class = 'text-center '><button class = 'cancel btn btn-default '>close</button></div>";
  var subjectAppend = "<label>Subject</label><select name = 'subject' id = 'subject' class = 'form-control' required>";

  for (var index = 0; index < subjectArray.length; index++) {
    if (index == 0) {
      subjectAppend += "<option value = ''>" + subjectArray[index] + "</option>";
    }
    else {
      subjectAppend += "<option value = '" + subjectArray[index] + "'>" + subjectArray[index] + "</option>";
    }
  }
  subjectAppend += "</select>";

  $("#mock-exam").on("change", "#subject", function() {
    var codes;
    var codeAppend;

    switch ($("#subject").val()) {
      case "Chemistry": codes = chemSubjects; break;
      case "Mathematics": codes = mathSubjects; break;
      case "Physics": codes = physicsSubjects; break;
      case "Biology": codes = bioSubjects; break;
      case "Computer Science": codes = comsciSubjects; break;
      case "Communications": codes = commSubjects; break;
    }
  });

```

```

if ($("#subject").val() !== "") {
  codeAppend = "<select name = 'code' class =
'form-control' required>";
  for (var index = 0; index < codes.length;
index++) {
    codeAppend += "<option value = '" + codes[
index] + "'>" + codes[index] + "</
option>";
  }
  codeAppend += "</select>";
}

$("#code-append").empty();
$("#code-append").append(codeAppend);
});

$("#event-type").on("change", function() {
if ($("#event-type").val() === "MOCK_EXAM") {
mockExamString = subjectAppend;
mockExamString += "<div id = 'code-append
'></div>";
mockExamString += "<label>Mock exam time</
label>";
mockExamString += "<div id = 'time-picker-1'
class = 'input-group input-append date
'>";
mockExamString += "<input name = 'time'
value = '" + $("input:hidden[name='
timeHidden']").val() + "' data-format =
'hh:mm:ss' class = 'form-control'
required/>";
mockExamString += "<span class = 'input-
group-addon add-on'><i data-time-icon =
'glyphicon glyphicon-time' data-date-
icon = 'glyphicon glyphicon-calendar'></
i></span></div>";
mockExamString += "<label>Discussion date</
label>";
mockExamString += "<div id = 'date-picker'
class = 'input-group input-append date'
>";
mockExamString += "<input name =
discussionDate' value = '" + $("input:
hidden[name~='discussionDateHidden']").
val() + "' data-format = 'yyyy-MM-dd'
class = 'form-control' required/>";
mockExamString += "<span class = 'input-
group-addon add-on'><i data-time-icon =
'glyphicon glyphicon-time' data-date-
icon = 'glyphicon glyphicon-calendar'></
i></span></div>";
mockExamString += "<label>Discussion time</
label>";
mockExamString += "<div id = 'time-picker-2'
class = 'input-group input-append date
'>";
mockExamString += "<input name =
discussionTime' value = '" + $("input:
hidden[name~='discussionTimeHidden']").
val() + "' data-format = 'hh:mm:ss'
class = 'form-control' required/>";
mockExamString += "<span class = 'input-
group-addon add-on'><i data-time-icon =
'glyphicon glyphicon-time' data-date-
icon = 'glyphicon glyphicon-calendar'></
i></span></div>";
mockExamString += "<label>Slots</label>";
mockExamString += "<input type = 'number'
name = 'slots' value = '" + $("input:
hidden[name~='slotsHidden']").val() +
" min = '5' max = '30' class = 'form-
control' required/><br/>";
mockExamString += cancelButton;
$("#mock-exam").empty();
$("#mock-exam").append(mockExamString);
}
else {
$("#mock-exam").empty();
}
});

$("#mock-exam").on("click", ".cancel",
function() {
$("#mock-exam").empty();
$("#event-type").val("noType");
});

```

```
});
```

## endorsement.xml

```

<?xml version = "1.0" encoding="UTF-8"?>
<questions>
  <question>Knowledge of subject matter</
question>
  <question>Verbal communication skills</
question>
  <question>Ability to work with other people</
question>
  <question>Potential to teach others</question>
  <question>Dependability</question>
</questions>

```

## faculty.xml

