

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

UP-PGH DEPARTMENT OF
OTORHINOLARYNGOLOGY
OPERATING ROOM SCHEDULING SYSTEM

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

Submitted by:

Raphael Mari N. Lanuza

June 2015

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

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

ACCEPTANCE SHEET

The Special Problem entitled "UP-PGH Department of Otorhinolaryngology Operating Room Scheduling System" prepared and submitted by Raphael Mari N. Lanuza in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science has been examined and is recommended for acceptance.

Richard Bryann L. Chua, M.Sc.
Adviser

EXAMINERS:

	Approved	Disapproved
1. Gregorio B. Baes, Ph.D. (<i>candidate</i>)	_____	_____
2. Avegail D. Carpio, M.Sc.	_____	_____
3. Perlita E. Gasmen, M.Sc. (<i>candidate</i>)	_____	_____
4. Ma. Sheila A. Magboo, M.Sc.	_____	_____
5. Vincent Peter C. Magboo, M.D., M.Sc.	_____	_____
6. Bernie B. Terrado, M.Sc. (<i>candidate</i>)	_____	_____

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

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

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

Abstract

Mobile technology in medical institutions has been very useful to mankind. It makes lives less demanding and gives accommodation through the applications itself and has been a fundamental part to the change of the nature of medicinal services. In the Philippine General Hospital, the Department of Otorhinolaryngology (ORL) lacks such innovations in mobile healthcare, particularly in arranging and managing operating room (OR) schedules.

The ORL Operating Room Scheduler system is a mobile based application for the users of the Department of Otorhinolaryngology in the Philippine General Hospital. It provides the given functionalities of an operating room scheduler that solves the problem of the OR scheduling in the said institution. In this application, the user can create, organize and arrange OR schedule of patients electronically. The users can also appoint and manage the OR schedules through the application themselves conveniently reducing the complexity of the process. The system can be accessed simply by the users in the Department through smartphones with the Android operating system having an internet connection.

Keywords: operating room, scheduling, Android OS

Contents

Acceptance Sheet	i
Abstract	ii
List of Figures	v
List of Tables	ix
I. Introduction	1
A. Background of the Study	1
B. Statement of the Problem	2
C. Objectives of the Study	3
D. Significance of the Study	4
E. Scope and Limitations	5
F. Assumptions	6
II. Review of Related Literature	7
III. Theoretical Framework	12
IV. Design and Implementation	16
A. Context Diagram	16
B. Use-Case Diagram	18
C. Entity Relationship Diagram (ERD)	22
D. Data Dictionary	23
V. Architecture	25
A. System Architecture	25
B. Technical Architecture	25

VI. Results	26
A. Installation	26
B. ORL OR Scheduler System	27
VII. Discussions	43
VIII. Conclusion	45
IX. Recommendations	46
X. Bibliography	47
XI. Appendix	52
XII. Acknowledgement	264

List of Figures

1	Context Diagram	17
2	Use-case for Consultant	18
3	Use-case for Residents	19
4	Use-case for the Department of Anesthesiology Representative	20
5	Use-case for the Nursing Services Department Representative	21
6	Use-case for the Administrator	22
7	ERD for the ORL OR Scheduler System	23
8	Installation part 1	26
9	Installation part 2	26
10	Installation part 3	26
11	Installation part 4	27
12	Installation part 5	27
13	Installation part 6	27
14	Home Page 1	28
15	Home Page 2	28
16	Login Page	28
17	Register Page	28
18	Register Page cont.	28
19	Register Resident	29
20	Register Consultant	29
21	Register Successful	29
22	Register Alert	29
23	Register Alert 2	29
24	Login Resident	30

25	Login Consultant	30
26	Login Admin	30
27	Login Alert	30
28	Login Alert 2	30
29	Login Alert 3	30
30	Resident Page	31
31	Resident Page 3	31
32	View OR Schedule Page	32
33	View OR Schedule Page 2	32
34	View OR Schedule Page 3	32
35	Create OR Schedule Page	33
36	Create OR Schedule Page 2	33
37	View OR Schedule Page 4	33
38	View OR Schedule Page for Private	33
39	View OR Schedule Page for Private 2	33
40	View OR Schedule Page for Private 3	33
41	Create OR Schedule Page 3	34
42	Create OR Schedule Page 4	34
43	Create OR Schedule Page 5	34
44	Update OR Schedule Page 2	34
45	Update OR Schedule Page 3	34
46	Update OR Schedule Page 4	34
47	Delete OR Schedule Page	35
48	Delete OR Schedule Page 2	35
49	Create OR Schedule Page 6	35

50	Create OR Schedule alert	36
51	View OR Schedule Page for Private 4	36
52	View OR Schedule Page for Private 5	36
53	View OR Schedule Page for Private 6	37
54	User Change Password	37
55	User Change Password Success	37
56	Generate OR Schedule Report Page	37
57	Generate OR Schedule Report Page 2	37
58	Generate OR Schedule Report Page 3	37
59	Generate OR Schedule Report Page 4	38
60	Generate OR Schedule Report Page 5	38
61	Generate OR Schedule Report Page 6	38
62	Admin Page	38
63	User Account Management Page	38
64	Update User Account Page	38
65	Update User Account Page 2	39
66	Update User Account Page 3	39
67	Update User Account Page 4	39
68	Delete User Account Page	40
69	User Account Management Page 2	40
70	Messaging Page	40
71	Send Message Page	40
72	Send Message to Users	40
73	Send Message to Residents	40
74	Send Message to Residents 2	41

75	Inbox Page	41
76	Resident Page with New Message Notification	41
77	Inbox with Message Page	42
78	Message Page	42
79	Delete Message Page	42
80	Message Deleted Page	42

List of Tables

1	users table	23
2	ortab table	24
3	messaging table	24

I. Introduction

A. Background of the Study

The reception of versatile innovations in every measurement of life has been incredible. The troublesome force of these new advances and the going hand in hand with waves of advancement they have started are changing the medicinal services industry, pushing stakeholders to reassess and repurpose how they give administrations [1]. The health care and life sciences industry is recognized as one of the top three fields (along with consumer products and the financial services industry) likely to propel mobile device growth in the next five years [2].

Meanwhile in the operating room in hospitals, computer software is increasingly common in healthcare for operating room scheduling. Advancements have been made fewer in-roads than other applications of operations research, such as manufacturing and logistics [3].

Operating room scheduling includes a course of action in arranging of some operating rooms to surgeons in a period of time. In the wellbeing administration segment, for example, government or private hospitals, the scheduling of Operating Rooms assumes an imperative part towards attaining to their objectives [4]

In the UP-PGH Department of Otorhinolaryngology, the institution does not have a computer software or any application for OR scheduling. Rather, it employs a manual system or the paper based process for scheduling patients due for operation. There are four operating rooms, three charity OR and one pay/private OR, one timeslot for each rooms. The scheduling process depends on the type of the OR. In the charity OR, the assigned resident is contacted via mobile by the consultants who would like to use the OR at a particular timeslot. Failure to contact the resident, the consultants leave a note containing his desired OR timeslot on the residents table instead. Also, the resident himself can make the schedule providing he has the necessary data. The resident can write it first on a paper or directly encode the schedule on a particular OR template on the computer. The residents and his team are in-charge for operating the patients for each room.

In the pay/private OR, the consultants contact someone in the department

whoever has the logbook. The logbook contains a record of pre-made schedules of patients before and for upcoming appointments. The consultants are the ones who operate in the private/pay OR. In this operating room, the process here is a first come first served basis. This simply means that whoever first make the contact shall be the only one who will accommodate the slot for each room. Each room in the private/pay section can only appoint one consultant for each timeslot.

After all the necessary arrangements, it will now be printed having the schedules for both the charity and pay/private OR. The Department actually has a deadline for these schedules to be finished and must be sent immediately to the chairman and chief resident for acknowledgment having their signatures. Then it will be sent to the five sections of the department. These sections are the ear section, rhino section, oral, plastic and larynx section. These are the schedules for the next day.

B. Statement of the Problem

In the Philippine General Hospital, most departments experience difficulties in the management of operating room (OR) schedules. The Department of Otorhinolaryngology (ORL) is an example. At present, the ORL department experiences problems in operating room (OR) patient scheduling, which is brought about by the complicated manual and communications scheduling process that the department utilizes.

OR scheduling in the ORL department is done manually, involving a number of hospital personnel such as residents, consultants and secretary. Such a procedure is both time-consuming and relatively inefficient, considering the group of people needed in order to make the schedule and the complicated scheduling process itself.

One problem with the procedure in the charity OR is that there is no way of knowing OR schedules apart from personally contacting the resident. Another is that contacting the resident may prove difficult at times, particularly when the resident is engaged in a task.

On the other hand, in the pay/private OR, their process is time-consuming as there is no particular person assigned to keep the logbook and the consultant

still has to determine whoever has it at the moment before an OR schedule can be made. There is also no way of knowing if the slot is already taken unless looking in the logbook. The alternative process undertaken by the consultants when that particular person cannot be contacted - just leaving a note on the residents or secretary's table - is problem-ridden as well. It is highly probable that such notes, as they are left unattended, may become mixed-up with the other files on the table or misplaced somewhere else. Moreover, there is no guarantee that the resident will see the note before the supposed scheduled timeslot.

Considering the narrow resources in the UP-PGH, equipment to be used and OR personnel to be involved in the operating rooms may only be limited. These cannot be in the same timeslot parallel to the timeslot of the other operating room. In effect, there are conflict of resources between each personnel should they include. Knowing of the OR details is not possible unless someone contacts the assigned resident or personally look into the schedule itself. These conflict can cause disorganization of the process.

Furthermore, since the editing or updating of the schedule is done by hand, the modification of the schedules may render it illegible after several revisions. Normally, the consultants edit the schedules simply by writing over the previous ones written, making the schedule unreadable and subsequently creating disparities in the final schedule as well.

C. Objectives of the Study

General Objective:

The project aims to provide the Department of Otorhinolaryngology in the Philippine General Hospital with a flexible, efficient, organized and detailed operating room scheduling scheme in a mobile application.

The proposed system should provide a convenient and user-friendly application on mobile phones running in an Android operating system.

The ORL OR Scheduler has the following functionalities:

1. Allow the consultants to:
 - a. Register in the system
 - b. Log-in and Log-out their accounts

- c. View OR schedules
 - d. Update OR schedule details (pay/private OR only)
 - e. Send and Receive messages
 - f. Change password
 - g. Generate OR schedule reports
2. Allows the residents to:
- a. Register in the system
 - b. Log-in and Log-out their accounts
 - c. View OR schedules
 - d. Update OR schedule details (charity OR only)
 - e. Send and Receive messages
 - f. Change password
 - g. Generate OR schedule reports
3. Allows the Department of Anesthesiology Representative or the Nursing Services Department Representative to:
- a. View OR schedules
4. Allows the administrator to:
- a. Log-in and Log-out with his account
 - b. Perform user account management
 - c. Send and Receive messages

D. Significance of the Study

The efficient scheduling of patients due for an operation shall save time. This reduces time spent when using the system compared to manual processing. It removes the paper based for the initial organization and preparations of the OR schedules. The users do not have to manually create or revise paper records of the schedules of the operating rooms. Another is there is no more need for a middle man to compile and encode the schedules on the computer as the system

can generate reports. It avoids consuming time since there is no more encoding of the data, which are the schedules, on the computer.

Managing and viewing of the OR schedules are done online at real time. Meaning there is no need for the consultants to contact the resident assigned in the operating room or someone in the department whoever has the logbook. Using the system, the consultants can easily book themselves into each timeslot in every operating room (pay/private OR). Residents can arrange the schedules without stress in the charity OR as well as reducing the complexity and complications compared to manual scheduling. The users can also easily and readily view the OR details through the system itself and to know which slot is available or is already taken. Making the processing organized and structured.

Furthermore, allowing the representatives from the Nursing Services and Anesthesiology to notify and to know conveniently the nurses and anesthesiologist involved in the surgical management team.

The system, being a mobile phone application, shall make OR scheduling a user-friendly and convenient to use in editing and updating contents of the system for the consultants and residents. It is more organized and efficient. They shall be able to access all the data with regards to the areas mentioned above at any time they need to do so providing only they have an internet connection.

E. Scope and Limitations

1. The use of the system is limited to the consultants, residents, representative from the nursing services department, representative from the anesthesiology department and an administrator of the Department of Otorhinolaryngology in the Philippine General Hospital only.
2. The Anesthesiology Department Representative is the one responsible for informing the respective anesthesiologist of the OR schedule details.
3. The Nursing Services Representative is the one responsible for informing the respective nurse of the OR schedule details.
4. Updating of today's OR schedule is not allowed.

5. Actual change in the OR schedule for the day is outside the system (overtime of an earlier OR, electrical power problems, etc.)
6. The system runs only on Java-enabled mobile phones to support the mobile application.
7. The system is restricted to mobile phones with an Android operating system.
8. Creating and Viewing of OR schedules is only possible within a month.
9. Generated type of OR schedule reports is be limited to .pdf file extension format.
10. Registration in the system is not allowed for the Department of Anesthesiology and Nursing Services Department Representatives but must install the system to view the OR schedules.
11. The Department of Otorhinolaryngology in the Philippine General Hospital can provide the mobile application upon the request of the users which will be directly installed into their respective mobile phones.

F. Assumptions

1. The system concerns only with the arrangement and organization of the schedules of patients due for operation.
2. The users of the system have access to an adequate internet connection for the facilitation of the process.
3. The system is not responsible for the process delay caused by slow internet connection.
4. The consultant or resident must include their name in the surgical team in the OR details if they are part of the surgical team.
5. The system has only one administrator.
6. The system is not responsible in the event that a consultant or a resident becomes defunct due to circumstances (e.g. transfer of user to another hospital institution, resignation, sudden or immediate disappearance, etc.)

II. Review of Related Literature

The management of healthcare deals with the “resources, devices, and methods required in optimizing the acquisition, storage, retrieval, and use of information in health and biomedicine”. It is the intersection of information technology and healthcare, information science, and computer science. This includes not only computers but also formal medical technologies, clinical guidelines, and communication and information systems [5]. Research efforts within the healthcare industry and the rapid development in information and communication systems over the past few decades have brought about important advances in the quality of medical services. One such innovation is the application of mobile technology in healthcare.

Mobile technology has significantly expanded and developed globally. In a remarkably short span of time, the mobile technology and the internet have become a part of daily life for most people in the emerging and developing world [6]. With the advent of smart phones along with 3G, the utilization of this technology is quickly transforming fields such as communications, entertainment, commerce, and healthcare [7].

Mobile devices and applications are used by health care professionals for many purposes, which is grouped into five broad categories: administration, communications and consulting, health record maintenance and access, medical education, and reference and information gathering. Information and time management is one of the most frequent uses for mobile devices for health care professionals [8].

A type of mobile phone known as the “smartphone” combines advanced mobile portable computation and communication and is also capable of running third-party software [9]. At present, the Android Operating System (OS) is one of the most widely used mobile Operating Systems in smartphones [10]. Developed by Google, it is an open platform for mobile development based on the Linux kernel [11]. As an open source operating system, the Android OS is the most suitable platform where developers can create new applications, such as those regarding scheduling and health care management.

An example of such an application is the “Offline Android Application-

Appointment Scheduler and Location Helper” created by Pooja (2014) using file operation. The user of the application can schedule his appointment and identify the path and direction of any nearby places as well. The application thus enables the user to efficiently manage his time in setting up appointments and in looking for directions to a location [12].

Another is the “Quick Organizer” developed by Baxi (2012), an application for managing and organizing daily activities. It has three main features: (1) a calendar which synchronizes with Google calendar and the default calendar application in the Android device, allowing the user to create events in all the user calendar accounts; (2) notes management for creating a new note, viewing all notes for a selected month, and searching notes for the user accounts in sync with the Evernote application; and (3) tasks management for creating daily to-do lists with deadlines per task where the user can view all the tasks created, with a clear distinction between the completed and pending tasks. Through these features, activity setting and organizing is made simple and easy [13].

A range of applications have also been made integrating scheduling in the health care setting. Choudhari et al. (2014) developed the “Doctors Appointment Scheduler” application where the patient can book an appointment with the doctor of his choice through various filters used in the application, and the doctor can come to know the number of patients he has to attend to within the day. This eliminates the need for the patient to call or personally go to the clinic in order to book an appointment and for the doctor or his secretary to manually schedule the appointments. The application simplifies and makes the process of scheduling an appointment with ones doctor convenient [14].

The “Health Portal” application was developed by Gavastar, Sumithra and Saranya (2013) wherein patients can access and set an appointment with any doctor of their choice and doctors can register and maintain their patient database. Linked with the application is a GPS connectivity which enables the patient to find doctors near their location which is particularly useful during health emergencies. In cases of direct visit, setting appointments with the doctor is made easier by filtering the list based on the needed specification or complaint. Another feature is that reminders are also added during the time of prescription recording. The application thus acts as a portal enabling patients to care for their health

amidst their busy schedules while making patient management more convenient for doctors as well [15].

The aim of hospitals, essentially, is to reduce costs while maximizing the level of patient satisfaction. Hence, there is a need for scheduling to provide medical services in the most optimum way possible, particularly for Operating Room (OR) procedures, since it is the largest cost and revenue center in a hospital [16]. Hospital Operating Room Scheduling involves the arrangement of operating rooms to a particular medical surgeon or surgical teams for a given period of time. Poor OR scheduling may lengthen waiting time which may worsen the disease of the patient [17]. Logistical difficulties are also commonly encountered in scheduling procedures in hospitals due to a wide range of variables such as operating hours, doctor availability, doctor preference, and room functionality [18]. Thus, the creation of an optimal surgical schedule becomes a difficult task because of the many changing factors that are needed to be considered.

There are two common processes for the scheduling of surgeries, known as block-booking and open-booking. In the block-booking system, surgical groups or individual surgeons are assigned times in a particular or in a periodic schedule, such as weekly or monthly. Surgeons book cases into their assigned block time during each period. On the other hand, in the open-booking system, cases are submitted by surgeons up until the scheduled day for the surgery which is subject to approval. All accepted cases are scheduled, which is limited to the hospitals total capacity to accommodate such. Open-booking is more common in destination medical centers where a patient can only become a candidate for surgery after the completion of a medical examination upon arrival. Because it is generally difficult for patients to return at a later date, medical centers often try to offer surgery on short notice [19].

For hospitals, the block booking system is more common. The scheduling of surgeries in a hospital are generally done with definite time periods. Operating Rooms are listed with their total available regular capacity per time period. A certain amount of time is required for each surgery to be done. This includes the time the OR is occupied with activities in relation to the particular surgery, such as the time required to prepare the tools and machines, the time necessary to perform the surgery, and the time needed to clean the OR after the surgical

procedure. Every surgery must be allocated to one of the ORs deemed as suitable for a particular patient [20].

The problems encountered for OR scheduling can generally be divided into three, namely (1) the Case Mix Problem (CMP), (2) the Master Surgery Scheduling Problem (MSSP), and (3) the Surgery Scheduling Problem. The CMP involves determining the time of ORs for each surgical procedure to maximize profit and minimize cost; the MSSP involves allocating surgical specialties to OR times to maximize resource utilization; while the SSP involves selecting and sequencing patients to be accommodated in each OR to minimize waiting time, OR overtime, and cancelled cases and to maximize the utilization of the OR as well [21].

The creation and implementation of a systematic OR scheduling system is an efficient solution to such problems. One example is the three-step approach using a hierarchical method. The first step is Sessions Target Planning, where the number of hours to be assigned to each surgical specialty for the proposed schedule is determined. The next step is Master Surgical Scheduling (MSS), where each surgical team are assigned a clearly defined OR block with time duration. The last step is Case Scheduling, which involves the preparation of a detailed OR schedule daily that includes the cases to be performed, the estimated duration of the procedure, the start and finish time per surgery, the surgery type, and the names of the hospital personnel conducting the procedure such as the nurses and the anesthesiologist [22].

Another is Online Scheduling, a model which “aims to minimize cancellations of pre-scheduled elective patients whilst also allowing for additional scheduling of emergency cases”. Here, tentative schedules are determined “in the offline environment prior to the day of surgery for the online environment”. Adjustments made in the offline schedule during the project implementation are diminished through an online scheduling model that operates in real-time [23]. The Operating Room Management Software (ORMS) is an efficient OR scheduling system that creating of a digital daily schedule for the OR that may be printed early on the day of the surgery. The software performs what is termed as the “conflict check”, where a notification is generated whenever a room is booked twice for the same slot. Once a procedure is scheduled, a corresponding “Physicians Preference Card” is generated which contains details regarding the procedure such as the

necessary instruments, supplies and equipment needed for a particular case [24].

The Operating Room, essentially, should be managed efficiently to maximize service to patients and to reduce costs. This can best be done through optimized scheduling. In scheduling systems, changes in the OR processes must be given a clear target and the results monitored and relayed to everyone involved. An organized system aids in the motivation of the OR personnel as well, assisted by the clear management and leadership of their superiors [25].

III. Theoretical Framework

1. Otorhinolaryngology

Otorhinolaryngology or also known as Otolaryngology is an area in the medical field that involves surgical management and treatment of patients with diseases and disorders of the ear, nose, throat (ENT), and related structures of the head and neck [26]. It also include “diagnosing and managing diseases of the sinuses, larynx (voice box), oral cavity, and upper pharynx (mouth and throat), as well as structures of the neck and face [26].” “Otolaryngologists diagnose, treat, and manage specialty-specific disorders as well as many primary care problems in both children and adults [26].”

2. UP-PGH Department of Otorhinolaryngology

“The Department of Otorhinolaryngology, University of the Philippines Philippine General Hospital is the bastion of ear, nose, and throat (ENT) care.” As of today, the department is considered the largest tertiary Otorhinolaryngology service in the country having 26 residents and 26 consultants. Encompassing amenities and facilities to Out-Patient clinics and Emergency room, In and Out-patient Operating room for Surgery, Audio logic Laboratories and Prosthetic Fabrications Unit. The staff members are nationally and internationally recognized for their academic and clinical activities, publications and innovations, and community service [27].

According to Dr. Philip B. Fullante, a consultant in the ORL department in the Philippine General Hospital, the department has four operating rooms; three charity OR and one pay OR. The first year residents are mostly the ones who manage the records, admissions and clinical logs of the department. A head resident is assigned for each operating room.

3. Pay/Private OR and Charity OR

According to Dr. Philip B. Fullante, a consultant in the ORL department in the Philippine General Hospital, the pay/private it is an operation room in the department mainly for private patients only. It has different num-

ber of rooms and timeslots depending on the day. It only operates on Monday, Wednesday and Friday. For Monday, there are only two rooms having three timeslots each. For Wednesday and Friday, it has three rooms but only one timeslot each. Here, the consultants are the ones who operate in the OR. The charity OR handles the operation outside the range of private patients. It operates from Monday to Friday. It has three rooms for each day and three timeslots each. Here, the residents are the ones who operate in the OR.

4. Operating Room

Operating room is the entire area in which surgical operations are performed and materials are prepared and stored for surgery. “Operating room” and “OR” are also used to indicate an individual OR in which surgery is performed [28]. “Operating room (OR) is a cost-intensive environment, and it should be managed efficiently [25].”

5. Operating Room Scheduling

Operating room scheduling involves the assignment of surgeries of different types of patients and surgeons in the available ORs [22]. It is also the science of how to run an Operating Room Suite. “Operating room management focuses on maximizing operational efficiency at the facility, i.e. to maximize the number of surgical cases that can be done on a given day while minimizing the required resources and related costs [29].”

6. Smartphones

It is defined as “a fully-fledged computers that you can fit in your pocket. It can run programs, applications, and games, access the internet, send email and much more. Nearly all smartphones now use touchscreen controls. Instead of having hardware buttons like before, one side of the phone is taken up mostly by a touchscreen that you control using taps and gestures [30].” Smartphones have advanced computing capabilities in comparison with other typical mobile phones. It runs a complete type of operating system.

7. Mobile Application

Mobile application is “an add-on software for handheld devices, such as smartphones and personal digital assistants (PDA).” Among the most popular and are most commonly used applications are games, social networking, maps, news, business, weather and travel information. “All of these leverage at least one of the devices technical features: communications interfaces (Wi-Fi, WiBro/mobile WiMAX, GSM/EDGE, W-CDMA/UMTS/HSPA and Bluetooth), audio and video processors, camera, sensors or GPS module [31].” “From the technical point of view the different mobile applications are run able in different managed platforms like iPhone, BlackBerry, Android, Symbian, windows; also some virtual machine such as Java/J2ME, BREW, Flash Light, Silverlight [32].” Mobile application is a new and fast developing Segment of the global Information and Communication Technology. Mobile application is easy, user friendly, inexpensive, downloadable and run able in most of the mobile phone including inexpensive and entry level phone [32].

8. Mobile Healthcare

It is defined as the use of mobile phones to improve the quality of care and enhance efficiency of service delivery and is a sub-segment of the broader field of electronic health (eHealth). The world health organization has defined mobile healthcare as the “provision of health services and information via mobile technologies such as mobile phones and Personal Digital Assistants (PDAs). Creates cost efficiencies and improving the capacity of health systems to provide quality healthcare [33].” It includes the usage and capitalization on a mobile phones core effectiveness and efficacy of different complex functionalities and technological aspects [33]. “Mobile healthcare is the use of mobile networks and devices in supporting e-care. Stresses leveraging health-focused applications on general-purpose tools such as smartphones [34].” “It is a new frontier in health innovation. This area covers applications, devices and communications networks that allow clinicians and consultants to give care anywhere at any time [34].”

9. Android Operating System

Android is defined as “a mobile operating system, initially developed by Android Inc. and is based on a modified Linux 2.6 kernel [11].” Android operating system is primarily designed for smartphones and tablets. Since Android is an open source it has become the fastest growing mobile operating system making it one of the most widely used mobile Operating System these days [10]. It is an open-source and is the operating system to be used in this project.

10. iText

It is “an open source library for creating and manipulating PDF files in Java. It was written by Bruno Lowagie, Paulo Soares, and others. Ohloh reports more than 10,000 commits by 26 different contributors since 2001 representing more than 1 million lines of code. iText has a well established, mature codebase maintained by a large development team with stable Y-O-Y commits. Using the COCOMO model, it took an estimated 250 man-years of effort starting with its first commit in November, 2000.” [36].

In this system, the application used iText to generate the OR schedule reports in pdf format.

11. Android SDK and IDE

SDK stands for software development kit. “It includes a comprehensive set of development tools. These include a debugger, libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. Currently supported development platforms include computers running Linux (any modern desktop Linux distribution), Mac OS X10.5.8 or later, and Windows XP or later [35].” IDE stands for integrated development environment. “For the moment one can also develop Android software on Android itself by using this IDE. The officially supported integrated development environment (IDE) is Android Studio using the Android Development Tools (ADT) Plugin, though IntelliJ IDEA IDE (all editions) fully supports Android development out of the box, and NetBeans IDE also supports Android development via a plugin [35].”

IV. Design and Implementation

ORL OR Scheduler System has 5 users:

1. Consultants
2. Residents
3. Administrator
4. Department of Anesthesiology Representative
5. Nursing Services Department Representative

The consultants are the target users of the system regarding the private/pay OR. They input the necessary OR details in the system. They can also send and receive messages to the residents within the system

The Resident is be the one in charge for the schedules in the charity OR. They input the necessary details of the schedule in the system. The resident is also in charge of finalizing of both the OR schedules and generating the OR schedule reports. Notifying the consultants is possible through sending a message within the application if there will be any concerns regarding the OR schedules.

The administrator is the one in charge for the user account management within the system. Maintaining and checking the availability of the system is also the responsibility of the administrator.

Being part of the operation team, it is also very important for the anesthesiologists and nurses to know the schedule as to avoid conflicts in their designated timeslots and to know the schedules beforehand. A representative from each of the Department has access on viewing only the OR schedules and the details.

A. Context Diagram

This segment, contains the discussion of the overview of the system. We take a look at the components of the system regarding the usage. Figure 1 shows the context diagram for the ORL OR Scheduler System.

Figure 1 has six sections. The consultant section, resident section, department of anesthesiology representative section, nursing services department representative section, administrator section and the database section. These sections

defines the users and the constituents of the system. It presents and demonstrates the overview of the flow of the system.

Figure 1 (a) shows the consultant section. Here, the consultants input the contents of the schedule. This is the pay/private OR perspective. Take note that only the consultants can make schedule in the pay/private OR and the number of consultants allowed for each timeslot is only one. A consultant may take the other slots in each operating room as long as it does not constitute to the same timeslot.

Figure 1 (b) shows the resident section. The residents input the contents of the schedule. This is the charity OR perspective. There is only one resident in charge and assigned each day to create the OR schedule.

Figure 1 (c) shows the Department of Anesthesiology Representative section. The representative can only view the OR schedule details to inform the respective and assigned anesthesiologist in the OR.

Figure 1 (d) shows the Nursing Services Department representative section. The representative can only view the OR schedule details to inform the respective and assigned nurse in the OR.

Figure 1 (e) shows the administrator section. In this section, the adminis-

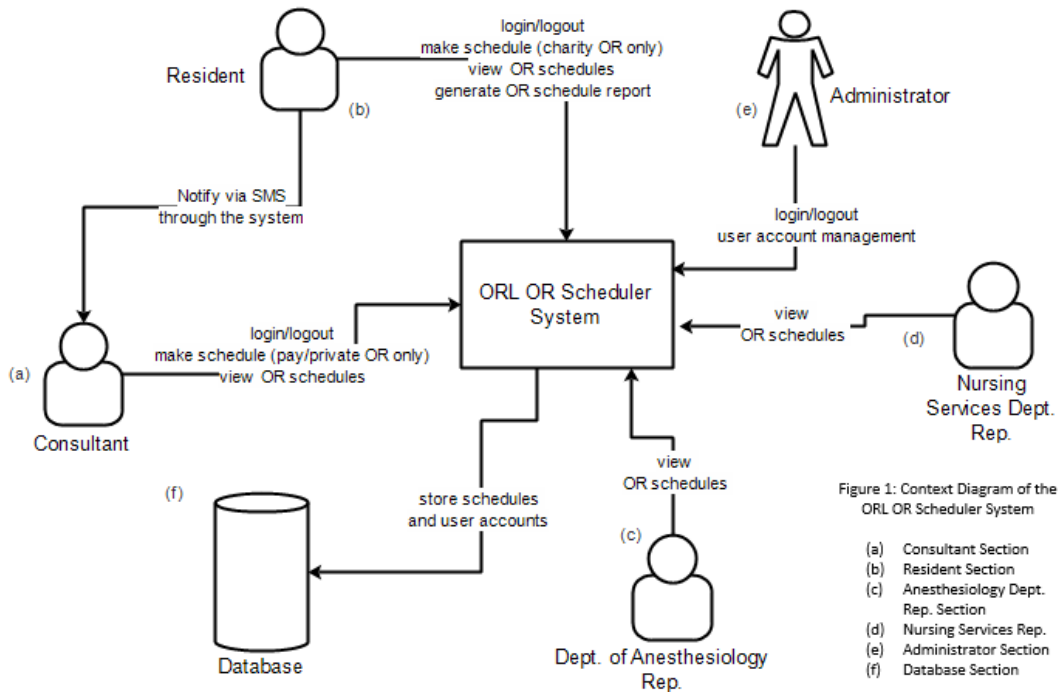


Figure 1: Context Diagram

trator monitors the system and is in charge of the user account management.

Figure 1 (f) shows the database section. In this section, the information inputted are stored in a database for record purposes.

B. Use-Case Diagram

1. Use Case for the consultants

The consultants register first in the system if they dont have an account yet. After which, they can login their account with their username and password to make a schedule. The consultants input the necessary OR details in the field. They can add, delete and update the details of the schedule. The said timeslot will now be closed after making the schedule.

Disabling any other consultants to update the timeslot. Only one specific consultant can be appointed for each slot for every room. However, the said consultant cannot schedule himself on the other operating room with the same timeslot. Only the consultant who made the schedule can update the timeslot

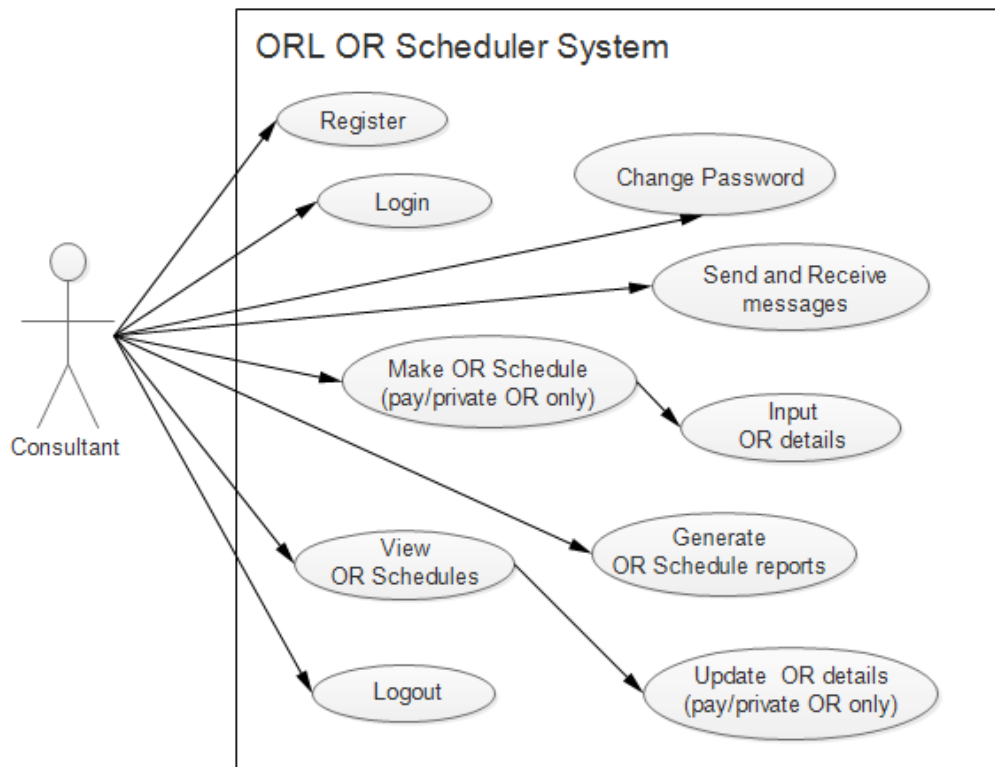


Figure 2: Use-case for Consultant

or enable it again. Should there be any changes or modifications that need to be done, they can be messaged by the residents within the system. They can also send messages to other user accounts. Generating the OR schedule reports is also possible within their panel. Viewing is allowed for both the OR schedules but updating the charity OR details are not allowed. Figure 2 shows the use-case for the consultants.

2. Use Case for the Residents

Another user that can make a schedule are the residents. The residents register first in the system if they dont have an account yet. Logging in with their username and password is required before creation of OR schedule. They can only make schedules for the charity OR. They input necessary OR details in the field. The resident can add, delete and update the details of the OR schedule. Only one resident is assigned each day for all the rooms in the charity. The resident is also in charge in finalizing the OR schedules including in the pay/private OR. Then, he can generate the OR schedule report or

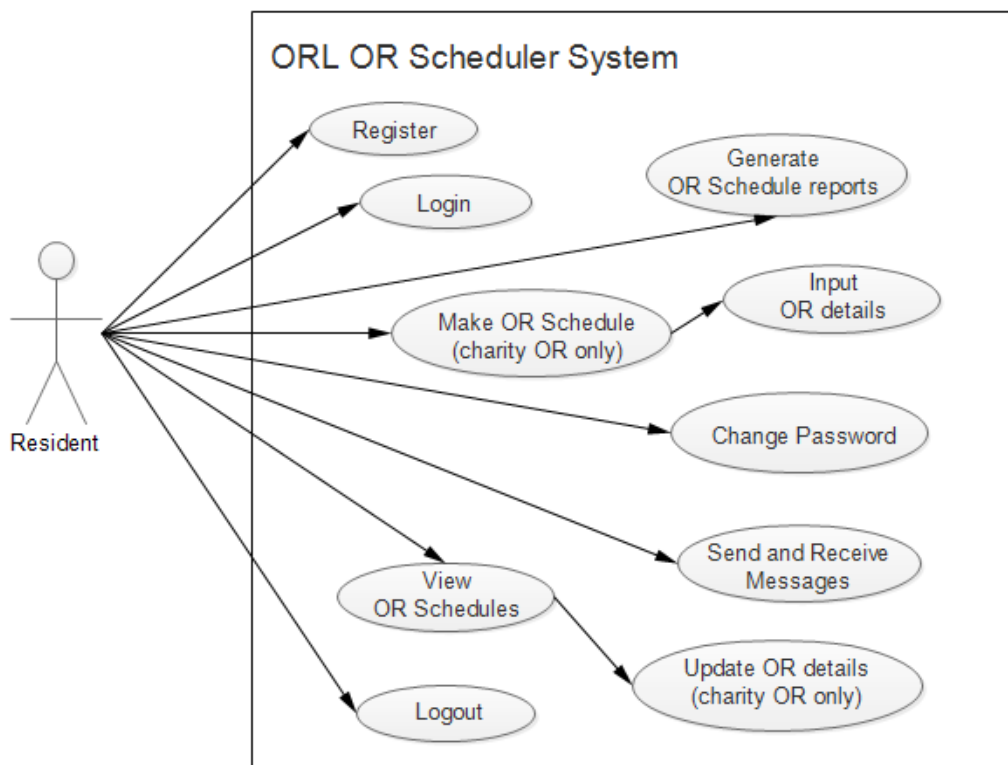


Figure 3: Use-case for Residents

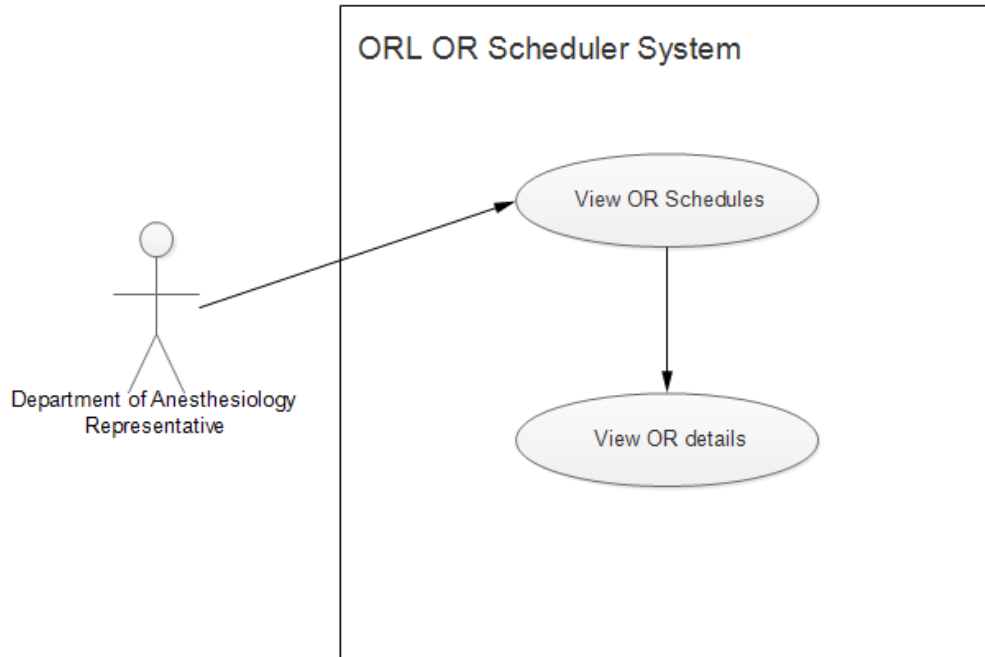


Figure 4: Use-case for the Department of Anesthesiology Representative

the template and is now ready for printing. Should there be any changes or modifications that need to be done, the resident can notify the consultants by sending messages within the system. They can also receive messages from the other user accounts. Viewing is allowed for both the OR schedules but updating the pay/private OR details are not allowed. Figure 3 shows the use-case for the residents.

