A project report on

A SMART APPLICATION FOR ONLINE MEDICAL SERVICES

Submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

by

PRAVALLIKA K
(1210314132)

Under the esteemed guidance of

Dr . J. HYMA, Assistant Professor,
Department of Computer Science and Engineering
GITAM (Deemed- to- be University)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GITAM Institute of Technology, GITAM (Deemed - to – be University)
Accredited by NAAC with 'A+'Grade
Visakhapatnam – 530045
2014-2018

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GITAM INSTITUE OF TECHNOLOGY
GITAM (Deemed - to - be University)
Accredited by **NAAC** with '**A+**' Grade
Visakhapatnam – 530045

CERTIFICATE

This is to certify that the project entitled **"A SMART APPLICATION FOR ONLINE MEDICAL SERVICES"** is being submitted by **SANTOSH KUMAR K (1210314134), HEMANTH KUMAR K (1210314129), ISHAN C S S (1210314125)** and **PRAVALLIKA (1210314132)** in partial fulfilment of the academic requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering to GITAM (Deemed- to- be University), is a record of bona fide work carried out by them under my guidance and supervision. The result obtained in the project has not been submitted to any other industry or institute for the award of degree.

Project Guide

Dr. J.Hyma

Assistant Professor, Dept. of CSE
GIT
Visakhapatnam 530045

Head of the Department

Prof. K. Thammi Reddy

Professor, Dept. of CSE
GIT
Visakhapatnam 530045

DECLARATION

We hereby declare that this project titled as "A SMART APPLICATION FOR ONLINE MEDICAL SERVICES" is an original and authentic work done in the Department of Computer Science and Engineering, GITAM Institute of Technology, GITAM (Deemed-to- be University), Visakhapatnam, under the guidance of Dr. J.HYMA, Assistant Professor, Department of Computer Science and Engineering, GITAM (Deemed- to- be University) submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering during the year 2017-2018. The matter embodied in this project work has not been submitted earlier for award of any degree to the best of our knowledge.

PRAVALLIKA K **1210314132**

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without mentioning the people who made it possible and whose constant guidance and encouragement crown all the effort with success.

We would like to express our gratitude and sincere thanks to **Dr. J.HYMA,** Assistant Professor, Department of Computer Science and Engineering for his constant guidance, supervision and motivation in completing the project work.

We would like to personally place on our record our heartfelt thanks to **Shri.T. SRINIVAS,** Assistant Professor, Department of Computer Science and Engineering and with the help of **Dr. P. SANKARA RAO**, Assistant Professor, Department of Computer Science and Engineering, being our Project Coordinators and for their direct involvement and overwhelming cooperation in the completion of project.

We feel elated to express our gratitude to **Prof. K. THAMMI REDDY**, Head of the Department, Computer Science and Engineering, for his encouragement all the way during this project. His annotations, insinuations and criticisms are keys behind the successful completion of this project.

We also take this opportunity to express our heartfelt thanks for his time and guidance to **Prof. P.LAKSHMI PRASAD**, Principal, GITAM Institute of Technology, GITAM, Visakhapatnam the teaching and non-teaching staff of the department for their perspective comments and suggestions.

PRAVALLIKA (1210314132)

ABSTRACT

Life is becoming too busy to get medical appointments in person and to maintain a proper health care. The main idea of this work is to provide ease and comfort to patients while taking appointment from doctors and it also resolves the problems that the patients has to face while making an appointment. The android application “A Smart Application for Online Medical Services” acts as a client whereas the database containing the doctor’s details, patient’s details and appointment details is maintained by a website that acts as a server.The purpose of the application is to automate the existing manual system by the help of computerized equipment and full-fledged computer software, fulfilling their requirements, so that their valuable data or information can be stored confidentially with easy accessing and manipulation at the same time. The required software and hardware are easily available and easy to work with.

The proposed project is a smart appointment booking system that provides patients or any user an easy way of booking a doctor’s appointment online. This application overcomes the issue of managing and booking appointments according to user’s choice or demands. The task sometimes becomes very tedious for the compounder or doctor himself in manually allotting appointments for the users as per their availability. Hence this project offers an effective solution where users can view various booking slots available and select the preferred time. This system helps to reduce the waiting time of the patient. User can select the appointment time according to his preference.

Keywords: Appointment, online application, android, hospital, scheduling, track, healthcare.

INDEX

INDEX PAGE NO

1. **INTRODUCTION**

This Online Appointment System has been developed to override the major problem in manual system which is gigantic queuing wait times. This software is supported to eliminate and in some cases reduce the hardships faced by this existing systems. Moreover this system is designed for patients to carry out operations in a seamless and effective manner. The application is reduced as much as possible to avoid errors while entering the data. It also provides error message while entering invalid data. No formal knowledge is needed for the user to use a system. Thus by this it proves it is user-friendly. Doctor appointment system, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on record keeping. Thus, it will help organization in better utilization of resources. Every organization, whether big or small, has challenges to overcome and managing the information of Appointment, Doctor, Booking. Every Doctor Appointment System has different doctor needs, therefore we designed a system that is adapted to our requirements. This is designed to help user to ensure that their organization is equipped with right level of details. The proposed project is a smart appointment booking system that provides patients or any user an easy way of booking a doctor's appointment online. This is a web based application that overcomes the issue of managing and booking appointments according to user's choice or demands. The task sometimes becomes very tedious for the compounder or doctor himself in manually allotting appointments for the users as per their availability.

Research on outpatient clinics shows that waiting times are patients' main dissatisfaction with hospital services. According to doctors and personnel, overtime and peak workloads are potential threats for the quality of care and the quality of labor, because they increase stress and time pressure. This research focuses on outpatient scheduling as a means to solve these problems for outpatient clinics. According to ,access time is the time between the patient's request for an appointment and his arrival at the outpatient clinic. A patient's internal waiting time is the period between the scheduled starting time and the actual starting time of his consultation. Waiting time due to a patient's early arrival is extracted from the internal waiting time, since it is not a consequence of the appointment system .Hence this project offers an effective solution where users can view various booking slots available and select the preferred date and time. The already booked space will be marked yellow and will not be available for anyone else for the specified time. This system also allows users to cancel their booking anytime. The system provides an additional feature of calculating monthly earnings of doctor. Doctor has to just feed the system regularly with daily earnings and the system automatically generates a report of total amount earned at the end of the month. The application uses "Android" as a front-end and "SQLite" database as the back-end.

2. **LITERATURE REVIEW**

Waiting time simply means a period of time which one must wait in order for a specific action to occur, after that action is requested or mandated (Fernandes et al., 1994). Patients' waiting time has been defined as "the length of time from when the patient entered the outpatient clinic to the time the patient actually received his or her prescription" (Jamaiah, 2003). It is defined as the total time from registration until consultation with a doctor. There were two waiting times, the first is time taken to see a physician and the second is time to obtain medicine (Suriani, 2003). This paper deals with the waiting time to see physicians. Long waiting times are a serious problem for patients using urban health centre's in developing countries (Bachmann, 1998). A block appointment system was introduced and evaluated in a large South African health centre.

Waiting times of all patients were measured over one-week period before and after the implementation of appointments. Focus groups and individual interviews were conducted with staff and patients. After introducing appointments, patients with acute and chronic illnesses and having appointments had significantly shorter waits time than similar patients without appointments (Mahomed, 1998). Appointments had no benefits for patients not seeing doctors or collecting repeat medication. There was, however, an overall increase in patients' waiting times after introducing the system, mainly due to one typical day in the follow-up study. Focus groups and interviews revealed that staff were skeptical at baseline but at follow-up were positive about the system. Patients were enthusiastic about the appointment system at all stages. The study shows that block appointments can reduce patients' waiting times for acute patients, but may not be suitable for all patients. Staff and patients had different views, which converged with experience of the new system (Mahomed, 1998). From this we have learned about the waiting time which can be developed.

A patient appointment system or appointment schedule for health care center started long time ago (Harper, 2003). Management of patients' appointments has earlier works and has developed simplified queuing models and fairly static scheduling conditions. Another attempt was made to calculate the waiting time between patient and doctor using the mathematical queuing models to minimize waiting time (Gamlin, 2003). However; traditionally the appointment system has considered that the doctor time is more important than patient time (Wijewickrama, 2005). So an appointment system was designed to minimize the doctor idle time but current designing of an appointment system is based on decisive factors with respect to both the patient and doctor (Takakuwa, 2005). The patient appointment system has complex structures because it represents the patient appointment time in the healthcare center and controls the patient waiting time based on the type and the period of patient appointment (Gamlin, 2003). Moreover, a patient appointment system is International Journal of Computer Science & Information Technology (IJCSIT) Vol 6, No 4, August 2014 62 meant for: managing doctor's ideal time. From this we have learned about the patients appointment system which can be developed.

Past research shows that the longer the appointment delay which is defined as the time between the day a patient requests an appointment and her actual appointment date, the higher the chances that he/she will cancel or not show up (Gallucci et al. 2005). This suggests

an obvious way of minimizing no-shows and cancellations: this is done by asking the patients to come right away or make appointment requests on the day they want to be seen (Murray, 2000). This is called an open access (OA) or advanced access policy (Tantau, 2000), and of late it has become a popular paradigm in practice and the subject of active research. Several authors report on their experiences in implementing OA, both positive and negative (Dixon et al. 2006). Some practitioners strongly advocate OA (Murray and Tantau 2000), and there are some who are strongly against it (Lamb, 2002). 2.2.2 Managing Patients' Appointment system According to Dexter (1999), managing patient appointment system is a computer application used to manage and reduce the patient waiting time in the health care center. Some health care centers do not use any appointment system. So it has a longer average patients' waiting time than the health care center that adopts the patients' appointment system.

While patients can wait for more than one hour to be attended to by a physician in a health care center, they also can feel that they are being disregarded and treated unfairly. So when patients are given the time of appointment in a health care centre, they can evaluate the quality of service in the centre (Dexter, 1999). Hence, developing patients' appointment process for health care centres necessitates the use of a sophisticated queuing model that captures much of the real system's features (saving time, reducing idle time, etc). Therefore the appointment schedule represents the real situation in the health care centre faced by patient appointment schedulers (Rohleder, 2002). On the other hand, the standard practice for scheduling and processing patient appointments are based on the nature of treatments of the patients and that better approaches more sensitive to patient needs are desirable (Klassen, 2002). From this we have learned about the appointment delay which can be developed.

An online system is also known as a web based system. A web is made up of page that is commonly known as web page or web site, and a web site is a computer program that runs a web server that provides access to a group of related web pages (Alex, 2000). A system is a set of independent components working together to achieve a common objective. Therefore a web based system is a system that is accessible over the internet by a user in order to achieve a particular task for a given purpose. The Internet is a system that is use to connect computers and computer networks. It helps to link millions of computer networks all over the world and it allows the users to get information stored on other computers from a long distance (James, 1999). According to Chua (2010) the public demand for better healthcare system and the alarming number of missed appointments have forced the healthcare sector to recognize how they deliver care services. From this we have learned about the online booking system which can be developed.

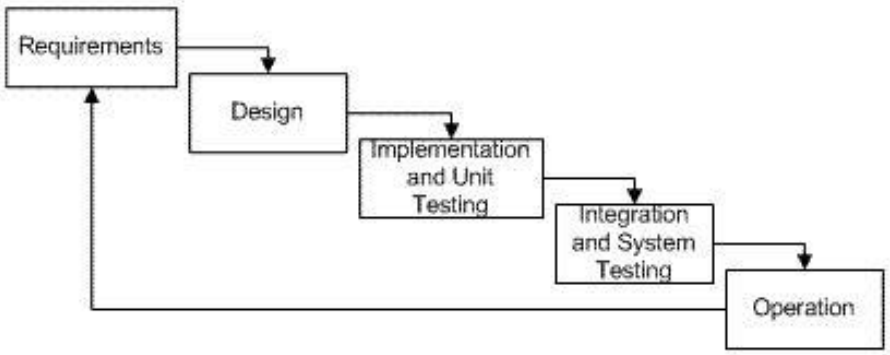
3. PROBLEM STATEMENT

To make an application that eliminate the gigantic queuing wait times which is major problem in manual system.

3.1. **OBJECTIVE**

- The main objective of the proposed system on Doctor Appointment System is to manage the details of Doctor, Appointment, Patient, Booking.
- It manages all the information about the Doctor and Hospitals. The administrator has a total control over the data.

3.2. **SYSTEM MODEL**



The methodology used for developing this system is INCREMENTAL. This model is used here because detailed input, output requirements are present in the beginning, though some details can evolve during process. Also, the work is divided into parallel development cycles where each one undergoes all the phases of development. Thus, the process for development will be iterative.