```

<?xml version = "1.0" encoding="UTF-8"?>
<faculties>
  <faculty><name>Agapito, Josephine D. DrPH</
name><email>jd_agapito@yahoo.com</email></
faculty>
  <faculty><name>Alcantara, Mark Jun M.</name><
email>mmalcantara3@up.edu.ph</email></
faculty>
  <faculty><name>Baclig, Alvin C.</name><email>
acbaclig@up.edu.ph</email></faculty>
  <faculty><name>Balana, Aaron John T.</name><
email>atbalana@up.edu.ph</email></faculty>
  <faculty><name>Benjamin, Kimberly B.</name><
email>ksbeltran@post.upm.edu.ph</email></
faculty>
  <faculty><name>Billones, Junie B.</name><email>
jbbillones@up.edu.ph</email></faculty>
  <faculty><name>Billones, Liza T.</name><
email>ltbillones@up.edu.ph</email></
faculty>
  <faculty><name>Brillantes, Rochelle Y.</name><
email>rochbrillantes@gmail.com</email></
faculty>
  <faculty><name>Carpio, Avegail D.</name><email>
avegail.carpio@gmail.com</email></faculty>
  <faculty><name>Carrillo, Maria Constancia O.</
name><email>mcocarrillo@yahoo.com</email></
faculty>
  <faculty><name>Chua, Richard Bryann L.</name><
email>rlchua@up.edu.ph</email></faculty>
  <faculty><name>Co, Elisa L.</name><email>elisa
.l.co@gmail.com</email></faculty>
  <faculty><name>Cordova, Dmitri Leo M.</name><
email>dmcordova@up.edu.ph</email></faculty>
  <faculty><name>Dalet, Jay T.</name><email>
jtdalet@up.edu.ph</email></faculty>
  <faculty><name>De Guzman, Margaret C.</name><
email>meggie.deguzman@gmail.com</email></
faculty>
  <faculty><name>De Luna, Marie Josephine M.</
name><email>mmdeluna@up.edu.ph</email></
faculty>
  <faculty><name>De Ocampo, Alyssa Jose S.</name>
<email>asdeocampo@up.edu.ph</email></
faculty>
  <faculty><name>De Vera, Miriam P. PhD</name><
email>desamir@yahoo.com</email></faculty>
  <faculty><name>Dela Cruz, Jobert J.</name><
email>jjdelacruz@up.edu.ph</email></
faculty>
  <faculty><name>Domingo, Herbert B.</name><
email>hbdomingo@up.edu.ph</email></faculty>
  <faculty><name>Endriga, Marla A.</name><email>
maendriga@post.upm.edu.ph</email></faculty>
  <faculty><name>Evangelista, Fredeslinda C.</
name><email>fredes25@yahoo.com</email></
faculty>
  <faculty><name>Fernandez, Randy Joseph G.</
name><email>rgfernandez1@up.edu.ph</email>

```