3. Use Case for the Department of Anesthesiology Representative

A representative from the Department of Anesthesiology can only view the OR schedules to notify and inform the respective anesthesiologist assigned in the OR in their Department. Since updating is not allowed, user account is not necessary. Figure 4 shows the use-case for the Department of Anesthesiology Representative.

4. Use Case for the Nursing Services Department Representative

A representative from the Nursing Services Department can only view the OR schedules to notify and inform the respective nurses assigned in the

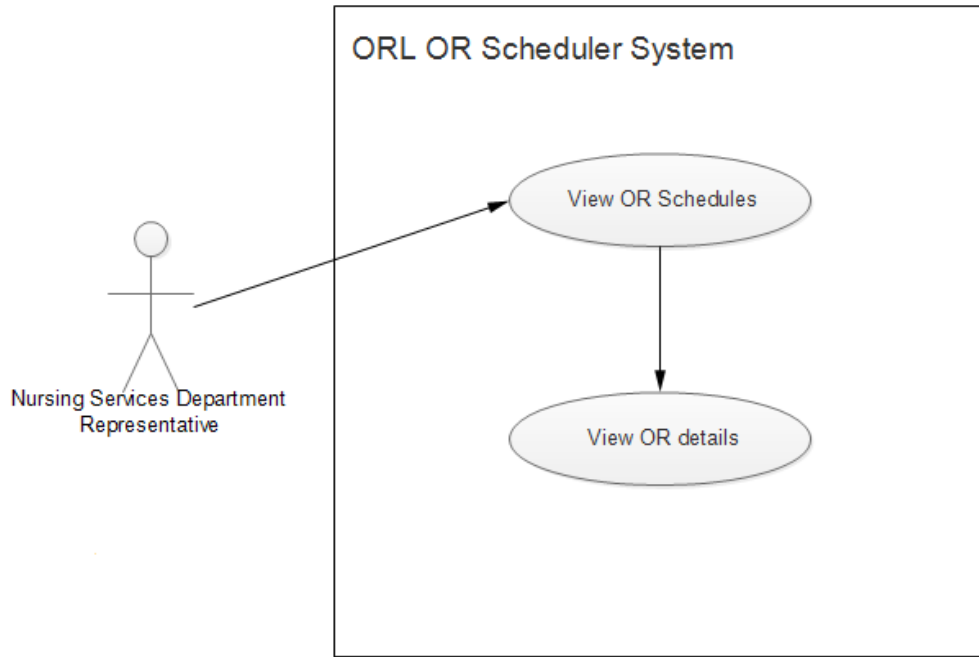


Figure 5: Use-case for the Nursing Services Department Representative

OR in their Department. Since updating is not allowed, user account is not necessary. Figure 4 shows the use-case for the Nursing Services Department Representative.

5. Use Case for the Administrator

The administrator handles the user account management within the system. With their username and password, they can login on the system to perform the task. The administrator can view and update user accounts. The administrator can also send and receive messages within the users in the system. Figure 6 shows the use-case for the administrator.

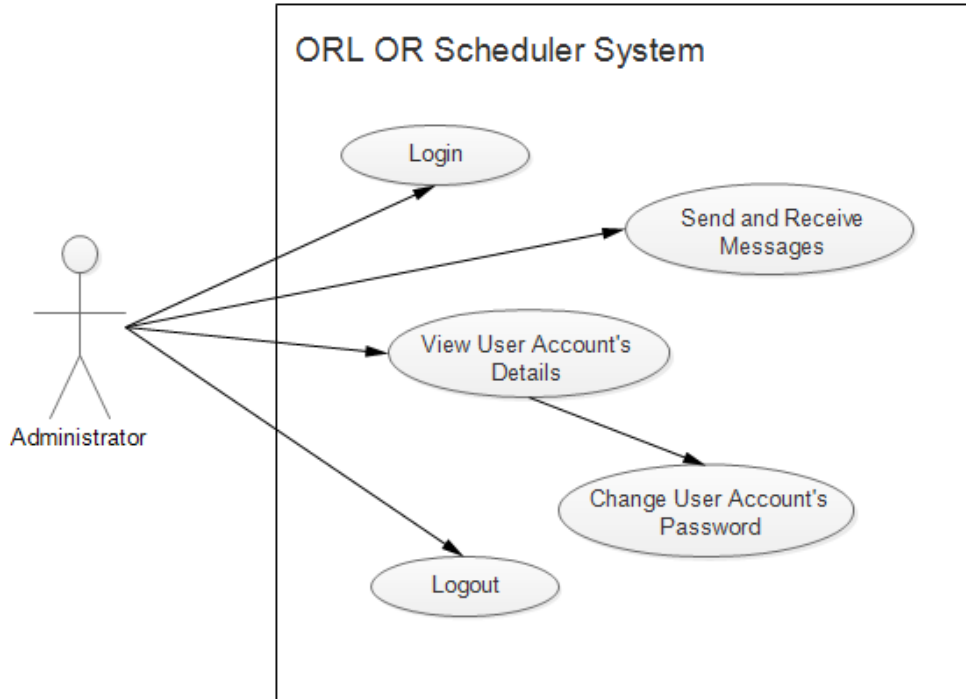


Figure 6: Use-case for the Administrator

C. Entity Relationship Diagram (ERD)

Data inputs are stored in a relational database management system. The following diagram shows the entity-relationship model of the system.

Figure 7 shows the ERD for the ORL OR Scheduler system. It presents the association of the objects and units within the system. As mentioned earlier, the system has only two users who can register and login their accounts: the consultants, the residents and the administrator.

The "user_type" determines the type of user of the system. The values may only be consultant, resident or administrator. Consultants and Residents can input many OR schedules or none and an OR schedule must be accounted to only one user.

The "sched_type" determines the type of the OR schedule. It has only two types. This can either be pay/private or charity OR.

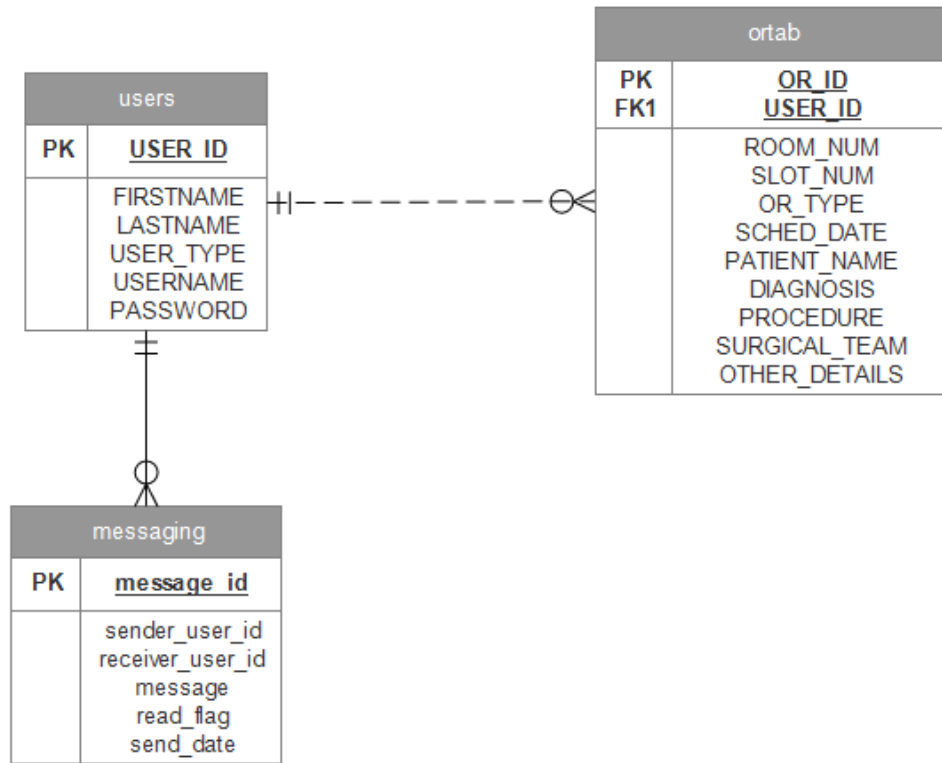


Figure 7: ERD for the ORL OR Scheduler System

D. Data Dictionary

This section describes the data in the ERD found on Figure 7.

Data Field	Data Type	Description
USER_ID (PK)	int(11)	ID of the user
FIRSTNAME	varchar(100)	First name of the user
LASTNAME	varchar(100)	Last name of the user
USER_TYPE	varchar(20)	Type of the user
USERNAME	varchar(100)	Username of the user
PASSWORD	varchar(100)	Password of the user

Table 1: users table

The users table contains the information of the users.

Data Field	Data Type	Description
OR.ID (PK)	int(11)	ID of the OR schedule
ROOM_NUM	int(11)	Room number of the OR
SLOT_NUM	varchar(50)	Slot number of a particular slot in the OR
OR_TYPE	varchar(100)	Type of the OR
SCHED_DATE	date	Date of the OR schedule
PATIENT_NAME	varchar(200)	Full name of the patient
DIAGNOSIS	varchar(1000)	Diagnosis of the patient
PROCEDURE	varchar(200)	Procedure or Treatment for the patient
SURGICAL_TEAM	varchar(600)	Surgical Team for the operation
OTHER_DETAILS	varchar(1000)	Other details or remarks
USER_ID (FK1)	int(11)	ID of the user who made the OR schedule

Table 2: ortab table

The ortab table contains the information about the OR schedule.

Data Field	Data Type	Description
message_id (PK)	int(11)	ID of the message
sender_user_id	int(11)	ID of the sender
receiver_user_id	int(11)	ID of the receiver
message	varchar(2000)	The message itself
read_flag	int(1)	Flag to determine if the message is already read or not
send_date	timestamp	Date and time of the message sent and received

Table 3: messaging table

The messaging table contains the information about the messaging feature.

V. Architecture

A. System Architecture

The mobile application is developed using Java programming language in Android Studio and the necessary Android SDK Tools. It supports smartphones with Android operating system having platforms such as Ice Cream Sandwich, Jelly Bean, KitKat and Lollipop.

B. Technical Architecture

The minimum system requirements for the application are the following:

1. 256MB of RAM
2. 1.00 GHz processor
3. 100 MB free space
4. Wi-Fi 802.11 connectivity
5. Ice Cream Sandwich or API 15 version 4.1

The system requirements for the server are the following:

1. 2.00 GHz Intel Core 2 Duo CPU
2. 2.00 GB RAM
3. More than 1 GB hard drive free space
4. Internet connection
5. MySQL Database
6. PHP Server (Apache 5.4.7)

VI. Results

A. Installation

This section contains the guide and process for installation the application.

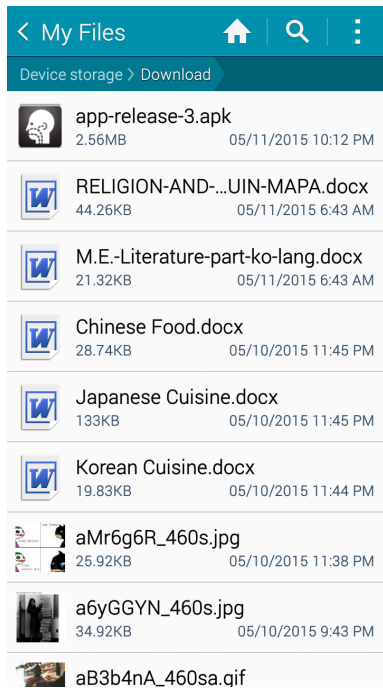


Figure 8: Installation part 1

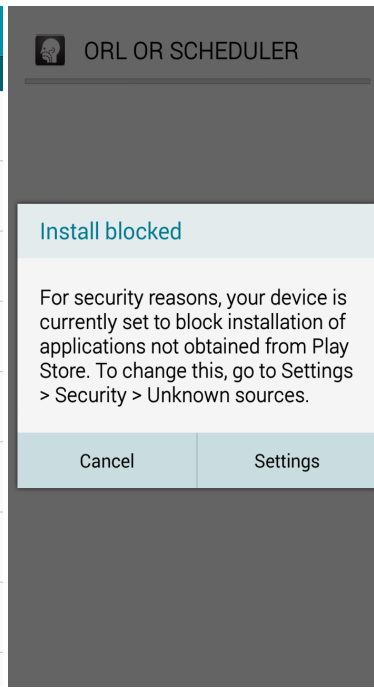


Figure 9: Installation part 2

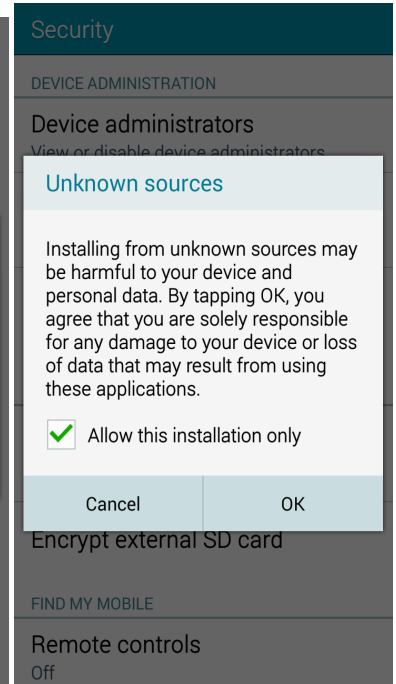


Figure 10: Installation part 3

The installation file or the APK (Android Application Package) will be sent to the android device via bluetooth from the administrator or through USB cable directly from the computer. The APK file will now be sent and stored on the device as shown in Figure 8. The icon of the application is also shown in the figure. Upon receiving the APK, open the file and an alert will be pop out as shown in Figure 9. This is just a reminder about security purposes to block other applications outside the Play Store. The ORL OR Scheduler system is not uploaded and out in the Google Play Store. The only way to install the application is by going to settings and to allow installation from unknown sources as shown in Figure 10.

The APK is now ready to install after selecting OK as shown in Figure 11. The application needs an internet connection to fully use the features. Figure

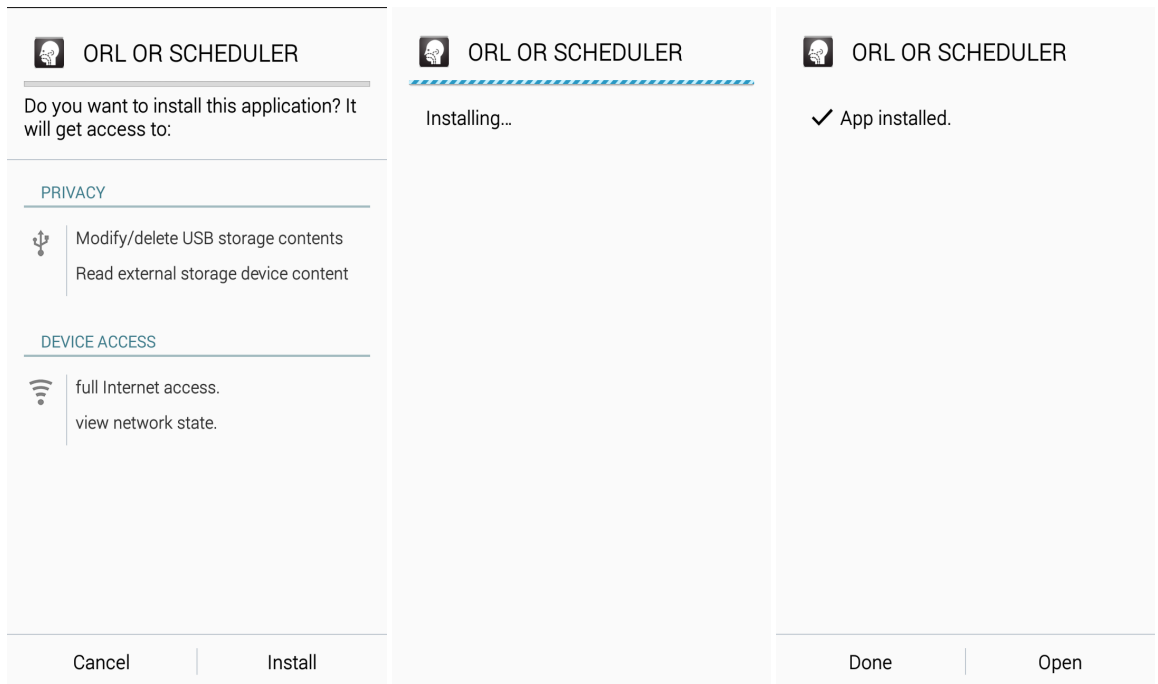


Figure 11: Installation part 4

Figure 12: Installation part 5

Figure 13: Installation part 6

12 shows the APK being installed. It will only take a while depending on the processor or speed of the android device. Figure 13 shows a succesful installation and is now ready for use.

B. ORL OR Scheduler System

Figure 14 shows the home page of the ORL OR Scheduler system. This view contains the information about the Department of Otorhinolaryngology in UP-PGH. Figure 15 is a continuation view for home page. The proceed button will direct the user to the login/view page.

Figure 16 shows the login/view page. The register button directs the user to the registration page shown in Figure 17. Below the login part, there is the view operating room schedule form wherein you can choose to view either the pay/private or the charity OR. This part is for the nursing representative and anesthesiologist representative. Upon clicking the view button, they will be brought to the view schedules only page. Figure 17 shows the registration page. Figure 18 is just a continuation of the registration page.

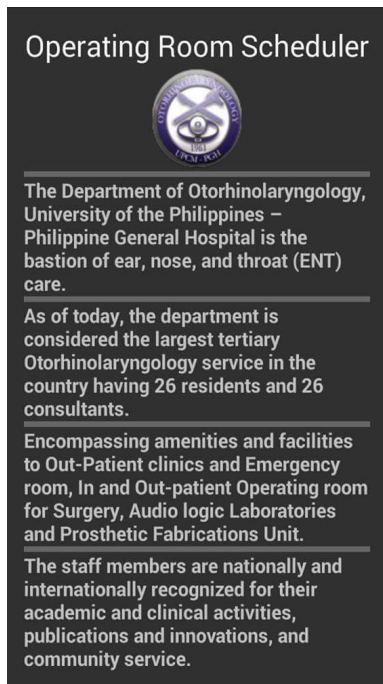


Figure 14: Home Page 1

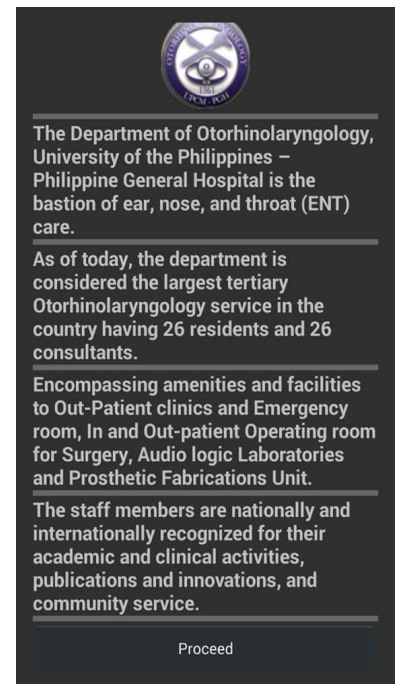


Figure 15: Home Page 2

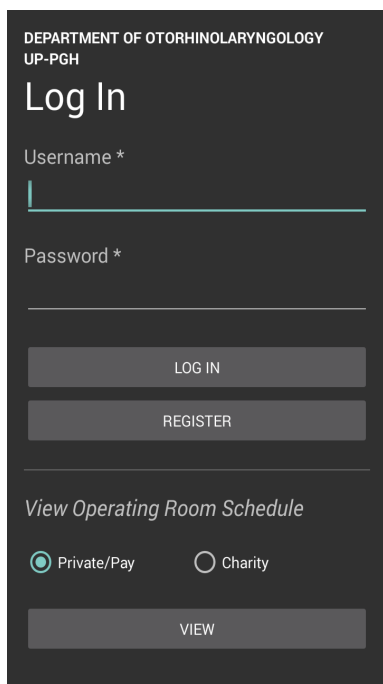


Figure 16: Login Page

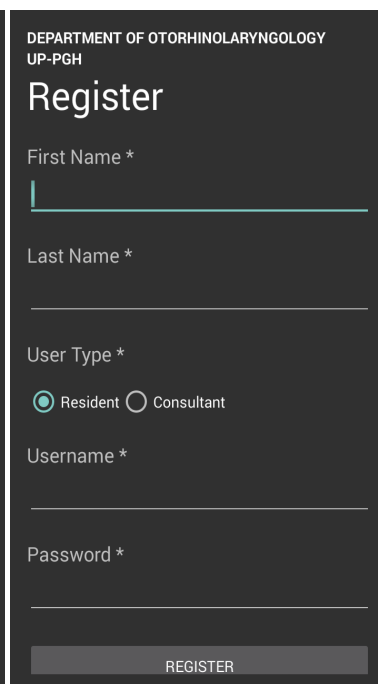


Figure 17: Register Page

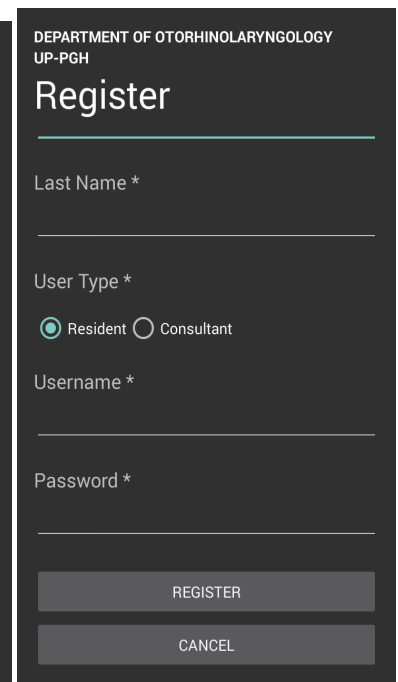


Figure 18: Register Page cont.

We create a resident account as shown in Figure fig:regres and a consultant account as shown in Figure fig:regcon with following the credentials. Upon clicking the register button, an alert will come out informing the user for a successful registration.

DEPARTMENT OF OTORHINOLARYNGOLOGY
UP-PGH

Register

First Name *
Michael

Last Name *
Johnson

User Type *
 Resident Consultant

Username *
mjohnson

Password *
password1

REGISTER

Figure 19: Register Resident

DEPARTMENT OF OTORHINOLARYNGOLOGY
UP-PGH

Register

First Name *
John

Last Name *
Smith

User Type *
 Resident Consultant

Username *
jsmith

Password *
password

REGISTER

Figure 20: Register Consultant

DEPARTMENT OF OTORHINOLARYNGOLOGY
UP-PGH

Register

First Name *
John

Alert
User Successfully CREATED!

OK

Username *
jsmith

Password *
password

REGISTER

Figure 21: Register Successful

DEPARTMENT OF OTORHINOLARYNGOLOGY
UP-PGH

Register

First Name *
|

Last Name *
|

User Type *
 Resident Consultant

Username *
|

Password *
|

Please enter all required fields (*).

REGISTER

Figure 22: Register Alert

DEPARTMENT OF OTORHINOLARYNGOLOGY
UP-PGH

Register

First Name *
Michael

Alert
Username is already in use!

OK

Username *
jsmith

Password *
password

REGISTER

Figure 23: Register Alert 2

Figure 22 shows an alert if the required field (*) is empty and will not proceed. Figure 23 shows an alert if the user tries to enter a username which is already taken.

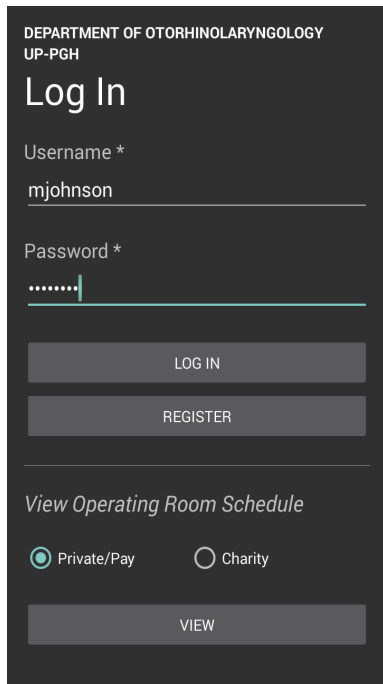


Figure 24: Login Resident

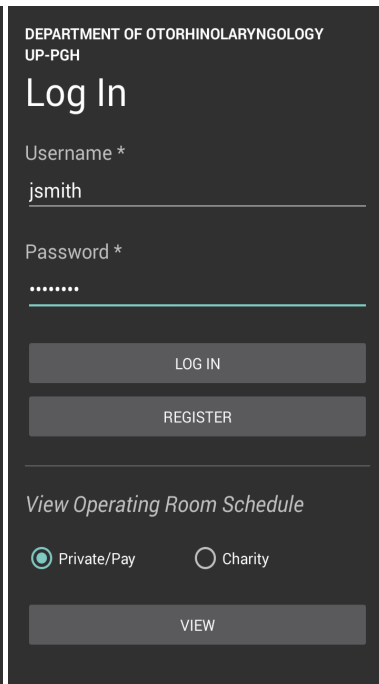


Figure 25: Login Consultant

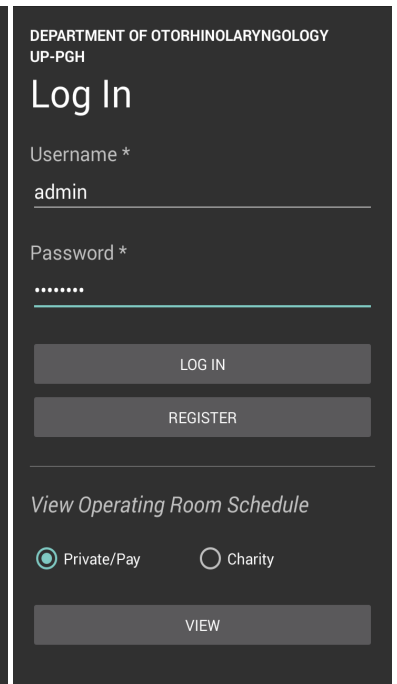


Figure 26: Login Admin

Figure 24, Figure 25 and 26 shows the resident, the consultant and the administrator respectively ready for login. The administrator account is manually inserted in the database. The system has only one administrator.

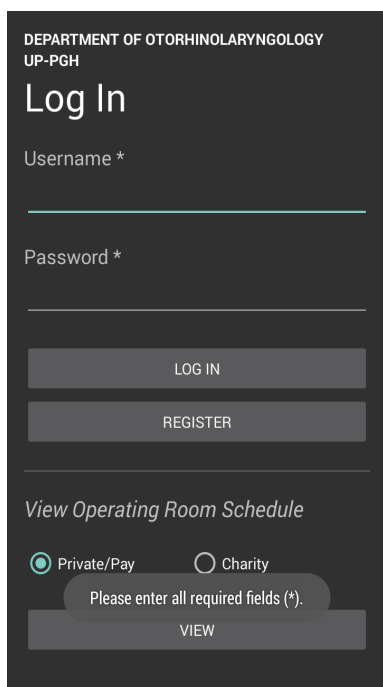


Figure 27: Login Alert

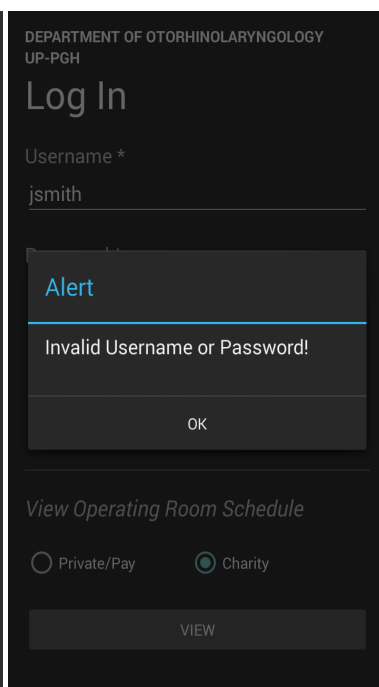


Figure 28: Login Alert 2

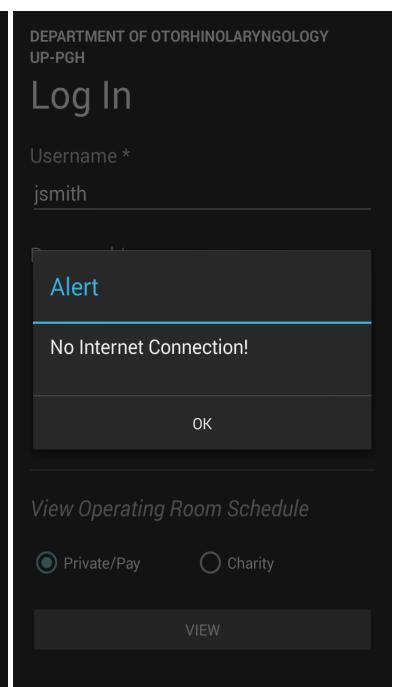


Figure 29: Login Alert 3

Figure 27 shows an alert if the fields are empty and Figure 28 shows an alert if the username or password entered is invalid. Figure 29 shows an alert if there is no internet connection.

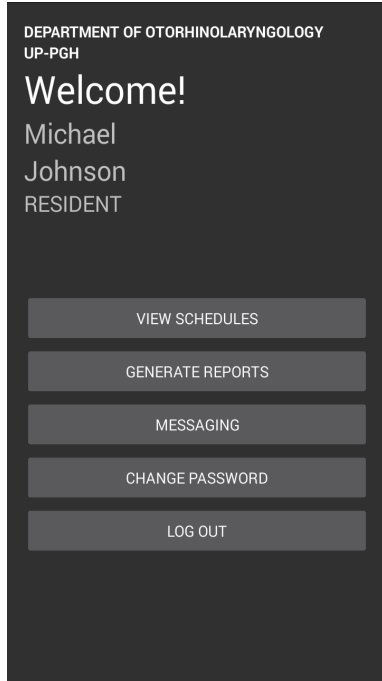


Figure 30: Resident Page

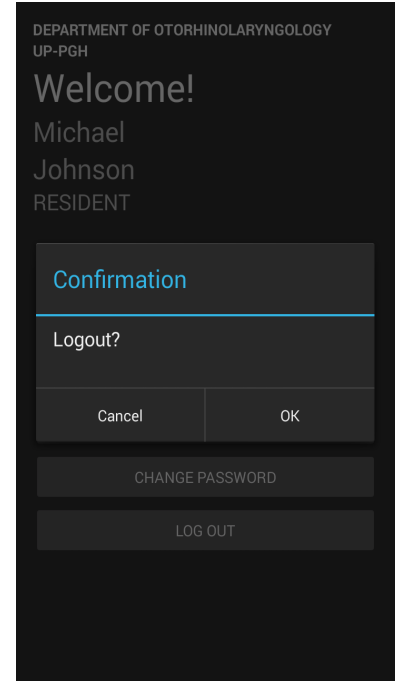


Figure 31: Resident Page 3

Figure 30 shows the home page or the dashboard for the residents. The dashboard or the user panel is the same as for the consultants also. Figure 31 shows a confirmation before logging out. This goes to all the users before logging out or by pressing the back button from the device.

The scheme of this UI for creating OR schedules is that the slots aligned vertically are slots 1, the next is slot 2 and the third is for slots 3. For an instance; Room 1 - slot 1, Room 2 - slot 1, Room 3 - slot 1 and so on. This goes for both the private/pay and charity operating rooms.

Figure 32 shows the view schedule page for the day. In this case, Monday is the day today. Updating OR schedules on the said day is not allowed. Only viewing is allowed. This page has no schedules made. Hence the slots are empty. In Figure 33, this time let's say the day today is Friday. So this is the page if either one of the slots has schedules. This page is the same as for the nurse and anes-

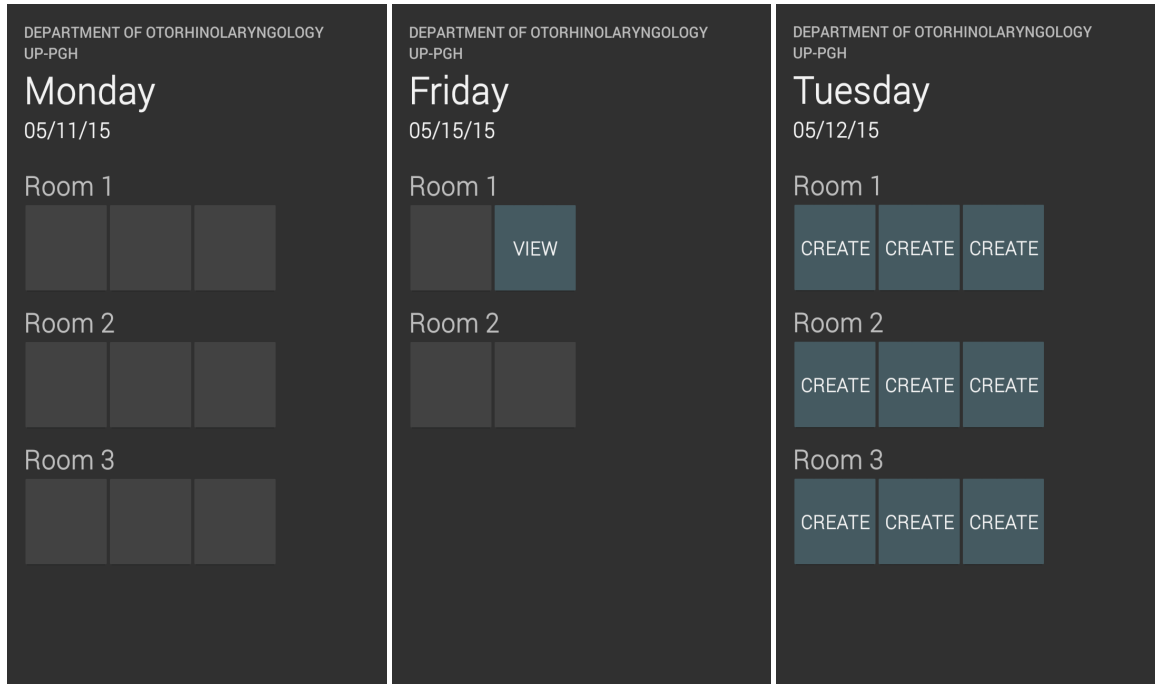


Figure 32: View OR Schedule Page

Figure 33: View OR Schedule Page 2

Figure 34: View OR Schedule Page 3

thesiologist representative outside the user panel for viewing the OR schedules only. Other slots are empty or disabled. Figure 34 shows the page for Tuesday. This interface is the same as for Wednesday and Thursday. Unlike Fridays, it has only 2 rooms. Here you can see the slots are not yet taken. Residents can create schedules easily just by clicking either one of the slots. The Department only assigns one resident to create the schedule per day.

Figure 35 shows the making of an OR schedule. The surgical team is typed in that format wherein: surgeon/consultant/anesthesiologist. Upon completing the form, an alert will pop out to inform the user for a successful creation as shown in Figure 36. Figure 37 shows the view schedule page for Friday. We created the schedule on room 1 and slot 2. Only the resident who created on that particular slot can update the OR schedule.