Advantages of Incremental model:

• Generates working software quickly and early during the software life cycle.

4. EXISTING SYSTEM

The current system which is used massively is manual system. The problem of the existing system restricts the independency and also is time consuming for patients. The patient has to follow the queue line which is usually very long especially in a country like India. If a patient has any injury on arms or legs, filling an appointment form or standing in the long queues would be a huge problem, and the patient alone cannot do such tasks unless accompanied by somebody else to help them with these issues, which might not happen always.

One application developed to manage patients' appointment scheduling has used exponential enter arrival times. This model assumes that the exponential enter arrival times could not be directly validated by date, and it is limited due to the nature of the appointment scheduling. Since appointments are scheduled in the future, the exact model of call arrivals will only have limited impact on measures related to the time between the call and the appointment time. For this reason, the challenge for making appointment system is designing a suitable system based on the health care procedure environment.

Hence, the appointment provider in the health care centre can schedule a patient into an appropriate time slot on a given day. Klassen developed another method for managing patients' appointment using multiple schedule appointment in multiple period environments. Patients can call for any appointment time but if the period time is full, they should replace the appointment to another time.

Moreover, various combinations for multi appointment and double booking are measured and recommended for different operational use depending on the health care environment because the varying appointment request has little effect on appointment system performance, especially maintaining acceptable performance, except when the system has the overloaded option.

Many studies about patients' appointment have found that there are rules or policies for scheduling appointment system such as no scheduling for more than 20 or 30 clients and the best schedule is to place two patients in the first appointment and spread the rest consistently over a period based on average service times. On the other hand, a patient can call for an appointment without knowledge of the type of appointment and appointment queue number and the patient is not aware whether the appointment is variable or not. Sometimes the exact duration for each patient can be known but at other times this is unknown.

Another system developed by Mustafa, allows a registered patient, having user name and password, to access and explore the list of physicians alphabetically and select a physician whose email contact and profile are also provided. A patient can also view the physician working calendar to find out his/her working and nonworking day to make an appointment. When the patient selects, view calendar the patient can then choose any valid day in any month to make an appointment.

5. **PROPOSED SYSTEM**

The main objective of the proposed system is independency, of the patient. The system allows the user to book an appointment, know the doctors available timings, list of hospitals available in the particular city, user's scheduled appointments, alerts according to the appointments.The proposed project is a smart appointment booking system that provides patients or any user an easy way of booking a doctor's appointment online. This application overcomes the issue of managing and booking appointments according to user's choice or demands.

This project is a smart appointment booking system that provides patients or any user an easy way of booking a doctor's appointment online. This is a android application that overcomes the issue of managing and booking appointments according to user's choice or demands. The task sometimes becomes very tedious for the compounder or doctor himself in manually allotting appointments for the users as per their availability.

This system helps to reduce the waiting time of the patient. User can select the appointment time according to his preference.

6. **FEASIBILITY STUDY**

After doing the project, study and analysing all the existing or required functionalities of the system, the next task is to do the feasibility study for the project. All the projects are feasible- given unlimited resources and infinity time.

Feasibility study includes consideration of all the possible ways to provide a solution to the given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that the future changes can be easily done based on the future upcoming requirements.

6.1 **Economical Feasibility:**

This is a very important aspect to be considered while developing a project. We decided the technology based on minimum possible cost factor.

This project is economically feasible, because the software with which the project is developed are open source and are easily available. After deployment the user doesn't have to bear any charges except the data charges as this application requires internet.

6.2 **Technical Feasibility:**

This includes the study of functions, performance and constraints that may affect the ability to achieve an acceptable system. For this feasible study, we studied complete functionality to be provided in the system, as described in the System Requirement Specification, and checked if everything was possible, using different type of front-end and back-end platforms.

This project is technically feasible, because it is developed using Android Studio and SQLite database softwares which can be easily available on the internet, that are also very user-friendly. And this project is made for Kitkat and above versions of Android which can be used by any further version devices of Android.

6.3 **Operational Feasibility:**

No doubt, the proposed system is fully GUI based that is very user-friendly and all inputs to be taken all self-explanatory even to a layman. Besides, a proper training has been conducted to let know the essence of the system to the user so that, they feel comfortable with the new system. As far as studies concerned, the clients are comfortable and happy as the systems has cut down their loads and doing.

This project is operational feasible. Because the records of patients can be easily listed out by SQLite software and the GUI is user-friendly as it is developed by Android Studio.

7. SYSTEM REQUIREMETS AND SPECIFICATIONS

7.1 Hardware Requirements:

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware, A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of hardware requirements.

Hardware Requirements For Present Project:

1. VDU: Mobile phones, tablets, phablets
2. Input Devices: Smart phone devices
3. RAM: 512 MB
4. Processor: Quad-Core processor
5. Storage: 50-100 Mb of Micro SD card or internal memory

7.2 Software Requirements:

Software Requirements deal with defining software resource requirements and pre-requisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or pre-requisites are generally not included in the software installation package and need to be installed separately before the software is installed.

Software Requirements For Present Project:

1. Operating System: Android Lollipop or above
2. Android Studio
3. Firebase
4. SQLite

7.3 Android Studio:

Android is a mobile operating system which was developed by Google. It is based on a modified version of the Linux kernel and other open source software and designed primarily for touchscreen mobile devices such as smartphones and tablets.

Android studio is the official IDE (integrated development environment). Android studios help us to build the highest-quality apps for every Android device. The programming language you will be using is Java and this will be installed separately on your machine. Android Studio is simply where you will write, edit and save your projects and the files that comprise said projects.

Some of the features of android are:

Multi-tasking	User can jump from one task to another and same time various applications can run simultaneously.
Multi-Language	Supports single direction and bi-directional text.
Multi-touch	Android has native support for multi-touch which was initially made available in handsets such as the HTC Hero.
User interface	Android OS basic screen provides a beautiful and intuitive user interface.

Resizable widgets	Widgets are resizable, so users can expand them to show more content or shrink them to save space.
Storage	SQLite, a lightweight relational database, is used for data storage purposes.
Connectivity	GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC and WiMAX.
Android Beam	A popular NFC-based technology that lets users instantly share, just by touching two NFC-enabled phones together.

We have used android studio for the development of our front-end design which contains all the doctor details. We have used various UI controls provided by Android that allow us to build the graphical user interface for our application. In android the user interface of an app is made with a collection of **View** and **ViewGroup** objects.

The **View** is a base class for all UI components in android and it is used to create an interactive UI components such as Text View, Edit Text, Checkbox, Radio Button, etc. and it responsible for event handling and drawing.

The **View Group** is a subclass of **View** and it will act as a base class for layouts and layout parameters. The View Group will provide an invisible container to hold other Views or View Groups and to define the layout properties.

Input controls are the interactive components in our app's user interface. Android provides a wide variety of controls you can use in your UI, such as buttons, text fields, seek bars, check box and toggle buttons.

Some of the UI controls have been listed below:

Edit text	Edit Text is a predefined subclass of Text View that includes rich editing capabilities.
Text view	This control is used to display text to the user.
Auto complete text view	The Auto Complete Text View is a view that is similar to Edit Text, except that it shows a list of completion suggestions automatically while the user is typing.
Button	A push-button that can be pressed, or clicked, by the user to perform an action.
Image Button	An Image Button is an Absolute Layout which enables you to specify the exact location of its children. This shows a button with an image (instead of text) that can be pressed or clicked by the user.

Check Box	An on/off switch that can be toggled by the user. You should use check box when presenting users with a group of selectable options that are not mutually exclusive.
Radio Button	The Radio Button has two states: either checked or unchecked.
Radio Group	A Radio Group is used to group together one or more Radio Buttons.
Progress Bar	The Progress Bar view provides visual feedback about some ongoing tasks, such as when you are performing a task in the background.
spinner	A drop-down list that allows users to select one value from a set.

7.4 SQLite Database:

SQLite is a software library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. SQLite is the most widely deployed SQL database engine in the world. The source code for SQLite is in the public domain. This tutorial will give you a quick start with SQLite and make you comfortable with SQLite programming.

SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. It is a database, which is zero-configured, which means like other databases you do not need to configure it in your system.

SQLite engine is not a standalone process like other databases, you can link it statically or dynamically as per your requirement with your application. SQLite accesses its storage files directly.

- SQLite does not require a separate server process or system to operate (serverless).
- SQLite comes with zero-configuration, which means no setup or administration needed.
- A complete SQLite database is stored in a single cross-platform disk file.
- SQLite is very small and light weight, less than 400KiB fully configured or less than 250KiB with optional features omitted.
- SQLite is self-contained, which means no external dependencies.
- SQLite transactions are fully ACID-compliant, allowing safe access from multiple processes or threads.
- SQLite supports most of the query language features found in SQL92 (SQL2) standard.
- SQLite is written in ANSI-C and provides simple and easy-to-use API.
- SQLite is available on UNIX (Linux, Mac OS-X, Android, iOS) and Windows (Win32, WinCE, WinRT).

8. MODULE DESCRIPTION

The main objective of the project on Doctor Appointment System is to manage the details of Doctor, Appointment, Patient, Booking. It manages all the information about the Doctor and Hospitals. The project is totally built at administrative end and thus the administrator is guaranteed the access.

The purpose of the project is to build the application program to reduce the manual work for managing the Doctor, Appointment, Patient.

FUNCTIONALITIES

1. Provides all the information of list of hospitals in a particular city.
2. The Doctor Appointment System also manages the details of Doctor, Hospitals and Booking.
3. It also provides the information of Doctors and their availabilities in specific hospitals.
4. Manages the information of Appointment.

8.1 USER MODULE:

1. The user can register with their email address and create a password through this mobile application.
2. The user can view list of hospitals and select a desired hospital and make an appointment at a specific mentioned time.
3. The user can also view list of types of doctors and select a desired doctor in a specific available specializations and make an appointment at a specific mentioned time.
4. The user can give rating to the doctors based on their treatment.

8.2 ADMIN MODULE:

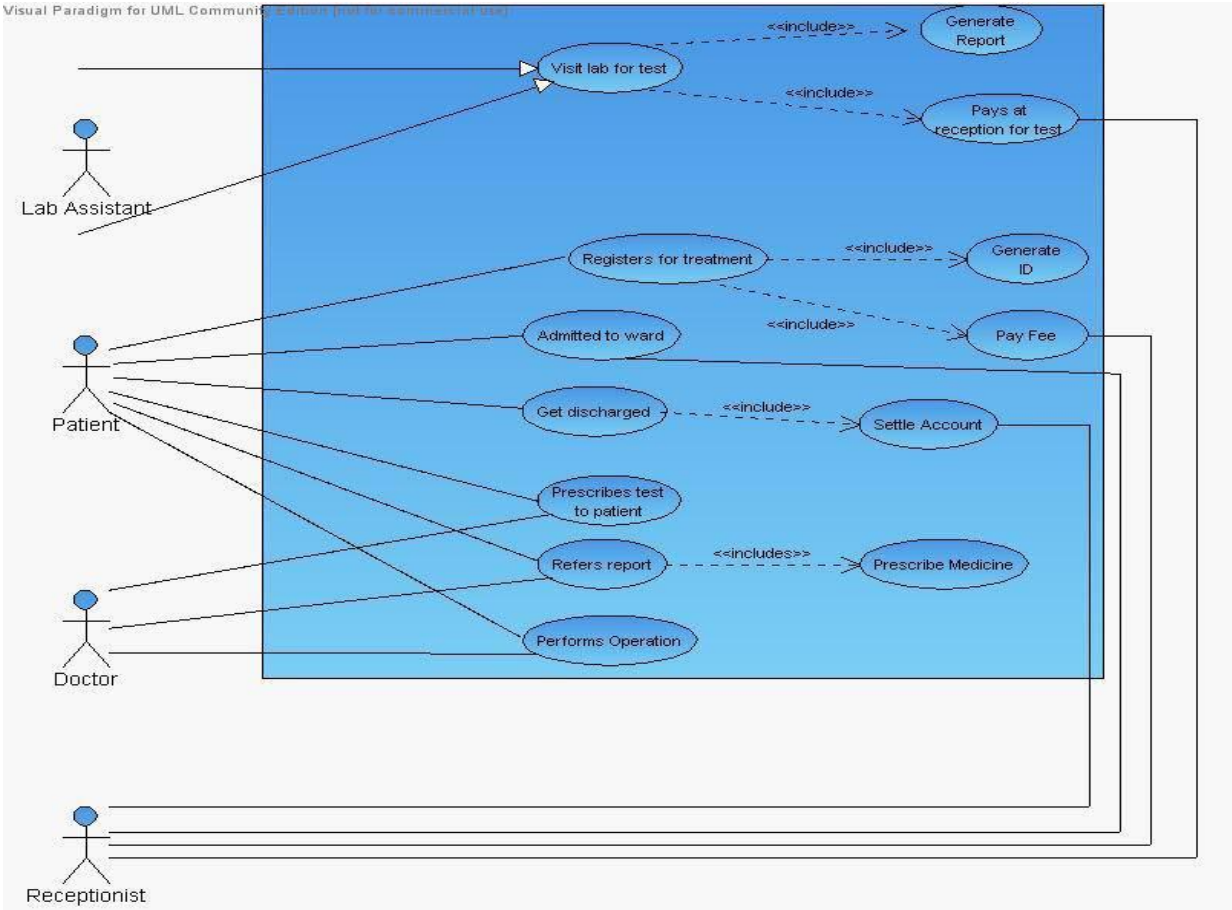
1. Admin can manage the data provided by the users.
2. Admin can view, modify and delete the records in the database.
3. Admin checks the security of the database.