```

</faculty>
<faculty><name>Frio, Enrique Jose L.</name><
  email>enrique.frio@gmail.com</email></
  faculty>
<faculty><name>Go, Samuel M.</name><email>
  samuelmgo@yahoo.com</email></faculty>
<faculty><name>Gonzaga, Alex C.</name><email>
  acgonzaga@up.edu.ph</email></faculty>
<faculty><name>Hallare, Arnold V.</name><email>
  >avhallare@upm.edu.ph</email></faculty>
<faculty><name>Ignacio, Marvin John C.</name><
  email>mcignacio3@up.edu.ph</email></
  faculty>
<faculty><name>Ila0, Luciana V.</name><email>
  lvilao@up.edu.ph</email></faculty>
<faculty><name>Johnson, Sarah C.</name><email>
  scjohnson@upm.edu.ph</email></faculty>
<faculty><name>Lacdan, Natividad E.</name><
  email>nlf2662@yahoo.com</email></faculty>
<faculty><name>Magboo, Ma. Sheila A.</name><
  email>mamagboo@up.edu.ph</email></faculty>
<faculty><name>Magboo, Vincent Peter C.</name>
  <email>vcmagboo@up.edu.ph</email></faculty
  >
<faculty><name>Marquez, Edanjarlo J.</name><
  email>reophax.ejm@gmail.com</email></
  faculty>
<faculty><name>Mendoza, John Althom A.</name><
  email>jamendoza9@up.edu.ph</email></
  faculty>
<faculty><name>Neri, Gabriel Luis L.</name><
  email>glneri@up.edu.ph</email></faculty>
<faculty><name>Ng, Janice V.</name><email>jah.
  velasco@gmail.com</email></faculty>
<faculty><name>Nicolas, Marilou G.</name>
  <email>mgnicolas@upm.edu.ph</email></
  faculty>
<faculty><name>Nu ez, Julius Andrew P.</name>
  <email>julius.andrew.nunez@upd.edu.ph</
  email></faculty>
<faculty><name>Obico, Jasper John A.</name><
  email>jjaobico@gmail.com</email></faculty>
<faculty><name>Ocampo, Melody Anne B.</name><
  email>melodyocampo@yahoo.com</email></
  faculty>
<faculty><name>Organo, Voltaire G.</name><
  email>vgorgano@up.edu.ph</email></faculty>
<faculty><name>Ragragio, Elena M.</name><email>
  >lenmraragragio@gmail.com</email></faculty>
<faculty><name>Ramirez, Neil Edsel C.</name><
  email>ramirez.neil@gmail.com</email></
  faculty>
<faculty><name>Rubite, Rosario R.</name><email>
  >rosariorubite@yahoo.com</email></faculty>
<faculty><name>Salunga, Thucydides L.</name><
  email>salunga@hotmail.com</email></faculty
  >
<faculty><name>Sia Su, Glenn L.</name><email>
  glss76@yahoo.com</email></faculty>
<faculty><name>Solano, Geoffrey A.</name><
  email>gasolano@up.edu.ph</email></faculty>
<faculty><name>Talana, Arlene D.</name><email>
  arlene.talana@gmail.com</email></faculty>
<faculty><name>Tumlos, Roy B.</name><email>
  rbtumlos@up.edu.ph</email></faculty>
<faculty><name>Torres, John Ephraim E.</
  name><email>jetorres2@up.edu.ph</email></
  faculty>
<faculty><name>Robidillo, Christopher
  Jay T.</name><email>ctrobidillo@up.edu.ph<
  /email></faculty>
</faculties>
<organization>Alpha Sigma Fraternity</
  organization>
<organization>Anakbayan UP Manila</
  organization>
<organization>ASAP Katipunan</organization>
<organization>Bigkis UPM</organization>
<organization>Buklod UPCN</organization>
<organization>Campus Crusade for Christ - UPM
  Chapter</organization>
<organization>Cerulean</organization>
<organization>CFC Youth For Christ UP Manila</
  organization>
<organization>Christ Youth in Action - UPM</
  organization>
<organization>Christian Brotherhood
  International</organization>
<organization>Circle K Club of UP Manila</
  organization>
<organization>Concerned Students for Justice
  and Peace</organization>
<organization>Daebak UPM</organization>
<organization>Freshie Leads</organization>
<organization>Gabriela-Youth</organization>
<organization>Kabataan Para Sa Tribung
  Pilipino</organization>
<organization>Lakbay UPM</organization>
<organization>Lifebox UPM</organization>
<organization>Maralitang Innamorato</
  organization>
<organization>Mu Sigma Phi Fraternity</
  organization>
<organization>Mu Sigma Phi Sorority</
  organization>
<organization>NNARA-Youth Pangkalusugang
  Lingkod Bayan</organization>
<organization>Phi Kappu Mu Fraternity</
  organization>
<organization>Phi Lambda Delta Sorority</
  organization>
<organization>Rotaract Club of UP Manila</
  organization>
<organization>Shoot UPM</organization>
<organization>Sigma Alpha Nu Sorority</
  organization>
<organization>Sigma Delta Pi Sorority</
  organization>
<organization>Sigma Kappa Pi Fraternity</
  organization>
<organization>SOAR Manila</organization>
<organization>Society of Computer Scientists</
  organization>
<organization>Spectrum UP Manila</organization
  >
<organization>State Varsity Christian
  Fellowship Manila</organization>
<organization>Tau Omega Mu and It's Ladies
  Circle</organization>
<organization>The Hope Project</organization>
<organization>Touch Move UPM</organization>
<organization>UP Biological Sciences Society</
  organization>
<organization>UP Caduceus</organization>
<organization>UP Health Sciences and Pre-
  Medicine Society</organization>
<organization>UP Manila Basketball Club</
  organization>
<organization>UP Manila Belle</organization>
<organization>UP Manila Chorale</organization>
<organization>UP Manila Classical Ensemble</
  organization>
<organization>UP Manila Dramatista</
  organization>
<organization>UP Manila Seeds of the Nation</
  organization>
<organization>UP Manila Students of Destiny</
  organization>
<organization>UP Minggan</organization>
<organization>UP M0rg</organization>
<organization>UP One Earth</organization>
<organization>UP Physician Students
  Association</organization>
<organization>UP Saribuhay</organization>
<organization>UP-HOPE</organization>
<organization>UPM Debate Circle</organization>
<organization>UPM HYDRA</organization>
<organization>UPM Indayog Dance Varsity</
  organization>

```

## organizations.xml