The Figures 38, 39, 40 shows the view schedule pages for Monday, Wednesday and Friday respectively. These are the days for the private OR's. As mentioned earlier, private or pay OR's operates only during Mondays, Wednesdays and Fridays. Monday has two rooms and three timeslots each. Wednesday has only one room but also has three timeslots. Friday has three rooms but with only

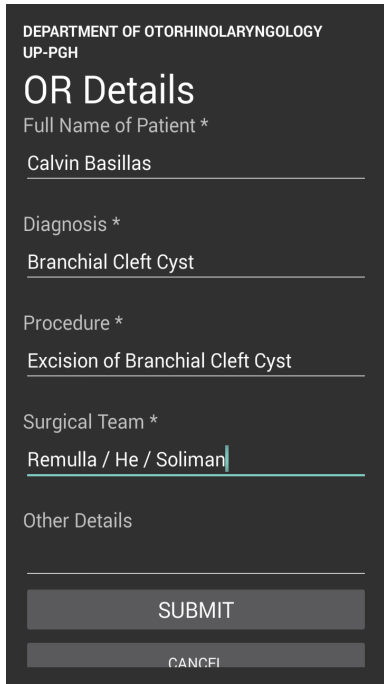


Figure 35: Create OR Schedule Page

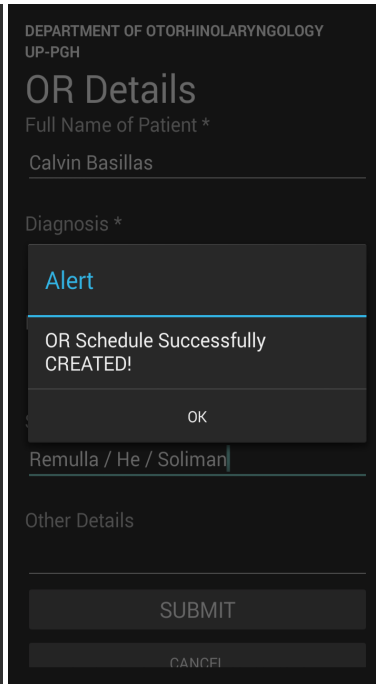


Figure 36: Create OR Schedule Page 2

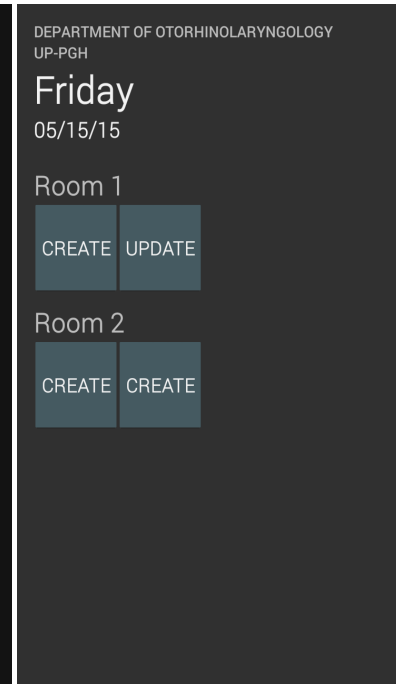


Figure 37: View OR Schedule Page 4

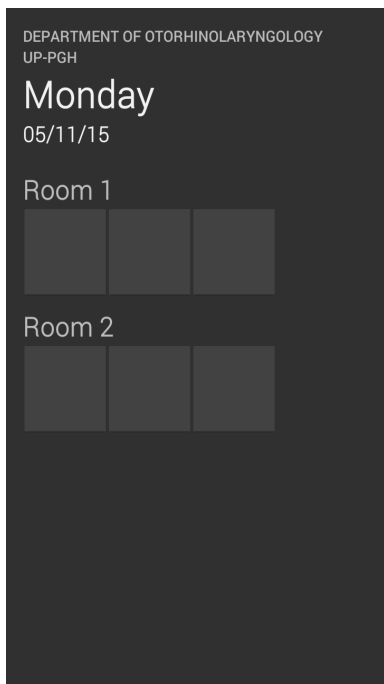


Figure 38: View OR Schedule Page for Private

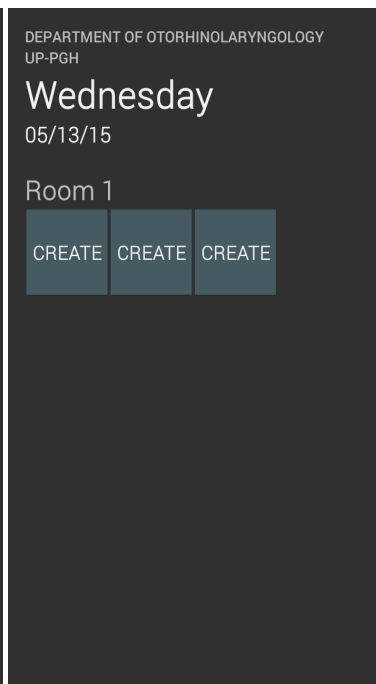


Figure 39: View OR Schedule Page for Private 2

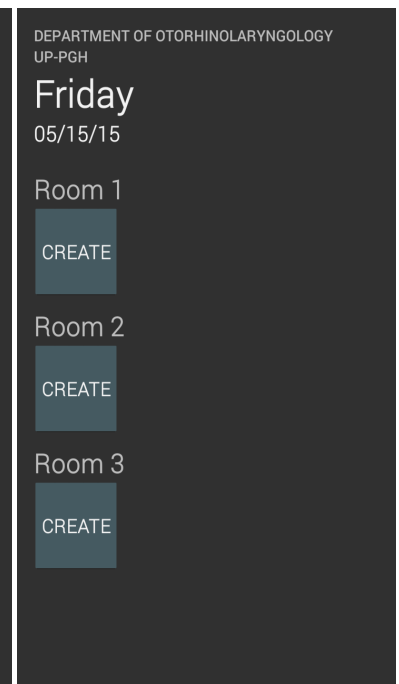


Figure 40: View OR Schedule Page for Private 3

one timeslot each. Monday OR slots have been disabled because as mentioned above just like in Figure 32, in that case, the day for today is Monday.

Figure 41 shows another creation of OR schedule. Here it shows an alert

DEPARTMENT OF OTORHINOLARYNGOLOGY
UP-PGH

OR Details

Full Name of Patient *

Paquito Castillo

Diagnosis *

Obstructive Sleep Apnea

Procedure *

Surgical Team *

Carrillo / Zantua / Nonato

Other Details

Please enter all required fields (*).
SUBMIT

CANCEL

DEPARTMENT OF OTORHINOLARYNGOLOGY
UP-PGH

OR Details

Full Name of Patient *

Paquito Castillo

Diagnosis *

Obstructive Sleep Apnea

Procedure *

Tracheostomy

Surgical Team *

Carrillo / Zantua / Nonato

Other Details

SUBMIT

CANCEL

DEPARTMENT OF OTORHINOLARYNGOLOGY
UP-PGH

OR Details

Full Name of Patient *

Paquito Castillo

Diagnosis *

Obstructive Sleep Apnea

Procedure *

Tracheostomy

Surgical Team *

Carrillo / Zantua / Nonato

Other Details

Alert

OR Schedule Successfully
CREATED!

OK

SUBMIT

CANCEL

Figure 41: Create OR Schedule Page 3

Figure 42: Create OR Schedule Page 4

Figure 43: Create OR Schedule Page 5

that the required fields must not be empty. In Figure 42 we added the detail in the Procedure. Figure 43 shows a successfully created OR schedule.

DEPARTMENT OF OTORHINOLARYNGOLOGY
UP-PGH

UPDATE

Full Name of Patient *

Paquito Castillo

Diagnosis *

Obstructive Sleep Apnea

Procedure *

Tracheostomy

Surgical Team *

Carrillo / Zantua / Nonato

Other Details

created by: MICHAEL JOHNSON

UPDATE

DELETE

CANCEL

DEPARTMENT OF OTORHINOLARYNGOLOGY
UP-PGH

UPDATE

Obstructive Sleep Apnea

Procedure *

Tracheostomy

Surgical Team *

Carrillo / Zantua / Nonato

Other Details

created by: MICHAEL JOHNSON

UPDATE

DELETE

CANCEL

DEPARTMENT OF OTORHINOLARYNGOLOGY
UP-PGH

UPDATE

Paquito Castillo

Diagnosis *

Obstructive Sleep Apnea

Procedure *

Tracheostomy

Surgical Team *

Carrillo / Zantua / Nonato

Other Details

none

created by: MICHAEL JOHNSON

UPDATE

Figure 44: Update OR Schedule Page 2

Figure 45: Update OR Schedule Page 3

Figure 46: Update OR Schedule Page 4

Figure 44 shows the view for updating OR schedule. The page also shows

who created the OR schedule. In this case it is created by a resident named Michael Johnson. Figure 45 shows the continuation page of the Update OR schedule. Two buttons were added; update and delete. Figure 46 shows a successfully updated OR schedule. In this case, we added 'none' on the other details field.

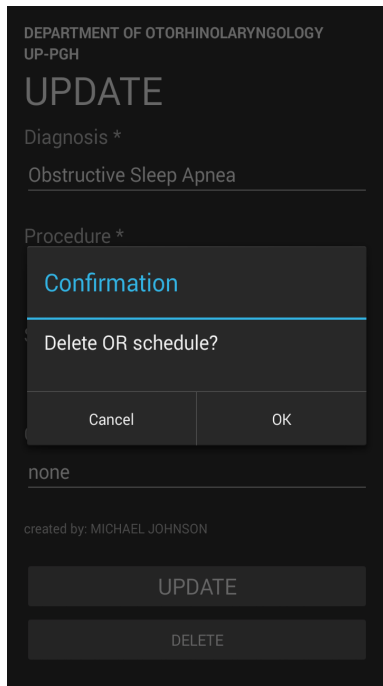


Figure 47: Delete OR Schedule Page

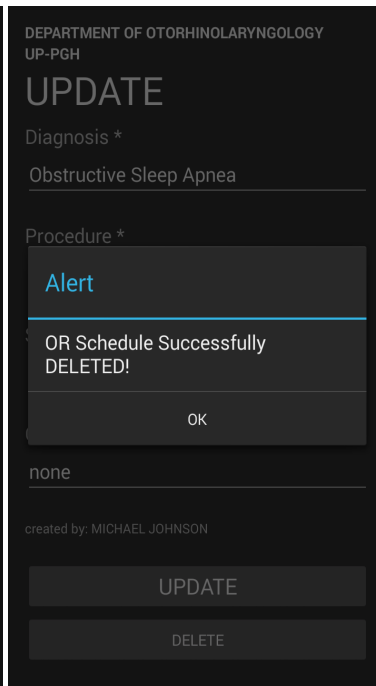


Figure 48: Delete OR Schedule Page 2

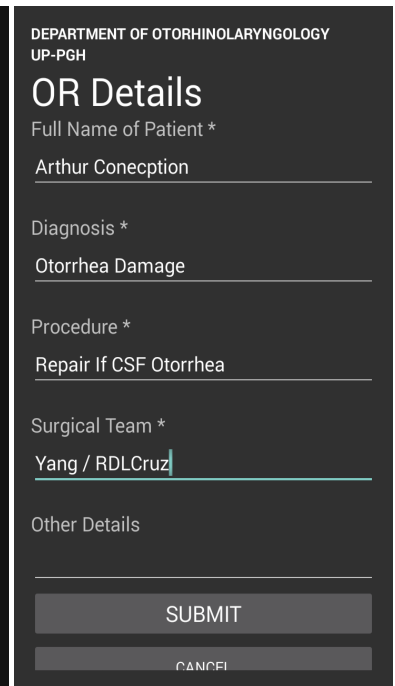


Figure 49: Create OR Schedule Page 6

Figure 47 shows a confirmation dialog before deleting an OR schedule and Figure ?? shows an alert for a successful deletion of OR schedule. In Figure 49 we create again another schedule under the pay or private section by a consultant named John Smith.

In the situation when two users, specifically consultants, happen to create an OR schedule at the same time on the same slot, whoever first finishes inputting the schedule will get the slot. Figure 50 shows an alert for the consultant that the slot is already taken. Figure 51 shows the Friday view page for private. As mentioned earlier, a consultant cannot operate at the same time in the other rooms with the same slot number. So, making a schedule within the same slot number within the operating rooms is not allowed. As shown in Figure 51, the consultant can no longer create a schedule on the same slot numbers for the slots

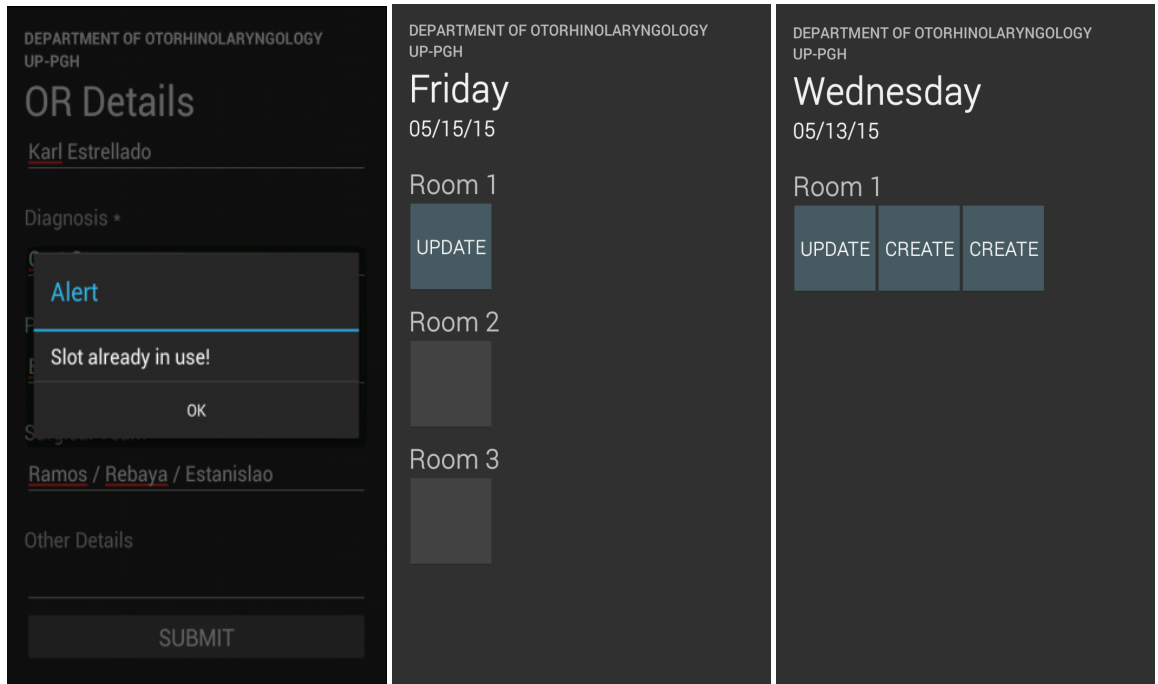


Figure 50: Create OR Schedule alert

Figure 51: View OR Schedule Page for Private 4

Figure 52: View OR Schedule Page for Private 5

are already disabled but other consultants can. Updating a particular OR schedule is only possible for that consultant who created that particular OR schedule. Other consultants can only view the said OR schedule. Figure ?? is the private OR Schedule Page for Wednesday. Here, the consultant can create on all the timeslots shown for they do not constitutes the same slot number.

Figure 53 shows other consultant's account viewing the page. Figure 54 shows the user's change password page. In this case we want to change the password of John Smith from 'password' to 'smith'. An alert comes out notifying the user a successfully changed password as shown in Figure 55.

Figure 56 shows the Generate OR Schedule Report Page. The generate button is disabled until it checked and retrieved the dates that has OR schedules. The generate button is also disabled if there are no OR schedules made. Therefore the dates to be selected for generating are empty. Figure 57 shows the page after retrieving, if there are any, the dates. In Figure 58, the user can choose the date of the OR schedule he wishes to generate and print. Only the dates that have OR schedules are shown.

We chose the date 5/15/15, Friday as shown in Figure 59. An alert dia-

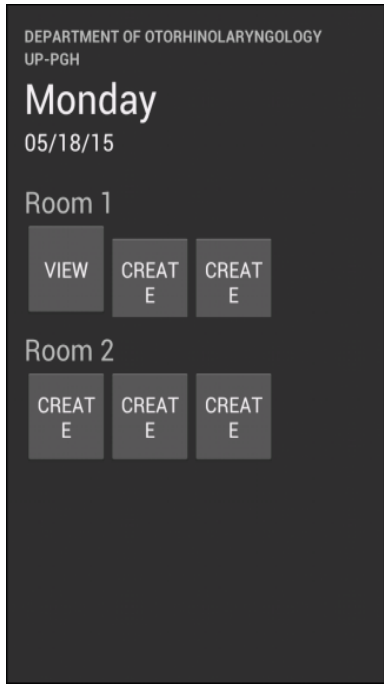


Figure 53: View OR Schedule Page for Private 6

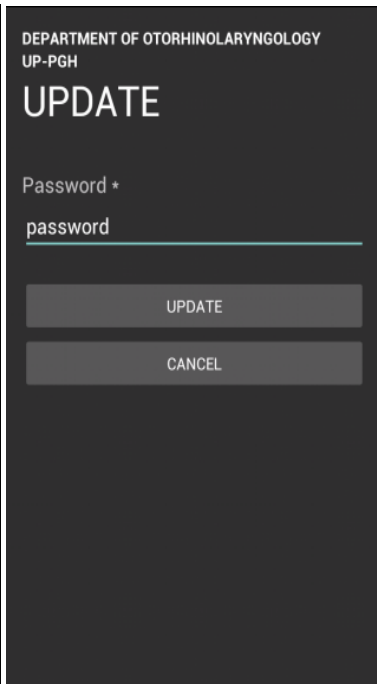


Figure 54: User Change Password

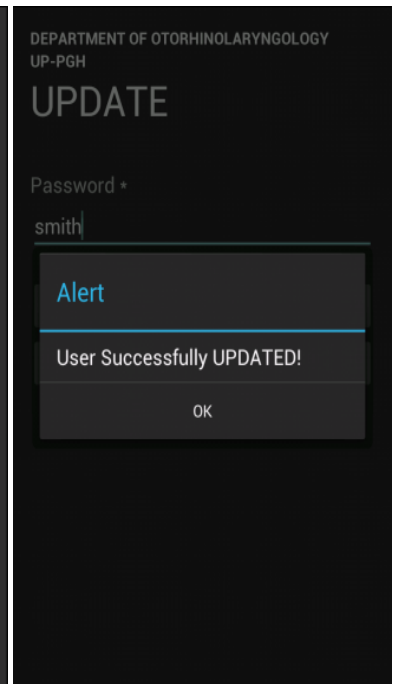


Figure 55: User Change Password Success

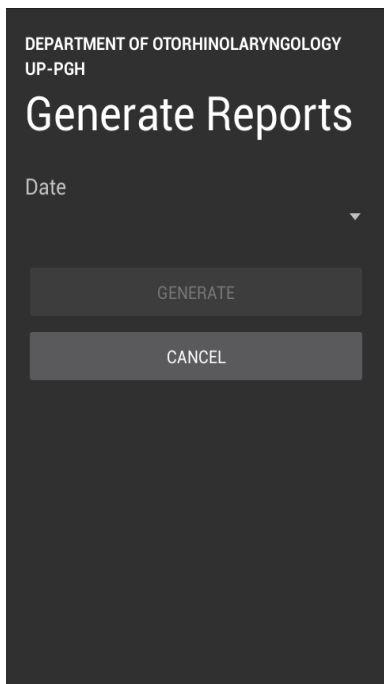


Figure 56: Generate OR Schedule Report Page

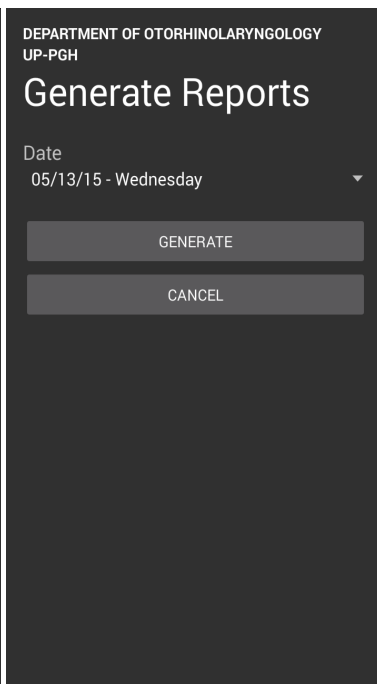


Figure 57: Generate OR Schedule Report Page 2

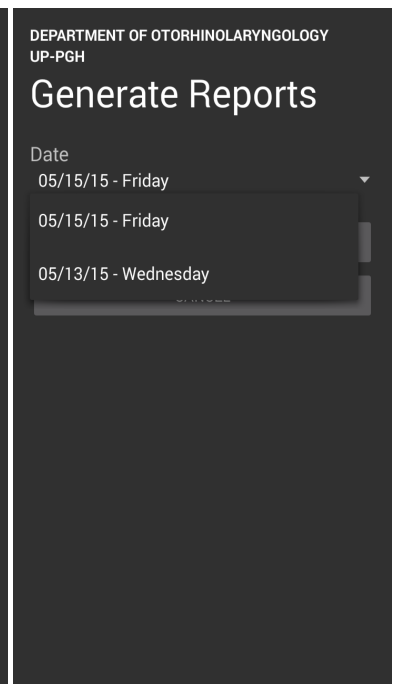


Figure 58: Generate OR Schedule Report Page 3

log pops out informing a successfully generated report. The OR schedule report will open and the user will be directed to the PDF file as shown in Figure 60. As mentioned earlier, we created two OR schedules. One from each side of the Private and Charity OR's. This is an automated generation of report and does

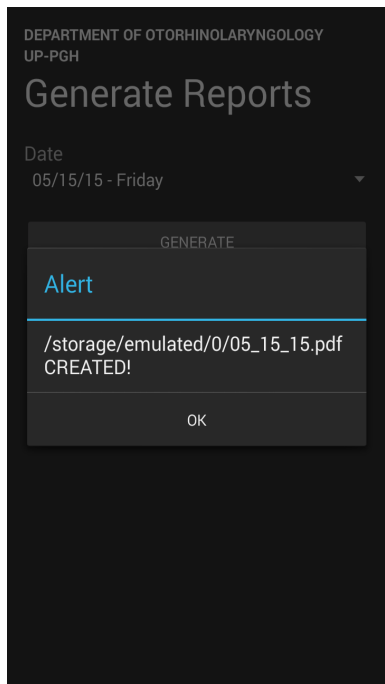


Figure 59: Generate OR Schedule Report Page 4

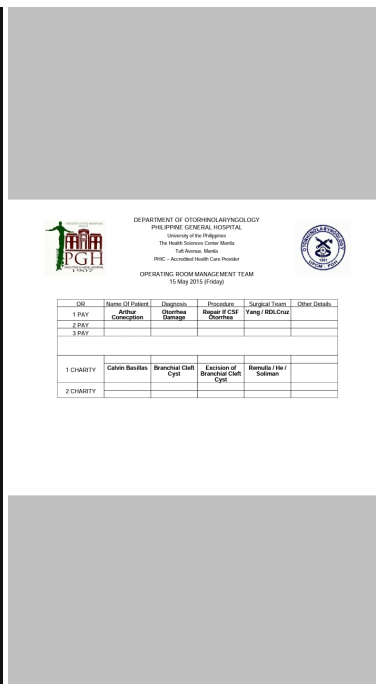


Figure 60: Generate OR Schedule Report Page 5

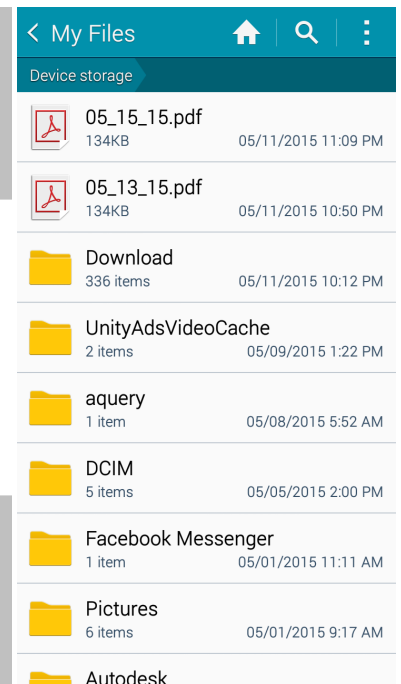


Figure 61: Generate OR Schedule Report Page 6

not require manual typing of OR details into the PDF file. Figure 61 shows the location where the PDF file or the OR schedule report is stored.

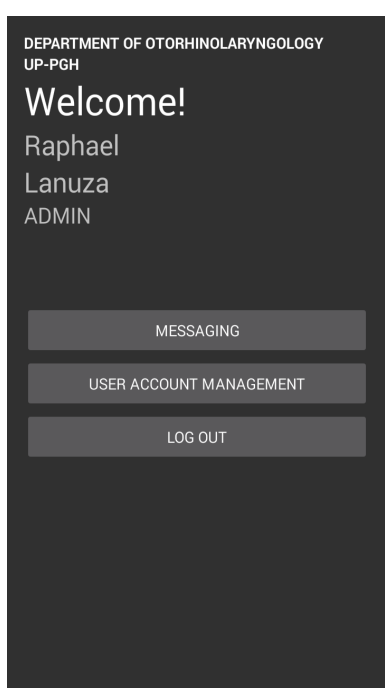


Figure 62: Admin Page

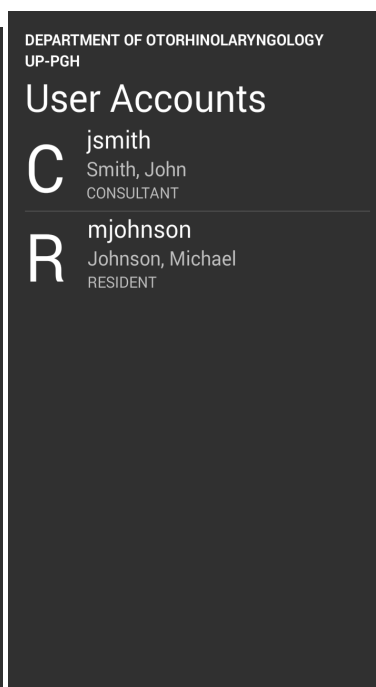


Figure 63: User Account Management Page

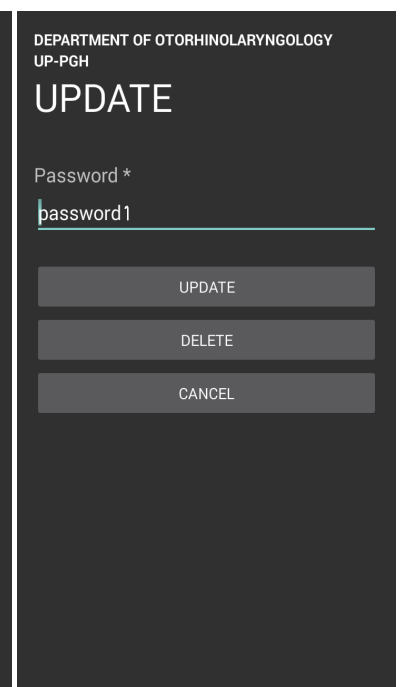


Figure 64: Update User Account Page

Figure 62 shows the panel or dashboard for the Administrator. Figure 63

shows the User Account Management Page. It displays the list of all the user accounts. The administrator can see all the user's details. The big letter 'C' stands for consultant and 'R' stands for resident. Figure 64 shows the update page for the user account. In this case, this is the account of Michael Johnson, a resident. The administrator's can only change their passwords or delete them.

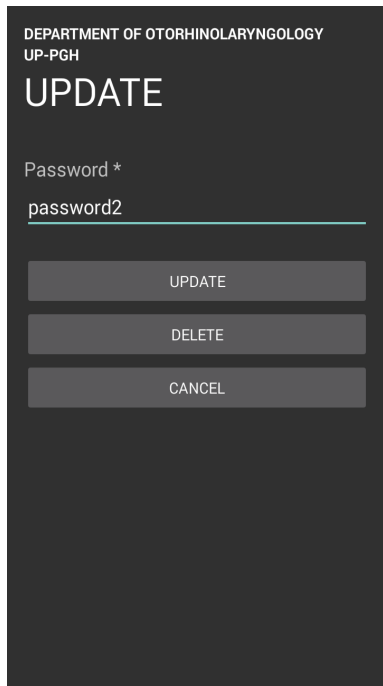


Figure 65: Update User Account Page 2

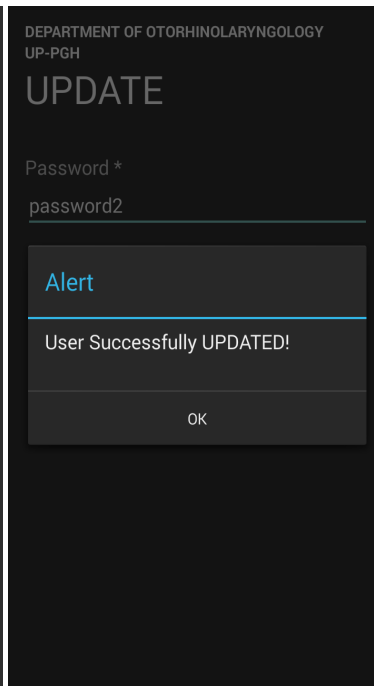


Figure 66: Update User Account Page 3

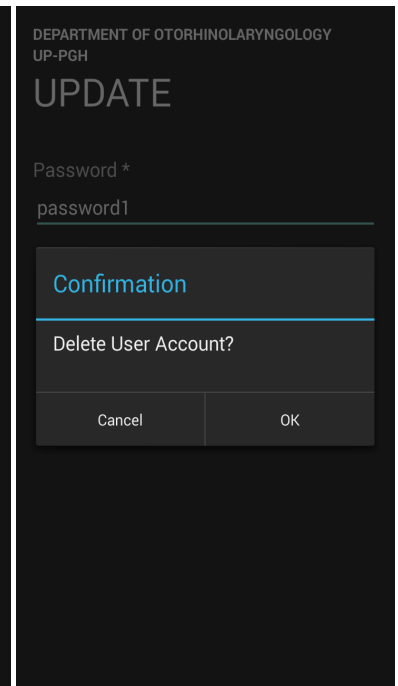


Figure 67: Update User Account Page 4

We update the resident's password with 'password2'. Before it was 'password1' as shown in Figure 65. Figure 66 shows an alert for a successfully updated account by changing the password of the user. Figure 67 shows the confirmation dialog before deleting the account.

An alert for a successful deletion of the account pops out as shown in Figure 68. The user account, Michael Johnson, a resident is now deleted and removed from the list of user accounts as shown in Figure 69. Figure 70 shows the Messaging page.

Figure 71 shows the user interface of send message page. This is the same for the user accounts in the system. In Figure 72 shows all the user accounts registered in the system including the administrator. The user cannot send a

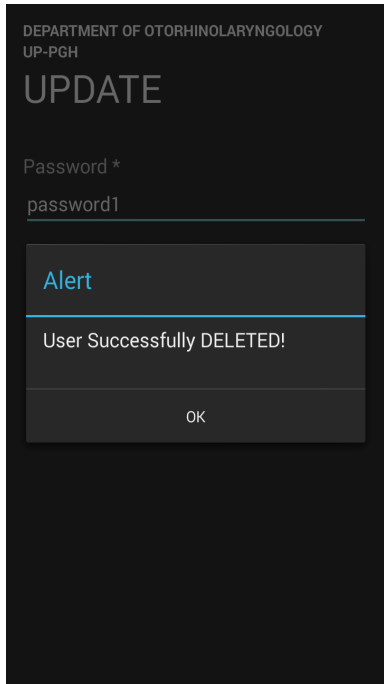


Figure 68: Delete User Account Page

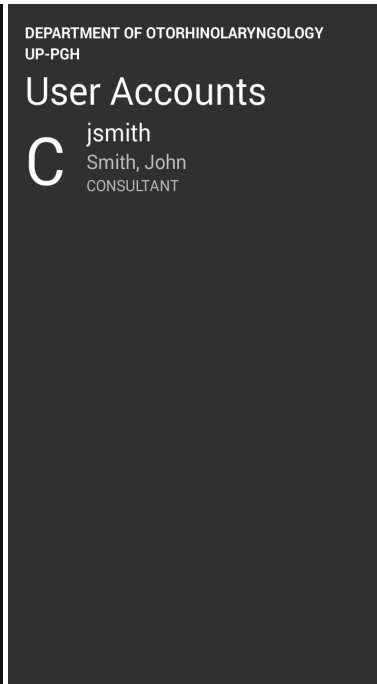


Figure 69: User Account Management Page 2

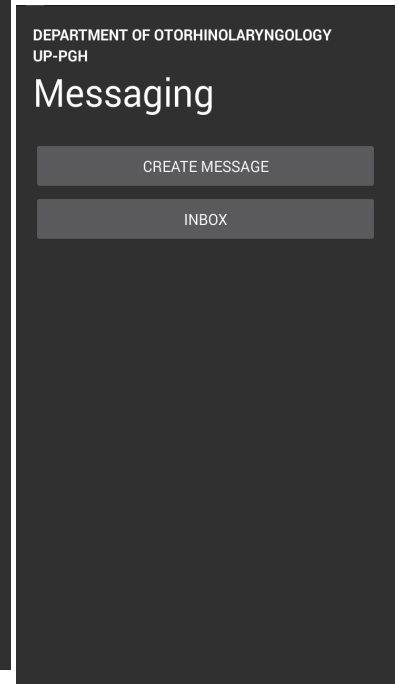


Figure 70: Messaging Page

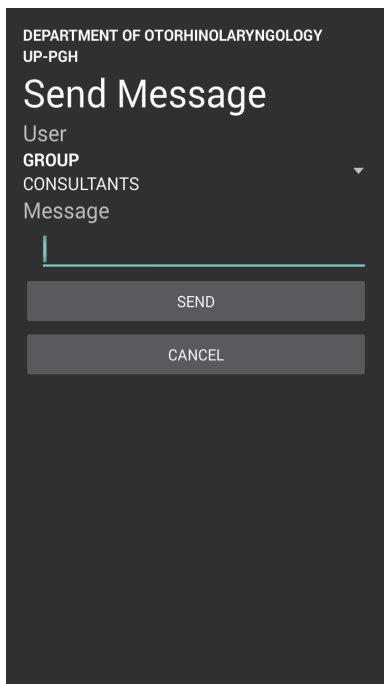


Figure 71: Send Message Page

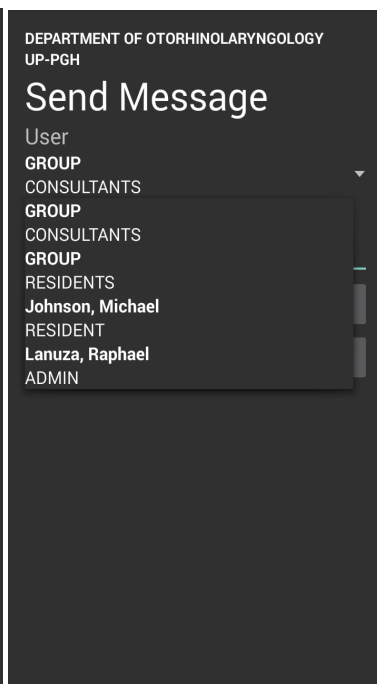


Figure 72: Send Message to Users

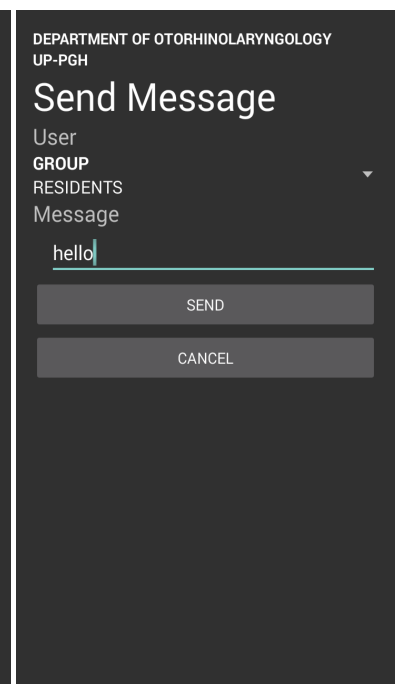


Figure 73: Send Message to Residents

message to himself and so he will be not on the list. The group means choosing all the residents only or all the consultants only. In this case, it is John Smith, a consultant, who is trying to send a message. Figure 73 shows the user sending a message to all the residents only. In this case it is the administrator who is trying

to send a 'hello' message to all the residents.

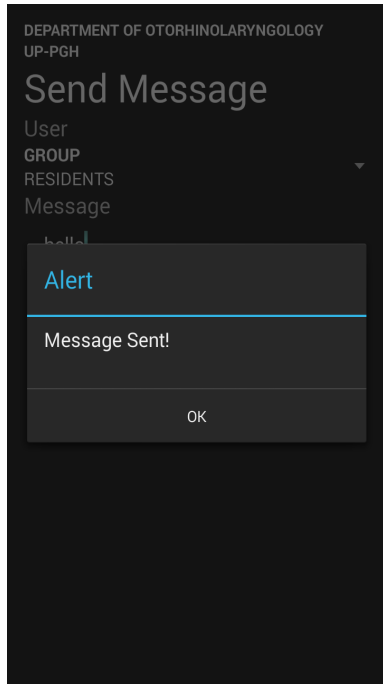


Figure 74: Send Message to Residents 2

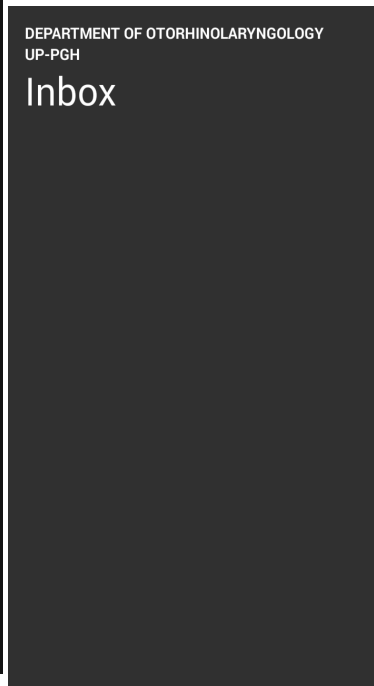


Figure 75: Inbox Page

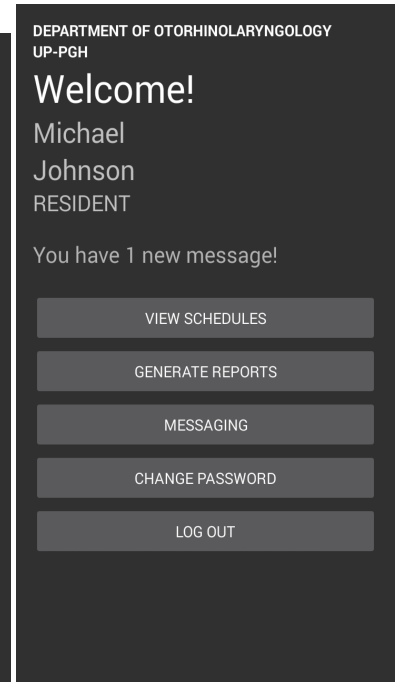


Figure 76: Resident Page with New Message Notification

Figure 74 shows an alert with a successfully sent message. Figure 75 shows the Inbox page. In this case the inbox is empty. Figure 76 shows a the dashboard or panel of Michael Johnson, a resident, with a new message received notification. This will only be gone after the user opened the message from the inbox.

Figure 77 shows an inbox with a message received. This is Michael Johnson's inbox and he is a resident. The 'new' located on the upper right corner of the message will be removed only after opening the message. The message is opened and it now displays the message 'hello' sent by the administrator as shown in Figure 78. The details of the sender is also shown. In Figure 79, it shows a confirmation before deleting the message.

Figure 80 shows an alert for a successfully deleted message.