9. UML DIAGRAMS

9.1 USECASE DIAGRAM:

A use case diagram in the Unified Modelling Language (UML) is a type of behavioural diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

Use Case diagrams are formally included in two modeling languages defined by the OMG: the Unified Modeling Language (UML) and the Systems Modeling Language (SysML).



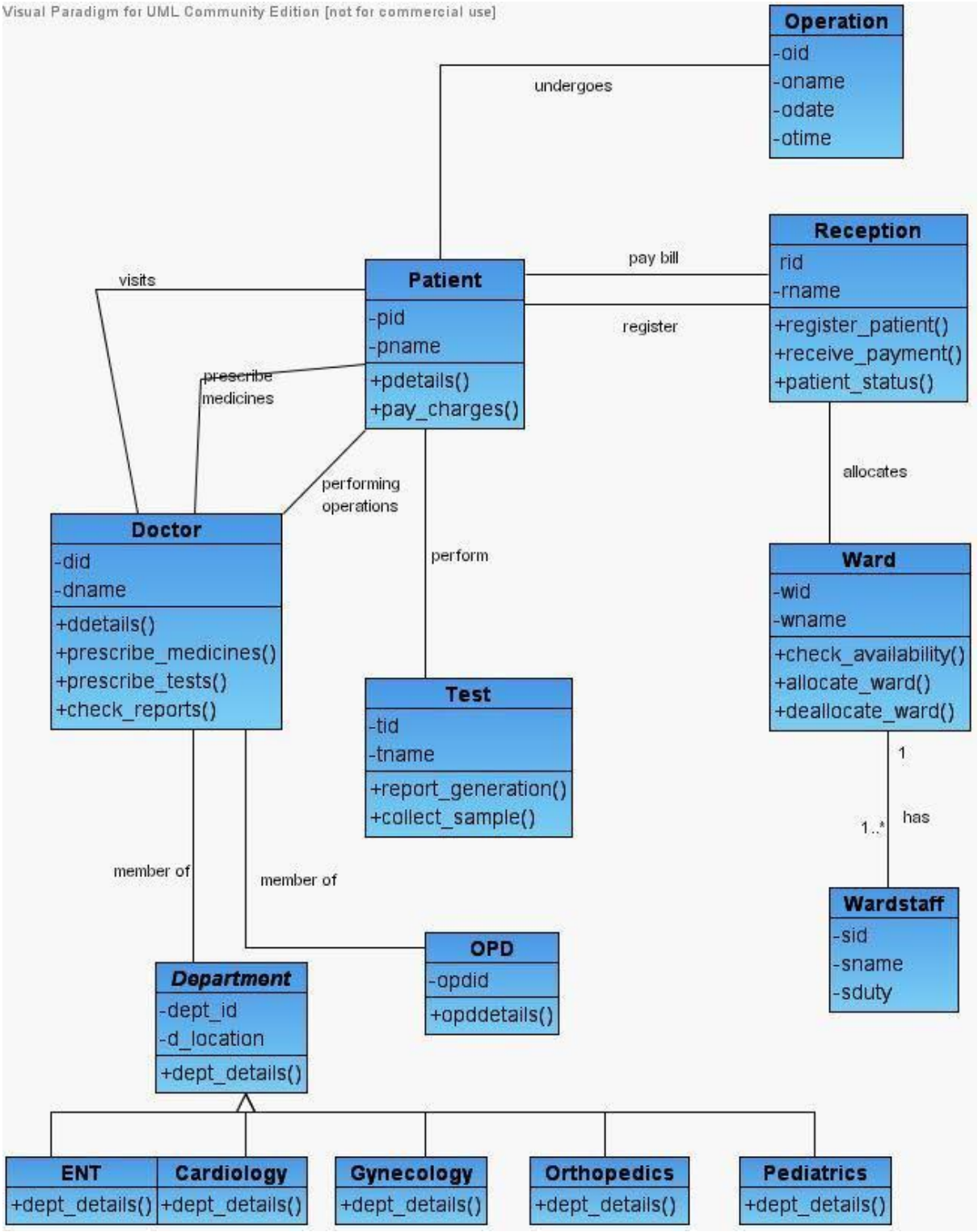
9.2 **CLASS DIAGRAM:**

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes. The class diagram is the main building block in object oriented modeling. They are being used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. The classes in a class diagram represent both the main objects and or interactions in the application and the objects to be programmed. In the class diagram these classes are represented with boxes which contain three parts:

A class with three sections.

- The upper part holds the name of the class
- The middle part contains the attributes of the class
- The bottom part gives the methods or operations the class can take or undertake

In the system design of a system, a number of classes are identified and grouped together in a class diagram which helps to determine the static relations between those objects. With detailed modeling, the classes of the conceptual design are often split in a number of subclasses.


Visual Paradigm for UML Community Edition [not for commercial use]
Operation
-oid
-oname
-odate
-otime
undergoes
Reception
rid
-rname
+register_patient()
+receive_payment()
+patient_status()
pay bill
register
Patient
-pid
-pname
+pdetails()
+pay_charges()
visits
prescribe medicines
performing operations
perform
allocates
Doctor
-did
-dname
+ddetails()
+prescribe_medicines()
+prescribe_tests()
+check_reports()
Test
-tid
-tname
+report_generation()
+collect_sample()
Ward
-wid
-wname
+check_availability()
+allocate_ward()
+deallocate_ward()
1
1..*
has
Wardstaff
-sid
-sname
-sduty
member of
member of
OPD
-opdid
+opddetails()
Department
-dept_id
-d_location
+dept_details()
ENT
+dept_details()
Cardiology
+dept_details()
Gynecology
+dept_details()
Orthopedics
+dept_details()
Pediatrics
+dept_details()

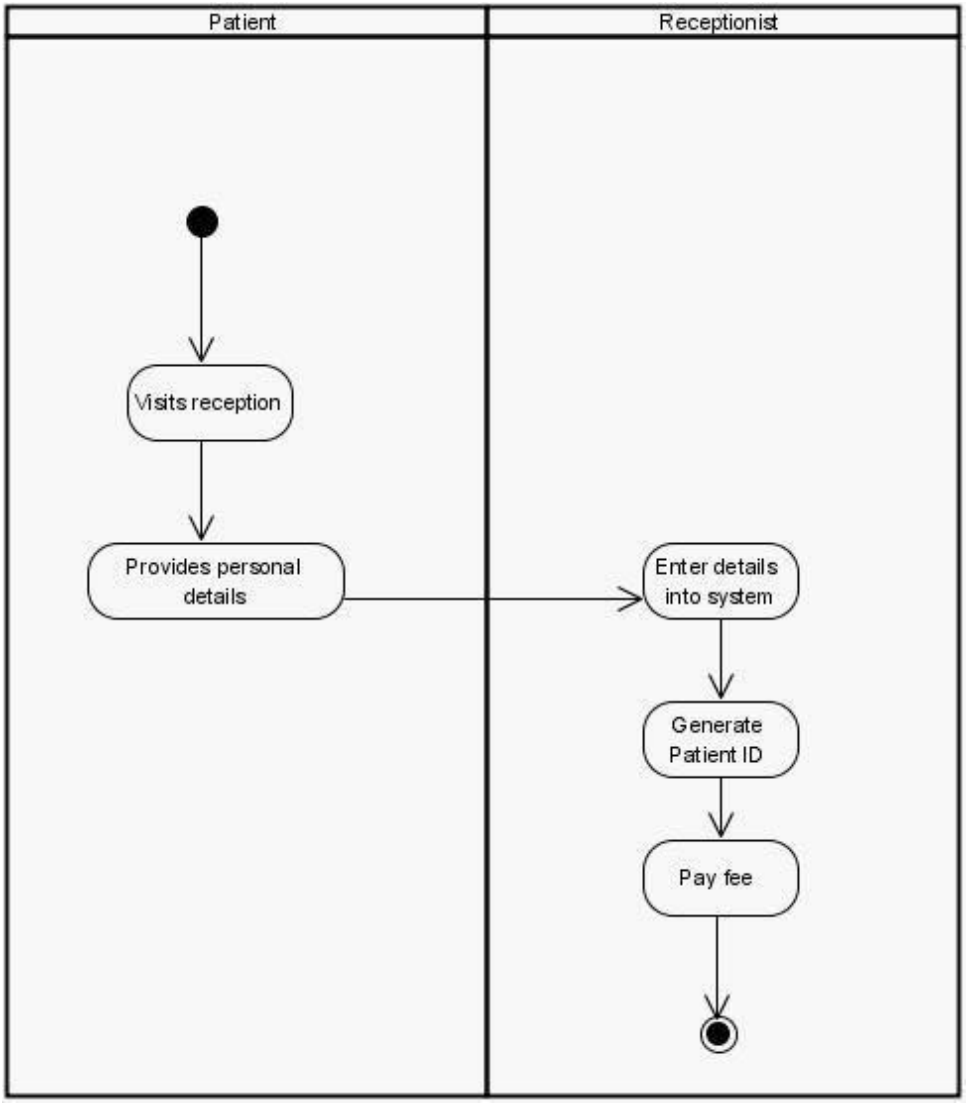
9.3 **ACTIVITY DIAGRAM:**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. [1] In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

Activity diagrams are constructed from a limited repertoire of shapes, connected with arrows. The most important shape types:

* Rounded rectangles represent activities;
* Diamonds represent decisions;
* Bars represent the start (split) or end (join) of concurrent activities
* A black circle represents the start (initial state) of the workflow;
* An encircled black circle represents the end (final state).

Arrows run from the start towards the end and represent the order in which activities happen. Hence they can be regarded as a form of flowchart. Typical flowchart techniques lack constructs for expressing concurrency. However, the join and split symbols in activity diagrams only resolve this for simple cases; the meaning of the model is not clear when they are arbitrarily combined with decisions or loops. While in UML 1.x, activity diagrams were a specialized form of state diagrams, in UML 2.x, the activity diagrams were reformalized to be based on Petri net-like semantics, increasing the scope of situations that can be modeled using activity diagrams. These changes cause many UML 1.x activity diagrams to be interpreted differently in UML 2.x