```

<?xml version = "1.0" encoding="UTF-8"?>
<organizations>
  <organization>AISEC UPM</organization>
  <organization>Alpha Phi Omega Fraternity</
  organization>
  <organization>Alpha Phi Omega Sorority</
  organization>

```

```

    organization>
<organization>UPM MAO</organization>
<organization>UPM Omake</organization>
<organization>UPM Reader's Circle</
organization>
<organization>UPM Scholar's Club</organization
>
<organization>Upsilon Sigma Phi Fraternity</
organization>
</organizations>

```

## questions.xml

```

<?xml version = "1.0" encoding="UTF-8"?>
<questions>
<question>The tutorial program has met my
positive expectation.</question>
<question>The tutorial program helped me
understand the topics in this course.</
question>
<question>The LRC staff fulfilled my request
in a timely manner.</question>
<question>I am satisfied with my tutorial.</
question>
<question>I am more likely to recommend the
LRC tutorial to others.</question>
<question>The tutor encouraged my interest in
the course topic.</question>
<question>The tutor effectively managed the
tutorial interaction.</question>
<question>The tutor developed good rapport
with the tutee.</question>
<question>The tutor respects the tutees and
responds to their needs.</question>
<question>The tutor's manner of teaching is
easy to understand.</question>
<question>The tutor gave examples to help the
tutee understand the topic.</question>
<question>The tutor felt comfortable in asking
questions.</question>
<question>The tutor encouraged the tutee to
participate actively during the tutorial
.</question>
<question>The tutor was prompt and
professional.</question>
<question>The tutor was well prepared for the
tutorial.</question>
</questions>

```

## lrc\_tms.sql

```

-- phpMyAdmin SQL Dump
-- version 4.1.14
-- http://www.phpmyadmin.net
--
-- Host: 127.0.0.1
-- Generation Time: May 25, 2016 at 09:42 AM
-- Server version: 5.6.17
-- PHP Version: 5.5.12

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=
@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=
@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=
@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;

--
-- Database: 'lrc_tms'
--
-- Table structure for table 'application'
--

CREATE TABLE IF NOT EXISTS 'application' (
'applicationId' int(5) NOT NULL,
'userId' int(5) DEFAULT NULL,
'applicationType' smallint(1) DEFAULT NULL,
'organizations' varchar(500) DEFAULT NULL,

```

```

'positions' varchar(300) DEFAULT NULL,
'faculty' varchar(50) DEFAULT NULL,
'applicationStatus' smallint(1) DEFAULT NULL,
'subjectRank_1' varchar(16) DEFAULT NULL,
'subjectRank_2' varchar(16) DEFAULT NULL,
'subjectRank_3' varchar(16) DEFAULT NULL,
'subjectRank_4' varchar(16) DEFAULT NULL,
'subjectRank_5' varchar(16) DEFAULT NULL,
'subjectRank_6' varchar(16) DEFAULT NULL,
'subjectRank_1_taken' varchar(50) DEFAULT NULL
,
'subjectRank_2_taken' varchar(50) DEFAULT NULL
,
'subjectRank_3_taken' varchar(50) DEFAULT NULL
,
'subjectRank_4_taken' varchar(50) DEFAULT NULL
,
'subjectRank_5_taken' varchar(50) DEFAULT NULL
,
'subjectRank_6_taken' varchar(50) DEFAULT NULL
,
'academicStanding' varchar(100) DEFAULT NULL,
'lastModified' timestamp NOT NULL DEFAULT
CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
PRIMARY KEY ('applicationId'),
KEY 'application_ibfk_1' ('userId')
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Table structure for table 'appointment'
--