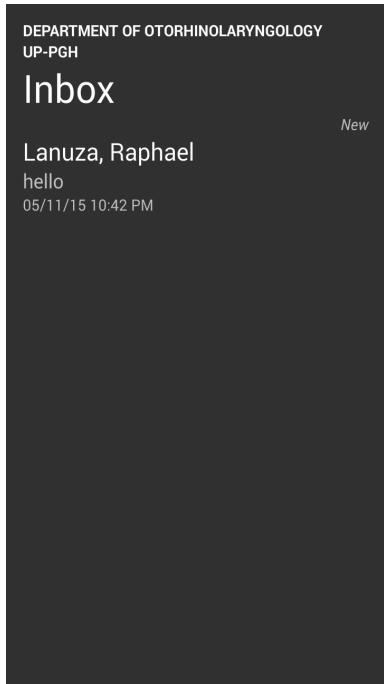


Figure 77: Inbox with Message Page

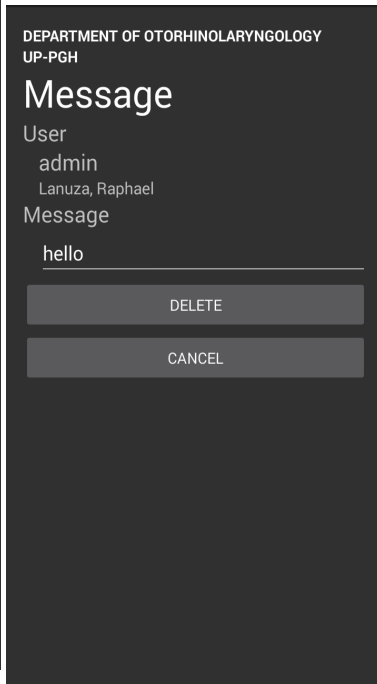


Figure 78: Message Page

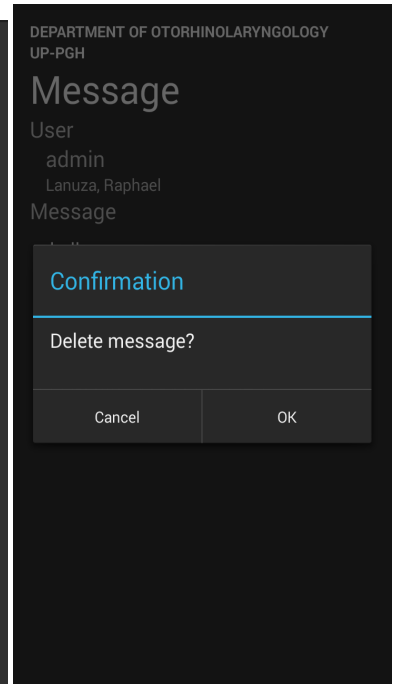


Figure 79: Delete Message Page

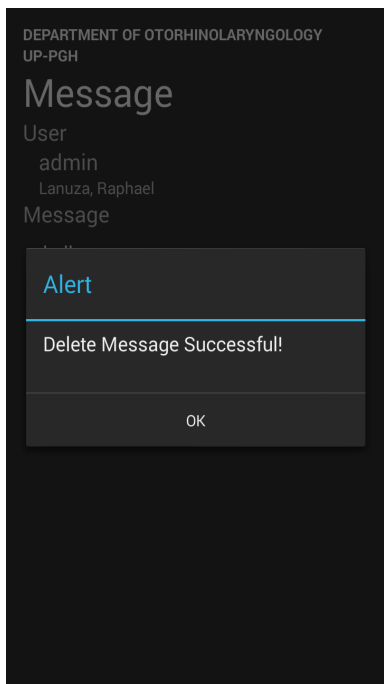


Figure 80: Message Deleted Page

VII. Discussions

The Otorhinolaryngology Operating Room Scheduler (ORL OR Scheduler) is a mobile application that helps the users to efficiently create and manage OR schedules. Since we removed the paper based or manual scheduling, the application gives the users a flexible and organized scheme in making an OR schedule at ease. It is also a user-friendly application making it convenient to use.

The days are presented with the range of a month excluding Saturdays and Sundays. Days are shown regardless of whether the days belong to the same month. Once the day has passed, the application will add or increment another day and the previous day will be removed from the list. Hence, viewing and creating OR schedules is no longer possible for that particular day that was removed. Say for instance, in the charity OR, if the day is Monday, the schedule that will be presented on the user interface is as follows: Monday, Tuesday, Wednesday, Thursday, Friday, the days next week and so on. So long as it has a range of a month. Once Monday has passed, the schedule for Monday shall be removed from the list and the next day for the end day is now included on the list, as such: Tuesday, Wednesday, Thursday, Friday, Monday, the days next week and so on. For the pay/private OR, the implementation is the same as the charity OR but having only the days of Monday, Wednesday and Friday.

This is to give allowance and time for the Department to check the availability and capacity of the Operating Rooms for the next days. This also gives the users the ability to view and create OR schedules ahead of time. But since the records of the OR schedules are not the concern of the system, we only apply a month long period of scheduling.

The tools and items needed in surgical procedures in the Department are already assigned to a specific room and they do not change unless there are circumstances that need to be solved.

As mentioned earlier, the system has five types of users. These are the consultants, residents, administrator and one representative each from the Anesthesiology Department and Nursing Services Department.

The consultants are the ones in charge for the OR schedules in the pay or private OR's. They should be registered to access the system. Upon login in,

the consultants can easily create and view OR schedules. They can also update only the OR schedules that they created. Pay or private OR is only available for Mondays, Wednesdays and Fridays.

On the other hand, the residents are the ones in charge for the charity OR schedules. Same goes for the consultants, they can easily create and view OR schedules. In the charity part, there is only one resident assigned to each day to make the schedules and complete the slots due for tomorrow. Charity OR is available from Mondays to Fridays.

The administrator manages the user accounts and is also in charge for the installation of the application upon the request of the users. The representatives from the Anesthesiology and Nursing Services Departments do not need registration and login authentication to view the OR schedules.

The application is a real time system wherein it is updated whenever a user creates a schedule. Providing only it has an internet connection. Another is that users can send and receive messages to each other within the application itself. Meaning there is no more need for the text and call features provided by the mobile phone. This is to notify through sending messages to the residents by the consultants and vice versa if there are any concerns regarding the OR schedules.

By the time the OR schedules are already completed, the resident or the consultant can generate the OR schedule reports. This is a ready-to-print made OR schedule template in PDF format. They can choose to print or save it in the device for later use.

VIII. Conclusion

The mobile application ORL OR Scheduler is a very efficient and helpful tool for the users in the Department of Otorhinolaryngology of Philippine General Hospital especially the ENT doctors in terms of OR scheduling. A lot of surveys and studies showed that the number of people using smartphones are rapidly increasing from time to time nowadays. Thus, we take advantage to utilize and maximize the usage of smartphones especially running on an Android Operating Systems. As well as the advancement of technologies in mobile phones are also increasing.

The application is designed to have a simple interface making it a user-friendly system. It has made OR scheduling and preparation easier and organized. Paperless creation and management of OR schedules were made possible by the application. The users can easily make schedules and update them at any time electronically provided they are connected to an internet connection. The details of the OR schedules can be viewed easily and conveniently at real time. Also, they can send and receive messages to each other through the application itself. Means of communication such as text messaging and calls were no longer necessary to fulfill the whole OR scheduling. Moreover, as soon as the OR schedules are completed and ready, the application can generate OR schedule reports for a particular day and it is already ready for printing.

IX. Recommendations

The ORL OR Scheduler mobile application is a scheduling tool that provides the users the capability to create, manage and view OR schedules as well as generating reports. Also, it has messaging feature to notify each users. Creating a schedule can be extended by adding a list of only all the available nurses, surgeons and anesthesiologists for the residents and the consultants to select who they want to include in the OR surgical team. Another is adding a search feature to look for a specific OR schedule and details.

Other extensions are the users have the ability to remove themselves from the group list in the messaging feature and adding a reply feature in the messages.

To further improve the system, it should implement the calendar view in the interface giving the residents and consultants to pick any day from the calendar to view and create OR schedules.

X. Bibliography

- [1] Greenspun, Henry (2012). *US Deloitte Center for Health Solutions. mHealth in an mWorld, How mobile technology is transforming health care*, 1, 3. Retrieved 23 November 2014 from <http://www2.deloitte.com/us/en/pages/life-sciences-and-health-care/articles/center-for-health-solutions-mhealth-in-an-mworld.html>
- [2] Wilson, S. (2012). *Open Mobile: The Growth Era Accelerates. The Deloitte Open Mobile Survey 2012, Deloitte Research*. Retrieved 23 November 2014 from http://www.deloitte.com/view/en_US/us/Industries/Telecom-Telecommunications-Technology/69f289e50b484310VgnVCM2000001b56f00aRCRD.htm
- [3] <https://www.informs.org/ORMS-Today/Public-Articles/June-Volume-39-Number-3/Scheduling-for-better-healthcare2> Last Accessed Dec. 21, 2014
- [4] Sufahani, S.F., Mohd Razali S.N.A. & Ismail, Z. (2012). *A Scheduling Problem for Hospital Operating Theatre*, 1, 1-2. Retrieved from 24 November 2014 from <http://arxiv.org/ftp/arxiv/papers/1205/1205.2108.pdf>
- [5] Burney, S.M., Mahmood, N. & Abbas, Z. (2010). Information and Communication Technology in Healthcare Management Systems: Prospects for Developing Countries. *International Journal of Computer Applications*, 4(2), 27-32. Retrieved 22 November 2014 from <http://ijcaonline.net/volume4/number2/pxc3871138.pdf>
- [6] Wike, R. & Oates, R. (2014). *Emerging Nations Embrace Internet, Mobile Technology: Cell Phones Nearly Ubiquitous in Many Countries*. Retrieved 22 November 2014 from <http://www.pewglobal.org/files/2014/02/Pew-Research-Center-Global-Attitudes-Project-Technology-Report-\FINAL-February-13-20146.pdf>

- [7] West, D. (2012). How Mobile Devices are Transforming Healthcare. *Issues in Technology Innovation*, 18, 1-14. Retrieved 22 November 2014 from http://www.insidepolitics.org/brookingsreports/mobile_health_52212.pdf
- [8] Ventola, C.L. (2014). Mobile Devices and Apps for Health Care Professionals: Uses and Benefits. *Pharmacy and Therapeutics*, 39(5), 356364. Retrieved 22 November 2014 from <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4029126/>
- [9] Mosa, A.S., Yoo, I. & Sheets, L. (2012). A Systematic Review of Healthcare Applications for Smartphones. *BMC Medical Informatics and Decision Making*, 12(67), 1-31. DOI:10.1186/1472-6947-12-67.
- [10] Singh, R. (2014). *An Overview of Android Operating System and Its Security Features*. *Journal of Engineering Research and Applications*, 2(1), 519-521. Retrieved 22 November 2014 from http://www.ijera.com/papers/Vol14_issue2/Version%201/CC4201519521.pdf
- [11] Heger, D.A. (n.d.). *Mobile Devices: An Introduction to the Android Operating Environment*. Retrieved 22 November 2014 from <http://www.dhtusa.com/media/AndroidInternals.pdf>
- [12] Pooja, K.P. (2014). Android Based Appointment Scheduler and Location Helper using file operation. *Journal of Computer Engineering*, 16(2), 71-75. Retrieved 22 November 2014 from <http://www.iosrjournals.org/iosr-jce/papers/Vol16-issue2/Version-11/N0162117175.pdf>
- [13] Baxi, P.S. (2008). *Android Application of Quick Organizer*. Masters Thesis, Maharashtra Institute of Technology College Of Engineering, India. Retrieved 22 November 2014 from <https://krex.k-state.edu/dspace/bitstream/handle/2097/13800/PoonamBaxi2012.pdf?sequence=1>
- [14] Choudhari, S.B. et al. (2014). Android Application for Doctors Appointment. *International Journal of Innovative Research in Computer and Communication Engineering*, 2(1), 2472-2474. Retrieved 29 November 2014 from http://www.ijircce.com/upload/2014/january/11_Android.pdf
- [15] Gavaskar, S., Sumithra, A. & Saranya, A. (2013). Health Portal: An Android Smarter Healthcare Application. *International Journal of Research in*

- Engineering and Technology*, 2(9), 291-295. Retrieved 22 November 2014 from http://ijret.org/Volumes/V02/I09/IJRET_110209043.pdf
- [16] Cardoen, B., Demeulemeester, E. & Belin, J. (n.d.). *Operating room planning and scheduling: A literature review*. Retrieved 29 November 2014 from https://lirias.kuleuven.be/bitstream/123456789/165923/1/kbi_0807.pdf
- [17] Sufahani, S.F., Mohd Razali, S.N. & Ismail, Z. (2001, November 1-3). A Scheduling Problem for Hospital Operating Theatre. Paper presented at the International Seminar on the Application of Science & Mathematics. Retrieved 22 November 2014 from <http://arxiv.org/ftp/arxiv/papers/1205/1205.2108.pdf>
- [18] Romanyuk, A. & Silva, A. (2012). *Optimization of an Operating Room Surgical Schedule*. Masters Thesis, Washington University, St. Louis, Missouri. Retrieved 29 November 2014 from <http://ese.wustl.edu/ContentFiles/Research/UndergraduateResearch/CompletedProjects/WebPages/sp12/FinalReportSilvaRomanyuk.pdf>
- [19] Denton, B.T. et al. (2010). Optimal Allocation of Surgery Blocks to Operating Rooms Under Uncertainty. *Operations Research*, 58(4), 802816. DOI 10.1287/opre.1090.0791.
- [20] Agdestein, S.D. (2012). *Operating Room Scheduling Problem*. Masters Thesis, Norwegian University of Science and Technology, Trondheim, Norway. Retrieved 30 November 2014 from <http://www.diva-portal.org/smash/get/diva2:626507/FULLTEXT01.pdf>
- [21] Abdelrasol, Z.Y., Harraz, N. & Eltawil, A. (2013, October 23-25). *A Proposed Solution Framework for the Operating Room Scheduling Problems*. Paper presented at the World Congress on Engineering and Computer Science, San Francisco, USA. Retrieved 20 November 2014 from http://www.iaeng.org/publication/WCECS2013/WCECS2013_pp1149-1157.pdf
- [22] Rafaliya, N.R. (2013). *Scheduling Elective Surgeries in Operation Room with Optimization of Post-Surgery Recovery Unit Capacity*. Masters Thesis, University of Windsor, Canada. Retrieved 30 November 2014 from <http://scholar.uwindsor.ca/cgi/viewcontent.cgi?article=5755&context=etd>

- [23] Stuart, K. & Kozan, E. (2009, December 14-16). *Online scheduling in the operating theatre*. Paper presented at the Kitakyushu International Conference Center, Kitakyushu, Japan. Retrieved 30 November 2014 from <http://eprints.qut.edu.au/29825/1/c29825.pdf>
- [24] Pitteger, P. (n.d.). Operating Room Scheduling Systems. Retrieved 30 November 2014 from <https://www.distance.purdue.edu/training/cssp/cis/pdf/CIS226.pdf>
- [25] Marjamaa, R., Vakkuri, A. & Kirvel, O. (2008). Operating room management: why, how and by whom? *Acta Anaesthesiologica Scandinavica*, 52(5), 596-600. DOI: 10.1111/j.1399-6576.2008.01618.x.
- [26] <http://entnet.org/content/about-us> Last Accessed Nov. 25, 2014
- [27] http://www.upcm.ph/?q=dept_otorhinolaryngology Last Accessed Nov. 25, 2014
- [28] http://nursing411.org/Courses/MD0923_Intro_Operating_Room/introduction_Intro_Operating_Room.html Last Accessed Nov. 25, 2014
- [29] http://en.wikipedia.org/wiki/Operating_room_management#cite_ref-1 Last Accessed Nov. 25, 2014
- [30] Tech Savvy Seniors (n.d.). Introduction to Smartphones Part 2. Beginners Guide. Retrieved 21 November 2014 from <https://www.telstra.com.au/telstra-seniors/download/document/beginners-intro-smartphones.pdf>
- [31] ITU-T TechWatch Alert (2009). Mobile Applications. Retrieved 27 November 2014 from http://www.itu.int/dms_pub/itu-t/oth/23/01/T230100000C0004PDFE.pdf
- [32] Islam R.Md., Islam R.Md., Mazumder T.A. & Boras H.M.Sc. (2010). Mobile Application and Its Global Impact *International Journal of Engineering & Technology*, 10(6), 1. IJET-IJENS. Retrieved 24 November 2014 from <http://www.ijens.org/107506-0909%20ijet-ijens.pdf>
- [33] mHealth: New horizons for health through mobile technologies: secondary global survey on eHealth. WHO Library Cataloguing-in-Publication Data. 1,6.

Retrieved 22 November 2014 from http://www.who.int/goe/publications/goe_mhealth_web.pdf

[34] Connecting America: The National Broadband Plan. Health Care: Chapter 10. Federal Communications Commissions. www.broadband.gov 10,200. Retrieved 23 November 2014 from <http://www.americantelemed.org/docs/default-source/policy/connecting-america-the-national-broadband-plan.pdf?sfvrsn=8>

[35] http://en.wikipedia.org/wiki/Android_software_development Last Accessed Jan. 21, 2015

[36] <http://en.wikipedia.org/wiki/IText> Last Accessed May 12, 2015

XI. Appendix

config.inc.php

```
<?php

$username = "scheduler";
$password = "scheduler";
$host = "localhost";
$dbname = "ORLORSCHED";

$options = array(PDO::MYSQL_ATTR_INIT_COMMAND => 'SET NAMES
    utf8');

try
{
    $db = new
        PDO("mysql:host={$host};dbname={$dbname};charset=utf8",
            $username, $password, $options);
}
catch(PDOException $ex)
{
    die("Failed to connect to the database: " .
        $ex->getMessage());
}

$db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

$db->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE,
    PDO::FETCH_ASSOC);

if(function_exists('get_magic_quotes_gpc') &&
    get_magic_quotes_gpc())
{
```

```

function undo_magic_quotes_gpc(&$array)
{
    foreach($array as &$value)
    {
        if(is_array($value))
        {
            undo_magic_quotes_gpc($value);
        }
        else
        {
            $value = stripslashes($value);
        }
    }
}

undo_magic_quotes_gpc($_POST);
undo_magic_quotes_gpc($_GET);
undo_magic_quotes_gpc($_COOKIE);
}

header('Content-Type: text/html; charset=utf-8');

session_start();

```

?>

createorsched.php

<?php

```
require("config.inc.php");
```

```
if (!empty($_POST)) {
```

```

if($_POST['operation']=="CREATE"){
    $query = " SELECT 1 FROM ortab WHERE SCHED_DATE =
        STR_TO_DATE(:sched_date, '%m/%d/%Y') and OR_TYPE =
        :or_type and ROOM_NUM = :room_num and SLOT_NUM =
        :slot_num";

    $query_params = array(
        ':or_type' => $_POST['or_type'],
        ':sched_date' => $_POST['sched_date'],
        ':room_num' => $_POST['room_num'],
        ':slot_num' => $_POST['slot_num']
    );

    try {
        $stmt = $db->prepare($query);
        $result = $stmt->execute($query_params);
    }
    catch (PDOException $ex) {
        $response["success"] = 0;
        $response["message"] = "Database Error. Please Try
            Again!";
        die(json_encode($response));
    }
    $row = $stmt->fetch();
    if ($row) {
        $response["success"] = 0;
        $response["message"] = "Slot already in use!";
        die(json_encode($response));
    }

    $query = "INSERT INTO 'ortab'('ROOM_NUM', 'SLOT_NUM',
        'OR_TYPE', 'SCHED_DATE', 'PATIENT_NAME', 'DIAGNOSIS',
        'PROCEDURE', 'SURGICAL_TEAM', 'OTHER_DETAILS', 'USER_ID')

```



```

VALUES
    (:room_num,:slot_num,:or_type,STR_TO_DATE(:sched_date,
        '%m/%d/%Y'),:patient_name,:diagnosis,:procedure,
:surgical_team,:other_details,:user_id) ";

$message_operation = "Created";
$query_params = array(
    ':room_num' => $_POST['room_num'],
    ':slot_num' => $_POST['slot_num'],
    ':or_type' => $_POST['or_type'],
    ':sched_date' => $_POST['sched_date'],
    ':patient_name' => $_POST['patient_name'],
    ':diagnosis' => $_POST['diagnosis'],
    ':procedure' => $_POST['procedure'],
    ':surgical_team' => $_POST['surgical_team'],
    ':other_details' => $_POST['other_details'],
    ':user_id' => $_POST['user_id']
);
} else if($_POST['operation']=="UPDATE"){
    $query = "UPDATE `ortab` SET
        `PATIENT_NAME`=:patient_name,`DIAGNOSIS`=:diagnosis,`PROCEDURE`
=:procedure,`SURGICAL_TEAM`=:surgical_team,`OTHER_DETAILS`
=:other_details WHERE `OR_ID`=:or_id";

$message_operation = "Updated";
$query_params = array(
    ':or_id' => $_POST['or_id'],
    ':patient_name' => $_POST['patient_name'],
    ':diagnosis' => $_POST['diagnosis'],
    ':procedure' => $_POST['procedure'],
    ':surgical_team' => $_POST['surgical_team'],
    ':other_details' => $_POST['other_details']
);

```

```

} else if($_POST['operation']=="DELETE"){
    $query = "DELETE FROM 'ortab' where 'OR_ID' = :or_id ";
    $message_operation = "Deleted";
    $query_params = array(
        ':or_id' => $_POST['or_id']
    );
}

try {
    $stmt = $db->prepare($query);
    $result = $stmt->execute($query_params);
}
catch (PDOException $ex) {
    $response["success"] = 0;
    $response["message"] = $ex;
    die(json_encode($response));
}

$response["success"] = 1;
$response["message"] = "OR Schedule Successfully " .
    $_POST['operation'] . "D!";
echo json_encode($response);

}
?>

```

getcontacts.php

```

<?php

require("config.inc.php");

$query = "SELECT USERNAME, FIRSTNAME, LASTNAME, USER_TYPE FROM
    'users' ORDER BY CONCAT(LASTNAME,FIRSTNAME) ASC";

```

```

try {
    $stmt = $db->prepare($query);
    $result = $stmt->execute();
}
catch (PDOException $ex) {
    $response["success"] = 0;
    $response["message"] = $ex;
    die(json_encode($response));
}

$rows = $stmt->fetchAll();
$response["success"] = 1;
$response["message"] = "Get Contacts Successful!";
$response["contacts"] = array();
if ($rows) {
    foreach ($rows as $row) {
        $user = array();
        $user["USERNAME"] = $row["USERNAME"];
        $user["FIRSTNAME"] = $row["FIRSTNAME"];
        $user["LASTNAME"] = $row["LASTNAME"];
        $user["USER_TYPE"] = $row["USER_TYPE"];
        array_push($response["contacts"], $user);
    }
}

echo json_encode($response);

```

?>

getmessages.php

<?php

```
require("config.inc.php");
```

```

if (!empty($_POST)) {
    if(!empty($_POST['count'])){
        $query = "SELECT count(*) as COUNT FROM messaging LEFT JOIN
            users ON sender_user_id = user_id WHERE receiver_user_id
            = :receiver_user_id and read_flag = 0 ";
        $query_params = array(
            ':receiver_user_id' => $_POST['receiver_user_id']
        );
        try {
            $stmt = $db->prepare($query);
            $result = $stmt->execute($query_params);
        }
        catch (PDOException $ex) {
            $response["success"] = 0;
            $response["message"] = "Database Error. Please Try
                Again!";
            die(json_encode($response));
        }

        $row = $stmt->fetch();
        $response["success"] = 1;
        $response["message"] = "Get Message Count Successful!";
        $response["COUNT"] = $row["COUNT"];
        die(json_encode($response));
    } else if(!empty($_POST['message_id'])){
        $query = "DELETE FROM messaging WHERE message_id =
            :message_id";
        $query_params = array(
            ':message_id' => $_POST['message_id']
        );
        try {
            $stmt = $db->prepare($query);

```

```

        $result = $stmt->execute($query_params);
    }
    catch (PDOException $ex) {
        $response["success"] = 0;
        $response["message"] = "Database Error. Please Try
            Again!";
        die(json_encode($response));
    }
    $response["success"] = 1;
    $response["message"] = "Delete Message Successful!";
    echo json_encode($response);
} else {
    $query = "SELECT * FROM messaging LEFT JOIN users ON
        sender_user_id = user_id WHERE receiver_user_id =
        :receiver_user_id ORDER BY SENT_DATE DESC ";
    $query_params = array(
        ':receiver_user_id' => $_POST['receiver_user_id']
    );
    try {
        $stmt = $db->prepare($query);
        $result = $stmt->execute($query_params);
    }
    catch (PDOException $ex) {
        $response["success"] = 0;
        $response["message"] = "Database Error. Please Try
            Again!";
        die(json_encode($response));
    }

    $rows = $stmt->fetchAll();
    $response["success"] = 1;
    $response["message"] = "Get Messages Successful!";
    $response["messages"] = array();
}

```

```

if ($rows) {
    foreach ($rows as $row) {
        $message = array();
        $message["message_id"] = $row["message_id"];
        $message["message"] = $row["message"];
        $message["read_flag"] = $row["read_flag"];
        $message["sent_date"] = date( 'm/d/y h:i A',
            strtotime($row["sent_date"]));

        $user = array();
        $user["user_id"] = $row["USER_ID"];
        $user["firstname"] = $row["FIRSTNAME"];
        $user["lastname"] = $row["LASTNAME"];
        $user["user_type"] = $row["USER_TYPE"];
        $user["username"] = $row["USERNAME"];

        $message["sender_user"] = $user;

        array_push($response["messages"], $message);
    }
}

$query = "UPDATE messaging set read_flag = 1 WHERE
    receiver_user_id = :receiver_user_id";
$query_params = array(
    ':receiver_user_id' => $_POST['receiver_user_id']
);

try {
    $stmt = $db->prepare($query);
    $result = $stmt->execute($query_params);
}
catch (PDOException $ex) {

```

```

    }

    echo json_encode($response);
}
}

?>

```

getorsched.php

```

<?php

require("config.inc.php");

if (!empty($_POST)) {
    if (!empty($_POST['sched_date'])) {

        $query = "SELECT * FROM ortab AS ortab INNER JOIN users AS
            users ON users.user_id = ortab.user_id where sched_date =
            STR_TO_DATE(:sched_date, '%m/%d/%Y')";

        $query_params = array(
            ':sched_date' => $_POST['sched_date']
        );
    } else if (empty($_POST['or_type'])) {
        $query = "SELECT * FROM ortab AS ortab INNER JOIN users AS
            users ON users.user_id = ortab.user_id where sched_date
            >= STR_TO_DATE(:start_date, '%m/%d/%Y') and sched_date <=
            STR_TO_DATE(:end_date, '%m/%d/%Y')";

        $query_params = array(
            ':start_date' => $_POST['start_date'],
            ':end_date' => $_POST['end_date']
        );
    }
}

```

```

} else {
    $query = "SELECT * FROM ortab AS ortab INNER JOIN users AS
        users ON users.user_id = ortab.user_id where sched_date
        >= STR_TO_DATE(:start_date, '%m/%d/%Y') and sched_date <=
        STR_TO_DATE(:end_date, '%m/%d/%Y') and or_type = :or_type
        ";

    $query_params = array(
        ':start_date' => $_POST['start_date'],
        ':end_date' => $_POST['end_date'],
        ':or_type' => $_POST['or_type']
    );
}

try {
    $stmt = $db->prepare($query);
    $result = $stmt->execute($query_params);
}
catch (PDOException $ex) {
    $response["success"] = 0;
    $response["message"] = "Database Error. Please Try Again!";
    die(json_encode($response));
}

$rows = $stmt->fetchAll();
$response["success"] = 1;
$response["message"] = "Get Schedules Successful!";
$response["schedules"] = array();
if ($rows) {
    foreach ($rows as $row) {
        $slot = array();
        $slot["or_id"] = $row["OR_ID"];
        $slot["room_num"] = $row["ROOM_NUM"];
    }
}

```



```

        $slot["slot_num"] = $row["SLOT_NUM"];
        $slot["or_type"] = $row["OR_TYPE"];
        $slot["sched_date"] = date( 'm/d/y',
            strtotime($row["SCHED_DATE"]));
        $slot["patient_name"] = $row["PATIENT_NAME"];
        $slot["diagnosis"] = $row["DIAGNOSIS"];
        $slot["procedure"] = $row["PROCEDURE"];
        $slot["surgical_team"] = $row["SURGICAL_TEAM"];
        $slot["other_details"] = $row["OTHER_DETAILS"];
        $slot["user_id"] = $row["USER_ID"];

        $user = array();
        $user["user_id"] = $row["USER_ID"];
        $user["firstname"] = $row["FIRSTNAME"];
        $user["lastname"] = $row["LASTNAME"];
        $user["user_type"] = $row["USER_TYPE"];
        $user["username"] = $row["USERNAME"];
        $user["password"] = $row["PASSWORD"];

        $slot["user"] = $user;

        array_push($response["scheds"], $slot);
    }
}

echo json_encode($response);
}

?>

```

getscheddates.php

<?php

```

require("config.inc.php");

$query = "SELECT distinct SCHED_DATE FROM ortab order by
        SCHED_DATE DESC";
try {
    $stmt = $db->prepare($query);
    $result = $stmt->execute();
}
catch (PDOException $ex) {
    $response["success"] = 0;
    $response["message"] = "Database Error. Please Try Again!";
    die(json_encode($response));
}

$rows = $stmt->fetchAll();
$response["success"] = 1;
$response["message"] = "Get Schedules Successful!";
$response["dates"] = array();
if ($rows) {
    foreach ($rows as $row) {
        $date = array();
        $date["date"] = date( 'm/d/y - l',
            strtotime($row["SCHED_DATE"]));
        error_log($date["date"], 0);
        array_push($response["dates"], $date);
    }
}

echo json_encode($response);

?>

```

getusers.php

```

<?php

require("config.inc.php");

$query = "SELECT * FROM 'users' WHERE USER_TYPE != 'ADMIN'
        ORDER BY USERNAME ASC ";
try {
    $stmt = $db->prepare($query);
    $result = $stmt->execute();
}
catch (PDOException $ex) {
    $response["success"] = 0;
    $response["message"] = $ex;
    die(json_encode($response));
}
$rows = $stmt->fetchAll();
$response["success"] = 1;
$response["message"] = "Get Contacts Successful!";
$response["users"] = array();
if ($rows) {
    foreach ($rows as $row) {
        $user = array();
        $user["user_id"] = $row["USER_ID"];
        $user["firstname"] = $row["FIRSTNAME"];
        $user["user_type"] = $row["USER_TYPE"];
        $user["lastname"] = $row["LASTNAME"];
        $user["username"] = $row["USERNAME"];
        $user["password"] = $row["PASSWORD"];
        array_push($response["users"], $user);
    }
}

echo json_encode($response);

```

?>

login.php

```
<?php
```

```
require("config.inc.php");
```

```
if (!empty($_POST)) {
```

```
    $query = "SELECT
```

```
        user_id,firstname,lastname,user_type,username, password
```

```
        FROM users WHERE username = :username";
```

```
    $query_params = array(
```

```
        ':username' => $_POST['username']
```

```
    );
```

```
    try {
```

```
        $stmt = $db->prepare($query);
```

```
        $result = $stmt->execute($query_params);
```

```
    }
```

```
    catch (PDOException $ex) {
```

```
        $response["success"] = 0;
```

```
        $response["message"] = "Database Error. Please Try Again!";
```

```
        die(json_encode($response));
```

```
    }
```

```
    $validated_info = false;
```

```
    $row = $stmt->fetch();
```

```
    if ($row) {
```

```
        $user = array();
```

```

$user["user_id"] = $row["user_id"];
$user["firstname"] = $row["firstname"];
$user["lastname"] = $row["lastname"];
$user["user_type"] = $row["user_type"];
$user["username"] = $row["username"];
$user["password"] = $row["password"];
    if ($_POST['password'] === $row['password']) {
        $response["success"] = 1;
        $response["message"] = "Login successful!";
        $response["user"] = $user;
    } else {
        $response["success"] = 0;
        $response["message"] = "Invalid Username or Password!";
    }
    die(json_encode($response));
} else {
    $response["success"] = 0;
    $response["message"] = "Invalid Username or Password!";
    die(json_encode($response));
}
}
?>

```

register.php

```

<?php

require("config.inc.php");

if (!empty($_POST)) {

    if($_POST['operation']=="CREATE"){

```

```

$query          = " SELECT 1 FROM users WHERE username =
                :username";

$query_params = array(
    ':username' => $_POST['username']
);

try {
    $stmt = $db->prepare($query);
    $result = $stmt->execute($query_params);
}
catch (PDOException $ex) {
    $response["success"] = 0;
    $response["message"] = "Database Error. Please Try
        Again!";
    die(json_encode($response));
}

$row = $stmt->fetch();
if ($row) {
    if($_POST['operation']=="CREATE"){
        $response["success"] = 0;
        $response["message"] = "Username is already in use!";
        die(json_encode($response));
    }
}

$query = "INSERT INTO users ( firstname, lastname,
                user_type, username, password ) VALUES ( :firstname,
                :lastname, :user_type, :username, :password) ";
$query_params = array(
    ':firstname' => $_POST['firstname'],
    ':lastname' => $_POST['lastname'],
    ':user_type' => $_POST['user_type'],

```

```

        ':username' => $_POST['username'],
        ':password' => $_POST['password']
    );
} else if ($_POST['operation']=="UPDATE"){
    $query = "UPDATE users set password = :password where
        user_id = :user_id";
    $query_params = array(
        ':user_id' => $_POST['user_id'],
        ':password' => $_POST['password']
    );
} else if ($_POST['operation']=="DELETE"){
    $query = "DELETE FROM users where user_id = :user_id; DELETE
        FROM messaging where sender_user_id = :user_id or
        receiver_user_id = :user_id";
    $query_params = array(
        ':user_id' => $_POST['user_id']
    );
}

try {
    $stmt = $db->prepare($query);
    $result = $stmt->execute($query_params);
}
catch (PDOException $ex) {
    $response["success"] = 0;
    $response["message"] = $ex;
    die(json_encode($response));
}
$response["success"] = 1;
$response["message"] = "User Successfully " .
    $_POST['operation'] . "D!";
echo json_encode($response);

```

```
}  
?>
```

sendmessage.php

```
<?php  
  
require("config.inc.php");  
  
if (!empty($_POST)) {  
    if (!empty($_POST['receiver_user_type'])) {  
        error_log($_POST['receiver_user_type'], 0);  
        $query = "SELECT USER_ID FROM 'users' where  
            CONCAT(user_type,'S') = :receiver_user_type and USER_ID!=  
            :sender_user_id";  
        $query_params = array(  
            ':sender_user_id' => $_POST['sender_user_id'],  
            ':receiver_user_type' => $_POST['receiver_user_type']  
        );  
        try {  
            $stmt = $db->prepare($query);  
            $result = $stmt->execute($query_params);  
        }  
        catch (PDOException $ex) {  
            $response["success"] = 0;  
            $response["message"] = "Database Error. Please Try  
                Again!";  
            die(json_encode($response));  
        }  
  
        $rows = $stmt->fetchAll();  
        if ($rows) {  
            foreach ($rows as $row) {
```



```

$query = "INSERT INTO 'messaging'('sender_user_id',
    'receiver_user_id', 'message') VALUES
    (:sender_user_id, :receiver_user_id, :message)";
$query_params = array(
    'sender_user_id' => $_POST['sender_user_id'],
    'receiver_user_id' => $row['USER_ID'],
    'message' => $_POST['message']
);

try {
    $stmt = $db->prepare($query);
    $result = $stmt->execute($query_params);
}
catch (PDOException $ex) {
    $response["success"] = 0;
    $response["message"] = $ex;
    die(json_encode($response));
}

error_log("message sent to".$row['USER_ID'], 0);
}
}

$response["success"] = 1;
$response["message"] = "Message Sent!";
echo json_encode($response);
} else {
    $query = "INSERT INTO 'messaging'('sender_user_id',
        'receiver_user_id', 'message') VALUES (:sender_user_id,
        (SELECT USER_ID FROM 'users' where USERNAME =
        :receiver_user_name), :message)";
    $query_params = array(
        'sender_user_id' => $_POST['sender_user_id'],

```

```

        ':receiver_user_name' => $_POST['receiver_user_name'],
        ':message' => $_POST['message']
    );

    try {
        $stmt = $db->prepare($query);
        $result = $stmt->execute($query_params);
    }
    catch (PDOException $ex) {
        $response["success"] = 0;
        $response["message"] = $ex;
        die(json_encode($response));
    }
    $response["success"] = 1;
    $response["message"] = "Message Sent!";
    echo json_encode($response);
}
}
?>

```

CONSTANTS.java

```

package orl.orscheduler.objects.global;

/**
 * Created by Raphael Lanuza on 4/25/2015.
 */
public class CONSTANTS {
    //UI OPERATION
    public static final String OPERATION_VIEW = "VIEW";
    public static final String OPERATION_CREATE = "CREATE";
    public static final String OPERATION_UPDATE = "UPDATE";
    public static final String OPERATION_DELETE = "DELETE";
    public static final String OPERATION_DISABLED = "DISABLED";
}

```

```

//OR TYPE
public static final String OR_TYPE_CHARITY = "CHARITY";
public static final String OR_TYPE_PRIVATE = "PRIVATE";

//USER TYPE
public static final String USER_TYPE_RESIDENT = "RESIDENT";
public static final String USER_TYPE_CONSULTANT = "CONSULTANT";
public static final String USER_TYPE_ADMIN = "ADMIN";

//DB JSON RESULT TAG
public static final String TAG_SUCCESS = "success";
public static final String TAG_MESSAGE = "message";

//DAYS
public static final String DAY_MONDAY = "MONDAY";
public static final String DAY_TUESDAY = "TUESDAY";
public static final String DAY_WEDNESDAY = "WEDNESDAY";
public static final String DAY_THURSDAY = "THURSDAY";
public static final String DAY_FRIDAY = "FRIDAY";
public static final String DAY_SATURDAY = "SATURDAY";
public static final String DAY_SUNDAY = "SUNDAY";

public static final String[] VALID_CHARITYDAYS = {DAY_MONDAY,
    DAY_TUESDAY, DAY_WEDNESDAY, DAY_THURSDAY, DAY_FRIDAY};
public static final String[] VALID_ORDDAYS = {DAY_MONDAY,
    DAY_WEDNESDAY, DAY_FRIDAY};
}

\end{XML}

\bfseries GlobalVariables.java
\begin{JAVA}