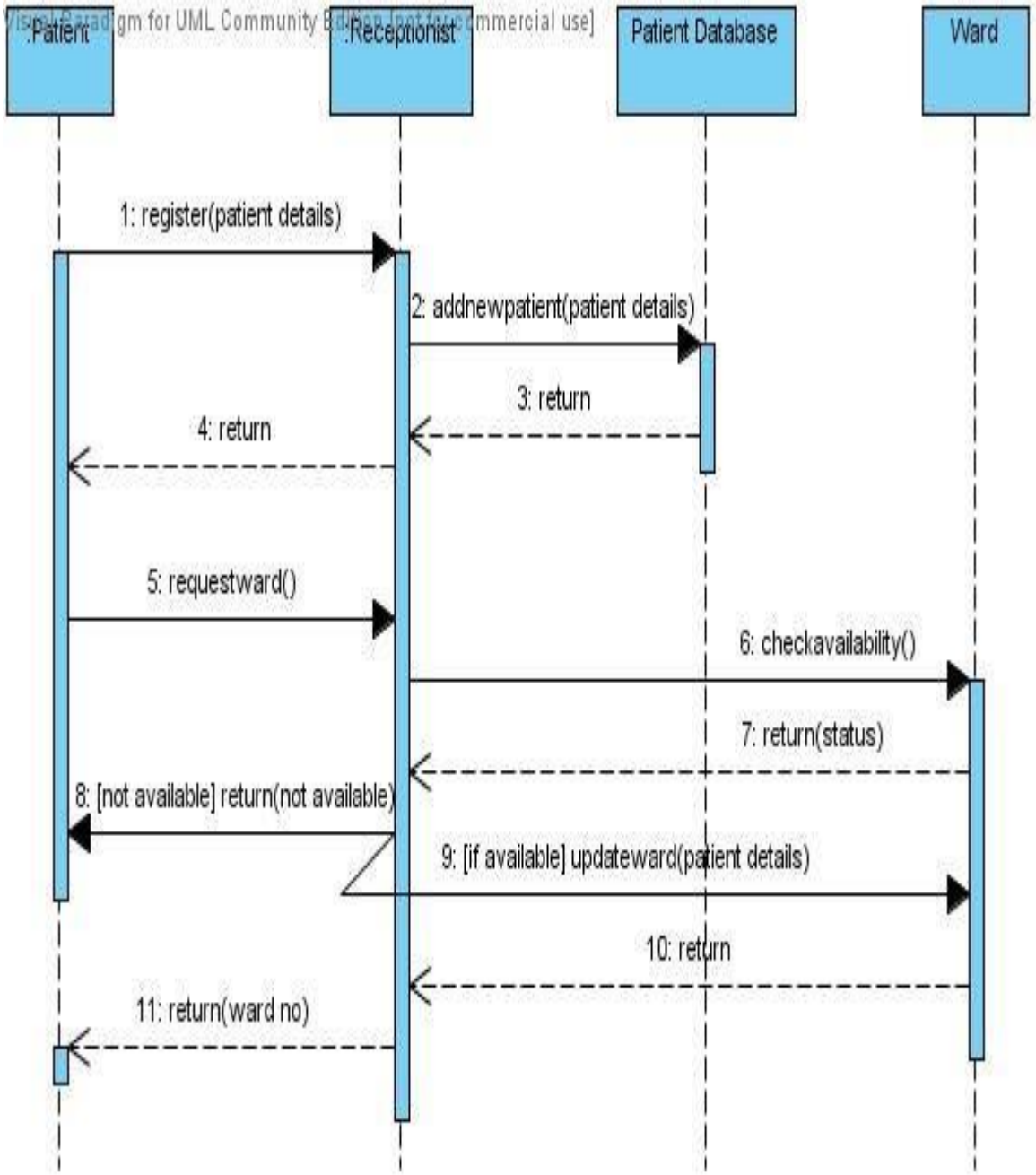


9.4 **SEQUENCE DIAGRAM:**

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams. A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

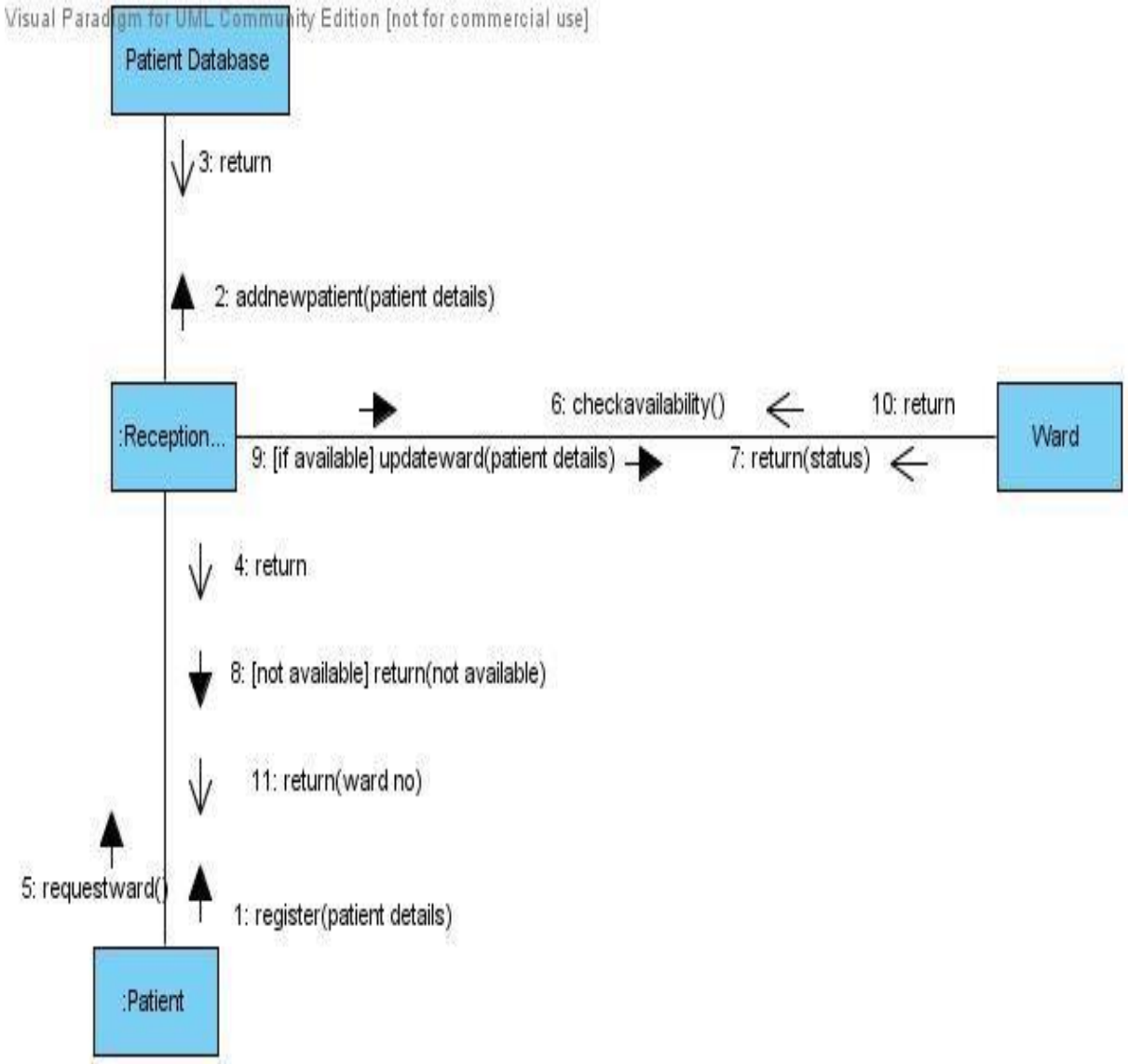
For instance, the UML 1.x diagram on the right describes the sequences of messages of a (simple) restaurant system. This diagram represents a Patron ordering food and wine, drinking wine then eating the food, and finally paying for the food. The dotted lines

extending downwards indicate the timeline. Time flows from top to bottom. The arrows represent messages (stimuli) from an actor or object to other objects. For example, the Patron sends message 'pay' to the Cashier. Half arrows indicate asynchronous method calls. The UML 2.0 Sequence Diagram supports similar notation to the UML 1.x Sequence Diagram with added support for modeling variations to the standard flow of events.



9.5 **COLLOBORATION DIAGRAM:**

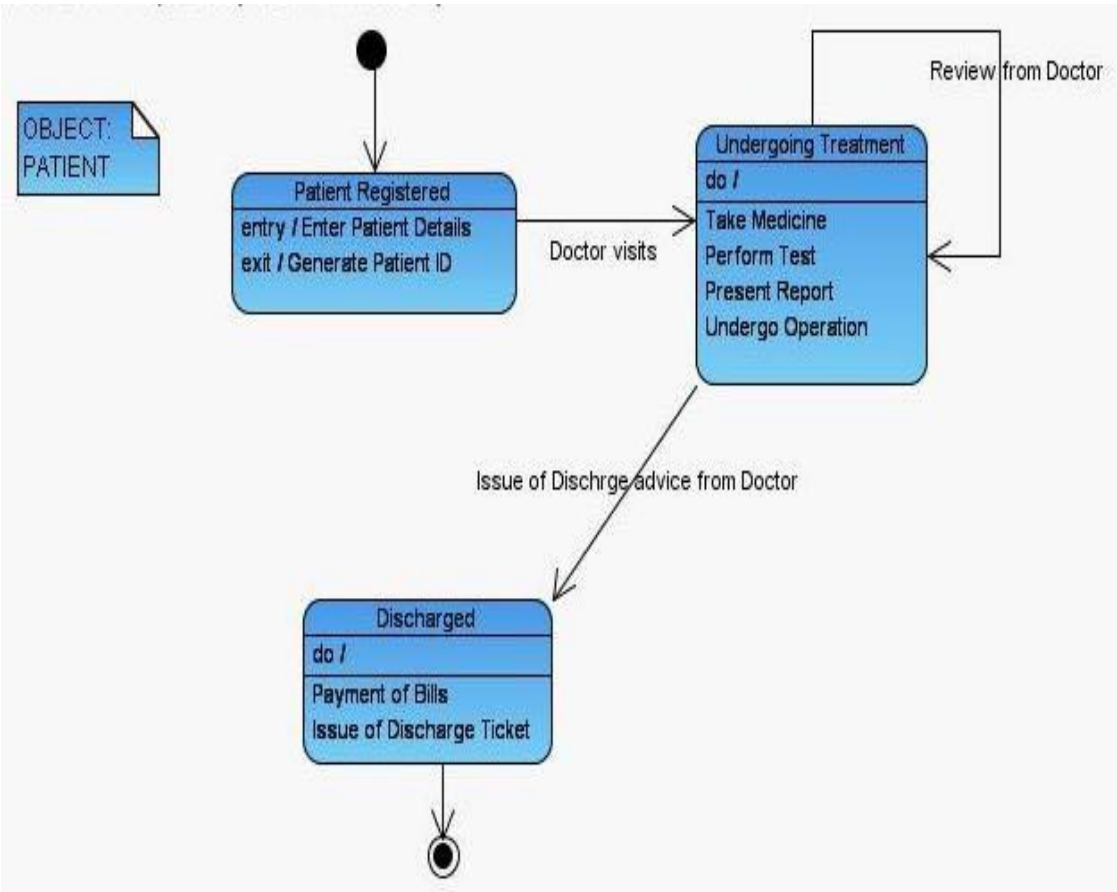
A Collaboration diagram is very similar to a Sequence diagram in the purpose it achieves; in other words, it shows the dynamic interaction of the objects in a system. A distinguishing feature of a Collaboration diagram is that it shows the objects and their association with other objects in the system apart from how they interact with each other. The association between objects is not represented in a Sequence diagram. A Collaboration diagram is easily represented by modeling objects in a system and representing the associations between the objects as links. The interaction between the objects is denoted by arrows. To identify the sequence of invocation of these objects, a number is placed next to each of these.



9.6 **STATECHART DIAGRAM:**

A state diagram is a type of diagram used in computer science and related fields to describe the behavior of systems. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. There are many forms of state diagrams, which differ slightly and have different semantics. The diagrams are used to give an abstract description of the behavior of a system. This behavior is analyzed and represented in series of events that could occur in one or more possible states. Hereby "each diagram usually represents objects of a single class and tracks the different states of its objects through the system".

State diagrams can be used to graphically represent finite state machines. This was introduced by Taylor Booth in his 1967 book "Sequential Machines and Automata Theory". Another possible representation is the State transition table.