CREATE TABLE IF NOT EXISTS 'appointment' (
'appointmentId' int(5) NOT NULL,
'userId' int(5) DEFAULT NULL,
'numberOfStudents' int(2) DEFAULT NULL,
'subject' varchar(16) DEFAULT NULL,
'code' varchar(10) DEFAULT NULL,
'topic' varchar(50) DEFAULT NULL,
'faculty' varchar(50) DEFAULT NULL,
'date' date DEFAULT NULL,
'time' varchar(12) DEFAULT NULL,
'status' smallint(1) DEFAULT NULL,
'tutor_userId' int(5) DEFAULT NULL,
'lastModified' timestamp NOT NULL DEFAULT
CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
PRIMARY KEY ('appointmentId'),
KEY 'userId' ('userId')
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Table structure for table 'calendar'
--

CREATE TABLE IF NOT EXISTS 'calendar' (
'calendarId' int(8) NOT NULL,
'eventName' varchar(100) DEFAULT NULL,
'eventType' varchar(30) DEFAULT NULL,
'eventId' int(5) NOT NULL,
'date' date DEFAULT NULL,
'additionalDetails' varchar(300) DEFAULT NULL,
'flag' smallint(1) NOT NULL DEFAULT '0',
'isAllowed' smallint(1) NOT NULL DEFAULT '1',
'lastModified' timestamp NOT NULL DEFAULT
CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
PRIMARY KEY ('calendarId')
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Table structure for table 'endorsement'
--

CREATE TABLE IF NOT EXISTS 'endorsement' (
'endorsementId' int(5) NOT NULL,
'endorsementLink' varchar(10) DEFAULT NULL,
'applicationId' int(5) DEFAULT NULL,
'userId' int(5) DEFAULT NULL,
'char_1_score' smallint(1) DEFAULT NULL,
'char_2_score' smallint(1) DEFAULT NULL,
'char_3_score' smallint(1) DEFAULT NULL,
'char_4_score' smallint(1) DEFAULT NULL,
'char_5_score' smallint(1) DEFAULT NULL,
'subjects' varchar(300) DEFAULT NULL,

```



```

CREATE TABLE IF NOT EXISTS 'user' (
    'userId' int(5) NOT NULL,
    'userType' smallint(1) DEFAULT NULL,
    'password' varchar(200) DEFAULT NULL,
    'studentNumber' int(9) DEFAULT NULL,
    'name' varchar(50) DEFAULT NULL,
    'email' varchar(50) DEFAULT NULL,
    'course' varchar(50) DEFAULT NULL,
    'college' varchar(4) DEFAULT NULL,
    'birthday' date DEFAULT NULL,
    'profilePicture' varchar(100) DEFAULT NULL,
    'hasAttended' varchar(50) DEFAULT '0',
    'hasApplication' smallint(1) NOT NULL DEFAULT
    '0',
    'approvalCount' int(2) NOT NULL DEFAULT '0',
    'isWarned' smallint(1) DEFAULT '0',
    'isBlocked' smallint(1) DEFAULT '0',
    'blockCount' int(3) DEFAULT '0',
    'batchNumber' int(5) NOT NULL DEFAULT '0',
    PRIMARY KEY ('userId')
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Dumping data for table 'user'
--

INSERT INTO 'user' ('userId', 'userType', '
password', 'studentNumber', 'name', 'email
', 'course', 'college', 'birthday', '
profilePicture', 'hasAttended', '
hasApplication', 'approvalCount', 'isWarned
', 'isBlocked', 'blockCount', 'batchNumber
') VALUES
(10000, 1, '$2a$08$93nWDFeYL7YGsNn/
qSg55001R1YaVWYf8AmctP37n1KpBY2maAu4G',
NULL, 'ADMIN', NULL, NULL, NULL, NULL, NULL
, '0', 0, 0, 0, 0, 0, 0);

--
-- Constraints for dumped tables
--

--
-- Constraints for table 'application'
--
ALTER TABLE 'application'
ADD CONSTRAINT 'application_ibfk_1' FOREIGN
KEY ('userId') REFERENCES 'user' ('userId')
;