```

```

package orl.orscheduler.objects.global;

import java.util.ArrayList;
import java.util.List;

import orl.orscheduler.Login;
import orl.orscheduler.objects.or.DaySchedule;
import orl.orscheduler.objects.or.ScheduleParameters;
import orl.orscheduler.objects.user.User;

/**
 * Created by Raphael Lanuza on 4/25/2015.
 */
public class GlobalVariables {
    public static User user = new User();
    public static ScheduleParameters scheduleParameters = new
        ScheduleParameters();
    public static Class<?> viewClass;
    public static List<User> userList = new ArrayList<User>();
    public static List<User> contactList = new ArrayList<User>();
    public static void initializeAll(){
        user = new User();
        scheduleParameters = new ScheduleParameters();
        userList = new ArrayList<User>();
        contactList = new ArrayList<User>();
        viewClass = Login.class;
    }
}

```

PREFERENCES.java

```

package orl.orscheduler.objects.global;

```

```

/**

```

```
* Created by Raphael Lanuza on 4/25/2015.
*/
public class PREFERENCES {
    public static final int DATE_RANGE_COUNT = 31;
}

```

DaySchedule.java

```
package orl.orscheduler.objects.or;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.text.ParseException;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Date;
import java.util.List;
import java.util.Locale;

import orl.orscheduler.objects.global.CONSTANTS;
import orl.orscheduler.objects.global.GlobalVariables;
import orl.orscheduler.objects.user.User;
import orl.orscheduler.utilities.DateUtil;

/**
* Created by Raphael Lanuza on 4/25/2015.
*/
public class DaySchedule {
    private String date;
    private String dayString;
    private String orType;
    private List<Room> roomList;
}

```

```
public String getDate() {
    return date;
}

public void setDate(String date) {
    this.date = date;
}

public String getDayString() {
    return dayString;
}

public void setDayString(String dayString) {
    this.dayString = dayString;
}

public List<Room> getRoomList() {
    return roomList;
}

public void setRoomList(List<Room> roomList) {
    this.roomList = roomList;
}

public String getOrType() {
    return orType;
}

public void setOrType(String orType) {
    this.orType = orType;
}
```

```

public Room getRoom(String roomNumber){
    Room room = new Room();
    for(Room listRoom : this.getRoomList()){
        if(listRoom.getRoomNum().equals(roomNumber)){
            room = listRoom;
        }
    }
    return room;
}

public Room getRoom(int roomNumberInt){
    String roomNumber = String.valueOf(roomNumberInt);
    Room room = new Room();
    for(Room listRoom : this.getRoomList()){
        if(listRoom.getRoomNum().equals(roomNumber)){
            room = listRoom;
        }
    }
    return room;
}

public Boolean isRoomExisting(String roomNumber){
    Boolean existing = false;
    for(Room room : this.getRoomList()){
        if(room.getRoomNum().equals(roomNumber)){
            existing = true;
        }
    }
    return existing;
}

public String getOperation(String roomNum, String slotNum){

```

```

String operation = CONSTANTS.OPERATION_VIEW;
if(GlobalVariables.user.getUser_id()==null){
    if(getRoom(roomNum).getSlot(slotNum).getUser()==null){
        operation = CONSTANTS.OPERATION_DISABLED;
    }
} else {
    operation = CONSTANTS.OPERATION_CREATE;
    if(isRoomExisting(roomNum)){
        if(getRoom(roomNum).isSlotExisting(slotNum)){
            User user =
                getRoom(roomNum).getSlot(slotNum).getUser();
            if(user!=null){
                if(user.getUser_id().equals
(GlobalVariables.user.getUser_id())){
                    operation = CONSTANTS.OPERATION_UPDATE;
                } else {
                    operation = CONSTANTS.OPERATION_VIEW;
                }
            }
        }
    }
}
if(getOrType().equals(CONSTANTS.OR_TYPE_PRIVATE) &&
operation.equals(CONSTANTS.OPERATION_CREATE)){
    for(Room room : getRoomList()){
        Slot slot = room.getSlot(slotNum);
        User user = slot.getUser();
        if(user!=null){
            if(user.getUser_id().equals
(GlobalVariables.user.getUser_id())){
                operation = CONSTANTS.OPERATION_DISABLED;
            }
        }
    }
}
}

```



```

    }

    if(DateUtil.isTodaysDate(getDate())){
        if(operation.equals(CONSTANTS.OPERATION_CREATE)){
            operation = CONSTANTS.OPERATION_DISABLED;
        } else
            if(operation.equals(CONSTANTS.OPERATION_UPDATE)){
                operation = CONSTANTS.OPERATION_VIEW;
            }
        }
    }
    return operation;
}
}
}

```

PdfDaySchedule.java

```

package orl.orscheduler.objects.or;

/**
 * Created by Raphael Lanuza on 5/3/2015.
 */
public class PdfDaySchedule {
    private DaySchedule privateDaySchedule;
    private DaySchedule charityDaySchedule;

    public DaySchedule getPrivateDaySchedule() {
        return privateDaySchedule;
    }

    public void setPrivateDaySchedule(DaySchedule
        privateDaySchedule) {
        this.privateDaySchedule = privateDaySchedule;
    }
}

```

```

public DaySchedule getCharityDaySchedule() {
    return charityDaySchedule;
}

public void setCharityDaySchedule(DaySchedule
    charityDaySchedule) {
    this.charityDaySchedule = charityDaySchedule;
}
}

```

Room.java

```

package orl.orscheduler.objects.or;

import java.util.List;

/**
 * Created by Raphael Lanuza on 4/25/2015.
 */
public class Room {
    private String roomNum;
    private List<Slot> slotList;

    public String getRoomNum() {
        return roomNum;
    }

    public void setRoomNum(String roomNum) {
        this.roomNum = roomNum;
    }

    public List<Slot> getSlotList() {
        return slotList;
    }
}

```

```

}

public void setSlotList(List<Slot> slotList) {
    this.slotList = slotList;
}

public Slot getSlot(String slotNumber){
    Slot slot = new Slot();
    for(Slot listSlot : this.getSlotList()){
        if(listSlot.getSlotNumber().equals(slotNumber)){
            slot = listSlot;
        }
    }
    return slot;
}

public Slot getSlot(int slotNumberInt){
    String slotNumber = String.valueOf(slotNumberInt);
    Slot slot = new Slot();
    for(Slot listSlot : this.getSlotList()){
        if(listSlot.getSlotNumber().equals(slotNumber)){
            slot = listSlot;
        }
    }
    return slot;
}

public Boolean isSlotExisting(String slotNumber){
    Boolean available = false;
    for(Slot slot : this.getSlotList()){
        if(slot.getSlotNumber().equals(slotNumber)){
            available = true;
        }
    }
}

```

```
    }
    return available;
}
}
```

ScheduleParameters.java

```
package orl.orscheduler.objects.or;

/**
 * Created by Raphael Lanuza on 4/25/2015.
 */
public class ScheduleParameters {
    private String roomNumber;
    private Slot slot;
    private String date;
    private String operation;

    public ScheduleParameters(){
    }

    public ScheduleParameters(String operation, String roomNumber,
        Slot slot, String date){
        this.roomNumber = roomNumber;
        this.slot = slot;
        this.date = date;
        this.operation = operation;
    }

    public String getRoomNumber() {
        return roomNumber;
    }

    public void setRoomNumber(String roomNumber) {
```

```

        this.roomNumber = roomNumber;
    }

    public String getDate() {
        return date;
    }

    public void setDate(String date) {
        this.date = date;
    }

    public Slot getSlot() {
        return slot;
    }

    public void setSlot(Slot slot) {
        this.slot = slot;
    }

    public String getOperation() {
        return operation;
    }

    public void setOperation(String operation) {
        this.operation = operation;
    }
}

```

Slot.java

```

package orl.orscheduler.objects.or;

import orl.orscheduler.objects.user.User;

```

```

/**
 * Created by Raphael Lanuza on 4/25/2015.
 */
public class Slot {
    private String or_id;
    private String slotNumber;
    private String patientName;
    private String diagnosis;
    private String procedure;
    private String surgicalTeam;
    private String otherDetails;
    private User user;

    public String getOr_id() {
        return or_id;
    }

    public void setOr_id(String or_id) {
        this.or_id = or_id;
    }

    public String getSlotNumber() {
        return slotNumber;
    }

    public void setSlotNumber(String slotNumber) {
        this.slotNumber = slotNumber;
    }

    public String getPatientName() {
        return patientName;
    }
}

```

```
public void setPatientName(String patientName) {
    this.patientName = patientName;
}

public String getDiagnosis() {
    return diagnosis;
}

public void setDiagnosis(String diagnosis) {
    this.diagnosis = diagnosis;
}

public String getProcedure() {
    return procedure;
}

public void setProcedure(String procedure) {
    this.procedure = procedure;
}

public String getSurgicalTeam() {
    return surgicalTeam;
}

public void setSurgicalTeam(String surgicalTeam) {
    this.surgicalTeam = surgicalTeam;
}

public String getOtherDetails() {
    return otherDetails;
}

public void setOtherDetails(String otherDetails) {
```

```

        this.otherDetails = otherDetails;
    }

    public User getUser() {
        return user;
    }

    public void setUser(User user) {
        this.user = user;
    }

    public void updateSlot(Slot slot){
        this.or_id = slot.getOr_id();
        this.patientName = slot.getPatientName();
        this.diagnosis = slot.getDiagnosis();
        this.procedure = slot.getProcedure();
        this.surgicalTeam = slot.getSurgicalTeam();
        this.otherDetails = slot.getOtherDetails();
        this.user = slot.getUser();
    }
}

```

Message.java

```

package orl.orscheduler.objects.user;

/**
 * Created by Raphael Lanuza on 5/6/2015.
 */
public class Message {

    private int    messageId;
    private User   sender;
    private String message;

```



```
private int    readFlag;
private String sentDate;

public int getMessageId() {
    return messageId;
}

public void setMessageId(int messageId) {
    this.messageId = messageId;
}

public User getSender() {
    return sender;
}

public void setSender(User sender) {
    this.sender = sender;
}

public String getMessage() {
    return message;
}

public void setMessage(String message) {
    this.message = message;
}

public int getReadFlag() {
    return readFlag;
}

public void setReadFlag(int readFlag) {
    this.readFlag = readFlag;
}
```

```

    }

    public String getSentDate() {
        return sentDate;
    }

    public void setSentDate(String sentDate) {
        this.sentDate = sentDate;
    }

    public Boolean isRead(){
        Boolean isRead = false;
        if(readFlag==1){
            isRead = true;
        }
        return isRead;
    }
}

```

User.java

```

package orl.orscheduler.objects.user;

import org.json.JSONException;
import org.json.JSONObject;

import java.util.List;

import orl.orscheduler.objects.global.CONSTANTS;
import orl.orscheduler.utilities.StringUtil;

/**
 * Created by Raphael Lanuza on 4/25/2015.
 */

```

```
public class User {
    private String user_id;
    private String firstname;
    private String lastname;
    private String usertype;
    private String username;
    private String password;
    private String fullname;
    private List<Message> messageList;

    public String getUser_id() {
        return user_id;
    }

    public void setUser_id(String user_id) {
        this.user_id = user_id;
    }

    public String getFirstname() {
        return firstname;
    }

    public void setFirstname(String firstname) {
        this.firstname = firstname;
    }

    public String getLastname() {
        return lastname;
    }

    public void setLastname(String lastname) {
        this.lastname = lastname;
    }
}
```

```

public String getUertype() {
    return uertype;
}

public void setUertype(String uertype) {
    this.uertype = uertype;
}

public String getUsername() {
    return username;
}

public void setUsername(String username) {
    this.username = username;
}

public String getPassword() {
    return password;
}

public void setPassword(String password) {
    this.password = password;
}

public void buildUserFromJson(JSONObject json) throws
    JSONException {
    String user_id = String.valueOf(json.getInt("user_id"));
    String firstname = json.getString("firstname");
    String lastname = json.getString("lastname");
    String uertype = json.getString("user_type");
    String username = json.getString("username");
}

```

```

        this.setUser_id(user_id);
        this.setFirstname(firstname);
        this.setLastname(lastname);
        this.setUser_type(usertype);
        this.setUsername(username);

        if(json.has("password")){
            String password = json.getString("password");
            this.setPassword(password);
        }

    }

    public String getORType(){
        return this.getUser_type().toUpperCase().equals
(CONSTANTS.USER_TYPE_CONSULTANT) ? CONSTANTS.OR_TYPE_PRIVATE :
CONSTANTS.OR_TYPE_CHARITY;
    }

    public String getFullname() {
        String fullName = "";
        if(!StringUtil.isNullOrEmpty(lastname)){
            fullName += lastname;
        }
        if(!StringUtil.isNullOrEmpty(lastname) &&
!StringUtil.isNullOrEmpty(firstname)){
            fullName += ", ";
        }
        if(!StringUtil.isNullOrEmpty(firstname)){
            fullName += firstname;
        }
        return fullName;
    }
}

```

```

    public List<Message> getMessageList() {
        return messageList;
    }

    public void setMessageList(List<Message> messageList) {
        this.messageList = messageList;
    }
}

```

ContactsUtil.java

```

package orl.orscheduler.utilities;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.List;

import orl.orscheduler.objects.global.CONSTANTS;
import orl.orscheduler.objects.global.GlobalVariables;
import orl.orscheduler.objects.user.User;

/**
 * Created by Raphael Lanuza on 5/6/2015.
 */
public class ContactsUtil {

    public static List<User> buildCotactsFromJson(JSONObject json)
        throws JSONException {
        List<User> userList = new ArrayList<User>();
        JSONArray jsonArray = json.getJSONArray("contacts");
    }
}

```

```

        User userAllConsultants = new User();
        userAllConsultants.setLastname("GROUP");
        userAllConsultants.setUserType
(CONSTANTS.USER_TYPE_CONSULTANT+"S");
        userList.add(userAllConsultants);
        User userAllResidents = new User();
        userAllResidents.setLastname("GROUP");
        userAllResidents.setUserType
(CONSTANTS.USER_TYPE_RESIDENT+"S");
        userList.add(userAllResidents);
        for(int i=0; i < jsonArray.length(); i++){
            JSONObject jsonObject = jsonArray.getJSONObject(i);
            User user = new User();
            user.setUsername(jsonObject.getString("USERNAME"));
            user.setFirstname(jsonObject.getString("FIRSTNAME"));
            user.setLastname(jsonObject.getString("LASTNAME"));
            user.setUserType(jsonObject.getString("USER_TYPE"));
            if(!user.getUsername().equals
(GlobalVariables.user.getUsername())){
                userList.add(user);
            }
        }
        return userList;
    }
}

```

DateUtil.java

```

package orl.orscheduler.utilities;

import android.os.Environment;

import java.text.ParseException;
import java.text.SimpleDateFormat;

```

```

import java.util.Calendar;
import java.util.Date;

/**
 * Created by Raphael Lanuza on 4/26/2015.
 */
public class DateUtil {

    private static SimpleDateFormat sdf = new
        SimpleDateFormat("MM/dd/yy");
    private static SimpleDateFormat sdf2 = new SimpleDateFormat("d
        MMM yyyy (EEEE)");
    private static SimpleDateFormat sdf3 = new
        SimpleDateFormat("MM/dd/yy - EEEE");
    private static SimpleDateFormat wkd = new
        SimpleDateFormat("EEEE");
    private static Calendar calendar = Calendar.getInstance();

    public static String getCurrentDayOfTheWeek() {
        return wkd.format(calendar.getTime()).toUpperCase();
    }

    public static String getCurrentDate() {
        return sdf.format(calendar.getTime()).toUpperCase();
    }

    public static String getMMDDYYDate(Date date) {
        return sdf.format(date);
    }

    public static String getMMDDYYDate(Calendar calendar){
        Date date = calendar.getTime();
        return sdf.format(date);
    }
}

```



```

}

public static String getDayOfTheWeek(Date date){
    return wkd.format(date);
}

public static String getDayOfTheWeek(Calendar calendar){
    return wkd.format(calendar.getTime());
}

public static String getDayOfTheWeek(String string) throws
    ParseException {
    Date date = getDateFromString(string);
    return wkd.format(date);
}

public static String getUppercaseDayOfTheWeek(Calendar
    calendar){
    return wkd.format(calendar.getTime()).toUpperCase();
}

public static Date getDateFromString(String date) throws
    ParseException {
    return sdf.parse(date);
}

public static String getDateForReportHeader(String string)
    throws ParseException {
    Date date = getDateFromString(string);
    return sdf2.format(date);
}

public static Boolean isTodaysDate(String string){

```

```

        return getCurrentDate().equals(string) ? true : false;
    }

    public static String convertMMDDYYDayToMMDDYY(String string)
        throws ParseException {
        Date date = sdf3.parse(string);
        return sdf.format(date);
    }

    public static String getPdfDateFilename(String string){
        return Environment.getExternalStorageDirectory() + "/" +
            string.replace("/", "_") + ".pdf";
    }
}

```

DayScheduleBuilder.java

```

package orl.orscheduler.utilities;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.text.ParseException;
import java.util.ArrayList;
import java.util.List;

import orl.orscheduler.objects.global.CONSTANTS;
import orl.orscheduler.objects.global.GlobalVariables;
import orl.orscheduler.objects.or.DaySchedule;
import orl.orscheduler.objects.or.PdfDaySchedule;
import orl.orscheduler.objects.or.Room;
import orl.orscheduler.objects.or.Slot;
import orl.orscheduler.objects.user.User;

```

```

/**
 * Created by Raphael Lanuza on 4/26/2015.
 */
public class DayScheduleBuilder {
    public static List<String> buildDateListFromJson(JSONObject
        json) throws JSONException {
        List<String> dates = new ArrayList<String>();
        JSONArray jsonArray = json.getJSONArray("dates");
        for(int i=0; i < jsonArray.length(); i++){
            JSONObject jsonObject = jsonArray.getJSONObject(i);
            String date = jsonObject.getString("date");
            dates.add(date);
        }
        return dates;
    }

    public static DaySchedule buildDayScheduleFromJson(JSONObject
        json, String date, String orType) throws JSONException,
        ParseException {
        DaySchedule daySchedule = new DaySchedule();
        JSONArray schedules = json.getJSONArray("scheds");
        daySchedule.setDate(date);
        daySchedule.setDayString(DateUtil.getDayOfTheWeek(date));
        daySchedule.setOrType(orType);
        List<Room> roomList = new ArrayList<Room>();
        for(int roomNumber = 1; roomNumber <= 3; roomNumber++){
            Room room = new Room();
            room.setRoomNum(String.valueOf(roomNumber));
            List<Slot> slotList = new ArrayList<Slot>();
            for(int slotNumber = 1; slotNumber <= 3; slotNumber++){
                Slot slot = new Slot();
                slot.setSlotNumber(String.valueOf(slotNumber));
            }
        }
    }
}

```

```

        slotList.add(slot);
    }
    room.setSlotList(slotList);
    roomList.add(room);
}

daySchedule.setRoomList(roomList);

for(int i =0; i < schedules.length(); i++){
    JSONObject sched = schedules.getJSONObject(i);
    String or_id = String.valueOf(sched.getInt("or_id"));
    String roomNumber =
        String.valueOf(sched.getInt("room_num"));
    String slotNumber =
        String.valueOf(sched.getInt("slot_num"));
    String ortype = sched.getString("or_type");
    String sched_date = sched.getString("sched_date");
    String patient_name = sched.getString("patient_name");
    String diagnosis = sched.getString("diagnosis");
    String procedure = sched.getString("procedure");
    String surgical_team = sched.getString("surgical_team");
    String other_details = sched.getString("other_details");
    String user_id =
        String.valueOf(sched.getInt("user_id"));
    if(sched_date.equals(date)){

        User user = new User();
        user.buildUserFromJson(sched.getJSONObject("user"));

        Slot slot = new Slot();
        slot.setOr_id(or_id);
        slot.setSlotNumber(slotNumber);
        slot.setPatientName(patient_name);
    }
}

```

```

        slot.setDiagnosis(diagnosis);
        slot.setProcedure(procedure);
        slot.setSurgicalTeam(surgical_team);
        slot.setOtherDetails(other_details);
        slot.setUser(user);

        daySchedule.getRoom(roomNumber)
.getSlot(slotNumber).updateSlot(slot);
    }
}
return daySchedule;
}

public static PdfDaySchedule
buildPdfDayScheduleFromJson(JSONObject json, String date)
throws JSONException, ParseException {
    PdfDaySchedule pdfDaySchedule = new PdfDaySchedule();
    DaySchedule privateDaySchedule = new DaySchedule();
    DaySchedule charityDaySchedule = new DaySchedule();
    JSONArray schedules = json.getJSONArray("scheds");

    privateDaySchedule.setDate(date);
    privateDaySchedule.setDayString(DateUtil.getDayOfTheWeek(date));
    privateDaySchedule.setOrType(CONSTANTS.OR_TYPE_PRIVATE);

    charityDaySchedule.setDate(date);
    charityDaySchedule.setDayString(DateUtil.getDayOfTheWeek(date));
    charityDaySchedule.setOrType(CONSTANTS.OR_TYPE_CHARITY);

    List<Room> privateRoomList = new ArrayList<Room>();
    for(int roomNumber = 1; roomNumber <= 3; roomNumber++){
        Room room = new Room();
        room.setRoomNum(String.valueOf(roomNumber));

```

```

List<Slot> slotList = new ArrayList<Slot>();
for(int slotNumber = 1; slotNumber <= 3; slotNumber++){
    Slot slot = new Slot();
    slot.setSlotNumber(String.valueOf(slotNumber));
    slotList.add(slot);
}
room.setSlotList(slotList);
privateRoomList.add(room);
}
privateDaySchedule.setRoomList(privateRoomList);

List<Room> charityRoomList = new ArrayList<Room>();
for(int roomNumber = 1; roomNumber <= 3; roomNumber++){
    Room room = new Room();
    room.setRoomNum(String.valueOf(roomNumber));
    List<Slot> slotList = new ArrayList<Slot>();
    for(int slotNumber = 1; slotNumber <= 3; slotNumber++){
        Slot slot = new Slot();
        slot.setSlotNumber(String.valueOf(slotNumber));
        slotList.add(slot);
    }
    room.setSlotList(slotList);
    charityRoomList.add(room);
}
charityDaySchedule.setRoomList(charityRoomList);

for(int i =0; i < schedules.length(); i++){
    JSONObject sched = schedules.getJSONObject(i);
    String or_id = String.valueOf(sched.getInt("or_id"));
    String roomNumber =
        String.valueOf(sched.getInt("room_num"));
    String slotNumber =
        String.valueOf(sched.getInt("slot_num"));

```

```

String orte = sched.getString("or_type");
String sched_date = sched.getString("sched_date");
String patient_name = sched.getString("patient_name");
String diagnosis = sched.getString("diagnosis");
String procedure = sched.getString("procedure");
String surgical_team = sched.getString("surgical_team");
String other_details = sched.getString("other_details");
String user_id =
    String.valueOf(sched.getInt("user_id"));

User user = new User();
user.buildUserFromJson(sched.getJSONObject("user"));

Slot slot = new Slot();
slot.setOr_id(orte);
slot.setSlotNumber(slotNumber);
slot.setPatientName(patient_name);
slot.setDiagnosis(diagnosis);
slot.setProcedure(procedure);
slot.setSurgicalTeam(surgical_team);
slot.setOtherDetails(other_details);
slot.setUser(user);
if(orte.equals(CONSTANTS.OR_TYPE_PRIVATE)){
    privateDaySchedule.getRoom(roomNumber)
.getSlot(slotNumber).updateSlot(slot);
} else if(orte.equals(CONSTANTS.OR_TYPE_CHARITY)){
    charityDaySchedule.getRoom(roomNumber)
.getSlot(slotNumber).updateSlot(slot);
}
}

pdfDaySchedule.setCharityDaySchedule(charityDaySchedule);
pdfDaySchedule.setPrivateDaySchedule(privateDaySchedule);
return pdfDaySchedule;

```

```
    }  
}
```

DialogUtil.java

```
package orl.orscheduler.utilities;  
  
import android.app.AlertDialog;  
import android.content.Context;  
import android.content.DialogInterface;  
import android.content.Intent;  
  
import orl.orscheduler.Welcome;  
  
/**  
 * Created by Raphael Lanuza on 5/2/2015.  
 */  
public class DialogUtil {  
    public static void showDialog(Context context, String message,  
        DialogInterface.OnClickListener onClickListener){  
        AlertDialog.Builder alert = new  
            AlertDialog.Builder(context);  
        alert.setTitle("Alert");  
        alert.setMessage(message);  
        alert.setPositiveButton("OK",onClickListener);  
        alert.setCancelable(false);  
        alert.show();  
    }  
  
    public static void showConfirmationDialog(Context context,  
        String message, DialogInterface.OnClickListener  
        onClickListenerOk, DialogInterface.OnClickListener  
        onClickListenerCancel){
```



```

        AlertDialog.Builder alert = new
            AlertDialog.Builder(context);
        alert.setTitle("Confirmation");
        alert.setMessage(message);
        alert.setPositiveButton("OK",onClickListenerOk);
        alert.setNegativeButton("Cancel", onClickListenerCancel);
        alert.show();
    }
}

```

JSONParser.java

```

package orl.orscheduler.utilities;

/**
 * Created by Raphael Lanuza on 4/24/2015.
 */
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.UnsupportedEncodingException;
import java.util.List;

import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.client.utils.URLEncodedUtils;
import org.apache.http.impl.client.DefaultHttpClient;
import org.json.JSONException;

```

```

import org.json.JSONObject;

import android.util.Log;

public class JSONParser {

    static InputStream is = null;
    static JSONObject jsonObj = null;
    static String json = "";

    // constructor
    public JSONParser() {

    }

    public JSONObject getJSONFromUrl(final String url) {

        // Making HTTP request
        try {
            // Construct the client and the HTTP request.
            DefaultHttpClient httpClient = new DefaultHttpClient();
            HttpPost httpPost = new HttpPost(url);

            // Execute the POST request and store the response
            locally.
            HttpResponse httpResponse =
                httpClient.execute(httpPost);
            // Extract data from the response.
            HttpEntity httpEntity = httpResponse.getEntity();
            // Open an inputStream with the data content.
            is = httpEntity.getContent();

        } catch (UnsupportedEncodingException e) {

```

```

        e.printStackTrace();
    } catch (ClientProtocolException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}

try {
    // Create a BufferedReader to parse through the
        inputStream.
    BufferedReader reader = new BufferedReader(new
        InputStreamReader(
            is, "iso-8859-1"), 8);
    // Declare a string builder to help with the parsing.
    StringBuilder sb = new StringBuilder();
    // Declare a string to store the JSON object data in
        string form.
    String line = null;

    // Build the string until null.
    while ((line = reader.readLine()) != null) {
        sb.append(line + "\n");
    }

    // Close the input stream.
    is.close();
    // Convert the string builder data to an actual string.
    json = sb.toString();
} catch (Exception e) {
    Log.e("Buffer Error", "Error converting result " +
        e.toString());
}

```

```

    // Try to parse the string to a JSON object
    try {
        jsonObj = new JSONObject(json);
    } catch (JSONException e) {
        Log.e("JSON Parser", "Error parsing data " +
            e.toString());
    }

    // Return the JSON Object.
    return jsonObj;
}

// function get json from Url
// by making HTTP POST or GET mehtod
public JSONObject makeHttpRequest(String url, String method,
    List<NameValuePair> params) {

    // Making HTTP request
    try {

        // check for request method
        if(method == "POST"){
            // request method is POST
            // defaultHttpClient
            DefaultHttpClient httpClient = new
                DefaultHttpClient();
            HttpPost httpPost = new HttpPost(url);
            httpPost.setEntity(new
                URLEncodedFormEntity(params));

            HttpResponse httpResponse =
                httpClient.execute(httpPost);

```

```

        HttpEntity httpEntity = httpResponse.getEntity();
        is = httpEntity.getContent();

    }else if(method == "GET"){
        // request method is GET
        DefaultHttpClient httpClient = new
            DefaultHttpClient();
        String paramString = URLEncodedUtils.format(params,
            "utf-8");
        url += "?" + paramString;
        HttpGet httpGet = new HttpGet(url);

        HttpResponse httpResponse =
            httpClient.execute(httpGet);
        HttpEntity httpEntity = httpResponse.getEntity();
        is = httpEntity.getContent();
    }

} catch (UnsupportedEncodingException e) {
    e.printStackTrace();
} catch (ClientProtocolException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}

try {
    BufferedReader reader = new BufferedReader(new
        InputStreamReader(
            is, "iso-8859-1"), 8);
    StringBuilder sb = new StringBuilder();
    String line = null;
    while ((line = reader.readLine()) != null) {

```

```

        sb.append(line + "\n");
    }
    is.close();
    json = sb.toString();
} catch (Exception e) {
    Log.e("Buffer Error", "Error converting result " +
        e.toString());
}

// try parse the string to a JSON object
try {
    jsonObj = new JSONObject(json);
} catch (JSONException e) {
    Log.e("JSON Parser", "Error parsing data " +
        e.toString());
}

// return JSON String
return jsonObj;

}
}

```

MessageBuilder.java

```

package orl.orscheduler.utilities;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.List;

```

```

import orl.orscheduler.objects.user.Message;
import orl.orscheduler.objects.user.User;

/**
 * Created by Raphael Lanuza on 5/7/2015.
 */
public class MessageBuilder {
    public static List<Message>
        buildMessageListFromjson(JSONObject json) throws
        JSONException {
        List<Message> messageList = new ArrayList<Message>();
        JSONArray jsonArray = json.getJSONArray("messages");

        for(int i=0; i < jsonArray.length(); i++){
            JSONObject jsonObject = jsonArray.getJSONObject(i);
            User user = new User();
            user.buildUserFromJson(jsonObject
                .getJSONObject("sender_user"));
            Message message = new Message();
            message.setMessageId(jsonObject.getInt("message_id"));
            message.setMessage(jsonObject.getString("message"));
            message.setReadFlag(jsonObject.getInt("read_flag"));
            message.setSentDate(jsonObject.getString("sent_date"));
            message.setSender(user);
            messageList.add(message);
        }
        return messageList;
    }
}

```

NetworkUtil.java

```

package orl.orscheduler.utilities;

```

```

import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;

import orl.orscheduler.Login;
import orl.orscheduler.objects.global.GlobalVariables;

/**
 * Created by Raphael Lanuza on 5/5/2015.
 */
public class NetworkUtil {
    public static boolean haveNetworkConnection(Context context) {
        boolean haveConnectedWifi = false;
        boolean haveConnectedMobile = false;

        ConnectivityManager cm = (ConnectivityManager)
            context.getSystemService(Context.CONNECTIVITY_SERVICE);
        NetworkInfo activeNetwork = cm.getActiveNetworkInfo();
        if( activeNetwork != null &&
            activeNetwork.isConnectedOrConnecting()){
            haveConnectedWifi = activeNetwork.getType() ==
                ConnectivityManager.TYPE_WIFI;
            haveConnectedMobile = activeNetwork.getType() ==
                ConnectivityManager.TYPE_MOBILE;
        }
        return haveConnectedWifi || haveConnectedMobile;
    }

    public static void logOutWhenNoConnection(Context context) {
        GlobalVariables.initializeAll();
        Intent i = new Intent(context, Login.class);
    }
}

```



```
        context.startActivity(i);
    }
}
```

ORPdfReportUtil.java

```
package orl.orscheduler.utilities;

/**
 * Created by Raphael Lanuza on 5/2/2015.
 */

import android.content.Context;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.net.Uri;
import android.os.Environment;

import com.itextpdf.text.Chunk;
import com.itextpdf.text.Document;
import com.itextpdf.text.DocumentException;
import com.itextpdf.text.Element;
import com.itextpdf.text.Font;
import com.itextpdf.text.PageSize;
import com.itextpdf.text.Paragraph;
import com.itextpdf.text.Rectangle;
import com.itextpdf.text.pdf.PdfPCell;
import com.itextpdf.text.pdf.PdfPTable;
import com.itextpdf.text.pdf.PdfWriter;
import com.itextpdf.text.Image;

import java.io.ByteArrayOutputStream;
import java.io.File;
```

```

import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.text.ParseException;
import java.util.Date;

import org.orscheduler.objects.global.CONSTANTS;
import org.orscheduler.objects.or.DaySchedule;
import org.orscheduler.objects.or.PdfDaySchedule;
import org.orscheduler.objects.or.Slot;

public class ORPdfReportUtil {
    private String path;
    private Document document;
    private static Font fontBold = new
        Font(Font.FontFamily.HELVETICA, 12, Font.BOLD);
    private static Font fontNormal = new
        Font(Font.FontFamily.HELVETICA, 12, Font.NORMAL);
    private static Font fontSmall = new
        Font(Font.FontFamily.HELVETICA, 10, Font.NORMAL);
    public static void createORPdfReport(PdfDaySchedule
        pdfDaySchedule, String date, Context context) throws
        DocumentException, ParseException, IOException {

        DaySchedule privateDaySchedule =
            pdfDaySchedule.getPrivateDaySchedule();
        DaySchedule charityDaySchedule =
            pdfDaySchedule.getCharityDaySchedule();
        String fpath = DateUtil.getPdfDateFilename(date);
        File file = new File(fpath);
        if (!file.exists()) {
            file.createNewFile();
        }
    }
}

```

```
}
```

```
Document document = new Document(PageSize.LETTER.rotate(),  
    30f, 30f, 30f, 30f);
```

```
PdfWriter.getInstance(document, new  
    FileOutputStream(file.getAbsolutePath()));  
document.open();
```

```
int private_maxRoomNumber = 0;
```

```
int private_maxSlotNumber = 0;
```

```
int charity_maxRoomNumber = 0;
```

```
int charity_maxSlotNumber = 0;
```

```
switch (DateUtil.getDayOfTheWeek(date).toUpperCase()){
```

```
    case CONSTANTS.DAY_MONDAY:
```

```
        private_maxRoomNumber = 2;
```

```
        private_maxSlotNumber = 3;
```

```
        charity_maxRoomNumber = 3;
```

```
        charity_maxSlotNumber = 3;
```

```
        break;
```

```
    case CONSTANTS.DAY_TUESDAY:
```

```
        charity_maxRoomNumber = 3;
```

```
        charity_maxSlotNumber = 3;
```

```
        break;
```

```
    case CONSTANTS.DAY_WEDNESDAY:
```

```
        private_maxRoomNumber = 1;
```

```
        private_maxSlotNumber = 3;
```

```
        charity_maxRoomNumber = 3;
```

```
        charity_maxSlotNumber = 3;
```

```
        break;
```

```
    case CONSTANTS.DAY_THURSDAY:
```

```
        charity_maxRoomNumber = 3;
```

```
        charity_maxSlotNumber = 3;
```

```
        break;
```

```
    case CONSTANTS.DAY_FRIDAY:
```

```

        private_maxRoomNumber = 3;
        private_maxSlotNumber = 1;
        charity_maxRoomNumber = 2;
        charity_maxSlotNumber = 2;
        break;
    default:
        break;
}

//Logo
InputStream ims = context.getAssets().open("pgh.png");
Bitmap bmp = BitmapFactory.decodeStream(ims);
ByteArrayOutputStream stream = new ByteArrayOutputStream();
bmp.compress(Bitmap.CompressFormat.PNG, 100, stream);
Image pgh = Image.getInstance(stream.toByteArray());
pgh.scaleAbsoluteHeight(115);
pgh.scaleAbsoluteWidth(130);
pgh.setAbsolutePosition(75,455);

InputStream ims2 = context.getAssets().open("orl.png");
Bitmap bmp2 = BitmapFactory.decodeStream(ims2);
ByteArrayOutputStream stream2 = new
    ByteArrayOutputStream();
bmp2.compress(Bitmap.CompressFormat.PNG, 100, stream2);
Image orl = Image.getInstance(stream2.toByteArray());
orl.scaleAbsoluteHeight(115);
orl.scaleAbsoluteWidth(115);
orl.setAbsolutePosition(600,455);

//Header
Paragraph header = new Paragraph();
header.setAlignment(Element.ALIGN_CENTER);

```

```

header.add(new Chunk("DEPARTMENT OF OTORHINOLARYNGOLOGY",
    fontNormal));
header.add(Chunk.NEWLINE);
header.add(new Chunk("PHILIPPINE GENERAL HOSPITAL",
    fontNormal));
header.add(Chunk.NEWLINE);
header.add(new Chunk("University of the Philippines",
    fontSmall));
header.add(Chunk.NEWLINE);
header.add(new Chunk("The Health Sciences Center Manila",
    fontSmall));
header.add(Chunk.NEWLINE);
header.add(new Chunk("Taft Avenue, Manila", fontSmall));
header.add(Chunk.NEWLINE);
header.add(new Chunk("PHIC Accredited Health Care
    Provider", fontSmall));
header.add(Chunk.NEWLINE);
header.add(Chunk.NEWLINE);
header.add(new Chunk("OPERATING ROOM MANAGEMENT TEAM",
    fontNormal));
header.add(Chunk.NEWLINE);
header.add(new Chunk(DateUtil.getDateForReportHeader
(privateDaySchedule.getDate()), fontNormal));
header.add(Chunk.NEWLINE);
header.add(Chunk.NEWLINE);
header.add(Chunk.NEWLINE);

//Table
PdfPTable table = new PdfPTable(6);

//Table - First Row
PdfPCell cell1 = new PdfPCell(new Paragraph("OR",
    fontNormal));

```

```

cell1.setHorizontalAlignment(Element.ALIGN_CENTER);
PdfPCell cell2 = new PdfPCell(new Paragraph("Name Of
    Patient", fontNormal));
cell2.setHorizontalAlignment(Element.ALIGN_CENTER);
PdfPCell cell3 = new PdfPCell(new Paragraph("Diagnosis",
    fontNormal));
cell3.setHorizontalAlignment(Element.ALIGN_CENTER);
PdfPCell cell4 = new PdfPCell(new Paragraph("Procedure",
    fontNormal));
cell4.setHorizontalAlignment(Element.ALIGN_CENTER);
PdfPCell cell5 = new PdfPCell(new Paragraph("Surgical
    Team", fontNormal));
cell5.setHorizontalAlignment(Element.ALIGN_CENTER);
PdfPCell cell6 = new PdfPCell(new Paragraph("Other
    Details", fontNormal));
cell6.setHorizontalAlignment(Element.ALIGN_CENTER);

table.addCell(cell1);
table.addCell(cell2);
table.addCell(cell3);
table.addCell(cell4);
table.addCell(cell5);
table.addCell(cell6);

//PAY OR
for(int roomNumber = 1; roomNumber <=
    private_maxRoomNumber; roomNumber++){
    PdfPCell roomCell = new PdfPCell(new
        Paragraph(roomNumber+ " PAY", fontNormal));
    roomCell.setRowspan(private_maxSlotNumber);
    roomCell.setHorizontalAlignment(Element.ALIGN_CENTER);
    roomCell.setVerticalAlignment(Element.ALIGN_MIDDLE);
    table.addCell(roomCell);