10. SYSTEM IMPLEMENTATION

10.1 **Activity_main:**

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
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"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.dell.project3.MainActivity"
    android:background="@drawable/main">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <LinearLayout
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal"
            android:gravity="center"
            android:layout_marginTop="196dp"
            android:layout_alignParentTop="true"
            android:layout_alignParentStart="true"
            android:id="@+id/linearLayout">

            <Button
                android:id="@+id/button1"
                android:layout_width="match_parent"
```

```
            android:layout_height="wrap_content"
            android:background="#ADD8E6"
            android:text="Hospitals"
            android:onClick="hbutton"/>

    </LinearLayout>

    <Button
        android:id="@+id/button2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="13dp"
        android:background="#ff3658"
        android:text="Doctors"
        android:layout_below="@+id/linearLayout"
        android:layout_centerHorizontal="true"
        android:onClick="dbutton"/>

  </RelativeLayout>

</android.support.constraint.ConstraintLayout>
```

JAVA CLASS:

```
package com.example.dell.project3;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;

public class MainActivity extends AppCompatActivity {
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    public void hbutton(View view){
        Intent i= new Intent(MainActivity.this,hospitallist.class);
        startActivity(i);
    }
    public void dbutton(View view){
        Intent a= new Intent(MainActivity.this,doctortype.class);
        startActivity(a);
    }

}
```

10.2 **Hospitals List:**

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView 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"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.dell.project3.hospitallist"
    android:background="@drawable/listhos">
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <LinearLayout
            android:orientation="vertical"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="13dp"
            android:layout_alignParentTop="true"
```

```
android:layout_alignParentStart="true"
android:id="@+id/linearLayout">
<Button
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:text="HOSPITALS LIST"/>

<TextView
   android:id="@+id/textView"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:width="10dp"
   android:padding="10dp"
   android:text="NAME OF THE HOSPITAL :KGH "
   />

<TextView
   android:id="@+id/textView1"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:width="10dp"
   android:padding="10dp"
   android:text="ADDRESS : Near Jagadamba theatre,visakhapatnam." />

<TextView
   android:id="@+id/textView2"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:width="10dp"
   android:padding="10dp"
   android:text="Opens at:9:00AM - 10:00PM " />
<LinearLayout
   android:layout_width="match_parent"
   android:layout_height="wrap_content">

   <Button
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Back"
      android:id="@+id/scan_btn1a"
      android:onClick="bbutton"
      />
<Button
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="select"
   android:id="@+id/scan_btn1"
   android:onClick="oneb"
   />
</LinearLayout>
```

```
<TextView
   android:id="@+id/textView3"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:width="10dp"
   android:padding="10dp"
   android:text="NAME OF THE HOSPITAL : SEVEN HILLS "
   />

<TextView
   android:id="@+id/textView4"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:width="10dp"
   android:padding="10dp"
   android:text="ADDRESS : Near assilmetta,visakhapatnam. " />

<TextView
   android:id="@+id/textView5"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:width="10dp"
   android:padding="10dp"
   android:text="Opens at: 9:00AM - 8:00PM " />
<LinearLayout
   android:layout_width="match_parent"
   android:layout_height="wrap_content">
   <Button
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Back"
      android:onClick="bbutton"
      />
<Button
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="select"
   android:onClick="twob"/>
</LinearLayout>

<TextView
   android:id="@+id/textView6"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:width="10dp"
   android:padding="10dp"
   android:text="NAME OF THE HOSPITAL : APOLO HOSPITALS"
   />
```

```
<TextView
   android:id="@+id/textView7"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:width="10dp"
   android:padding="10dp"
   android:text="ADDRESS : Near assilmetta,visakhapatnam." />

<TextView
   android:id="@+id/textView8"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:width="10dp"
   android:padding="10dp"
   android:text="Opens at: 9:00AM - 10:00pm" />
<LinearLayout
   android:layout_width="match_parent"
   android:layout_height="wrap_content">
<Button
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="Back"
   android:id="@+id/scan_btn"
   android:onClick="bbutton"

   />

<Button
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="select"
   android:id="@+id/scan_btnc"
   android:onClick="threeb"
   />
</LinearLayout>

<TextView
   android:id="@+id/textView9"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:width="10dp"
   android:padding="10dp"
   android:text="NAME OF THE HOSPITAL : PRADHAMA HOSPITALS "
   />

<TextView
   android:id="@+id/textView10"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:width="10dp"
```

```
    android:padding="10dp"
    android:text="ADDRESS : Near MVP colony,visakhapatnam." />

<TextView
    android:id="@+id/textView11"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:width="10dp"
    android:padding="10dp"
    android:text="Opens at: 10:00AM - 10:00PM" />
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Back"
    android:onClick="bbutton"
    />
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="select"
    android:onClick="fourb"/>
</LinearLayout>
<TextView
    android:id="@+id/textView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:width="10dp"
    android:padding="10dp"
    android:text="NAME OF THE HOSPITAL :KGH "
    />
<TextView
    android:id="@+id/textView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:width="10dp"
    android:padding="10dp"
    android:text="NAME OF THE HOSPITAL :KGH "
    />

<TextView
    android:id="@+id/textView1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:width="10dp"
    android:padding="10dp"
    android:text="ADDRESS : Near Jagadamba theatre,visakhapatnam." />
```

```
<TextView
   android:id="@+id/textView2"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:width="10dp"
   android:padding="10dp"
   android:text="Opens at:9:00AM - 10:00PM " />

<LinearLayout
   android:layout_width="match_parent"
   android:layout_height="wrap_content">

   <Button
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Back"
      android:id="@+id/scan_btn1a"
      android:onClick="bbutton"
      />
<Button
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="select"
   android:id="@+id/scan_btn1"
   android:onClick="oneb"
   />


<TextView
   android:id="@+id/textView1"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:width="10dp"
   android:padding="10dp"
   android:text="ADDRESS : Near Jagadamba theatre,visakhapatnam." />

<TextView
   android:id="@+id/textView2"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:width="10dp"
   android:padding="10dp"
   android:text="Opens at:9:00AM - 10:00PM " />
<LinearLayout
   android:layout_width="match_parent"
   android:layout_height="wrap_content">

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

```
            android:text="Back"
            android:id="@+id/scan_btn1a"
            android:onClick="bbutton"
            />
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="select"
            android:id="@+id/scan_btn1"
            android:onClick="oneb"
            />


    </LinearLayout>
  </RelativeLayout>

</ScrollView>
```

JAVA CLASS:

```
package com.example.dell.project3;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;

public class hospitallist extends AppCompatActivity {

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_hospitallist);
  }
  public void oneb(View view){
    Intent a= new Intent(hospitallist.this,login.class);
    startActivity(a);
  }
  public void twob(View view){
    Intent a= new Intent(hospitallist.this,login.class);
    startActivity(a);
  }
  public void threeb(View view){
    Intent a= new Intent(hospitallist.this,login.class);
    startActivity(a);
  }
  public void fourb(View view){
    Intent a= new Intent(hospitallist.this,login.class);
    startActivity(a);
  }
```

```
    public void bbutton(View view){
        Intent a= new Intent(hospitallist.this,MainActivity.class);
        startActivity(a);
    }
}
```

10.3 **Doctors List:**

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
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"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.dell.project3.doctortype"
    android:background="@drawable/type">
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">


        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            >
            <Button
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:textAlignment="center"
                android:text="CARDIOLOGIST"
                android:onClick="d1page"/>

            <Button
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:textAlignment="center"
                android:text="NEUROLOGIST"
                android:onClick="d2page"/>

            <Button
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:textAlignment="center"
                android:text="PAEDIATRICIAN"
                android:onClick="d3page"/>
            <Button
                android:layout_width="match_parent"
```

```
            android:layout_height="wrap_content"
            android:textAlignment="center"
            android:text="GENERAL SURGERY"
            android:onClick="d4page"/>
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Back"
            android:onClick="homepage1"/>


    </LinearLayout>
  </RelativeLayout>

</android.support.constraint.ConstraintLayout>
```

JAVA CLASS:

```
package com.example.dell.project3;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;

public class doctortype extends AppCompatActivity {

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_doctortype);
  }
  public void d1page(View view){
    Intent i= new Intent(doctortype.this,doctype1.class);
    startActivity(i);
  }
  public void d2page(View view){
    Intent i= new Intent(doctortype.this,doctype2.class);
    startActivity(i);
  }
  public void d3page(View view){
    Intent i= new Intent(doctortype.this,doctype3.class);
    startActivity(i);
  }
  public void d4page(View view){
    Intent i= new Intent(doctortype.this,doctype4.class);
    startActivity(i);
  }
  public void homepage1(View view){
    Intent i= new Intent(doctortype.this,MainActivity.class);
    startActivity(i);
```

```
    }
}
```

10.4 **Login page:**

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
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"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.dell.project3.login"
    android:background="@drawable/login"
    >
    <LinearLayout
        android:orientation="vertical"
        android:layout_centerVertical="true"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content">
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textAppearance="@style/Base.ThemeOverlay.AppCompat.Dark"
                android:textColor="#FFFFFF"
                android:text="Email:"/>

        <EditText
            android:layout_margin="15dp"
            android:inputType="textEmailAddress"
            android:hint="Enter your email"
            android:id="@+id/editTextEmail"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textColor="#FFFFFF"/>
        </LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Password:"
        android:textColor="#FFFFFF"/>
```

```
        <EditText
            android:layout_margin="15dp"
            android:inputType="textPassword"
            android:hint="Enter password"
            android:id="@+id/editTextPassword"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textColor="#FFFFFF"/>
</LinearLayout>
        <Button
            android:layout_margin="15dp"
            android:id="@+id/buttonRegister"
            android:text="Register User"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:onClick="apage"/>

        <TextView
            android:textAlignment="center"
            android:text="Already Registered? Sign in here"
            android:id="@+id/textViewSignin"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textColor="#FFFFFF"/>
    </LinearLayout>


</android.support.constraint.ConstraintLayout>
```

JAVA CLASS:

```
package com.example.dell.project3;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;

public class login extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
    }
    public void apage(View view)
    {
        Intent i= new Intent(login.this,application.class);
        startActivity(i);
    }
}
```

10.5 **Application Page:**

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
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"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context="com.example.dell.project3.application"
   android:background="@drawable/images">
   <RelativeLayout
      android:layout_width="match_parent"
      android:layout_height="match_parent">
      <LinearLayout
         android:layout_width="match_parent"
         android:layout_height="match_parent"
         android:orientation="vertical">
         <TextView
            android:id="@+id/textView10"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Name:"
            android:textColor="#000000"
            />

         <EditText
            android:id="@+id/editTextName"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:ems="10"
            android:inputType="textPersonName"
            android:text=""
            tools:layout_editor_absoluteX="94dp"
            tools:layout_editor_absoluteY="4dp"
            android:textColor="#000000"
            />

         <TextView
            android:id="@+id/textView11"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Age:"
            android:textColor="#000000"

            />

         <EditText
```

```
    android:id="@+id/editTextAge"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textPersonName"
    android:text=""
    tools:layout_editor_absoluteX="100dp"
    tools:layout_editor_absoluteY="65dp"
    android:textColor="#000000"
    />

<TextView
    android:id="@+id/textView12"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="parent/gaurdian"
    tools:layout_editor_absoluteX="16dp"
    tools:layout_editor_absoluteY="154dp"
    android:textColor="#000000"
    />

<EditText
    android:id="@+id/editTextParent"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textPersonName"
    android:text=""
    tools:layout_editor_absoluteX="100dp"
    tools:layout_editor_absoluteY="140dp"
    android:textColor="#000000"
    />
<TextView
    android:id="@+id/textView13"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Affected with:"
    android:textColor="#000000"

    />

<EditText
    android:id="@+id/editTextAffect"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textPersonName"
    android:text=""
    tools:layout_editor_absoluteX="94dp"
    tools:layout_editor_absoluteY="4dp"
```

```
    android:textColor="#000000"
    />
<TextView
    android:id="@+id/textView14"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="phone"
    tools:layout_editor_absoluteX="16dp"
    tools:layout_editor_absoluteY="230dp"
    android:textColor="#000000"
    />

<EditText
    android:id="@+id/editTextPhone"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="phone"
    tools:layout_editor_absoluteX="100dp"
    tools:layout_editor_absoluteY="232dp"
    android:textColor="#000000"
    />
<TextView
    android:id="@+id/textView15"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Contact Details:"
    android:textColor="#000000"

    />

<EditText
    android:id="@+id/editTextContact"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textPersonName"
    android:text=""
    tools:layout_editor_absoluteX="94dp"
    tools:layout_editor_absoluteY="4dp"
    android:textColor="#000000"
    />
<Button
    android:id="@+id/ButtonSave"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Save"
    android:textColor="#000000"

    />
```

```
        <Button
            android:id="@+id/button4"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Next"
            tools:layout_editor_absoluteX="100dp"
            tools:layout_editor_absoluteY="336dp"
            android:onClick="epage"
            android:textColor="#000000"
            />
      </LinearLayout>
   </RelativeLayout>

</android.support.constraint.ConstraintLayout>
```

JAVA CLASS:

```
package com.example.dell.project3;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;

public class application extends AppCompatActivity {
   database mydb;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_application);
      mydb = new database(this);
   }
   public void epage(View view)
   {
      Intent i= new Intent(application.this,estimation.class);
      startActivity(i);
   }
}
```

10.6 **Doctor Type:**

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView 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"
   android:layout_width="match_parent"
```

```
android:layout_height="match_parent"
tools:context="com.example.dell.project3.doctype1"
android:background="@drawable/listdoc">

<RelativeLayout
   android:layout_width="match_parent"
   android:layout_height="match_parent">

   <LinearLayout
      android:orientation="vertical"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_marginTop="13dp"
      android:layout_alignParentTop="true"
      android:layout_alignParentStart="true"
      android:id="@+id/linearLayout">

      <Button
         android:id="@+id/button2"
         android:layout_width="match_parent"
         android:layout_height="wrap_content"
         android:layout_centerHorizontal="true"
         android:layout_marginTop="13dp"
         android:background="#365bff"
         android:onClick="homepage"
         android:text="CARDIOLOGIST" />



      <TextView
         android:id="@+id/textView"
         android:layout_width="match_parent"
         android:layout_height="wrap_content"
         android:width="10dp"
         android:padding="10dp"
         android:text="NAME :K.SRINIVASA RAO"
         android:textColor="#000000"
         />