--
-- Constraints for table 'appointment'
--
ALTER TABLE 'appointment'
ADD CONSTRAINT 'appointment_ibfk_1' FOREIGN
KEY ('userId') REFERENCES 'user' ('userId')
;

--
-- Constraints for table 'endorsement'
--
ALTER TABLE 'endorsement'
ADD CONSTRAINT 'endorsement_ibfk_1' FOREIGN
KEY ('applicationId') REFERENCES '
application' ('applicationId');

--
-- Constraints for table 'feedback'
--
ALTER TABLE 'feedback'
ADD CONSTRAINT 'feedback_ibfk_1' FOREIGN KEY
('appointmentId') REFERENCES 'appointment'
('appointmentId');

--
-- Constraints for table 'mockexam_students'
--
ALTER TABLE 'mockexam_students'
ADD CONSTRAINT 'mockExam_students_ibfk_1'
FOREIGN KEY ('mockExamId') REFERENCES '
mockexam' ('mockExamId'),
ADD CONSTRAINT 'mockExam_students_ibfk_2'
FOREIGN KEY ('userId') REFERENCES 'user' ('
userId');

--
-- Constraints for table 'schedule'
--
ALTER TABLE 'schedule'
ADD CONSTRAINT 'schedule_ibfk_1' FOREIGN KEY
('userId') REFERENCES 'user' ('userId');

/*!40101 SET CHARACTER_SET_CLIENT=
@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=
@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=
@OLD_COLLATION_CONNECTION */;

```

## **XI. Acknowledgement**

Working with an SP has proven that it is not only a challenge for the mind, but also, it is a feat that requires unrelenting determination and sleepless nights. Now that I have arrived at the most-awaited conclusion of this document, I will waste this page no further in thanking the people who supported me throughout my stay in this University.

Block 12 - Computer Science 2012, I am really thankful and proud that I am a part of the block. All throughout the past four years, we were able to hold on together, despite numerous requirements, head-pounding academics, and that constant fear of failure looming over our heads. You guys showed me that an extra bit of effort and determination will definitely go a long way.

I also thank Block 12 Hotties for being my closest group of friends in my college life. Thank you for being a source of support, especially during the times our future looked a little bleak. Thank you for making me feel the sense of belongingness throughout my college life. A special mention goes to Brendan for being my number one companion. Thank you so much for giving a lot of your time and support, especially when my stress levels are running through the roof. You helped me a lot in academics as well. I know I can't express this well enough, but trust me, I'm really grateful for everything.

To the Computer Science faculty, thank you for being our source of knowledge for the past four years. Also, you were the guiding force that helped me accomplish this SP, especially through critiques and the much-needed pressure to finish the entire project on time. Also, thank you Ma'am Eden for guiding me in other academic-related concerns. RH 114 will always be considered a comforting haven for me and other Computer Science students, thanks to you.

To my adviser, Ma'am Carpio, thank you for your guidance in this SP as well. Your suggestions were vital in the improvement of this project. In one way or another, I am grateful for the patience and time you have given me. Sir Althom, I am very grateful for helping me make this SP possible! Thank you very much for helping me with the SP topic, and also, thank you for checking on my progress from time to time. Your help gave me a greater sense of determination to push harder and achieve the goal of graduating on time. I definitely would have not made this far without you.

During times of great stress, my family was always the number one source of

support and motivation. Thank you for always taking care of me and helping me get myself together. To my mother, I will never forget that you were always by my side. Thank you also for being a source of strength and inspiration. Words are not enough to express everything you have done for me. I am proud to be your daughter, and I love you very much.

I always pray to God, especially when I felt that things were becoming shaky along the way. I would like to thank God for always guiding me towards my goals in life. Trusting You helped me stay calm despite the problems I have faced. I always believe that Your Will will be done, and the chain of events that have unfolded will always be for the better. In everything I do, God be glorified.

After all has been said and done, I want to give myself at least a pat on the back. I never thought I'd make it this far. I have to admit, being a Computer Science student is grueling, to say the least. However, I'm grateful that I have accepted and conquered this challenge. This is one great achievement, and I'm definitely going to savor it.

If there is one lesson I've learned in college, it's staying determined.