```

```

for(int slotNumber = 1; slotNumber
    <=private_maxSlotNumber; slotNumber++){
    Slot slot = privateDaySchedule.getRoom(roomNumber)
.getSlot(slotNumber);
    PdfPCell patientCell = new PdfPCell(new
        Paragraph(slot.getPatientName()==null ? " " :
            slot.getPatientName(), fontBold));
    patientCell.setHorizontalAlignment
(Element.ALIGN_CENTER);
    PdfPCell diagnosisCell = new PdfPCell(new
        Paragraph(slot.getDiagnosis()==null ? " " :
            slot.getDiagnosis(), fontBold));
    diagnosisCell.setHorizontalAlignment
(Element.ALIGN_CENTER);
    PdfPCell procedureCell = new PdfPCell(new
        Paragraph(slot.getProcedure()==null ? " " :
            slot.getProcedure(), fontBold));
    procedureCell.setHorizontalAlignment
(Element.ALIGN_CENTER);
    PdfPCell surgicalTeamCell = new PdfPCell(new
        Paragraph(slot.getSurgicalTeam()==null ? " " :
            slot.getSurgicalTeam(), fontBold));
    surgicalTeamCell.setHorizontalAlignment
(Element.ALIGN_CENTER);
    PdfPCell otherDetailsCell = new PdfPCell(new
        Paragraph(slot.getOtherDetails()==null ? " " :
            slot.getOtherDetails(), fontBold));
    otherDetailsCell.setHorizontalAlignment
(Element.ALIGN_CENTER);
    table.addCell(patientCell);
    table.addCell(diagnosisCell);
    table.addCell(procedureCell);
    table.addCell(surgicalTeamCell);

```

```

        table.addCell(otherDetailsCell);
    }
}

if(private_maxRoomNumber>0){
    PdfPCell spacingCell = new PdfPCell();
    spacingCell.setFixedHeight(40);
    spacingCell.setRowspan(2);
    spacingCell.setColspan(6);
    table.addCell(spacingCell);
}

//Charity
for(int roomNumber = 1; roomNumber <=
    charity_maxRoomNumber; roomNumber++){
    PdfPCell roomCell = new PdfPCell(new
        Paragraph(roomNumber+ " CHARITY", fontNormal));
    roomCell.setRowspan(charity_maxRoomNumber);
    roomCell.setHorizontalAlignment(Element.ALIGN_CENTER);
    roomCell.setVerticalAlignment(Element.ALIGN_MIDDLE);
    table.addCell(roomCell);
    for(int slotNumber = 1; slotNumber
        <=charity_maxSlotNumber; slotNumber++){
        Slot slot = charityDaySchedule.getRoom(roomNumber)
.getSlot(slotNumber);
        PdfPCell patientCell = new PdfPCell(new
            Paragraph(slot.getPatientName()==null ? " " :
                slot.getPatientName(), fontBold));
        patientCell.setHorizontalAlignment
(Element.ALIGN_CENTER);
        PdfPCell diagnosisCell = new PdfPCell(new
            Paragraph(slot.getDiagnosis()==null ? " " :
                slot.getDiagnosis(), fontBold));

```



```

        diagnosisCell.setHorizontalAlignment
(Element.ALIGN_CENTER);
        PdfPCell procedureCell = new PdfPCell(new
            Paragraph(slot.getProcedure()==null ? " " :
                slot.getProcedure(), fontBold));
        procedureCell.setHorizontalAlignment
(Element.ALIGN_CENTER);
        PdfPCell surgicalTeamCell = new PdfPCell(new
            Paragraph(slot.getSurgicalTeam()==null ? " " :
                slot.getSurgicalTeam(), fontBold));
        surgicalTeamCell.setHorizontalAlignment
(Element.ALIGN_CENTER);
        PdfPCell otherDetailsCell = new PdfPCell(new
            Paragraph(slot.getOtherDetails()==null ? " " :
                slot.getOtherDetails(), fontBold));
        otherDetailsCell.setHorizontalAlignment
(Element.ALIGN_CENTER);
        table.addCell(patientCell);
        table.addCell(diagnosisCell);
        table.addCell(procedureCell);
        table.addCell(surgicalTeamCell);
        table.addCell(otherDetailsCell);
    }
}

document.add(pgh);
document.add(ori);
document.add(header);
document.add(table);

document.close();
}
}

```

SchedDateList.java

```
package orl.orscheduler.utilities;

import java.lang.reflect.Array;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Calendar;
import java.util.Date;
import java.util.List;

import orl.orscheduler.objects.global.CONSTANTS;
import orl.orscheduler.objects.global.GlobalVariables;

/**
 * Created by Raphael Lanuza on 4/25/2015.
 */
public class SchedDateList {
    public static List<String> getDatesForViewing(int
        dateRangeSize, String orType){
        Calendar calendar = Calendar.getInstance();
        List<String> dates = new ArrayList<String>();
        for(int i = 0; i < dateRangeSize; i++){
            String day =
                DateUtil.getUppercaseDayOfTheWeek(calendar);
            if(orType.equals(CONSTANTS.OR_TYPE_CHARITY) &&
                Arrays.asList(CONSTANTS.VALID_CHARITYDAYS)
                .indexOf(day)>=0){
                dates.add(DateUtil.getMMDDYYDate(calendar));
            } else if(orType.equals(CONSTANTS.OR_TYPE_PRIVATE) &&
                Arrays.asList(CONSTANTS.VALID_ORDDAYS).indexOf(day)>=0){
                dates.add(DateUtil.getMMDDYYDate(calendar));
            }
            calendar.add(Calendar.DATE,1);
        }
    }
}
```

```
    }
    return dates;
}
}
```

StringUtil.java

```
package orl.orscheduler.utilities;

/**
 * Created by Raphael Lanuza on 5/9/2015.
 */
public class StringUtil {
    public static Boolean isNullOrEmpty(String string){
        Boolean isNull = false;
        Boolean isEmpty = false;
        if(string==null){
            isNull = true;
        } else if (string.isEmpty()){
            isEmpty = true;
        }
        return isNull || isEmpty;
    }
}
```

Validator.java

```
package orl.orscheduler.utilities;

import java.util.List;

/**
 * Created by Raphael Lanuza on 5/2/2015.
 */
public class Validator {
```

```

public static Boolean validateFieldList(List<String>fields){
    boolean isValid = true;
    for(String string: fields){
        if(string == null || string.isEmpty()){
            isValid = false;
        }
    }
    return isValid;
}
}

```

Url.java

```

package orl.orscheduler.web;

/**
 * Created by Raphael Lanuza on 4/25/2015.
 */
public class Url {
    public static final String DOMAIN = "http://192.168.43.42:80";
    //public static final String DOMAIN =
        "http://www.funwebdesigns.0fees.net/orlorscheduler";
    public static final String LOGIN = DOMAIN + "/login.php";
    public static final String REGISTER = DOMAIN + "/register.php";
    public static final String CREATESCCHEDULE = DOMAIN +
        "/createorsched.php";
    public static final String GETSCHEDULES = DOMAIN +
        "/getorsched.php";
    public static final String GETDISTINCTSCHEDATES = DOMAIN +
        "/getscheddates.php";
    public static final String GETMESSAGES = DOMAIN +
        "/getmessages.php";
    public static final String SENDMESSAGE = DOMAIN +
        "/sendmessage.php";
}

```

```
    public static final String GETCONTACTS = DOMAIN +
        "/getcontacts.php";
    public static final String GETUSERS = DOMAIN + "/getusers.php";
}
```

CreateSchedule.java

```
package orl.orscheduler;

/**
 * Created by Raphael Lanuza on 4/30/2015.
 */

import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.AsyncTask;
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import org.apache.http.NameValuePair;
import org.apache.http.message.BasicNameValuePair;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.List;
```

```

import orl.orscheduler.objects.global.CONSTANTS;
import orl.orscheduler.objects.global.GlobalVariables;
import orl.orscheduler.objects.or.Slot;
import orl.orscheduler.utilities.DialogUtil;
import orl.orscheduler.utilities.JSONParser;
import orl.orscheduler.utilities.NetworkUtil;
import orl.orscheduler.utilities.Validator;
import orl.orscheduler.web.Url;

public class CreateSchedule extends ActionBarActivity {

    private Context context;
    private TextView text_title, text_user, header1, header2;
    private EditText input_patientName, input_diagnosis,
        input_procedure, input_surgicalTeam, input_otherDetails;
    private Button button_submit, button_cancel, button_delete;
    private ProgressDialog pDialog;

    JSONParser jsonParser = new JSONParser();

    public String operation, room_num, slot_num, sched_date, or_id;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_create_schedule);
        context = this.getApplicationContext();

        header1 = (TextView) findViewById(R.id.header1);
        header2 = (TextView) findViewById(R.id.header2);

```

```

text_title = (TextView)
    findViewById(R.id.createSchedule_text_title);
input_patientName = (EditText)
    findViewById(R.id.createSchedule_input_patientName);
input_diagnosis = (EditText)
    findViewById(R.id.createSchedule_input_diagnosis);
input_procedure = (EditText)
    findViewById(R.id.createSchedule_input_procedure);
input_surgicalTeam = (EditText)
    findViewById(R.id.createSchedule_input_surgicalTeam);
input_otherDetails = (EditText)
    findViewById(R.id.createSchedule_input_otherDetails);

text_user = (TextView)
    findViewById(R.id.createSchedule_text_user);

button_submit = (Button)
    findViewById(R.id.createSchedule_button_submit);
button_submit.setOnClickListener(onClick);

button_cancel = (Button)
    findViewById(R.id.createSchedule_button_cancel);
button_cancel.setOnClickListener(onClick);

button_delete = (Button)
    findViewById(R.id.createSchedule_button_delete);
button_delete.setOnClickListener(onClick);

Slot slot = GlobalVariables.scheduleParameters.getSlot();
room_num =
    GlobalVariables.scheduleParameters.getRoomNumber();
slot_num = slot.getSlotNumber();

```

```

operation =
    GlobalVariables.scheduleParameters.getOperation();
sched_date = GlobalVariables.scheduleParameters.getDate();

if(operation.toUpperCase().equals(CONSTANTS.OPERATION_CREATE)
    || operation.equals(CONSTANTS.OPERATION_VIEW)){
    button_delete.setVisibility(View.GONE);
}

if(operation.toUpperCase().equals(CONSTANTS.OPERATION_VIEW)
    ||
    operation.toUpperCase().equals(CONSTANTS.OPERATION_UPDATE)){
    text_title.setText(operation.toUpperCase());

    input_patientName.setText(slot.getPatientName());
    input_diagnosis.setText(slot.getDiagnosis());
    input_procedure.setText(slot.getProcedure());
    input_surgicalTeam.setText(slot.getSurgicalTeam());
    input_otherDetails.setText(slot.getOtherDetails());
    or_id = slot.getOr_id();

    button_submit.setText(CONSTANTS.OPERATION_UPDATE);
text_user.setVisibility(View.VISIBLE);
    text_user.setText("created by: " +
        slot.getUser().getFirstname().toUpperCase() + " " +
        slot.getUser().getLastname().toUpperCase());

    if(operation.toUpperCase().equals
(CONSTANTS.OPERATION_VIEW)){
        input_patientName.setEnabled(false);
        input_diagnosis.setEnabled(false);
        input_procedure.setEnabled(false);
        input_surgicalTeam.setEnabled(false);

```



```

        input_otherDetails.setEnabled(false);
        button_submit.setEnabled(false);
    }
}

private View.OnClickListener onClick = new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(!NetworkUtil.haveNetworkConnection(CreateSchedule.this)){
            DialogUtil.showDialog(CreateSchedule.this, "No
                Internet Connection!", new
                DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int
                        which) {
                        NetworkUtil.logOutWhenNoConnection
(CreateSchedule.this);
                    }
                });
        } else {
            switch (v.getId()) {
                case R.id.createSchedule_button_submit:
                    if(validateRequiredFields()){
                        new AttemptCreate().execute();
                    } else {
                        Toast.makeText(CreateSchedule.this,
                            "Please enter all required fields
                            (*).", Toast.LENGTH_LONG).show();
                    }
                    break;
                case R.id.createSchedule_button_delete:

```

```

        DialogUtil.showConfirmationDialog(CreateSchedule.this,
            "Delete OR schedule?", new
            DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface
                    dialog, int which) {
                    operation =
                        CONSTANTS.OPERATION_DELETE;
                    new AttemptCreate().execute();
                }
            }, null);
        break;
    case R.id.createSchedule_button_cancel:
        onBackPressed();
        break;
    default:
        break;
    }
}

}

};

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if
    it is present.
    getMenuInflater().inflate(R.menu.menu_create_schedule,
        menu);
    return true;
}

@Override

```

```

public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so
    // long
    // as you specify a parent activity in
    // AndroidManifest.xml.
    int id = item.getItemId();

    //noinspection SimplifiableIfStatement
    if (id == R.id.action_settings) {
        return true;
    }

    return super.onOptionsItemSelected(item);
}

```

```

private boolean validateRequiredFields(){
    List<String> fields = new ArrayList<String>();
    String patient_name =
        input_patientName.getText().toString();
    String diagnosis = input_diagnosis.getText().toString();
    String procedure = input_procedure.getText().toString();
    String surgical_team =
        input_surgicalTeam.getText().toString();
    fields.add(patient_name);
    fields.add(diagnosis);
    fields.add(procedure);
    fields.add(surgical_team);
    return Validator.validateFieldList(fields);
}

```

```

class AttemptCreate extends AsyncTask<String, String,
    JSONObject> {

```

```

boolean failure = false;

@Override
protected void onPreExecute() {
    super.onPreExecute();
    pDialog = new ProgressDialog(CreateSchedule.this);
    pDialog.setMessage("Please wait...");
    pDialog.setIndeterminate(false);
    pDialog.setCancelable(true);
    pDialog.show();
}

@Override
protected JSONObject doInBackground(String... args) {
    String or_type = GlobalVariables.user.getORType();
    String patient_name =
        input_patientName.getText().toString();
    String diagnosis = input_diagnosis.getText().toString();
    String procedure = input_procedure.getText().toString();
    String surgical_team =
        input_surgicalTeam.getText().toString();
    String other_details =
        input_otherDetails.getText().toString();

    List<NameValuePair> params = new
        ArrayList<NameValuePair>();
    params.add(new BasicNameValuePair("operation",
        operation.toUpperCase()));
    params.add(new BasicNameValuePair("or_id", or_id));
    params.add(new BasicNameValuePair("room_num",
        room_num));
}

```

```

params.add(new BasicNameValuePair("slot_num",
    slot_num));
params.add(new BasicNameValuePair("or_type", or_type));
params.add(new BasicNameValuePair("sched_date",
    sched_date));
params.add(new BasicNameValuePair("patient_name",
    patient_name));
params.add(new BasicNameValuePair("diagnosis",
    diagnosis));
params.add(new BasicNameValuePair("procedure",
    procedure));
params.add(new BasicNameValuePair("surgical_team",
    surgical_team));
params.add(new BasicNameValuePair("other_details",
    other_details));
params.add(new BasicNameValuePair("user_id",
    GlobalVariables.user.getUser_id()));

return jsonParser.makeHttpRequest(Url.CREATESCHEDULE,
    "POST", params);
}

```

```

protected void onPostExecute(JSONObject json) {
    pDialog.dismiss();
    try {
        if(json!=null){
            int success =
                json.getInt(CONSTANTS.TAG_SUCCESS);
            String message =
                json.getString(CONSTANTS.TAG_MESSAGE);
            if (success == 1) {
                DialogUtil.showDialog(CreateSchedule.this,
                    message, new

```



```

import android.widget.Button;
import android.widget.ImageView;

/**
 * Created by Raphael Lanuza on 5/10/2015.
 */

public class Home extends Activity {

    private ImageView orl;
    private Button proceed;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home);

        orl = (ImageView) findViewById(R.id.orllogo);
        proceed = (Button) findViewById(R.id.proceed);

        proceed.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent myIntent = new Intent(view.getContext(),
                    Login.class);
                startActivityForResult(myIntent, 0);
                finish();
            }
        });
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {

```

```

        // Inflate the menu; this adds items to the action bar if
        // it is present.
        getMenuInflater().inflate(R.menu.menu_home, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so
        // long
        // as you specify a parent activity in
        // AndroidManifest.xml.
        int id = item.getItemId();
        if (id == R.id.action_settings) {
            return true;
        }
        return super.onOptionsItemSelected(item);
    }

    @Override
    public void onBackPressed() {
        Intent intent = new Intent(Intent.ACTION_MAIN);
        intent.addCategory(Intent.CATEGORY_HOME);
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(intent);
    }
}

```

Inbox.java

```

package orl.orscheduler;

/**
 * Created by Raphael Lanuza on 5/3/2015.

```



```

    */
import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.os.AsyncTask;
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.BaseAdapter;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import android.widget.TextView;

import org.apache.http.NameValuePair;
import org.apache.http.message.BasicNameValuePair;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import orl.orscheduler.objects.global.CONSTANTS;
import orl.orscheduler.objects.global.GlobalVariables;
import orl.orscheduler.objects.user.Message;
import orl.orscheduler.objects.user.User;
import orl.orscheduler.utilities.ContactsUtil;

```

```

import orl.orscheduler.utilities.DateUtil;
import orl.orscheduler.utilities.DialogUtil;
import orl.orscheduler.utilities.JSONParser;
import orl.orscheduler.utilities.MessageBuilder;
import orl.orscheduler.web.Url;

public class Inbox extends ActionBarActivity {

    private TextView header1, header2;
    private ListView listView_messages;
    private Context context;
    JSONParser jsonParser = new JSONParser();
    private ProgressDialog pDialog;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        GlobalVariables.viewClass = Inbox.class;
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_inbox);

        header1 = (TextView) findViewById(R.id.header1);
        header2 = (TextView) findViewById(R.id.header2);

        listView_messages = (ListView)
            findViewById(R.id.inbox_listView_messages);
        listView_messages.setOnItemClickListener(new
            AdapterView.OnItemClickListener() {
                @Override
                public void onItemClick(AdapterView<?> parent, View
                    view, int position, long id) {
                    Intent i = new Intent(Inbox.this,
                        SendMessage.class);
                    i.putExtra("message_index", position);
                }
            });
    }
}

```

```

        startActivity(i);
    }
});

    new AttemptGetMessages().execute();
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if
    // it is present.
    getMenuInflater().inflate(R.menu.menu_inbox, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so
    // long
    // as you specify a parent activity in
    // AndroidManifest.xml.
    int id = item.getItemId();

    //noinspection SimplifiableIfStatement
    if (id == R.id.action_settings) {
        return true;
    }

    return super.onOptionsItemSelected(item);
}

public class InboxAdapter extends BaseAdapter {

```

```

List<Message> messageList =
    GlobalVariables.user.getMessageList();

@Override
public int getCount() {
    // TODO Auto-generated method stub
    return messageList.size();
}

@Override
public Message getItem(int arg0) {
    // TODO Auto-generated method stub
    return messageList.get(arg0);
}

@Override
public long getItemId(int arg0) {
    // TODO Auto-generated method stub
    return arg0;
}

@Override
public View getView(int arg0, View arg1, ViewGroup arg2) {

    if(arg1==null)
    {
        LayoutInflater inflater = (LayoutInflater)
            Inbox.this.getSystemService
(Context.LAYOUT_INFLATER_SERVICE);
        arg1 = inflater.inflate(R.layout.message_item,
            arg2,false);
    }
}

```

```

        TextView textView_readFlag = (TextView)arg1.findViewById
(R.id.messageItem_text_readFlag);
        TextView textView_senderName =
            (TextView)arg1.findViewById
(R.id.messageItem_text_senderName);
        TextView textView_message = (TextView)arg1.findViewById
(R.id.messageItem_text_message);
        TextView textView_date = (TextView)arg1.findViewById
(R.id.messageItem_text_date);

        Message message = messageList.get(arg0);

        textView_readFlag.setText(message.isRead() ? " " :
            "New");
        textView_senderName.setText
(message.getSender().getFullname());
        textView_message.setText(message.getMessage());
        textView_date.setText(message.getSentDate());

        return arg1;
    }
}

```

```

class AttemptGetMessages extends AsyncTask<String, String,
    JSONObject> {

    boolean failure = false;

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
    }
}

```

```

@Override
protected JSONObject doInBackground(String... args) {
    List<NameValuePair> params = new
        ArrayList<NameValuePair>();
    params.add(new BasicNameValuePair("receiver_user_id",
        GlobalVariables.user.getUser_id()));
    return jsonParser.makeHttpRequest(Url.GETMESSAGES,
        "POST", params);
}

protected void onPostExecute(JSONObject json) {
    try {
        if(json!=null){
            int success =
                json.getInt(CONSTANTS.TAG_SUCCESS);
            String message =
                json.getString(CONSTANTS.TAG_MESSAGE);
            if (success == 1) {
                List<Message> messageList = new
                    ArrayList<Message>();
                messageList =
                    MessageBuilder.buildMessageListFromjson(json);
                GlobalVariables.user.setMessageList(messageList);
                InboxAdapter inboxAdapter = new
                    InboxAdapter();
                listView_messages.setAdapter(inboxAdapter);
            }else{
                DialogUtil.showDialog(Inbox.this, message,
                    null);
            }
        }
    }
    catch (JSONException e) {

```

```

        e.printStackTrace();
    }
}

@Override
public void onBackPressed() {
    Intent i = new Intent(Inbox.this,
        GlobalVariables.viewClass.equals(Inbox.class) ?
        Messaging.class : GlobalVariables.viewClass);
    startActivity(i);
}
}

```

Login.java

```

package orl.orscheduler;

/**
 * Created by Raphael Lanuza on 4/28/2015.
 */

import android.app.ProgressDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.net.Uri;
import android.os.AsyncTask;
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;

```

```

import android.widget.CompoundButton;
import android.widget.EditText;
import android.widget.RadioButton;
import android.widget.TextView;
import android.widget.Toast;

import org.apache.http.NameValuePair;
import org.apache.http.message.BasicNameValuePair;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.File;
import java.util.ArrayList;
import java.util.List;

import orl.orscheduler.objects.global.CONSTANTS;
import orl.orscheduler.objects.global.GlobalVariables;
import orl.orscheduler.objects.user.User;
import orl.orscheduler.utilities.DateUtil;
import orl.orscheduler.utilities.DialogUtil;
import orl.orscheduler.utilities.JSONParser;
import orl.orscheduler.utilities.NetworkUtil;
import orl.orscheduler.utilities.Validator;
import orl.orscheduler.web.Url;

public class Login extends ActionBarActivity implements
    View.OnClickListener {

    private TextView header1, header2;
    private EditText input_username, input_password;
    private Button button_login, button_register, button_view;
    private RadioButton radio_private, radio_charity;
    private Context context;

```



```

JSONParser jsonParser = new JSONParser();
private ProgressDialog pDialog;

@Override
protected void onCreate(Bundle savedInstanceState) {
    GlobalVariables.viewClass = Login.class;
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_log_in);

    header1 = (TextView) findViewById(R.id.header1);
    header2 = (TextView) findViewById(R.id.header2);

    input_username = (EditText)
        findViewById(R.id.login_input_username);
    input_password = (EditText)
        findViewById(R.id.login_input_password);

    button_login = (Button)
        findViewById(R.id.login_button_login);
    button_register = (Button)
        findViewById(R.id.login_button_register);
    button_view = (Button)
        findViewById(R.id.login_button_view);

    button_login.setOnClickListener(this);
    button_register.setOnClickListener(this);
    button_view.setOnClickListener(this);

    radio_charity = (RadioButton)
        findViewById(R.id.login_radio_charity);
    radio_private = (RadioButton)
        findViewById(R.id.login_radio_private);

```

```

        radio_private.setOnCheckedChangeListener(onCheck);
        radio_charity.setOnCheckedChangeListener(onCheck);

        radio_private.setChecked(true);

        context = this.getApplicationContext();
    }

private CompoundButton.OnCheckedChangeListener onCheck = new
    CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton buttonView,
        boolean isChecked) {
        if(isChecked){
            switch (buttonView.getId()) {
                case R.id.login_radio_charity:
                    radio_private.setChecked(false);
                    break;
                case R.id.login_radio_private:
                    radio_charity.setChecked(false);
                    break;
                default:
                    break;
            }
        }
    }
};

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_log_in, menu);
    return true;
}

```

```

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();

    if (id == R.id.action_settings) {
        return true;
    }

    return super.onOptionsItemSelected(item);
}

@Override
public void onClick(View v) {
    Intent i;
    if(!NetworkUtil.haveNetworkConnection(Login.this)){
        DialogUtil.showDialog(Login.this, "No Internet
        Connection!", null);
    } else {
        switch (v.getId()) {
            case R.id.login_button_login:
                if(validateRequiredFields()){
                    new AttemptLogin().execute();
                } else {
                    Toast.makeText(Login.this, "Please enter all
                    required fields (*)",
                    Toast.LENGTH_LONG).show();
                }
                break;
            case R.id.login_button_register:
                i = new Intent(this, Register.class);
                startActivity(i);
                break;
        }
    }
}

```

```

        case R.id.login_button_view:
            i = new Intent(this, ViewSchedules.class);
            i.putExtra("ORTYPE", radio_private.isChecked()
                ? CONSTANTS.OR_TYPE_PRIVATE :
                CONSTANTS.OR_TYPE_CHARITY);
            startActivity(i);
            break;
        default:
            break;
    }
}
}
}

```

```

class AttemptLogin extends AsyncTask<String, String,
    JSONObject> {

    boolean failure = false;

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        pDialog = new ProgressDialog(Login.this);
        pDialog.setMessage("Please wait...");
        pDialog.setIndeterminate(false);
        pDialog.setCancelable(true);
        pDialog.show();
    }

    @Override
    protected JSONObject doInBackground(String... args) {
        int success;
        String username = input_username.getText().toString();
    }
}

```

```

String password = input_password.getText().toString();

List<NameValuePair> params = new
    ArrayList<NameValuePair>();
params.add(new BasicNameValuePair("username",
    username));
params.add(new BasicNameValuePair("password",
    password));

return jsonParser.makeHttpRequest(Url.LOGIN, "POST",
    params);
}

protected void onPostExecute(JSONObject json) {
    pDialog.dismiss();
    try {
        if(json!=null){
            int success =
                json.getInt(CONSTANTS.TAG_SUCCESS);
            String message =
                json.getString(CONSTANTS.TAG_MESSAGE);
            if (success == 1) {
                User user = new User();
                user.buildUserFromJson(json.getJSONObject("user"));
                GlobalVariables.user = user;
                Intent i = new Intent(context,
                    Welcome.class);
                startActivity(i);
            }else{
                DialogUtil.showDialog(Login.this, message,
                    null);
            }
        }
    }
    catch (JSONException e) {

```

```

        e.printStackTrace();
    }
}

private boolean validateRequiredFields(){
    List<String> fields = new ArrayList<String>();
    String username = input_username.getText().toString();
    String password = input_password.getText().toString();
    fields.add(password);
    fields.add(username);
    return Validator.validateFieldList(fields);
}

@Override
public void onBackPressed() {
    Intent i = new Intent(Login.this, Home.class);
    startActivity(i);
}
}

```

Messaging.java

```

package orl.orscheduler;

/**
 * Created by Raphael Lanuza on 5/4/2015.
 */
import android.content.DialogInterface;
import android.content.Intent;
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;

```

```

import android.view.View;
import android.widget.Button;
import android.widget.TextView;

import orl.orscheduler.objects.global.GlobalVariables;
import orl.orscheduler.utilities.DialogUtil;
import orl.orscheduler.utilities.NetworkUtil;

public class Messaging extends ActionBarActivity {

    private Button button_create, button_inbox;
    private TextView header1, header2;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        GlobalVariables.viewClass = Messaging.class;
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_messaging);

        header1 = (TextView) findViewById(R.id.header1);
        header2 = (TextView) findViewById(R.id.header2);

        button_create = (Button)
            findViewById(R.id.messageHome_button_create);
        button_inbox = (Button)
            findViewById(R.id.messageHome_button_inbox);

        button_create.setOnClickListener(onClick);
        button_inbox.setOnClickListener(onClick);
    }
}

```

```

private View.OnClickListener onClick = new
    View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent i;
            if(!NetworkUtil.haveNetworkConnection(Messaging.this)){
                DialogUtil.showDialog(Messaging.this, "No Internet
                    Connection!", new
                    DialogInterface.OnClickListener() {
                        @Override
                        public void onClick(DialogInterface dialog, int
                            which) {
                            NetworkUtil.logOutWhenNoConnection(Messaging.this);
                        }
                    });
            } else {
                switch (v.getId()) {
                    case R.id.messageHome_button_create:
                        i = new Intent(Messaging.this,
                            SendMessage.class);
                        startActivity(i);
                        break;
                    case R.id.messageHome_button_inbox:
                        i = new Intent(Messaging.this, Inbox.class);
                        startActivity(i);
                        break;
                    default:
                        break;
                }
            }
        }
    };

```



```

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if
    // it is present.
    getMenuInflater().inflate(R.menu.menu_messaging, menu);
    return true;
}

```

```

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so
    // long
    // as you specify a parent activity in
    // AndroidManifest.xml.
    int id = item.getItemId();

    //noinspection SimplifiableIfStatement
    if (id == R.id.action_settings) {
        return true;
    }

    return super.onOptionsItemSelected(item);
}

```

```

@Override
public void onBackPressed() {
    Intent i = new Intent(Messaging.this,
        GlobalVariables.viewClass.equals(Messaging.class) ?
        Welcome.class : GlobalVariables.viewClass);
    startActivity(i);
}

```

```
}
```

PrintPdf.java

```
package orl.orscheduler;

import android.app.ProgressDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Environment;
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.Spinner;

import org.apache.http.NameValuePair;
import org.apache.http.message.BasicNameValuePair;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.File;
import java.text.ParseException;
import java.util.ArrayList;
import java.util.List;

import orl.orscheduler.objects.global.CONSTANTS;
```

```

import orl.orscheduler.objects.global.GlobalVariables;
import orl.orscheduler.objects.or.PdfDaySchedule;
import orl.orscheduler.objects.user.User;
import orl.orscheduler.utilities.DateUtil;
import orl.orscheduler.utilities.DayScheduleBuilder;
import orl.orscheduler.utilities.DialogUtil;
import orl.orscheduler.utilities.JSONParser;
import orl.orscheduler.utilities.NetworkUtil;
import orl.orscheduler.utilities.ORPdfReportUtil;
import orl.orscheduler.utilities.Validator;
import orl.orscheduler.web.Url;
/**
 * Created by Raphael Lanuza on 5/2/2015.
 */
public class PrintPdf extends ActionBarActivity {

    private Spinner spinner_date;
    private Button button_print, button_cancel;
    private ProgressDialog pDialog;
    JSONParser jsonParser = new JSONParser();
    private Context context;
    private String sched_date;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        GlobalVariables.viewClass = PrintPdf.class;
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_print_pdf);

        spinner_date = (Spinner)
            findViewById(R.id.print_spinner_date);
        button_print = (Button)
            findViewById(R.id.print_button_print);

```

```

button_cancel = (Button)
    findViewById(R.id.print_button_cancel);
button_print.setEnabled(false);
String selectedDate = "";

button_print.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v) {
        if(!NetworkUtil.haveNetworkConnection(PrintPdf.this)){
            DialogUtil.showDialog(PrintPdf.this, "No
                Internet Connection!", new
                DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog,
                        int which) {
                        NetworkUtil.logOutWhenNoConnection
(PrintPdf.this);
                    }
                });
        } else {
            if(validateRequiredFields()){
                new AttemptGetSchedules().execute();
            }
        }
    }
});

button_cancel.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        onBackPressed();
    }
});

```

```

        }
    });

    context = this.getApplicationContext();

    new AttemptGetDates().execute();
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if
    it is present.
    getMenuInflater().inflate(R.menu.menu_print_pdf, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so
    long
    // as you specify a parent activity in
    AndroidManifest.xml.
    int id = item.getItemId();

    //noinspection SimplifiableIfStatement
    if (id == R.id.action_settings) {
        return true;
    }

    return super.onOptionsItemSelected(item);
}

```

```

class AttemptGetDates extends AsyncTask<String, String,
    JSONObject> {

    boolean failure = false;

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
    }

    @Override
    protected JSONObject doInBackground(String... args) {
        return
            jsonParser.makeHttpRequest(Url.GETDISTINCTSCHEDATES,
                "POST", new ArrayList<NameValuePair>());
    }

    protected void onPostExecute(JSONObject json) {
        try {
            if(json!=null){
                int success =
                    json.getInt(CONSTANTS.TAG_SUCCESS);
                String message =
                    json.getString(CONSTANTS.TAG_MESSAGE);
                if (success == 1) {
                    List<String> dates = new ArrayList<String>();
                    dates =
                        DayScheduleBuilder.buildDateListFromJson(json);
                    ArrayAdapter<String> dataAdapter = new
                        ArrayAdapter<String>(PrintPdf.this,
                            android.R.layout.simple_spinner_item,
                                dates);
                    dataAdapter.setDropDownViewResource
(
                        android.R.layout.simple_spinner_dropdown_item);

```



```

String date = spinner_date.getSelectedItem().toString();
List<NameValuePair> params = new
    ArrayList<NameValuePair>();
try {
    sched_date =
        DateUtil.convertMMDDYYDayToMMDDYY(date);
    params.add(new BasicNameValuePair("sched_date",
        sched_date));
} catch (ParseException e) {
    e.printStackTrace();
}
return jsonParser.makeHttpRequest(Url.GETSCHEDULES,
    "POST",params);
}

protected void onPostExecute(JSONObject json) {
    pDialog.dismiss();
    try {
        if(json!=null){
            int success =
                json.getInt(CONSTANTS.TAG_SUCCESS);
            String message =
                json.getString(CONSTANTS.TAG_MESSAGE);
            if (success == 1) {
                PdfDaySchedule pdfDaySchedule = new
                    PdfDaySchedule();
                pdfDaySchedule =
                    DayScheduleBuilder.buildPdfDayScheduleFromJson
(json, sched_date);
                ORPdfReportUtil.createORPdfReport(pdfDaySchedule,
                    sched_date, PrintPdf.this);
                DialogUtil.showDialog(PrintPdf.this,
                    DateUtil.getPdfDateFilename(sched_date) +
                    " CREATED!",/*null*/ new

```



```

DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface
        dialog, int which) {
        Intent intent = new
            Intent(Intent.ACTION_VIEW);
        String fpath =
            DateUtil.getPdfDateFilename(sched_date);
        File file = new File(fpath);
        intent.setDataAndType(
            Uri.fromFile(file),
            "application/pdf" );
        intent.setFlags
(Intent.FLAG_ACTIVITY_NO_HISTORY);
        startActivity(intent);
    }
};
}else{
    DialogUtil.showDialog(PrintPdf.this,
        message, null);
}
}
} catch (Exception e){
    DialogUtil.showDialog(PrintPdf.this,
        e.getMessage(), null);
}
}

@Override
public void onBackPressed() {

```

```

        Intent i = new Intent(PrintPdf.this,
            GlobalVariables.viewClass.equals(PrintPdf.class) ?
            Welcome.class : GlobalVariables.viewClass);
        startActivity(i);
    }

    private boolean validateRequiredFields(){
        List<String> fields = new ArrayList<String>();
        String date = spinner_date.getSelectedItem() != null ?
            spinner_date.getSelectedItem().toString() : "";
        fields.add(date);
        return Validator.validateFieldList(fields);
    }
}

```

Register.java

```

package orl.orscheduler;

/**
 * Created by Raphael Lanuza on 4/27/2015.
 */

import android.app.ProgressDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.AsyncTask;
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.CompoundButton;
import android.widget.EditText;

```

```

import android.widget.RadioButton;
import android.widget.TextView;
import android.widget.Toast;

import org.apache.http.NameValuePair;
import org.apache.http.message.BasicNameValuePair;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.List;

import orl.orscheduler.objects.global.CONSTANTS;
import orl.orscheduler.objects.global.GlobalVariables;
import orl.orscheduler.objects.user.User;
import orl.orscheduler.utilities.DialogUtil;
import orl.orscheduler.utilities.JSONParser;
import orl.orscheduler.utilities.NetworkUtil;
import orl.orscheduler.utilities.Validator;
import orl.orscheduler.web.Url;

public class Register extends ActionBarActivity {

    private TextView text_title, text_firstName, text_lastName,
        text_userType, text_username, text_password, header1,
        header2;
    private EditText input_firstName, input_lastName,
        input_username, input_password;
    private RadioButton radio_residents, radio_consultants;
    private Button button_register, button_cancel, button_delete;
    private Context context;
    JSONParser jsonParser = new JSONParser();

```

```

private ProgressDialog pDialog;
private String operation;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_register);

    header1 = (TextView) findViewById(R.id.header1);
    header2 = (TextView) findViewById(R.id.header2);

    text_title = (TextView)
        findViewById(R.id.register_text_title);
    text_firstName = (TextView)
        findViewById(R.id.register_text_firstName);
    text_lastName = (TextView)
        findViewById(R.id.register_text_lastName);
    text_userType = (TextView)
        findViewById(R.id.register_text_userType);
    text_username = (TextView)
        findViewById(R.id.register_text_username);
    text_password = (TextView)
        findViewById(R.id.register_text_password);

    input_firstName = (EditText)
        findViewById(R.id.register_input_firstName);
    input_lastName = (EditText)
        findViewById(R.id.register_input_lastName);
    radio_residents = (RadioButton)
        findViewById(R.id.register_radio_residents);
    radio_consultants = (RadioButton)
        findViewById(R.id.register_radio_consultants);

```

```

input_username = (EditText)
    findViewById(R.id.register_input_username);
input_password = (EditText)
    findViewById(R.id.register_input_password);

radio_residents.setChecked(true);
radio_residents.setOnCheckedChangeListener(onCheck);

radio_consultants.setOnCheckedChangeListener(onCheck);

button_register = (Button)
    findViewById(R.id.register_button_register);
button_cancel= (Button)
    findViewById(R.id.register_button_cancel);
button_delete= (Button)
    findViewById(R.id.register_button_delete);
button_register.setOnClickListener(onClick);
button_cancel.setOnClickListener(onClick);
button_delete.setOnClickListener(onClick);
context = this.getApplicationContext();

User user = GlobalVariables.user;
operation = user !=null && user.getUser_id()!=null ?
    CONSTANTS.OPERATION_UPDATE : CONSTANTS.OPERATION_CREATE;
if(operation.equals(CONSTANTS.OPERATION_UPDATE)){
    text_title.setText(CONSTANTS.OPERATION_UPDATE);
    button_register.setText(CONSTANTS.OPERATION_UPDATE);
    text_firstName.setVisibility(View.GONE);
    input_firstName.setVisibility(View.GONE);
    text_lastName.setVisibility(View.GONE);
    input_lastName.setVisibility(View.GONE);
    text_username.setVisibility(View.GONE);
    input_username.setVisibility(View.GONE);