      <TextView
         android:id="@+id/textView2"
         android:layout_width="match_parent"
         android:layout_height="wrap_content"
         android:width="10dp"
         android:padding="10dp"
         android:text="Qualification :MD.DM "
         android:textColor="#000000"
         />

      <TextView
```

```
    android:id="@+id/textView1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:width="10dp"
    android:padding="10dp"
    android:text="Specialization : CARDIOLOGY"
    android:textColor="#000000"
    />


<TextView
    android:id="@+id/textView3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:width="10dp"
    android:padding="10dp"
    android:text="Contact Number : 9876543210"
    android:textColor="#000000"

    />
<TextView
    android:id="@+id/textView4"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:width="10dp"
    android:padding="10dp"
    android:text="Mail: srinivas@gmail.com"
    android:textColor="#000000"

    />
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="click"
    android:onClick="dapage"
    android:textColor="#000000"
    />



<TextView
    android:id="@+id/textView5"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:width="10dp"
    android:padding="10dp"
    android:text="NAME : M. RAMA RAO"
    android:textColor="#000000"

    />
```

```
<TextView
   android:id="@+id/textView6"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:width="10dp"
   android:padding="10dp"
   android:text="Qualification : MS.MCH"
   android:textColor="#000000"

   />
<TextView
   android:id="@+id/textView7"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:width="10dp"
   android:padding="10dp"
   android:text="Specialization: CARDIOLOGY"
   android:textColor="#000000"

   />
<TextView
   android:id="@+id/textView8"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:width="10dp"
   android:padding="10dp"
   android:text="Contact Number : 6785432789"
   android:textColor="#000000"

   />
<TextView
   android:id="@+id/textView9"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:width="10dp"
   android:padding="10dp"
   android:text="Mail Id : rama22@gmail.com"
   android:textColor="#000000"

   />
<Button
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="click"
   android:onClick="dapage"
   android:textColor="#000000"
   />
<TextView
   android:id="@+id/textView"
   android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
    android:width="10dp"
    android:padding="10dp"
    android:text="NAME :K.SRINIVASA RAO"
    android:textColor="#000000"
    />

<TextView
    android:id="@+id/textView2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:width="10dp"
    android:padding="10dp"
    android:text="Qualification :MD.DM "
    android:textColor="#000000"
    />

<TextView
    android:id="@+id/textView1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:width="10dp"
    android:padding="10dp"
    android:text="Specialization : CARDIOLOGY"
    android:textColor="#000000"
    />


<TextView
    android:id="@+id/textView3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:width="10dp"
    android:padding="10dp"
    android:text="Contact Number : 9876543210"
    android:textColor="#000000"

    />
<TextView
    android:id="@+id/textView4"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:width="10dp"
    android:padding="10dp"
    android:text="Mail: srinivas@gmail.com"
    android:textColor="#000000"

    />
<Button
    android:layout_width="wrap_content"
```

```
            android:layout_height="wrap_content"
            android:text="click"
            android:onClick="dapage"
            android:textColor="#000000"
            />


    </LinearLayout>
  </RelativeLayout>
</ScrollView>
```

JAVA CLASS:

```
package com.example.dell.project3;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;

public class doctype1 extends AppCompatActivity {

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_doctype1);
  }
  public void dapage(View view)
  {
    Intent i= new Intent(doctype1.this,doctoravailable.class);
    startActivity(i);
  }
  public void homepage(View view)
  {
    Intent i= new Intent(doctype1.this,doctortype.class);
    startActivity(i);
  }
}
```

```
package com.example.dell.project3;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;

public class doctype2 extends AppCompatActivity {

  @Override
  protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_doctype2);
    }

    public void dapage(View view)
    {
        Intent i= new Intent(doctype2.this,doctoravailable.class);
        startActivity(i);
    }
    public void homepage(View view)
    {
        Intent i= new Intent(doctype2.this,doctortype.class);
        startActivity(i);
    }
}


package com.example.dell.project3;


import android.content.Intent;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;


public class doctype3 extends AppCompatActivity {


    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_doctype3);

    }

    public void dapage(View view)

    {

        Intent i= new Intent(doctype3.this,doctoravailable.class);

        startActivity(i);

    }

    public void homepage(View view)
```

```
    {
        Intent i= new Intent(doctype3.this,doctortype.class);
        startActivity(i);
    }
}

package com.example.dell.project3;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;

public class doctype4 extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_doctype4);
    }
    public void dapage(View view)
    {
        Intent i= new Intent(doctype4.this,doctoravailable.class);
        startActivity(i);
    }
    public void homepage(View view)
    {
        Intent i= new Intent(doctype4.this,doctortype.class);
        startActivity(i);
    }
```

}

10.7 **Doctor Availability:**

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
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"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context="com.example.dell.project3.doctoravailable"
   android:background="@drawable/docavail">
   <ScrollView
      android:layout_width="match_parent"
      android:layout_height="match_parent">
      <RelativeLayout
         android:layout_width="match_parent"
         android:layout_height="match_parent">
         <LinearLayout
            android:orientation="vertical"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="13dp"
            android:layout_alignParentTop="true"
            android:layout_alignParentStart="true"
            android:id="@+id/linearLayout">
            <TextView
               android:id="@+id/textView13"
               android:layout_width="match_parent"
               android:layout_height="match_parent"
               android:width="10dp"
               android:padding="10dp"
               android:text="NAME :K. RAMA RAO "
               android:textColor="#000000"
               />

            <TextView
               android:id="@+id/textView22"
               android:layout_width="match_parent"
               android:layout_height="match_parent"
               android:width="10dp"
               android:padding="10dp"
               android:text="Qualification : MD.DM"
               android:textColor="#000000"
               />
```

```
<TextView
   android:id="@+id/textView15"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:width="10dp"
   android:padding="10dp"
   android:text="Specialization :CARDIOLOGIST "
   android:textColor="#000000"
   />


<TextView
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:width="10dp"
   android:padding="10dp"
   android:text="No of patients: 20"
   android:textColor="#000000"
   />
<LinearLayout
   android:layout_width="match_parent"
   android:layout_height="wrap_content">
<Button
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="Back"
   android:onClick="dtpage"
   android:textColor="#000000"
   />

<Button
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="click"
   android:onClick="lpage"
   android:textColor="#000000"
   />
</LinearLayout>


<RatingBar
   android:id="@+id/ratingBar1"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   tools:layout_editor_absoluteX="0dp"
   tools:layout_editor_absoluteY="250dp"
   android:rating="4.3"
   />
<TextView
```

```
    android:id="@+id/textView"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:width="10dp"
    android:padding="10dp"
    android:text="NAME : J.HYMA"
    android:textColor="#000000"

    />

<TextView
    android:id="@+id/textView2"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:width="10dp"
    android:padding="10dp"
    android:text="Qualification : MS.MD"
    android:textColor="#000000"
    />

<TextView
    android:id="@+id/textView1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:width="10dp"
    android:padding="10dp"
    android:text="Specialization :PAEDIATRICIAN "
    android:textColor="#000000"
    />


<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:width="10dp"
    android:padding="10dp"
    android:text="No of patients:50"
    android:textColor="#000000"
    />
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Back"
        android:onClick="dtpage"
        android:textColor="#000000"
        />
<Button
```

```
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="click"
            android:onClick="lpage"
            android:textColor="#000000"
            />
        </LinearLayout>

        <RatingBar
            android:id="@+id/ratingBar"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            tools:layout_editor_absoluteX="0dp"
            tools:layout_editor_absoluteY="250dp"
            android:rating="3.5"
            />
<TextView
            android:id="@+id/textView13"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:width="10dp"
            android:padding="10dp"
            android:text="NAME :K. RAMA RAO "
            android:textColor="#000000"
            />

        <TextView
            android:id="@+id/textView22"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:width="10dp"
            android:padding="10dp"
            android:text="Qualification : MD.DM"
            android:textColor="#000000"
            />

        <TextView
            android:id="@+id/textView15"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:width="10dp"
            android:padding="10dp"
            android:text="Specialization :CARDIOLOGIST "
            android:textColor="#000000"
            />


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

```
   android:width="10dp"
   android:padding="10dp"
   android:text="No of patients: 20"
   android:textColor="#000000"
   />
<LinearLayout
   android:layout_width="match_parent"
   android:layout_height="wrap_content">
<Button
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="Back"
   android:onClick="dtpage"
   android:textColor="#000000"
   />

<Button
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="click"
   android:onClick="lpage"
   android:textColor="#000000"
   />
</LinearLayout>


<RatingBar
   android:id="@+id/ratingBar1"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   tools:layout_editor_absoluteX="0dp"
   tools:layout_editor_absoluteY="250dp"
   android:rating="4.3"
   />

<TextView
   android:id="@+id/textView11"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:width="10dp"
   android:padding="10dp"
   android:text="NAME :M. VENKAT RAO "
   android:textColor="#000000"

   />

<TextView
   android:id="@+id/textView20"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
```

```
        android:width="10dp"
        android:padding="10dp"
        android:text="Qualification : MS.PM"
        android:textColor="#000000"
        />

    <TextView
        android:id="@+id/textView10"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:width="10dp"
        android:padding="10dp"
        android:text="Specialization : NEUROLOGIST" />


    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:width="10dp"
        android:padding="10dp"
        android:text="No of patients:25"
        android:textColor="#000000"
        />
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Back"
            android:onClick="dtpage"
            android:textColor="#000000"
            />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="click"
        android:onClick="lpage"
        android:textColor="#000000"
        />
    </LinearLayout>
    <RatingBar
        android:id="@+id/ratingBar4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        tools:layout_editor_absoluteX="0dp"
        tools:layout_editor_absoluteY="250dp"
        android:rating="3"
        />
</LinearLayout>
```

```
    </RelativeLayout>
  </ScrollView>

</android.support.constraint.ConstraintLayout>
```

JAVA CLASS:

```
package com.example.dell.project3;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;

public class doctoravailable extends AppCompatActivity {

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_doctoravailable);
  }
  public void lpage(View view)
  {
    Intent i= new Intent(doctoravailable.this,login.class);
    startActivity(i);
  }
  public void dtpage(View view)
  {
    Intent i= new Intent(doctoravailable.this,doctortype.class);
    startActivity(i);
  }
}
```

10.8 **Estimation Page:**

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
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"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  tools:context="com.example.dell.project3.estimation">
<RelativeLayout
  android:layout_width="match_parent"
  android:layout_height="wrap_content">
  <LinearLayout
    android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content">
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Estimation time is"/>
        <Button
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Allow Access"
            android:onClick="healthpage"/>

    </LinearLayout>

</RelativeLayout>
</android.support.constraint.ConstraintLayout>
```

10.9 **Health Facts:**

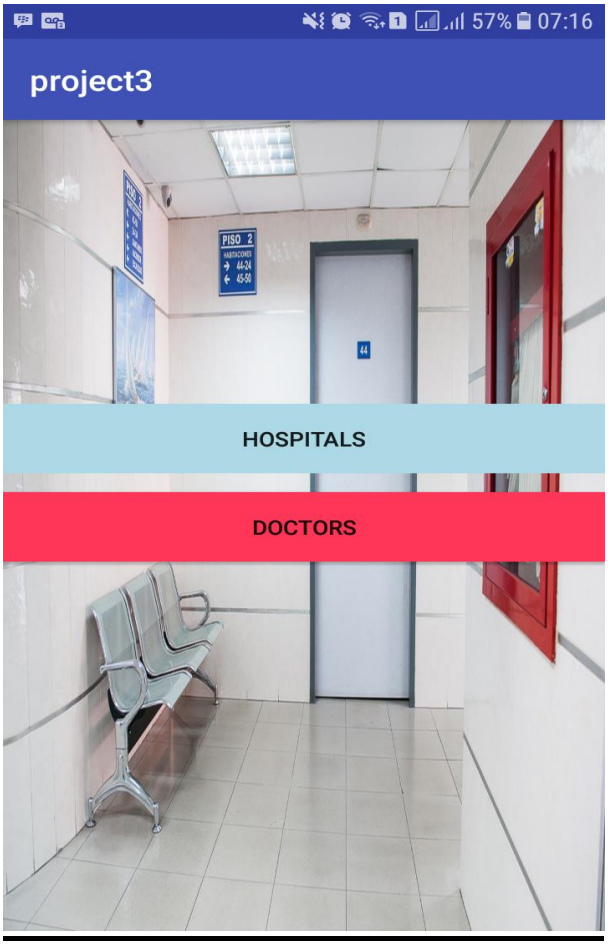
```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.dell.project3.health">

</android.support.constraint.ConstraintLayout>
```