```

```

        text_userType.setVisibility(View.GONE);
        radio_consultants.setVisibility(View.GONE);
        radio_residents.setVisibility(View.GONE);
        if(user.getUsertype().equals(CONSTANTS.USER_TYPE_ADMIN)){
            button_delete.setVisibility(View.VISIBLE);
            input_password.setText(GlobalVariables.userList.get
(getIntent().getExtras().getInt("user_index")).getPassword());
        } else {
            input_password.setText(user.getPassword());
        }
    }
}

private View.OnClickListener onClick = new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent i;
        if(!NetworkUtil.haveNetworkConnection(Register.this)){
            DialogUtil.showDialog(Register.this, "No Internet
            Connection!", new
            DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int
                which) {
                    NetworkUtil.logoutWhenNoConnection(Register.this);
                }
            });
        } else {
            switch (v.getId()) {
                case R.id.register_button_register:
                    if(validateRequiredFields()){

```

```

        new AttemptRegister().execute();
    } else {
        Toast.makeText(Register.this, "Please
            enter all required fields (*)",
            Toast.LENGTH_LONG).show();
    }
    break;
case R.id.register_button_cancel:
    onBackPressed();
    break;
case R.id.register_button_delete:
    DialogUtil.showConfirmationDialog(Register.this,
        "Delete User Account?", new
        DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface
                dialog, int which) {
                operation =
                    CONSTANTS.OPERATION_DELETE;
                new AttemptDelete().execute();
            }
        }, null);
    break;
default:
    break;
    }
}
}
};

```

```

private boolean validateRequiredFields(){
    List<String> fields = new ArrayList<String>();

```

```

if(operation.equals(CONSTANTS.OPERATION_CREATE)){
    String firstname = input_firstName.getText().toString();
    String lastname = input_lastName.getText().toString();
    String username = input_username.getText().toString();
    fields.add(firstname);
    fields.add(lastname);
    fields.add(username);
}
String password = input_password.getText().toString();
fields.add(password);
return Validator.validateFieldList(fields);
}

```

```

private CompoundButton.OnCheckedChangeListener onCheck = new
CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton buttonView,
        boolean isChecked) {
        if(isChecked){
            switch (buttonView.getId()) {
                case R.id.register_radio_consultants:
                    radio_residents.setChecked(false);
                    break;
                case R.id.register_radio_residents:
                    radio_consultants.setChecked(false);
                    break;
                default:
                    break;
            }
        }
    }
};

```



```

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_register, menu);
    return true;
}

```

```

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();

    if (id == R.id.action_settings) {
        return true;
    }

    return super.onOptionsItemSelected(item);
}

```

```

class AttemptRegister extends AsyncTask<String, String,
    JSONObject> {

    boolean failure = false;

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        pDialog = new ProgressDialog(Register.this);
        pDialog.setMessage("Please wait...");
        pDialog.setIndeterminate(false);
        pDialog.setCancelable(true);
        pDialog.show();
    }

    @Override

```

```

protected JSONObject doInBackground(String... args) {
    int success;
    String firstname = input_firstName.getText().toString();
    String lastname = input_lastName.getText().toString();
    String username = input_username.getText().toString();
    String password = input_password.getText().toString();
    String usertype = radio_consultants.isChecked() ?
        CONSTANTS.USER_TYPE_CONSULTANT :
        CONSTANTS.USER_TYPE_RESIDENT;
    List<NameValuePair> params = new
        ArrayList<NameValuePair>();
    params.add(new BasicNameValuePair("user_id",
        operation.equals(CONSTANTS.OPERATION_UPDATE) ?
        (GlobalVariables.user.getUsertype().equals
(CONSTANTS.USER_TYPE_ADMIN) ?
        GlobalVariables.userList.get(getIntent().getExtras()
.getInt("user_index")).getUser_id() :
        GlobalVariables.user.getUser_id()) : null));
    params.add(new BasicNameValuePair("firstname",
        firstname));
    params.add(new BasicNameValuePair("lastname",
        lastname));
    params.add(new BasicNameValuePair("user_type",
        usertype));
    params.add(new BasicNameValuePair("username",
        username));
    params.add(new BasicNameValuePair("password",
        password));
    params.add(new BasicNameValuePair("operation",
        operation));

    return jsonParser.makeHttpRequest(Url.REGISTER, "POST",
        params);
}

```

```

    }

    protected void onPostExecute(JSONObject json) {
        pDialog.dismiss();
        try {
            if(json!=null){
                int success =
                    json.getInt(CONSTANTS.TAG_SUCCESS);
                String message =
                    json.getString(CONSTANTS.TAG_MESSAGE);
                if (success == 1) {
                    if(operation.equals(CONSTANTS.OPERATION_UPDATE)
                        &&
                        !GlobalVariables.user.getUsertype().equals
(CONSTANTS.USER_TYPE_ADMIN)){
                            GlobalVariables.user.setPassword
(input_password.getText().toString());
                        }
                    DialogUtil.showDialog(Register.this,
                        message, new
                        DialogInterface.OnClickListener() {
                            @Override
                            public void onClick(DialogInterface
                                dialog, int which) {
                                    Intent i = new Intent(context,
                                        operation.equals
(CONSTANTS.OPERATION_UPDATE) ?
                                        (GlobalVariables.user.getUsertype().equals
(CONSTANTS.USER_TYPE_ADMIN) ? UserManagement.class :
                                        Welcome.class) : Login.class);
                                    startActivity(i);
                                }
                            }
                }
            }
        }
    }
}

```

```

        );

    }else{
        DialogUtil.showDialog(Register.this,
            message, null);
    }
}
} catch (JSONException e) {
    e.printStackTrace();
}
}

}

class AttemptDelete extends AsyncTask<String, String,
    JSONObject> {

    boolean failure = false;

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        pDialog = new ProgressDialog(Register.this);
        pDialog.setMessage("Please wait...");
        pDialog.setIndeterminate(false);
        pDialog.setCancelable(true);
        pDialog.show();
    }

    @Override
    protected JSONObject doInBackground(String... args) {
        List<NameValuePair> params = new
            ArrayList<NameValuePair>();

```

```

        params.add(new BasicNameValuePair("user_id",
            GlobalVariables.userList.get(getIntent().getExtras()
                .getInt("user_index")).getUser_id()));
        params.add(new BasicNameValuePair("operation",
            CONSTANTS.OPERATION_DELETE));
        return jsonParser.makeHttpRequest(Url.REGISTER, "POST",
            params);
    }

```

```

protected void onPostExecute(JSONObject json) {
    pDialog.dismiss();
    try {
        if(json!=null){
            int success =
                json.getInt(CONSTANTS.TAG_SUCCESS);
            String message =
                json.getString(CONSTANTS.TAG_MESSAGE);
            if (success == 1) {
                DialogUtil.showDialog(Register.this,
                    message, new
                    DialogInterface.OnClickListener() {
                        @Override
                        public void
                            onClick(DialogInterface
                                dialog, int which) {
                                Intent i = new
                                    Intent(context,
                                        UserManagement.class);
                                startActivity(i);
                            }
                        }
                    );
            }else{

```



```

import org.apache.http.NameValuePair;
import org.apache.http.message.BasicNameValuePair;
import org.json.JSONException;
import org.json.JSONObject;
import org.w3c.dom.Text;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import orl.orscheduler.objects.global.CONSTANTS;
import orl.orscheduler.objects.global.GlobalVariables;
import orl.orscheduler.objects.user.Message;
import orl.orscheduler.objects.user.User;
import orl.orscheduler.utilities.ContactsUtil;
import orl.orscheduler.utilities.DialogUtil;
import orl.orscheduler.utilities.JSONParser;
import orl.orscheduler.utilities.NetworkUtil;
import orl.orscheduler.web.Url;

public class SendMessage extends ActionBarActivity {

    private Spinner spinner_receiver;
    private TextView text_userName, text_fullName, text_title,
        header1, header2;
    private EditText input_message;
    private Button button_send, button_cancel, button_delete;

    private Context context;
    JSONParser jsonParser = new JSONParser();

```

```

private ProgressDialog pDialog;
private Message message;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_send_message);

    header1 = (TextView) findViewById(R.id.header1);
    header2 = (TextView) findViewById(R.id.header2);

    spinner_receiver = (Spinner)
        findViewById(R.id.sendMessage_spinner_receiver);
    input_message = (EditText)
        findViewById(R.id.sendMessage_editText_message);
    button_send = (Button)
        findViewById(R.id.sendMessage_button_send);
    button_cancel = (Button)
        findViewById(R.id.sendMessage_button_cancel);
    button_delete = (Button)
        findViewById(R.id.sendMessage_button_delete);
    button_cancel.setOnClickListener(onClick);
    button_send.setOnClickListener(onClick);
    button_delete.setOnClickListener(onClick);

    text_fullName = (TextView)
        findViewById(R.id.sendMessage_text_receiverFullname);
    text_userName = (TextView)
        findViewById(R.id.sendMessage_text_receiverUsername);
    text_title = (TextView)
        findViewById(R.id.sendMessage_text_title);
    if(getIntent().getExtras()!=null){
        text_fullName.setVisibility(View.VISIBLE);
    }
}

```



```

        text_userName.setVisibility(View.VISIBLE);
        button_delete.setVisibility(View.VISIBLE);
        spinner_receiver.setVisibility(View.GONE);
        button_send.setVisibility(View.GONE);

        text_title.setText("Message");

        int message_index =
            getIntent().getExtras().getInt("message_index");
        message = GlobalVariables.user.getMessageList()
.get(message_index);

        text_fullName.setText(message.getSender().getFullname());
        text_userName.setText(message.getSender().getUsername());
        input_message.setText(message.getMessage());
        input_message.setFocusable(false);
    } else {
        new AttemptGetContacts().execute();
    }
}

private View.OnClickListener onClick = new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent i;
        if(!NetworkUtil.haveNetworkConnection(SendMessage.this)){
            DialogUtil.showDialog(SendMessage.this, "No
                Internet Connection!", new
                DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int
                        which) {

```

```

        NetworkUtil.logOutWhenNoConnection
(SendMessage.this);
    }
    });
} else {
    switch (v.getId()) {
        case R.id.sendMessage_button_send:
            if(spinner_receiver
.getSelectedItemPosition()>2){
                new AttemptSendMessage().execute();
            } else {
                new AttemptSendGroupMessage().execute();
            }
            break;
        case R.id.sendMessage_button_cancel:
            onBackPressed();
            break;
        case R.id.sendMessage_button_delete:
            DialogUtil.showConfirmationDialog
(SendMessage.this, "Delete message?", new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface
dialog, int which) {
                    new AttemptDeleteMessage().execute();
                }
            }, null);
            break;
        default:
            break;
    }
}
}

```

```

    }
};

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if
    // it is present.
    getMenuInflater().inflate(R.menu.menu_send_message, menu);
    return true;
}

```

```

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so
    // long
    // as you specify a parent activity in
    // AndroidManifest.xml.
    int id = item.getItemId();

    //noinspection SimplifiableIfStatement
    if (id == R.id.action_settings) {
        return true;
    }

    return super.onOptionsItemSelected(item);
}

```

```

class AttemptGetContacts extends AsyncTask<String, String,
    JSONObject> {

    boolean failure = false;

```

```

@Override
protected void onPreExecute() {
    super.onPreExecute();
    pDialog = new ProgressDialog(SendMessage.this);
    pDialog.setMessage("Please wait...");
    pDialog.setIndeterminate(false);
    pDialog.setCancelable(true);
    pDialog.show();
}

@Override
protected JSONObject doInBackground(String... args) {

    return jsonParser.makeHttpRequest(Url.GETCONTACTS,
        "POST", new ArrayList<NameValuePair>());
}

protected void onPostExecute(JSONObject json) {
    pDialog.dismiss();
    try {
        if(json!=null){
            int success =
                json.getInt(CONSTANTS.TAG_SUCCESS);
            String message =
                json.getString(CONSTANTS.TAG_MESSAGE);
            if (success == 1) {
                List<User> contactList = new
                    ArrayList<User>();
                contactList =
                    ContactsUtil.buildCotactsFromJson(json);
                ArrayList<Map<String, String>> contactArray
                    = new ArrayList<Map<String, String>>();
                for(User contact : contactList){

```



```

protected void onPreExecute() {
    super.onPreExecute();
}

@Override
protected JSONObject doInBackground(String... args) {
    List<NameValuePair> params = new
        ArrayList<NameValuePair>();
    String sender_user_id =
        GlobalVariables.user.getUser_id();
    Map<String, String> stringStringMap = (Map<String,
        String>) spinner_receiver.getSelectedItemAt();

    String receiver_user_name =
        GlobalVariables.contactList.get
(spinner_receiver.getSelectedItemPosition()).getUsername();
    String message = input_message.getText().toString();
    params.add(new BasicNameValuePair("sender_user_id",
        sender_user_id));
    params.add(new BasicNameValuePair("receiver_user_name",
        receiver_user_name));
    params.add(new BasicNameValuePair("message", message));

    return jsonParser.makeHttpRequest(Url.SENDMESSAGE,
        "POST", params);
}

protected void onPostExecute(JSONObject json) {
    try {
        if(json!=null){
            int success =
                json.getInt(CONSTANTS.TAG_SUCCESS);

```



```

protected void onPreExecute() {
    super.onPreExecute();
}

@Override
protected JSONObject doInBackground(String... args) {
    List<NameValuePair> params = new
        ArrayList<NameValuePair>();
    String sender_user_id =
        GlobalVariables.user.getUser_id();
    Map<String, String> stringStringMap = (Map<String,
        String>) spinner_receiver.getSelectedItemAt();

    String receiver_user_type =
        GlobalVariables.contactList.get
(spinner_receiver.getSelectedItemPosition()).getUsertype();
    String message = input_message.getText().toString();
    params.add(new BasicNameValuePair("sender_user_id",
        sender_user_id));
    params.add(new BasicNameValuePair("receiver_user_type",
        receiver_user_type));
    params.add(new BasicNameValuePair("message", message));

    return jsonParser.makeHttpRequest(Url.SENDMESSAGE,
        "POST", params);
}

protected void onPostExecute(JSONObject json) {
    try {
        if(json!=null){
            int success =
                json.getInt(CONSTANTS.TAG_SUCCESS);

```



```

protected void onPreExecute() {
    super.onPreExecute();
}

@Override
protected JSONObject doInBackground(String... args) {
    List<NameValuePair> params = new
        ArrayList<NameValuePair>();
    String message_id =
        String.valueOf(message.getMessageId());

    params.add(new BasicNameValuePair("message_id",
        message_id));

    return jsonParser.makeHttpRequest(Url.GETMESSAGES,
        "POST", params);
}

protected void onPostExecute(JSONObject json) {
    try {
        if(json!=null){
            int success =
                json.getInt(CONSTANTS.TAG_SUCCESS);
            String message =
                json.getString(CONSTANTS.TAG_MESSAGE);
            if (success == 1) {
                DialogUtil.showDialog(SendMessage.this,
                    message, new
                    DialogInterface.OnClickListener() {
                        @Override
                        public void onClick(DialogInterface
                            dialog, int which) {

```



```

import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.BaseAdapter;
import android.widget.ListView;
import android.widget.TextView;

import org.apache.http.NameValuePair;
import org.apache.http.message.BasicNameValuePair;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.List;

import orl.orscheduler.objects.global.CONSTANTS;
import orl.orscheduler.objects.global.GlobalVariables;
import orl.orscheduler.objects.user.Message;
import orl.orscheduler.objects.user.User;
import orl.orscheduler.utilities.DialogUtil;
import orl.orscheduler.utilities.JSONParser;
import orl.orscheduler.utilities.MessageBuilder;
import orl.orscheduler.web.Url;

public class UserManagement extends ActionBarActivity {

    private ListView listView_users;
    private Context context;
    JSONParser jsonParser = new JSONParser();
    private ProgressDialog pDialog;
    private TextView header1, header2;

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_user_management);

    header1 = (TextView) findViewById(R.id.header1);
    header2 = (TextView) findViewById(R.id.header2);

    listView_users = (ListView)
        findViewById(R.id.admin_listView_users);
    listView_users.setOnItemClickListener(new
        AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View
                view, int position, long id) {
                Intent i = new Intent(UserManagement.this,
                    Register.class);
                i.putExtra("user_index", position);
                startActivity(i);
            }
        });
    new AttemptGetUsers().execute();
}

```

```

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if
    it is present.
    getMenuInflater().inflate(R.menu.menu_user_management,
        menu);
    return true;
}

```

```

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so
    long
    // as you specify a parent activity in
    AndroidManifest.xml.
    int id = item.getItemId();

    //noinspection SimplifiableIfStatement
    if (id == R.id.action_settings) {
        return true;
    }

    return super.onOptionsItemSelected(item);
}

public class UserAdapter extends BaseAdapter {

    List<User> userList = GlobalVariables.userList;

    @Override
    public int getCount() {
        // TODO Auto-generated method stub
        return userList.size();
    }

    @Override
    public User getItem(int arg0) {
        // TODO Auto-generated method stub
        return userList.get(arg0);
    }
}

```

```

@Override
public long getItemId(int arg0) {
    // TODO Auto-generated method stub
    return arg0;
}

@Override
public View getView(int arg0, View arg1, ViewGroup arg2) {

    if(arg1==null)
    {
        LayoutInflater inflater = (LayoutInflater)
            UserManagement.this.getSystemService
(Context.LAYOUT_INFLATER_SERVICE);
        arg1 = inflater.inflate(R.layout.admin_user_item,
            arg2,false);
    }

    TextView text_icon =
        (TextView)arg1.findViewById(R.id.admin_text_icon);
    TextView text_username =
        (TextView)arg1.findViewById(R.id.admin_text_username);
    TextView text_fullname =
        (TextView)arg1.findViewById(R.id.admin_text_fullname);
    TextView text_usertype =
        (TextView)arg1.findViewById(R.id.admin_text_usertype);

    User user = userList.get(arg0);

    text_icon.setText(user.getUsertype().equals
(CONSTANTS.USER_TYPE_RESIDENT) ? "R" : "C");
    text_username.setText(user.getUsername());

```

```

        text_fullname.setText(user.getFullname());
        text_usertype.setText(user.getUsertype());

        return arg1;
    }
}

class AttemptGetUsers extends AsyncTask<String, String,
    JSONObject> {

    boolean failure = false;

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
    }

    @Override
    protected JSONObject doInBackground(String... args) {
        List<NameValuePair> params = new
            ArrayList<NameValuePair>();
        return jsonParser.makeHttpRequest(Url.GETUSERS, "POST",
            params);
    }

    protected void onPostExecute(JSONObject json) {
        try {
            if(json!=null){
                int success =
                    json.getInt(CONSTANTS.TAG_SUCCESS);
                String message =
                    json.getString(CONSTANTS.TAG_MESSAGE);
                if (success == 1) {

```



```

        List<User> userList = new ArrayList<User>();
        JSONArray jsonArray =
            json.getJSONArray("users");
        for(int i=0;i<jsonArray.length();i++){
            JSONObject jsonUser =
                jsonArray.getJSONObject(i);
            User user = new User();
            user.buildUserFromJson(jsonUser);
            userList.add(user);
        }
        GlobalVariables.userList = userList;
        UserAdapter userAdapter = new UserAdapter();
        listView_users.setAdapter(userAdapter);
    }else{
        DialogUtil.showDialog(UserManagement.this,
            message, null);
    }
}
} catch (JSONException e) {
    e.printStackTrace();
}
}

}

@Override
public void onBackPressed() {
    Intent i = new Intent(UserManagement.this, Welcome.class);
    startActivity(i);
}
}
}

```

ViewSchedules.java

```

package orl.orscheduler;

/**
 * Created by Raphael Lanuza on 5/2/2015.
 */
import java.text.ParseException;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.List;

import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.os.AsyncTask;
import android.support.v4.view.PagerAdapter;
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.support.v4.view.ViewPager;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.RelativeLayout;
import android.widget.TextView;
import android.widget.Toast;

import org.apache.http.NameValuePair;
import org.apache.http.message.BasicNameValuePair;
import org.json.JSONException;
import org.json.JSONObject;

import orl.orscheduler.objects.global.CONSTANTS;
import orl.orscheduler.objects.global.GlobalVariables;
import orl.orscheduler.objects.global.PREFERENCES;

```

```

import orl.orscheduler.objects.or.DaySchedule;
import orl.orscheduler.objects.or.ScheduleParameters;
import orl.orscheduler.utilities.DateUtil;
import orl.orscheduler.utilities.DayScheduleBuilder;
import orl.orscheduler.utilities.DialogUtil;
import orl.orscheduler.utilities.JSONParser;
import orl.orscheduler.utilities.SchedDateList;
import orl.orscheduler.web.Url;

public class ViewSchedules extends ActionBarActivity {

    ViewPager viewPager;
    PagerAdapter adapter;
    JSONParser jsonParser = new JSONParser();
    private ProgressDialog pDialog;
    private Context viewScheduleContext;
    Calendar calendar = Calendar.getInstance();

    private Button button_r1s1, button_r1s2, button_r1s3,
        button_r2s1, button_r2s2, button_r2s3, button_r3s1,
        button_r3s2, button_r3s3;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        GlobalVariables.viewClass = ViewSchedules.class;
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_view_schedules);

        viewScheduleContext = this.getApplicationContext();
        new AttemptGetSchedules().execute();
    }
}

```

```

class AttemptGetSchedules extends AsyncTask<String, String,
    JSONObject> {

    boolean failure = false;

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        pDialog = new ProgressDialog(ViewSchedules.this);
        pDialog.setMessage("Please wait...");
        pDialog.setIndeterminate(false);
        pDialog.setCancelable(true);
        pDialog.show();
    }

    @Override
    protected JSONObject doInBackground(String... args) {
        GlobalVariables.viewClass = ViewSchedules.class;

        int success;
        String start_date ;
        String end_date ;
        String ortype = getIntent().getExtras() != null ?
            getIntent().getExtras().get("ORTYPE").toString() :
            GlobalVariables.user.getORType();

        start_date = DateUtil.getMDDYYDate(calendar);
        calendar.add(Calendar.DATE,
            PREFERENCES.DATE_RANGE_COUNT);
        end_date = DateUtil.getMDDYYDate(calendar);

        List<NameValuePair> params = new
            ArrayList<NameValuePair>();
    }
}

```

```

        params.add(new BasicNameValuePair("start_date",
            start_date));
        params.add(new BasicNameValuePair("end_date",
            end_date));
        params.add(new BasicNameValuePair("or_type", orte));

        return jsonParser.makeHttpRequest(Url.GETSCHEDULES,
            "POST", params);
    }

    protected void onPostExecute(JSONObject json) {
        pDialog.dismiss();
        try {
            if(json!=null){
                int success =
                    json.getInt(CONSTANTS.TAG_SUCCESS);
                String message =
                    json.getString(CONSTANTS.TAG_MESSAGE);
                if (success == 1) {
                    List<DaySchedule> dayScheduleList = new
                        ArrayList<DaySchedule>();
                    List<String> dateList =
                        SchedDateList.getDatesForViewing
(PREFERENCES.DATE_RANGE_COUNT, getIntent().getExtras()!=null ?
                    getIntent().getExtras().get("ORTYPE").toString() :
                    GlobalVariables.user.getORType());

                    for(int i = 0; i < dateList.size(); i++){
                        DaySchedule daySchedule = new
                            DaySchedule();
                        daySchedule = DayScheduleBuilder
.buildDayScheduleFromJson
(json, dateList.get(i), getIntent().getExtras()!=null ?
                    getIntent().getExtras().get("ORTYPE").toString() :

```



```

@Override
public int getCount() {
    return dayScheduleList.size();
}

@Override
public boolean isViewFromObject(View view, Object o) {
    return view == ((RelativeLayout) o);
}

@Override
public Object instantiateItem(ViewGroup container, int
    position) {
    TextView text_day, text_date, text_room1, text_room2,
        text_room3;
    Button button_r1s1, button_r1s2, button_r1s3,
        button_r2s1, button_r2s2, button_r2s3, button_r3s1,
        button_r3s2, button_r3s3;

    LayoutInflater = (LayoutInflater)
        context.getSystemService
(Context.LAYOUT_INFLATER_SERVICE);
    View schedPageItemView =
        inflater.inflate(R.layout.schedule_page_item,
            container, false);

    text_day = (TextView) schedPageItemView.findViewById
(R.id.schedPageItem_text_day);
    text_date = (TextView) schedPageItemView.findViewById
(R.id.schedPageItem_text_date);
    text_room1 = (TextView) schedPageItemView.findViewById
(R.id.schedPageItem_text_room1);

```

```

        text_room2 = (TextView) schedPageItemView.findViewById
(R.id.schedPageItem_text_room2);
        text_room3 = (TextView) schedPageItemView.findViewById
(R.id.schedPageItem_text_room3);

        button_r1s1 = (Button) schedPageItemView.findViewById
(R.id.schedPageItem_btn_rm1slt1);
        button_r1s2 = (Button) schedPageItemView.findViewById
(R.id.schedPageItem_btn_rm1slt2);
        button_r1s3 = (Button) schedPageItemView.findViewById
(R.id.schedPageItem_btn_rm1slt3);

        button_r2s1 = (Button) schedPageItemView.findViewById
(R.id.schedPageItem_btn_rm2slt1);
        button_r2s2 = (Button) schedPageItemView.findViewById
(R.id.schedPageItem_btn_rm2slt2);
        button_r2s3 = (Button) schedPageItemView.findViewById
(R.id.schedPageItem_btn_rm2slt3);

        button_r3s1 = (Button) schedPageItemView.findViewById
(R.id.schedPageItem_btn_rm3slt1);
        button_r3s2 = (Button) schedPageItemView.findViewById
(R.id.schedPageItem_btn_rm3slt2);
        button_r3s3 = (Button) schedPageItemView.findViewById
(R.id.schedPageItem_btn_rm3slt3);

        final DaySchedule daySchedule =
            dayScheduleList.get(position);

        text_day.setText(daySchedule.getDayString());
        text_date.setText(daySchedule.getDate());

        if(daySchedule.getOrType().equals

```



```

(CONSTANTS.OR_TYPE_PRIVATE)){
    if(daySchedule.getDayString().toUpperCase().equals
(CONSTANTS.DAY_MONDAY)){
        button_r1s1.setText(daySchedule
.getOperation("1","1"));
        button_r1s2.setText(daySchedule
.getOperation("1","2"));
        button_r1s3.setText(daySchedule
.getOperation("1","3"));

        button_r2s1.setText(daySchedule
.getOperation("2","1"));
        button_r2s2.setText(daySchedule
.getOperation("2","2"));
        button_r2s3.setText(daySchedule
.getOperation("2","3"));

        text_room3.setVisibility(View.GONE);
        button_r3s1.setVisibility(View.GONE);
        button_r3s2.setVisibility(View.GONE);
        button_r3s3.setVisibility(View.GONE);

    } else if
        (daySchedule.getDayString().toUpperCase().equals
(CONSTANTS.DAY_WEDNESDAY) ){
        button_r1s1.setText(daySchedule
.getOperation("1","1"));
        button_r1s2.setText(daySchedule
.getOperation("1","2"));
        button_r1s3.setText(daySchedule
.getOperation("1","3"));

        text_room2.setVisibility(View.GONE);

```

```

        button_r2s1.setVisibility(View.GONE);
        button_r2s2.setVisibility(View.GONE);
        button_r2s3.setVisibility(View.GONE);

        text_room3.setVisibility(View.GONE);
        button_r3s1.setVisibility(View.GONE);
        button_r3s2.setVisibility(View.GONE);
        button_r3s3.setVisibility(View.GONE);
    } else if
        (daySchedule.getDayString().toUpperCase().equals
(CONSTANTS.DAY_FRIDAY)){
        button_r1s1.setText(daySchedule
.getOperation("1","1"));
        button_r1s2.setVisibility(View.GONE);
        button_r1s3.setVisibility(View.GONE);

        button_r2s1.setText(daySchedule
.getOperation("2","1"));
        button_r2s2.setVisibility(View.GONE);
        button_r2s3.setVisibility(View.GONE);

        button_r3s1.setText(daySchedule
.getOperation("3","1"));
        button_r3s2.setVisibility(View.GONE);
        button_r3s3.setVisibility(View.GONE);
    }

    } else {
        if (daySchedule.getDayString().toUpperCase().equals
(CONSTANTS.DAY_FRIDAY)){
            button_r1s1.setText(daySchedule
.getOperation("1","1"));
            button_r1s2.setText(daySchedule

```

```

.getOperation("1","2"));
        button_r1s3.setVisibility(View.GONE);

        button_r2s1.setText(daySchedule
.getOperation("2","1"));
        button_r2s2.setText(daySchedule
.getOperation("2","2"));
        button_r2s3.setVisibility(View.GONE);

        text_room3.setVisibility(View.GONE);
        button_r3s1.setVisibility(View.GONE);
        button_r3s2.setVisibility(View.GONE);
        button_r3s3.setVisibility(View.GONE);
    } else {
        button_r1s1.setText(daySchedule
.getOperation("1","1"));
        button_r1s2.setText(daySchedule
.getOperation("1","2"));
        button_r1s3.setText(daySchedule
.getOperation("1","3"));

        button_r2s1.setText(daySchedule
.getOperation("2","1"));
        button_r2s2.setText(daySchedule
.getOperation("2","2"));
        button_r2s3.setText(daySchedule
.getOperation("2","3"));

        button_r3s1.setText(daySchedule
.getOperation("3","1"));
        button_r3s2.setText(daySchedule
.getOperation("3","2"));
        button_r3s3.setText(daySchedule

```

```

.getOperation("3","3"));
        }
    }

    List<Button> buttonList = new ArrayList<Button>();
    buttonList.add(button_r1s1);
    buttonList.add(button_r1s2);
    buttonList.add(button_r1s3);
    buttonList.add(button_r2s1);
    buttonList.add(button_r2s2);
    buttonList.add(button_r2s3);
    buttonList.add(button_r3s1);
    buttonList.add(button_r3s2);
    buttonList.add(button_r3s3);

    for(Button button : buttonList){
        String operation = button.getText().toString();
        if(operation==null ||
            operation.equals(CONSTANTS.OPERATION_DISABLED)){
            button.setEnabled(false);
            button.setText("");
        }
    }

    button_r1s1.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            String roomNumber="1";
            String slotNumber="1";
            String date = daySchedule.getDate();
            GlobalVariables.scheduleParameters = new
                ScheduleParameters(daySchedule.getOperation

```

```

(roomNumber,slotNumber),roomNumber, daySchedule.
getRoom(roomNumber).getSlot(slotNumber), date);
        startActivity(new Intent(viewScheduleContext,
                CreateSchedule.class));
    }
});

button_r1s2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String roomNumber="1";
        String slotNumber="2";
        String date = daySchedule.getDate();
        GlobalVariables.scheduleParameters = new
                ScheduleParameters(daySchedule.getOperation
(roomNumber,slotNumber),roomNumber, daySchedule.
getRoom(roomNumber).getSlot(slotNumber), date);
        startActivity(new Intent(viewScheduleContext,
                CreateSchedule.class));
    }
});

button_r1s3.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String roomNumber="1";
        String slotNumber="3";
        String date = daySchedule.getDate();
        GlobalVariables.scheduleParameters = new
                ScheduleParameters(daySchedule.getOperation
(roomNumber,slotNumber),roomNumber, daySchedule.
getRoom(roomNumber).getSlot(slotNumber), date);

```

```

        startActivity(new Intent(viewScheduleContext,
            CreateSchedule.class));
    }
});

button_r2s1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String roomNumber="2";
        String slotNumber="1";
        String date = daySchedule.getDate();
        GlobalVariables.scheduleParameters = new
            ScheduleParameters(daySchedule.getOperation
(roomNumber,slotNumber),roomNumber, daySchedule.
getRoom(roomNumber).getSlot(slotNumber), date);
        startActivity(new Intent(viewScheduleContext,
            CreateSchedule.class));
    }
});

button_r2s2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String roomNumber="2";
        String slotNumber="2";
        String date = daySchedule.getDate();
        GlobalVariables.scheduleParameters = new
            ScheduleParameters(daySchedule.getOperation
(roomNumber,slotNumber),roomNumber, daySchedule.
getRoom(roomNumber).getSlot(slotNumber), date);
        startActivity(new Intent(viewScheduleContext,
            CreateSchedule.class));
    }
});

```

```

});

button_r2s3.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String roomNumber="2";
        String slotNumber="3";
        String date = daySchedule.getDate();
        GlobalVariables.scheduleParameters = new
            ScheduleParameters(daySchedule.getOperation
(roomNumber,slotNumber),roomNumber, daySchedule.
getRoom(roomNumber).getSlot(slotNumber), date);
        startActivity(new Intent(viewScheduleContext,
            CreateSchedule.class));
    }
});

button_r3s1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String roomNumber="3";
        String slotNumber="1";
        String date = daySchedule.getDate();
        GlobalVariables.scheduleParameters = new
            ScheduleParameters(daySchedule.getOperation
(roomNumber,slotNumber),roomNumber, daySchedule.
getRoom(roomNumber).getSlot(slotNumber), date);
        startActivity(new Intent(viewScheduleContext,
            CreateSchedule.class));
    }
});

button_r3s2.setOnClickListener(new View.OnClickListener() {

```

```

@Override
public void onClick(View v) {
    String roomNumber="3";
    String slotNumber="2";
    String date = daySchedule.getDate();
    GlobalVariables.scheduleParameters = new
        ScheduleParameters(daySchedule.getOperation
(roomNumber,slotNumber),roomNumber, daySchedule.
getRoom(roomNumber).getSlot(slotNumber), date);
        startActivity(new Intent(viewScheduleContext,
            CreateSchedule.class));
    }
});

button_r3s3.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
    String roomNumber="3";
    String slotNumber="3";
    String date = daySchedule.getDate();
    GlobalVariables.scheduleParameters = new
        ScheduleParameters(daySchedule.getOperation
(roomNumber,slotNumber),roomNumber,
    daySchedule.getRoom(roomNumber)
.getSlot(slotNumber), date);
        startActivity(new Intent(viewScheduleContext,
            CreateSchedule.class));
    }
});

((ViewPager) container).addView(schedPageItemView);

return schedPageItemView;

```



```

    }

    @Override
    public void destroyItem(ViewGroup container, int position,
        Object object) {
        ((ViewPager) container).removeView((RelativeLayout)
            object);
    }
}

@Override
public void onBackPressed() {
    Intent i = new Intent(ViewSchedules.this,
        GlobalVariables.viewClass.equals(ViewSchedules.class) ?
        (GlobalVariables.user.getUser_id()==null? Login.class :
        Welcome.class) : GlobalVariables.viewClass);
    startActivity(i);
}

}

```

Welcome.java

```

package orl.orscheduler;

/**
 * Created by Raphael Lanuza on 4/29/2015.
 */
import android.app.ProgressDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.AsyncTask;
import android.support.v7.app.ActionBarActivity;

```

```

import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.animation.AlphaAnimation;
import android.view.animation.Animation;
import android.widget.Button;
import android.widget.TextView;

import org.apache.http.NameValuePair;
import org.apache.http.message.BasicNameValuePair;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.List;

import orl.orscheduler.objects.global.CONSTANTS;
import orl.orscheduler.objects.global.GlobalVariables;
import orl.orscheduler.objects.user.Message;
import orl.orscheduler.objects.user.User;
import orl.orscheduler.utilities.DialogUtil;
import orl.orscheduler.utilities.JSONParser;
import orl.orscheduler.utilities.MessageBuilder;
import orl.orscheduler.utilities.NetworkUtil;
import orl.orscheduler.web.Url;

public class Welcome extends ActionBarActivity {

    private Context context;
    private TextView text_firstname, text_lastname, text_usertype,
        text_message_count, header1, header2;

```

```

private Button button_create, button_view, button_updateUser,
    button_logOut, button_print, button_send;
JSONParser jsonParser = new JSONParser();
private ProgressDialog pDialog;

@Override
protected void onCreate(Bundle savedInstanceState) {
    GlobalVariables.viewClass = Welcome.class;
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_welcome);
    context = this.getApplicationContext();
    User user = GlobalVariables.user;

    header1 = (TextView) findViewById(R.id.header1);
    header2 = (TextView) findViewById(R.id.header2);

    text_firstname = (TextView)
        findViewById(R.id.welcome_text_firstname);
    text_lastname = (TextView)
        findViewById(R.id.welcome_text_lastname);
    text_usertype = (TextView)
        findViewById(R.id.welcome_text_usertype);
    text_message_count = (TextView)
        findViewById(R.id.welcome_text_messageCount);

    text_firstname.setText(user.getFirstname());
    text_lastname.setText(user.getLastname());
    text_usertype.setText(user.getUsertype());

    button_updateUser = (Button)
        findViewById(R.id.welcome_button_updateUser);
    button_view = (Button)
        findViewById(R.id.welcome_button_view);

```

```

button_logout = (Button)
    findViewById(R.id.welcome_button_logout);
button_print = (Button)
    findViewById(R.id.welcome_button_print);
button_send = (Button)
    findViewById(R.id.welcome_button_messaging);

button_logout.setOnClickListener(onClick);
button_updateUser.setOnClickListener(onClick);
button_view.setOnClickListener(onClick);
button_print.setOnClickListener(onClick);
button_send.setOnClickListener(onClick);

if (user.getUsertype().equals(CONSTANTS.USER_TYPE_ADMIN)){

    text_firstname.setText(user.getFirstname());
    text_lastname.setText(user.getLastname());
    text_usertype.setText(user.getUsertype());

    //text_firstname.setVisibility(View.GONE);
    //text_lastname.setVisibility(View.GONE);
    //text_usertype.setVisibility(View.GONE);

    button_logout.setOnClickListener(onClick);
    button_updateUser.setOnClickListener(onClick);
    button_view.setVisibility(View.GONE);
    button_print.setVisibility(View.GONE);
    button_send.setOnClickListener(onClick);
    button_updateUser.setText("User Account Management");
}

new AttemptGetMessageCount().execute();