JAVA CLASS:

```
package com.example.dell.project3;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class health extends AppCompatActivity {

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

11. **SNAPSHOTS**

11.1 Main Page:



The first page consists of two buttons-

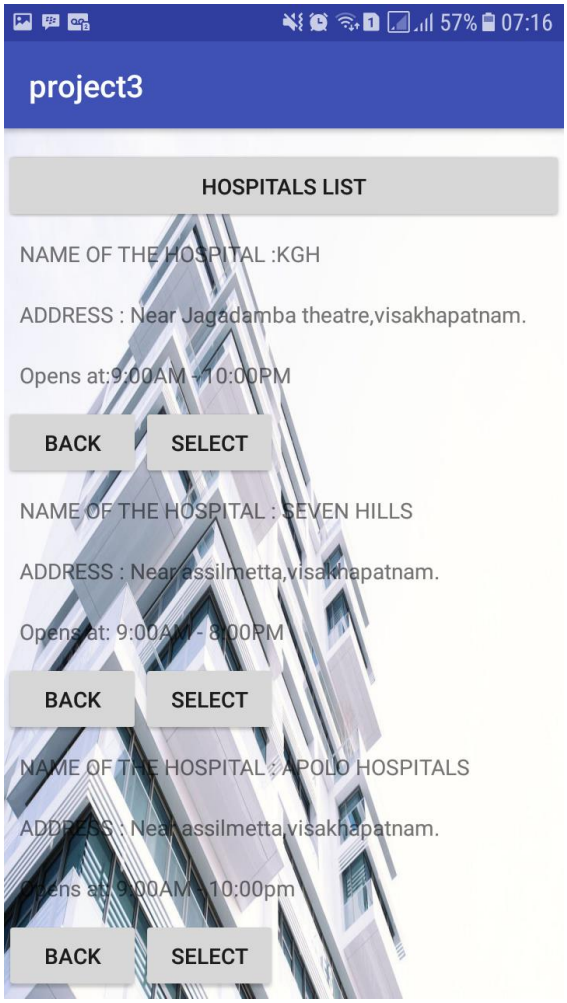
1. Hospitals

2. Doctors

If the user clicks on the Hospitals Button, a list of hospitals in the particular city.

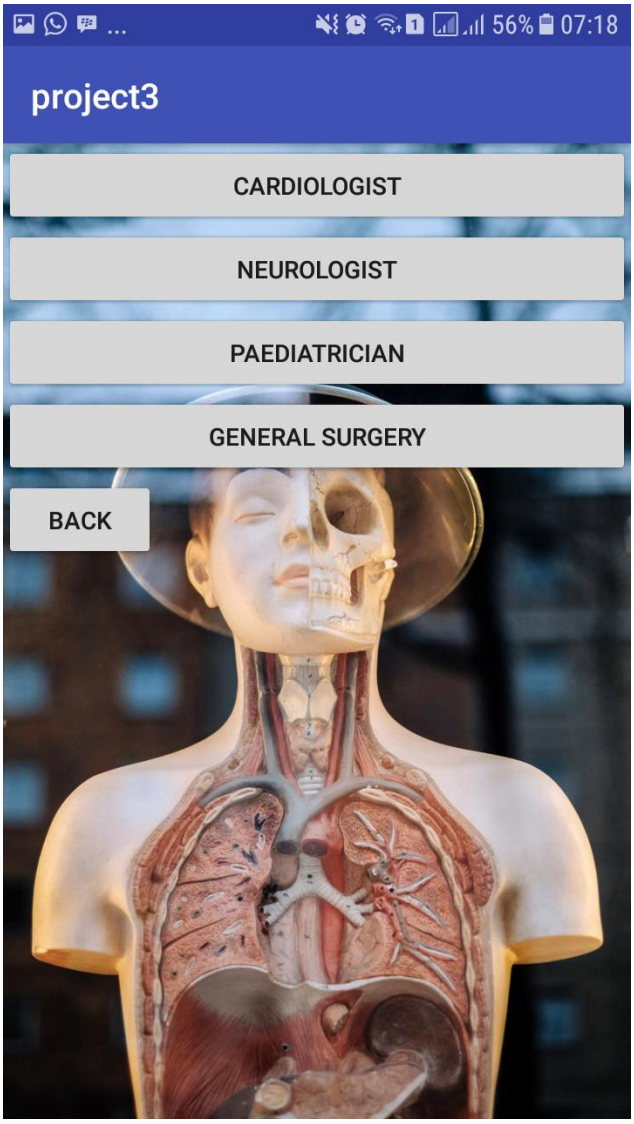
If the user clicks on the Doctors Button, all the types of Doctors are shown.

11.2 Hospitals List:



The second page consists of the list of hospitals and the information of every single hospitals. The information includes Name of the Hospital, address and timings.

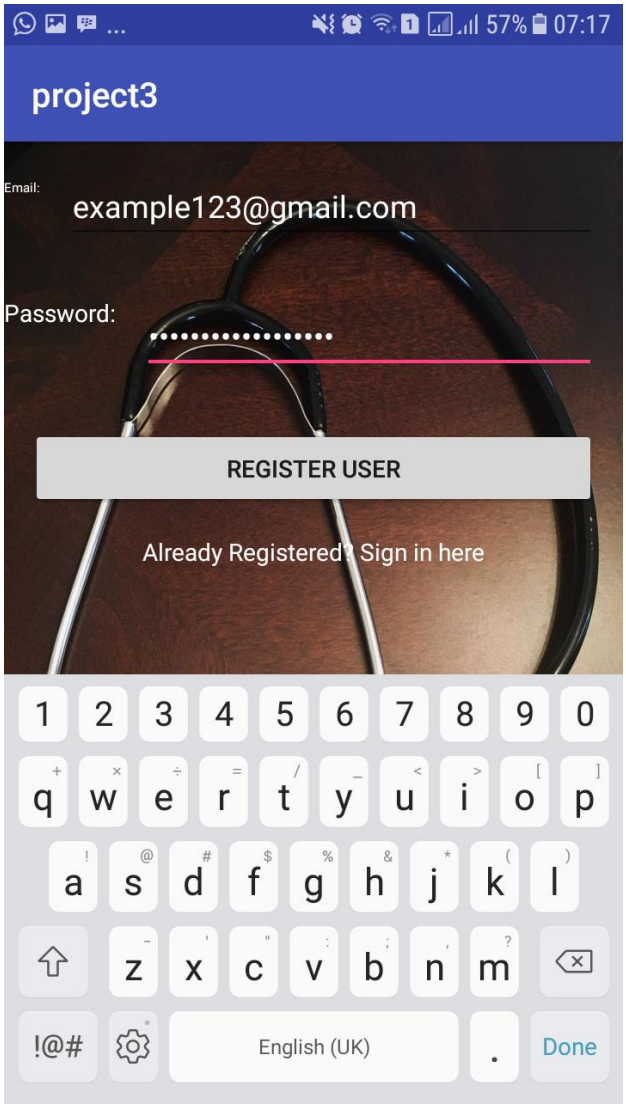
11.3 Types of Doctors:



This page is directed to user when the user clicks on Doctors button in the first page.

This page shows a list of types of Doctors (Specialization).

11.4 Login Page:

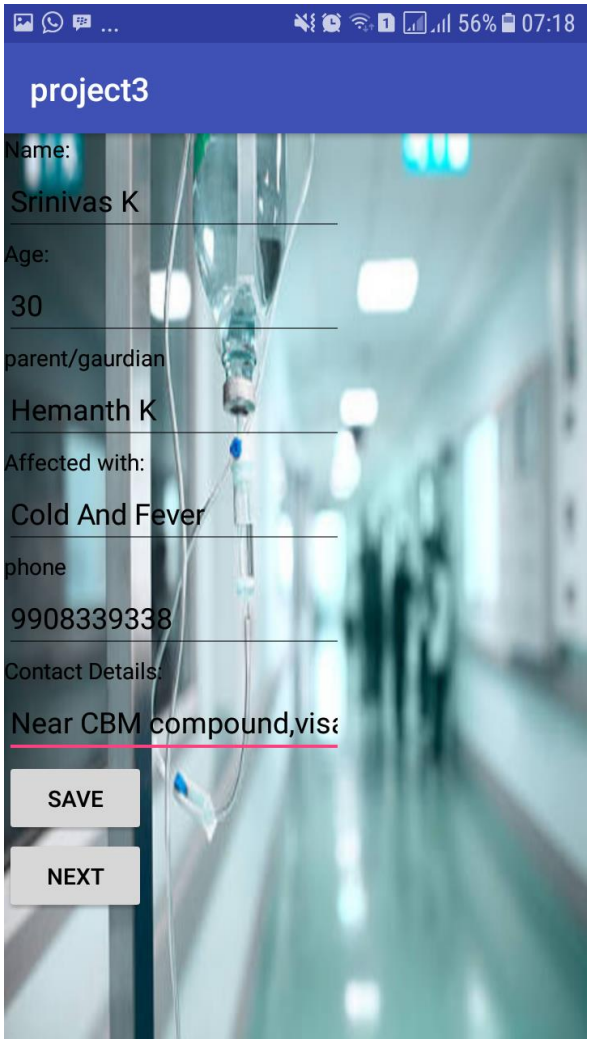


This is a registration page.

If the user is new to the application, then the user should register, providing the details of email ID and a new password.

If the user is already registered before, then he can skip this and click on the link- Sign in here.

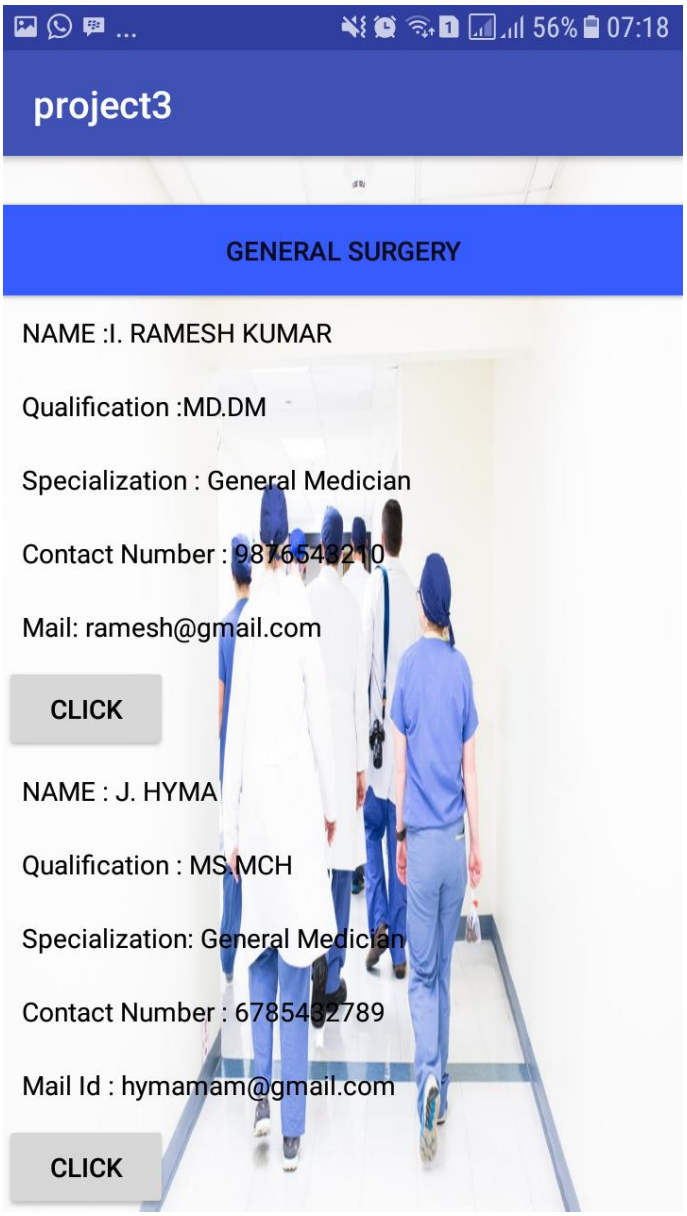
11.5 Application Form:



This is the application form page.

The user should provide the details regarding their Name, Age, Trusted contact, illness or injury, phone number and address.

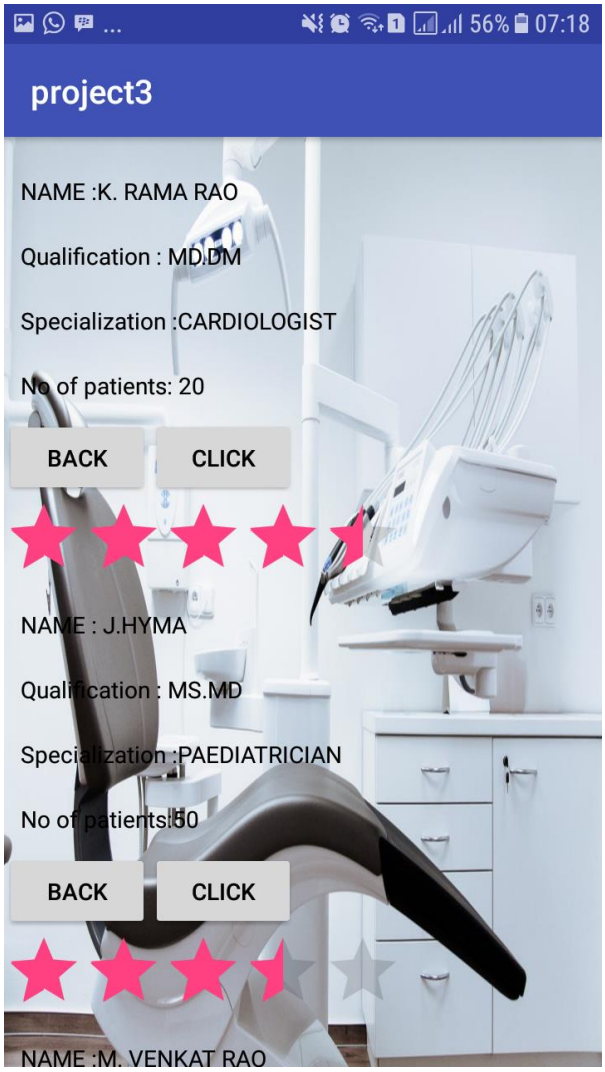
11.6 Doctors List :



In the Doctors page, where the list of types of Doctors is displayed, each type or specialization button redirects the user to this page, where the list of doctors under the selected specialization is displayed.

This page is an example of the user selecting General Surgery in Doctors page

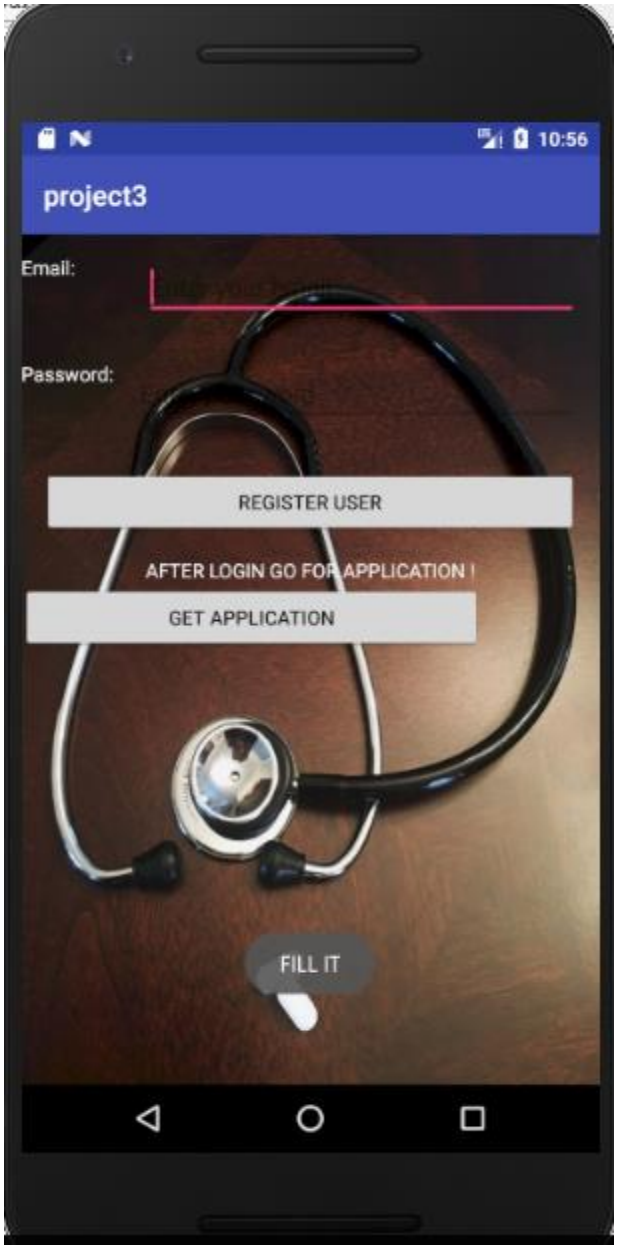
11.7 Availability of Doctors:



This page consists of availability of the Doctors, that is, the number of patients the Doctor is having, timings of them, and the user rating.

12. INPUT DATA AND VALIDATION OF PROJECT

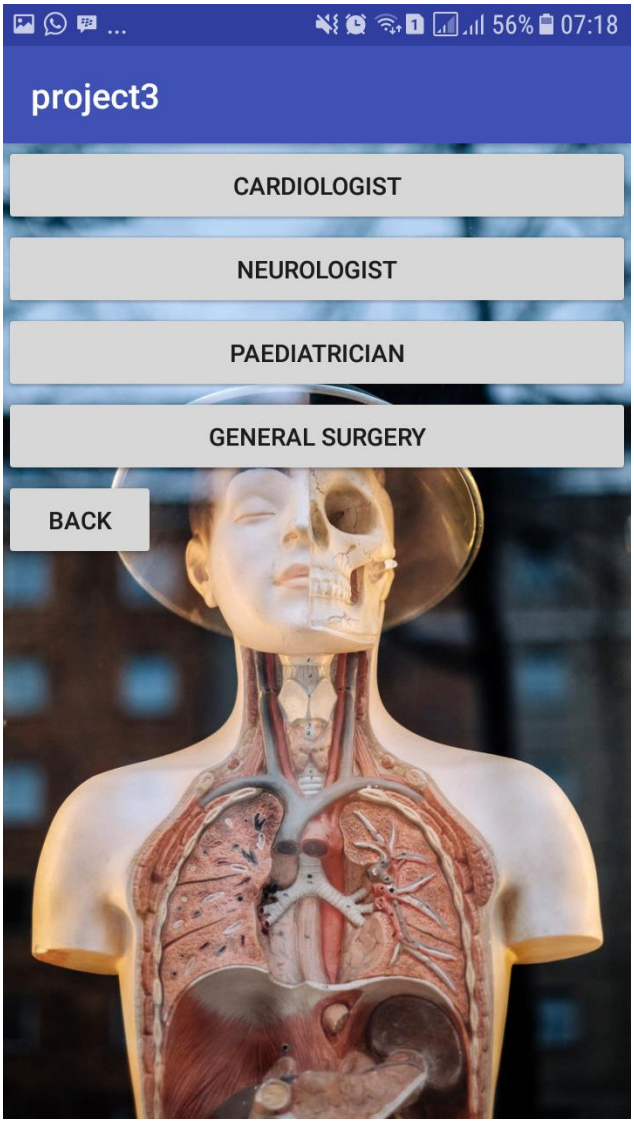
- All Fields must be filled in the registration form. If any field is left blank, then a toast message pops up saying ‘Fill It’.

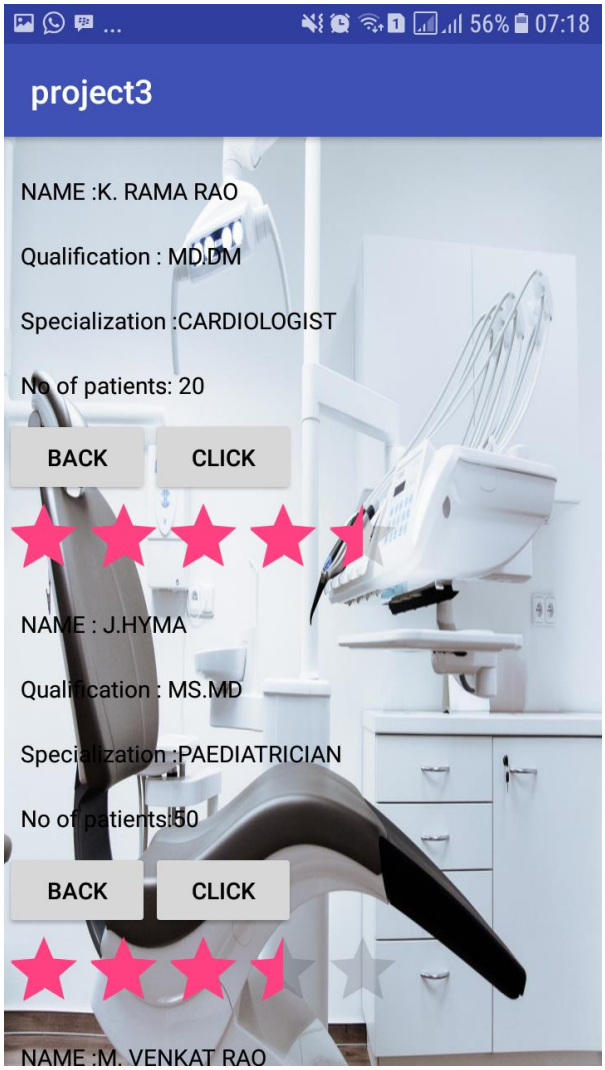


- Any field of Appointment form cannot accept blank values, If any field is left blank, then a toast message pops up saying ‘Fill It’.

- Avoiding errors in data.
- Actual testing done manually.
- Modifications done for the errors found during testing.

- Prepared the list of appointments and number of doctors.





- Validations for User input.
- Checking the coding standards to be maintained during coding.

13. CONCLUSION

The proposed online appointment system has been implemented in android studio for application development and website is developed using HTML and PHP. The tasks involved in this work are divided into modules. The data is approached and shared by using API'S between the website and the android application. The proposed system is efficient and has friendly user interface. That would help the Patient to register on the application and perform all the tasks on the app. The admin would be able to use the app for managing the details of the patients and the doctors instead of using the manual system. No payment will be charged to the users/patients while making an appointment to avoid the unethical users. As many users only register themselves just for fun and has no concern by making an appointment.