```

```

}

private View.OnClickListener onClick = new
    View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent i;
            if(!NetworkUtil.haveNetworkConnection(Welcome.this)){
                DialogUtil.showDialog(Welcome.this, "No Internet
                    Connection!", new
                        DialogInterface.OnClickListener() {
                            @Override
                            public void onClick(DialogInterface dialog, int
                                which) {
                                    NetworkUtil.logOutWhenNoConnection(Welcome.this);
                                }
                            });
            } else {
                switch (v.getId()) {
                    case R.id.welcome_button_view:
                        i = new Intent(context, ViewSchedules.class);
                        startActivity(i);
                        break;
                    case R.id.welcome_button_updateUser:
                        i = new Intent(context,
                            GlobalVariables.user.getUsertype().equals
(CONSTANTS.USER_TYPE_ADMIN) ? UserManagement.class :
Register.class);
                        startActivity(i);
                        break;
                    case R.id.welcome_button_logOut:
                        DialogUtil.showConfirmationDialog(Welcome.this,
                            "Logout?", new

```

```

        DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface
                dialog, int which) {
                GlobalVariables.initializeAll();
                Intent i = new Intent(Welcome.this,
                    Login.class);
                startActivity(i);
            }
        }, null);
        break;
    case R.id.welcome_button_print:
        i = new Intent(context, PrintPdf.class);
        startActivity(i);
        break;
    case R.id.welcome_button_messaging:
        i = new Intent(context, Messaging.class);
        startActivity(i);
        break;
    default:
        break;
    }
}

}

};

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if
    it is present.
    getMenuInflater().inflate(R.menu.menu_welcome, menu);
    return true;
}

```

```

}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so
    // long
    // as you specify a parent activity in
    // AndroidManifest.xml.
    int id = item.getItemId();

    //noinspection SimplifiableIfStatement
    if (id == R.id.action_settings) {
        return true;
    }

    return super.onOptionsItemSelected(item);
}

@Override
public void onBackPressed() {
    DialogUtil.showConfirmationDialog(Welcome.this, "Logout?",
        new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                GlobalVariables.initializeAll();
                Intent i = new Intent(Welcome.this, Login.class);
                startActivity(i);
            }
        }, null);
}

```

```

class AttemptGetMessageCount extends AsyncTask<String, String,
    JSONObject> {

    boolean failure = false;

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
    }

    @Override
    protected JSONObject doInBackground(String... args) {
        List<NameValuePair> params = new
            ArrayList<NameValuePair>();
        params.add(new BasicNameValuePair("count","count"));
        params.add(new BasicNameValuePair("receiver_user_id",
            GlobalVariables.user.getUser_id()));
        return jsonParser.makeHttpRequest(Url.GETMESSAGES,
            "POST", params);
    }

    protected void onPostExecute(JSONObject json) {
        try {
            if(json!=null){
                int success =
                    json.getInt(CONSTANTS.TAG_SUCCESS);
                String message =
                    json.getString(CONSTANTS.TAG_MESSAGE);
                if (success == 1) {
                    Animation anim = new AlphaAnimation(0.0f,
                        1.0f);
                    anim.setDuration(1000);
                    anim.setStartOffset(50);
                }
            }
        }
    }
}

```



```
}  
}
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>  
<manifest  
  xmlns:android="http://schemas.android.com/apk/res/android"  
  package="org.orscheduler" >  
  
  <uses-permission android:name="android.permission.INTERNET" />  
  <uses-permission  
    android:name="android.permission.WRITE_EXTERNAL_STORAGE" />  
  <uses-permission  
    android:name="android.permission.READ_EXTERNAL_STORAGE" />  
  <uses-permission  
    android:name="android.permission.ACCESS_NETWORK_STATE" />  
  
  <application  
    android:allowBackup="true"  
    android:icon="@mipmap/ic_launcher"  
    android:label="@string/app_name"  
    android:theme="@style/Theme.AppCompat.NoActionBar" >  
    <activity  
      android:name=".Home"  
      android:label="@string/app_name" >  
      <intent-filter>  
        <action android:name="android.intent.action.MAIN" />  
  
        <category  
          android:name="android.intent.category.LAUNCHER"  
        />  
      </intent-filter>  
    </activity>
```

```
<activity
    android:name=".Login"
    android:label="@string/title_activity_login" >
</activity>
<activity
    android:name=".Register"
    android:label="@string/title_activity_register" >
</activity>
<activity
    android:name=".ViewSchedules"
    android:label="@string/title_activity_view_schedules" >
</activity>
<activity
    android:name=".Welcome"
    android:label="@string/title_activity_welcome" >
</activity>
<activity
    android:name=".CreateSchedule"
    android:label="@string/title_activity_create_schedule" >
</activity>
<activity
    android:name=".PrintPdf"
    android:label="@string/title_activity_print_pdf" >
</activity>
<activity
    android:name=".SendMessage"
    android:label="@string/title_activity_send_message" >
</activity>
<activity
    android:name=".Messaging"
    android:label="@string/title_activity_messaging" >
</activity>
<activity
```

```

        android:name=".Inbox"
        android:label="@string/title_activity_inbox" >
</activity>
<activity
    android:name=".UserManagement"
    android:label="@string/title_activity_user_management" >
</activity>
</application>

```

```
</manifest>
```

activity_create_schedule.xml

```

<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context="orl.orscheduler.CreateSchedule">

```

```

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="fill_parent"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true">

```

```

    <TextView
        android:layout_width="wrap_content"

```

```
    android:layout_height="wrap_content"
    android:textStyle="bold"
    android:text="DEPARTMENT OF OTORHINOLARYNGOLOGY"
    android:id="@+id/header1" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textStyle="bold"
    android:text="UP-PGH"
    android:id="@+id/header2" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance=
"?android:attr/textAppearanceLarge"
    android:text="OR Details"
    android:id="@+id/createSchedule.text.title"
    android:textSize="35dp" />
```

```
<ScrollView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/createSchedule.scrollview" >
```

```
<LinearLayout
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
```

```
<TextView
    android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:textAppearance=
"?android:attr/textAppearanceMedium"
        android:text="Full Name of Patient *"
        android:id="@+id/createSchedule.text.patientName"
        android:paddingTop="20dp"/>
```

```
<EditText
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="textCapWords"
    android:id="@+id/createSchedule.input.patientName"
    />
```

```
<TextView
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance=
"?android:attr/textAppearanceMedium"
    android:text="Diagnosis *"
    android:id="@+id/createSchedule.text.diagnosis"
    android:paddingTop="20dp"/>
```

```
<EditText
```

```
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:inputType="textCapWords"
    android:ems="10"
    android:id="@+id/createSchedule.input.diagnosis"
    />
```

```
<TextView
```

```
    android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:textAppearance=
"?android:attr/textAppearanceMedium"
        android:text="Procedure *"
        android:id="@+id/createSchedule.text.procedure"
        android:paddingTop="20dp"/>
```

```
<EditText
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="textCapWords"
    android:id="@+id/createSchedule.input.procedure"
    />
```

```
<TextView
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance=
"?android:attr/textAppearanceMedium"
    android:text="Surgical Team *"
    android:id="@+id/createSchedule.text.surgicalTeam"
    android:paddingTop="20dp"/>
```

```
<EditText
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="textCapWords"
    android:id="@+id/createSchedule.input.surgicalTeam"
    />
```

```
<TextView
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```
        android:textAppearance=
"?android:attr/textAppearanceMedium"
        android:text="Other Details"
        android:id="@+id/createSchedule.text.otherDetails"
        android:paddingTop="20dp"/>
```

```
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/createSchedule.input.otherDetails"
    />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance=
"?android:attr/textAppearanceMedium"
    android:text=""
    android:id="@+id/createSchedule.text.user"
    android:paddingTop="20dp"
    android:paddingBottom="20dp"
    android:textSize="12sp"
    android:visibility="gone" />
```

```
<Button
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Submit"
    android:id="@+id/createSchedule.button.submit"
    android:textSize="20dp"/>
```

```
<Button
    android:layout_width="fill_parent"
```



```

        android:layout_height="wrap_content"
        android:text="Delete"
        android:id="@+id/createSchedule.button.delete"
    />

    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Cancel"
        android:id="@+id/createSchedule.button.cancel"
    />
</LinearLayout>
</ScrollView>

</LinearLayout>

</RelativeLayout>

```

activity_home.xml

```

<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Home">

    <ScrollView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:id="@+id/scrollView3" >

        <RelativeLayout
            android:orientation="vertical"

```

```

        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true">

<ImageView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/orllogo"
    android:src="@drawable/orl_logo"
    android:layout_below="@+id/textView"
    android:layout_centerHorizontal="true" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance=
"?android:attr/textAppearanceLarge"
    android:text="Operating Room Scheduler"
    android:id="@+id/textView"
    android:layout_gravity="center_horizontal"
    android:layout_weight="0.05"
    android:textSize="27dp"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance=
"?android:attr/textAppearanceMedium"
    android:textAlignment="inherit"
    android:textStyle="bold"

```

```
android:text="The Department of  
Otorhinolaryngology, University of the  
Philippines Philippine General Hospital is the  
bastion of ear, nose, and throat (ENT) care. As  
of today, the department is considered the  
largest tertiary Otorhinolaryngology service in  
the country having 26 residents and 26  
consultants. Encompassing amenities and  
facilities to Out-Patient clinics and Emergency  
room, In and Out-patient Operating room for  
Surgery, Audio logic Laboratories and Prosthetic  
Fabrications Unit. The staff members are  
nationally and internationally recognized for  
their academic and clinical activities,  
publications and innovations, and community  
service."
```

```
android:id="@+id/textView2"  
android:layout_below="@+id/orllogo"  
android:layout_alignParentLeft="true"  
android:layout_alignParentStart="true" />
```

```
<Button
```

```
style="?android:attr/buttonStyleSmall"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="Proceed"  
android:id="@+id/proceed"  
android:layout_below="@+id/textView2"  
android:layout_alignParentLeft="true"  
android:layout_alignParentStart="true"  
android:layout_alignParentRight="true"  
android:layout_alignParentEnd="true" />
```

```
</RelativeLayout>
```

```
</ScrollView>
```

```
</RelativeLayout>
```

activity_inbox.xml

```
<RelativeLayout
```

```
xmlns:android="http://schemas.android.com/apk/res/android"
```

```
xmlns:tools="http://schemas.android.com/tools"
```

```
android:layout_width="match_parent"
```

```
android:layout_height="match_parent"
```

```
android:paddingLeft="@dimen/activity_horizontal_margin"
```

```
android:paddingRight="@dimen/activity_horizontal_margin"
```

```
android:paddingTop="@dimen/activity_vertical_margin"
```

```
android:paddingBottom="@dimen/activity_vertical_margin"
```

```
tools:context="org.orscheduler.Inbox">
```

```
<LinearLayout
```

```
android:orientation="vertical"
```

```
android:layout_width="fill_parent"
```

```
android:layout_height="fill_parent">
```

```
<TextView
```

```
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
```

```
android:text="@string/header1"
```

```
android:textStyle="bold"
```

```
android:id="@+id/header1" />
```

```
<TextView
```

```
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
```

```

        android:text="@string/header2"
        android:textStyle="bold"
        android:id="@+id/header2" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance=
"?android:attr/textAppearanceLarge"
        android:text="Inbox"
        android:id="@+id/inbox.text.title"
        android:textSize="35dp" />

    <ListView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/inbox.listView.messages" />

</LinearLayout>

</RelativeLayout>

```

activity_log_in.xml

```

<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:paddingLeft="@dimen/activity_horizontal_margin"
        android:paddingRight="@dimen/activity_horizontal_margin"
        android:paddingTop="@dimen/activity_vertical_margin"
        android:paddingBottom="@dimen/activity_vertical_margin"
        tools:context="org.orscheduler.logInScreen">

```

```

<ScrollView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:id="@+id/scrollView3" >

<LinearLayout
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true">

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/header1"
    android:textStyle="bold"
    android:id="@+id/header1" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/header2"
    android:textStyle="bold"
    android:id="@+id/header2" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance=
"?android:attr/textAppearanceLarge"
    android:text="Log In"

```

```

        android:id="@+id/login.text.title"
        android:textSize="35dp" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance=
"?android:attr/textAppearanceMedium"
    android:text="Username *"
    android:id="@+id/login.text.username"
    android:paddingTop="20dp" />

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/login.input.username" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance=
"?android:attr/textAppearanceMedium"
    android:text="Password *"
    android:id="@+id/login.text.password"
    android:paddingTop="20dp" />

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="textPassword"
    android:ems="10"
    android:id="@+id/login.input.password" />
<Button

```

```
android:layout_width="fill_parent"  
android:layout_height="wrap_content"  
android:text="Log In"  
android:id="@+id/login.button.login"  
android:layout_marginTop="20dp"/>
```

```
<Button
```

```
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:text="Register"  
    android:id="@+id/login.button.register" />
```

```
<LinearLayout
```

```
    android:layout_width="fill_parent"  
    android:layout_height="1dp"  
    android:layout_marginTop="20dp"  
    android:orientation="horizontal"  
    android:background="#5fc3c3c3"  
    android:layout_marginBottom="20dp" />
```

```
<TextView
```

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:textAppearance=  
    "?android:attr/textAppearanceMedium"  
    android:text="View Operating Room Schedule"  
    android:id="@+id/login.text.orType"  
    android:paddingTop="0dp"  
    android:textStyle="italic"  
    android:textIsSelectable="false"  
    android:textSize="20dp" />
```

```
<LinearLayout
```



```

        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingTop="20dp">
        <RadioButton android:id="@+id/login.radio.private"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Private/Pay"
            android:layout_marginRight="50dp" />
        <RadioButton android:id="@+id/login.radio.charity"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Charity"/>
    </LinearLayout>

    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="View"
        android:id="@+id/login.button.view"
        android:layout_marginTop="20dp"/>

    </LinearLayout>
</ScrollView>

</RelativeLayout>

```

activity_messaging.xml

```

<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"

```

```
android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
android:paddingBottom="@dimen/activity_vertical_margin"
tools:context="org.orscheduler.Messaging">
```

```
<LinearLayout
```

```
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
```

```
<TextView
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/header1"
    android:textStyle="bold"
    android:id="@+id/header1" />
```

```
<TextView
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/header2"
    android:textStyle="bold"
    android:id="@+id/header2" />
```

```
<TextView
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance=
"?android:attr/textAppearanceLarge"
    android:text="Messaging"
    android:id="@+id/messageHome.text.title"
```

```

        android:textSize="35dp" />

<Button
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Create Message"
    android:id="@+id/messageHome.button.create"
    android:layout_marginTop="20dp"/>

<Button
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Inbox"
    android:id="@+id/messageHome.button.inbox" />
</LinearLayout>

</RelativeLayout>

```

activity_print_pdf.xml

```

<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context="org.orscheduler.PrintPdf">

<LinearLayout
    android:orientation="vertical"
    android:layout_width="fill_parent"

```

```
android:layout_height="fill_parent"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true">
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/header1"
    android:textStyle="bold"
    android:id="@+id/header1" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/header2"
    android:textStyle="bold"
    android:id="@+id/header2" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance=
"?android:attr/textAppearanceLarge"
    android:text="Generate Reports"
    android:id="@+id/print.text.title"
    android:textSize="35dp" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance=
"?android:attr/textAppearanceMedium"
    android:text="Date"
```

```

        android:id="@+id/print.text.date"
        android:paddingTop="20dp" />

    <Spinner
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:id="@+id/print.spinner.date" />

    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Generate"
        android:id="@+id/print.button.print"
        android:layout_marginTop="20dp" />

    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Cancel"
        android:id="@+id/print.button.cancel" />

</LinearLayout>

</RelativeLayout>

```

activity_register.xml

```

<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:paddingLeft="@dimen/activity_horizontal_margin"
        android:paddingRight="@dimen/activity_horizontal_margin"
        android:paddingTop="@dimen/activity_vertical_margin"

```

```
android:paddingBottom="@dimen/activity_vertical_margin"  
tools:context="org.orscheduler.Register">
```

```
<LinearLayout
```

```
    android:orientation="vertical"  
    android:layout_width="match_parent"  
    android:layout_height="fill_parent"  
    android:layout_alignParentTop="true"  
    android:layout_alignParentLeft="true"  
    android:layout_alignParentStart="true">
```

```
<TextView
```

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="@string/header1"  
    android:textStyle="bold"  
    android:id="@+id/header1" />
```

```
<TextView
```

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="@string/header2"  
    android:textStyle="bold"  
    android:id="@+id/header2" />
```

```
<TextView
```

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:textAppearance=  
"?android:attr/textAppearanceLarge"  
    android:text="Register"  
    android:id="@+id/register.text.title"  
    android:textSize="35dp" />
```

```

<ScrollView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/scrollView2" >

<LinearLayout
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance=
"?android:attr/textAppearanceMedium"
        android:text="First Name *"
        android:id="@+id/register.text.firstName"
        android:paddingTop="20dp"/>

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textCapWords"
        android:id="@+id/register.input.firstName" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance=
"?android:attr/textAppearanceMedium"
        android:text="Last Name *"
        android:id="@+id/register.text.lastName"

```

```
android:paddingTop="20dp"/>
```

```
<EditText
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
```

```
    android:inputType="textCapWords"
```

```
    android:id="@+id/register.input.lastName" />
```

```
<TextView
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:textAppearance=
```

```
    "?android:attr/textAppearanceMedium"
```

```
    android:text="User Type *"
```

```
    android:id="@+id/register.text.userType"
```

```
    android:paddingTop="20dp"/>
```

```
<LinearLayout
```

```
    android:orientation="horizontal"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
```

```
    android:paddingTop="15dp">
```

```
<RadioButton
```

```
    android:id="@+id/register.radio.residents"
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:text="Resident"/>
```

```
<RadioButton
```

```
    android:id="@+id/register.radio.consultants"
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:text="Consultant"/>
```

```
</LinearLayout>
```



```

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance=
"?android:attr/textAppearanceMedium"
    android:text="Username *"
    android:id="@+id/register.text.username"
    android:paddingTop="20dp"/>

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/register.input.username" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance=
"?android:attr/textAppearanceMedium"
    android:text="Password *"
    android:id="@+id/register.text.password"
    android:paddingTop="20dp"/>

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/register.input.password" />

<Button
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Register"

```

```

        android:id="@+id/register.button.register"
        android:layout_marginTop="20dp"/>
    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Delete"
        android:id="@+id/register.button.delete"
        android:visibility="gone" />

    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Cancel"
        android:id="@+id/register.button.cancel" />

    </LinearLayout>
</ScrollView>

</LinearLayout>
</RelativeLayout>

```

activity_send_message.xml

```

<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:paddingLeft="@dimen/activity_horizontal_margin"
        android:paddingRight="@dimen/activity_horizontal_margin"
        android:paddingTop="@dimen/activity_vertical_margin"
        android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context="org.orscheduler.SendMessage">

```

```

<LinearLayout
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/header1"
        android:textStyle="bold"
        android:id="@+id/header1" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textStyle="bold"
        android:text="@string/header2"
        android:id="@+id/header2" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance=
"?android:attr/textAppearanceLarge"
        android:text="Send Message"
        android:id="@+id/sendMessage.text.title"
        android:textSize="35dp" />

    <ScrollView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"

```

```
android:id="@+id/scrollView5" >
```

```
<LinearLayout
```

```
    android:orientation="vertical"
```

```
    android:layout_width="fill_parent"
```

```
    android:layout_height="fill_parent">
```

```
    <TextView
```

```
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
```

```
        android:textAppearance=
```

```
        "?android:attr/textAppearanceSmall"
```

```
        android:text="User"
```

```
        android:id="@+id/sendMessage.text.receiver"
```

```
        android:textSize="20dp"
```

```
        android:paddingTop="0dp" />
```

```
    <TextView
```

```
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
```

```
        android:textAppearance=
```

```
        "?android:attr/textAppearanceSmall"
```

```
        android:text="User"
```

```
        android:id="@+id/sendMessage.text.receiverUsername"
```

```
        android:textSize="20dp"
```

```
        android:paddingTop="0dp"
```

```
        android:layout_marginLeft="15dp"
```

```
        android:visibility="gone" />
```

```
    <TextView
```

```
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
```

```
        android:textAppearance=
```

```
"?android:attr/textAppearanceSmall"
    android:text="User"
    android:id="@+id/sendMessage.text.receiverFullname"
    android:textSize="15dp"
    android:paddingTop="0dp"
    android:layout_marginLeft="15dp"
    android:visibility="gone" />
```

```
<Spinner
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:id="@+id/sendMessage.spinner.receiver"
    />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance=
"?android:attr/textAppearanceSmall"
    android:text="Message"
    android:id="@+id/sendMessage.text.message"
    android:textSize="20dp"
    android:paddingTop="0dp" />
```

```
<EditText
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:inputType="textMultiLine"
    android:ems="10"
    android:id="@+id/sendMessage.editText.message"
    android:layout_marginLeft="15dp" />
```

```
<Button
```

```
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Delete"
    android:id="@+id/sendMessage.button.delete"
    android:layout_gravity="bottom"
    android:visibility="gone" />
```

```
<Button
```

```
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Send"
    android:id="@+id/sendMessage.button.send"
    android:layout_gravity="bottom" />
```

```
<Button
```

```
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Cancel"
    android:id="@+id/sendMessage.button.cancel"
    android:layout_gravity="bottom" />
```

```
</LinearLayout>
```

```
</ScrollView>
```

```
</LinearLayout>
```

```
</RelativeLayout>
```

activity_user_management.xml

```
<RelativeLayout
```

```
    xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    xmlns:tools="http://schemas.android.com/tools"
```

```
        android:layout_width="match_parent"
```

```
android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
android:paddingBottom="@dimen/activity_vertical_margin"
tools:context="org.orscheduler.UserManagement">
```

```
<LinearLayout
```

```
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
```

```
<TextView
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/header1"
    android:textStyle="bold"
    android:id="@+id/header1" />
```

```
<TextView
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/header2"
    android:textStyle="bold"
    android:id="@+id/header2" />
```

```
<TextView
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance=
"?android:attr/textAppearanceLarge"
    android:text="User Accounts"
    android:id="@+id/admin.text.title"
```

```
        android:textSize="35dp" />

        <ListView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/admin.listView.users" />

    </LinearLayout>

</RelativeLayout>
```

activity_view_schedules.xml

```
<android.support.v4.view.ViewPager
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
        android:id="@+id/pager"
    android:layout_width="match_parent"
        android:layout_height="match_parent"
    tools:context="org.orscheduler.ViewSchedules" >
</android.support.v4.view.ViewPager>
```

activity_welcome.xml

```
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
    android:layout_height="match_parent"
        android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
        tools:context="org.orscheduler.Welcome">
```



```

<LinearLayout
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/header1"
        android:textStyle="bold"
        android:id="@+id/header1" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/header2"
        android:textStyle="bold"
        android:id="@+id/header2" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance=
"?android:attr/textAppearanceLarge"
        android:text="Welcome!"
        android:id="@+id/welcome.text.title"
        android:textSize="35dp" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

```

```

        android:textAppearance=
"?android:attr/textAppearanceSmall"
        android:text="Firstname"
        android:id="@+id/welcome.text.firstname"
        android:textSize="25dp"
        android:paddingTop="0dp" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance=
"?android:attr/textAppearanceSmall"
    android:text="Lastname"
    android:id="@+id/welcome.text.lastname"
    android:textSize="25dp" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance=
"?android:attr/textAppearanceMedium"
    android:text="Usertype"
    android:id="@+id/welcome.text.usertype"
    android:textSize="20dp" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:text=""
    android:id="@+id/welcome.text.messageCount"
    android:layout_marginTop="20dp"
    android:layout_marginBottom="20dp"

```

```
android:textSize="20dp" />
```

```
<ScrollView
```

```
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:id="@+id/scrollView4" >
```

```
<LinearLayout
```

```
    android:orientation="vertical"  
    android:layout_width="fill_parent"  
    android:layout_height="fill_parent">
```

```
<Button
```

```
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:text="View Schedules"  
    android:id="@+id/welcome.button.view"  
    android:layout_gravity="center_horizontal" />
```

```
<Button
```

```
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:text="Generate Reports"  
    android:id="@+id/welcome.button.print"  
    android:layout_gravity="center_horizontal" />
```

```
<Button
```

```
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:text="Messaging"  
    android:id="@+id/welcome.button.messaging"  
    android:layout_gravity="center_horizontal" />
```

```

        <Button
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Change Password"
            android:id="@+id/welcome.button.updateUser"
            android:layout_gravity="center_horizontal" />

        <Button
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Log Out"
            android:id="@+id/welcome.button.logOut"
            android:layout_gravity="center_horizontal" />
    </LinearLayout>
</ScrollView>
</LinearLayout>
</RelativeLayout>

```

admin_user_item.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance=
            "?android:attr/textAppearanceLarge"
        android:text="A"
        android:id="@+id/admin.text.icon"
        android:textSize="60dp"
    >

```

```

        android:paddingLeft="0dp"
        android:paddingRight="20dp" />

<LinearLayout
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance=
"?android:attr/textAppearanceLarge"
        android:text="Large Text"
        android:id="@+id/admin.text.username" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance=
"?android:attr/textAppearanceMedium"
        android:text="Medium Text"
        android:id="@+id/admin.text.fullname" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceSmall"
        android:text="Small Text"
        android:id="@+id/admin.text.usertype" />

</LinearLayout>

```

```
</LinearLayout>
```

message_item.xml

```
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="100dp"
    >
```

```
    <LinearLayout
        android:orientation="vertical"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content">
```

```
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textAppearance=
"?android:attr/textAppearanceSmall"
            android:text="Small Text"
            android:id="@+id/messageItem.text.readFlag"
            android:layout_gravity="right"
            android:textStyle="italic" />
```

```
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textAppearance=
"?android:attr/textAppearanceLarge"
            android:text="Large Text"
            android:id="@+id/messageItem.text.senderName" />
```

```
        <TextView
```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance=
"?android:attr/textAppearanceMedium"
        android:text="Medium Text"
        android:id="@+id/messageItem.text.message"
        android:scrollHorizontally="true"
        android:ellipsize="end"
        android:singleLine="true"/>

```

```

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance=
"?android:attr/textAppearanceSmall"
    android:text="Small Text"
    android:id="@+id/messageItem.text.date" />

```

```
</LinearLayout>
```

```
</RelativeLayout>
```

schedule_page_item.xml

```

<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context="orl.orscheduler.ViewSchedules$PlaceholderFragment">

```

```

<LinearLayout
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="fill_parent"
    android:layout_alignParentTop="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/header1"
        android:textStyle="bold"
        android:id="@+id/header1" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/header2"
        android:textStyle="bold"
        android:id="@+id/header2" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance=
"?android:attr/textAppearanceLarge"
        android:text="Monday"
        android:id="@+id/schedPageItem.text.day"
        android:textSize="35dp" />

    <TextView

```



```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance=
"?android:attr/textAppearanceMedium"
        android:text="01 January"
        android:id="@+id/schedPageItem.text.date"
        android:textSize="20dp" />

```

```
<ScrollView
```

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/scrollView" >

```

```
<LinearLayout
```

```

        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="fill_parent">

```

```
<TextView
```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance=
"?android:attr/textAppearanceSmall"
        android:text="Room 1"
        android:id="@+id/schedPageItem.text.room1"
        android:textSize="25dp"
        android:paddingTop="20dp" />

```

```
<LinearLayout
```

```

        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

```

```
<Button
```

```

        android:layout_width="80dp"
        android:layout_height="80dp"
        android:id="@+id/schedPageItem.btn.rm1slt1"
    />
<Button
    android:layout_width="80dp"
    android:layout_height="80dp"
    android:id="@+id/schedPageItem.btn.rm1slt2"
    />
<Button
    android:layout_width="80dp"
    android:layout_height="80dp"
    android:id="@+id/schedPageItem.btn.rm1slt3"
    />
</LinearLayout>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance=
"?android:attr/textAppearanceSmall"
    android:text="Room 2"
    android:id="@+id/schedPageItem.text.room2"
    android:textSize="25dp"
    android:paddingTop="10dp" />

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <Button
        android:layout_width="80dp"

```

```

        android:layout_height="80dp"
        android:id="@+id/schedPageItem.btn.rm2slt1"
    />
<Button
    android:layout_width="80dp"
    android:layout_height="80dp"
    android:id="@+id/schedPageItem.btn.rm2slt2"
    />
<Button
    android:layout_width="80dp"
    android:layout_height="80dp"
    android:id="@+id/schedPageItem.btn.rm2slt3"
    />
</LinearLayout>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance=
"?android:attr/textAppearanceSmall"
    android:text="Room 3"
    android:id="@+id/schedPageItem.text.room3"
    android:textSize="25dp"
    android:paddingTop="10dp" />

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <Button
        android:layout_width="80dp"
        android:layout_height="80dp"

```

```

        android:id="@+id/schedPageItem.btn.rm3slt1"
        />
    <Button
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:id="@+id/schedPageItem.btn.rm3slt2"
        />
    <Button
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:id="@+id/schedPageItem.btn.rm3slt3"
        />
    </LinearLayout>
</LinearLayout>
</ScrollView>

</LinearLayout>
</RelativeLayout>

```

menu_create_schedule.xml

```

<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
        tools:context="orl.orscheduler.CreateSchedule">
    <item android:id="@+id/action_settings"
        android:title="@string/action_settings"
        android:orderInCategory="100" app:showAsAction="never" />
</menu>

```

menu_home.xml

```

<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"

```

```
xmlns:tools="http://schemas.android.com/tools"
    tools:context=".Home">
<item android:id="@+id/action_settings"
    android:title="@string/action_settings"
    android:orderInCategory="100" app:showAsAction="never" />
</menu>
```

menu_inbox.xml

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context="orl.orscheduler.Inbox">
<item android:id="@+id/action_settings"
    android:title="@string/action_settings"
    android:orderInCategory="100" app:showAsAction="never" />
</menu>
```

menu_log_in.xml

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context="orl.orscheduler.logInScreen">
<item android:id="@+id/action_settings"
    android:title="@string/action_settings"
    android:orderInCategory="100" app:showAsAction="never" />
</menu>
```

menu_messaging.xml

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context="orl.orscheduler.Messaging">
```

```
<item android:id="@+id/action_settings"
      android:title="@string/action_settings"
      android:orderInCategory="100" app:showAsAction="never" />
</menu>
```

menu_print_pdf.xml

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto"
      xmlns:tools="http://schemas.android.com/tools"
      tools:context="org.orscheduler.PrintPdf">
  <item android:id="@+id/action_settings"
        android:title="@string/action_settings"
        android:orderInCategory="100" app:showAsAction="never" />
</menu>
```

menu_register.xml

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto"
      xmlns:tools="http://schemas.android.com/tools"
      tools:context="org.orscheduler.Register">
  <item android:id="@+id/action_settings"
        android:title="@string/action_settings"
        android:orderInCategory="100" app:showAsAction="never" />
</menu>
```

menu_send_message.xml

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto"
      xmlns:tools="http://schemas.android.com/tools"
      tools:context="org.orscheduler.SendMessage">
  <item android:id="@+id/action_settings"
        android:title="@string/action_settings"
        android:orderInCategory="100" app:showAsAction="never" />
</menu>
```

</menu>

menu_user_management.xml

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto"
      xmlns:tools="http://schemas.android.com/tools"
      tools:context="org.orscheduler.UserManagement">
    <item android:id="@+id/action_settings"
          android:title="@string/action_settings"
          android:orderInCategory="100" app:showAsAction="never" />
</menu>
```

menu_view_schedules.xml

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto"
      xmlns:tools="http://schemas.android.com/tools"
      tools:context="org.orscheduler.ViewSchedules">
    <item android:id="@+id/action_settings"
          android:title="@string/action_settings"
          android:orderInCategory="100" app:showAsAction="never" />
</menu>
```

menu_welcome.xml

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto"
      xmlns:tools="http://schemas.android.com/tools"
      tools:context="org.orscheduler.Welcome">
    <item android:id="@+id/action_settings"
          android:title="@string/action_settings"
          android:orderInCategory="100" app:showAsAction="never" />
</menu>
```

dimens.xml

```
<resources>
    <!-- Default screen margins, per the Android Design
         guidelines. -->
    <dimen name="activity_horizontal_margin">16dp</dimen>
    <dimen name="activity_vertical_margin">16dp</dimen>
</resources>
```

dimens.xml (w820dp)

```
<resources>
    <!-- Example customization of dimensions originally defined
         in res/values/dimens.xml
         (such as screen margins) for screens with more than
         820dp of available width. This
         would include 7" and 10" devices in landscape (~960dp
         and ~1280dp respectively). -->
    <dimen name="activity_horizontal_margin">64dp</dimen>
</resources>
```

strings.xml

```
<resources>
    <string name="app_name">ORL OR SCHEDULER</string>

    <string name="hello_world">Hello world!</string>
    <string name="action_settings">Settings</string>
    <string name="title_activity_register">Register</string>
    <string name="title_activity_view_schedules">View
        Schedules</string>

    <string name="title_section1">Section 1</string>
    <string name="title_section2">Section 2</string>
    <string name="title_section3">Section 3</string>
    <string name="title_activity_welcome">Welcome</string>
```



```
<string name="title_activity_create_schedule">Create OR  
    Schedule</string>  
<string name="title_activity_print_pdf">Print OR  
    Schedule</string>  
<string name="title_activity_send_message">Send  
    Message</string>  
<string name="title_activity_messaging">Messaging</string>  
<string name="title_activity_inbox">Inbox</string>  
<string name="title_activity_user_management">User Account  
    Management</string>  
<string name="header1">DEPARTMENT OF  
    OTORHINOLARYNGOLOGY</string>  
<string name="header2">UP-PGH</string>  
<string name="title_activity_login">Log In</string>  
  
</resources>
```

styles.xml

```
<resources>  
  
    <!-- Base application theme. -->  
    <style name="AppTheme"  
        parent="Theme.AppCompat.Light.DarkActionBar">  
        <!-- Customize your theme here. -->  
    </style>  
  
</resources>
```

XII. Acknowledgement

First and for most, I would like to thank the Lord our God for guiding me through the years, for giving me the knowledge and wisdom in making tough decisions and for giving me all the strengths and courage I need everyday as I welcome new challenges in my life. You are one of the main reasons as to how I got here.

My biggest gratitude and thankfulness to the greatest, most amazing and the best parents in the world, Edita Lanuza and Gabriel Lanuza, for giving me all the support I need and all the needs a child should have. They never get tired understanding me especially in time of my failures. My dad drove me all the way to Manila from Legazpi City to enroll me in the university. It was an unexpected journey for me because my parents were not so sure to enroll me since we don't have the money for all the expenses. But I really wanted to study in the most prestigious and the best univeristy in the country. Fortunately, with the financial help of my uncle, Jojo Navera, and auntie, Alma Yap, I was able to start my dream. My biggest thanks and appreciation to my tito and tita. My uncle became my second Dad in the city. He took care of me while I was away with my parents. He and her wife, tita Mean always give me non-stop advices and it has been a great help for me to push through. My family is my biggest inspiration and motivation to pursue my dream.

A very special mention to my girlfriend, bestfriend and another inspiration, Mara Magsanoc, for helping me in every way possible. She never fails to make me feel okay whenever things get rough and is always there for me no matter what. Mara has been an integral part to my thesis as to how and why I accomplished it right from the start all the way to the finish line. The time when the thesis proposal is almost near and I was already cramming and don't have much time to continue, she did the chapter two. Particularly, the Review and Related Literature. As far as I know, chapter two is one of the most tedious and difficult chapters to deal with. I am so grateful and thankful to have her making me the most luckiest man in the world.

I want to express my gratitude to Dr. Philipp Fullante who is an ENT consultant in the Department of Otorhinolaryngology in the PGH for giving me

the thesis topic. If not for him, I wouldn't be doing anything and still don't have any thesis topic until now. I thank him so much for considering and responding to my request. It was all his idea from his brilliant mind what I am working on.

Another person I want to show my gratitude, John Cagampan, a good friend of mine. He helped and provided me some assistance in my excursion on my thesis. Android developing is a new thing for me. I have no idea how to make an android project. Although I researched and looked for things about Android developing, It is still not as easy as it takes. Thankfully, John taught me of the fundamentals and the essentials on Android programming. He is one of the main reason why I succeeded in my thesis.

My adviser, sir Bernie Terrado, who guided me on the road to victory. He had given me great insights and ideas to further improve my capability in finishing my thesis. Even though he was not present in my proposal nor in my defense, I know he will support me and cheer for me all the way to the end. My adviser is also one of the reason as to where I got all the courage to present and defend my thesis in front of the most respected and great professors in the DPSM.

To my fraternity brothers and my sorority sisters who gave me all the fun and happiness I never expected to have in my college life. Along with them, I learned so many amazing and beautiful things that I was able to apply every single thing of it in my life. I thank all the brods especially my batchmates in the fraternity for understanding and helping me in times of my struggle in school especially during my battle against my thesis. We have each others back and trust. There is no doubt about it. They were and will always be an important part in my life.

My dormates who became my good friends along the way. They also have been a big part in my journey. With just little conversation, bonding and a few drinks, somehow, everything is alright. From time to time, they have given me the emotional support that I need to get pass all sorts of hindrances during my stay in the dorm alongside with my college life.

To my highschool friends that had been giving me abundant support that a friend needed. They have been a great part of my struggle and one of the reasons why I haven't given up. There are a lot of times and situations that I almost quit and give up. Luckily for me, I have such an exceptional friends that exists. Eweer

since highschool, we have been helping and caring for each other on our gloomest and bluest days of our lives especially in our problems in school. They will surely be my friends forever.

All my years in the university, there are a lot of people that I've met and some of them I became friends with. Of course, I will not forget my blockmates which have been my companion in my expedition in school. They all become great experiences in my college life as we all together conquer all the fears and worries in the world. We have been helping each other on our problems especially in school as we face every hardships and get through every obstacles we encounter.

I've always been grateful and thankful for all these people I wrote. They are really the best and greatest things that ever happened to me as I pursue my dream and get what I want in life. They have been with me and always there for me whenever I need anything. They are also one of the reasons that I still continued and carry on with my combat in school and never ever gave up for I know they are with me and I have them. Studying and graduating in UP is the biggest and toughest challenge I ever had. All my hard works, sleepless nights, sufferings, triumphs and success are all worth it and all of these are all for my parents. I dedicate my thesis and everything to them.

Lastly, I would like to thank the University of the Philippines Manila for giving me the greatest and unforgettable experience that I will always remember and cherished each step I take in my life as I reach the top and become the best of the best.