Since patients' preference and the daily number of patient requests are generated randomly increasing, it may be less realistic than the situations in the real world. Patients could also track and monitor their own appointment record with this system. However, the display of bio-data such as X-rays and laboratory results are not included in the system due to technical constrain. The system will not be able to diagnose or prescribe drug for usage. The system is mainly designed to facilitate appointment booking between the patient and the health personnel. In compensation, additional modules such as Announcement, Medical case record and block/unblock schedule will further enhance the usability and functionality of the system and allow a flexible management of patients appointment Thus, real data from hospitals should be collected and used to conduct a future analysis based on this thesis. Finally, it is assumed in this thesis that patients' preference of time slots and doctors are equal. Further analysis of these factors will be helpful for the online appointment system of hospitals

14. REFERENCES

1. Adesina, D. (2009). The National Health Insurance Scheme. Retrieved October 3, from http://www.nigeriandoctor.Com/news.php?extend.85D.

2. Arodiogbu I.L. (2005). Introducing Social Health Insurance to solve problems of poor health Sector in Nigeria. MA Thesis. University of Leeds.

3. Alex Chaffee (2000-08-17). "What is a web application (or "webapp")?". Retrieved 2008-07-27.

4. Bailey, N. (1952). A study of queues and appointment systems in hospital outpatient departments, with special reference to waiting times. Journal of the Royal Statistical Society, A14, 185-199.

5. Bailey, N. (1954). Queuing for medical care. Applied Statistics 3, 137-145.

6. Bailey NTJ. A study of queues and appointment systems in hospital out-patient departments, with special reference to waiting times. J Royal Stat Soc 1952;14:185–99.

7. Cayirli, T, E. Veral, and H. Rosen. (2004). Assessment of patient classification in appointment systems. 1st Conference of the POMS College of Service Operations, New York, NY, USA.

8. Cayirli, T, E. Veral, and H. Rosen. (2006). Designing appointment scheduling systems for ambulatory care services. Health Care Management Science 9, 47–58.

9. Denton, B., and D. Gupta. 2003. Sequential bounding approach for optimal appointment scheduling, IIE Transactions, 35, 1003—1016.

10. Fernandes C, Daya M, Barry S, Palmer N (1994). Emergency department Patients who leave without seeing a Physician: The Toronto Hospital experience. Ann. Emer. Med., 24: 1092-1096.

11. James Duncan Davidson, Danny Coward (1999-12-17). Java Servlet Specification ("Specification") Version: 2.2 Final Release. Sun Microsystems. pp. 43–46. Retrieved 2008-07-27.

12. Johnson, J. A. & Stoskopt, C. (2009). Comparative Health Systems: Global Perspectives.

13. Kannegiesser, L. (2009). Nation Health Insurance Scheme to Boost Generics Market in Nigeria.

14. Klassen, K. J., T. R. Rohleder. 2004. Outpatient appointment scheduling with urgent clients in a dynamic, multi-period environment. Internat. J. Service Indust. Management 15(2) 167–186.