HOSPITAL MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted in partial fulfillment of the
Requirements for the award of the Degree of

BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)

BY

Mr. Nagarajan Nambirajan

Seat Number: 4027960

Under the esteemed guidance of

Prof. Priya Maurya
&
Prof. Krunali Petkar

DEPARTMENT OF INFORMATION TECHNOLOGY

SMT. PARMESHWARIDEVI DURGADUTT TIBREWALA LIONS JUHU COLLEGE OF ARTS, COMMERE AND SCIENCE

Affiliated to University of Mumbai

J.B. NAGAR, ANDHERI (E), MUMBAI-400059

Academic Year 2019-2020

SMT.Parmeshwaridevi Durgadutt Tibrewala Lions juhu College of Arts, Commerce &. Science J.B.Nagar, Andheri (E) Mumbai-59

CERTIFICATE OF APPROVAL

This is to certify that the project entitled, **"Hospital Management System"**, is bonafied work of **Mr. Nagarajan Nambirajan,** bearing **Seat No 4027960** submitted in partial fulfilment of the requirements for the award of degree of BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY from the University of Mumbai , for the Academic Year 2019-2020. It is further certified that he has completed all required phases of the project.

_____ _____

Internal Examiner External Examiner

_____ _____

Project Guide Co-ordinator

_____

College Stamp

PROFORMA FOR THE APPROVAL PROJECT PROPOSAL

PNR No.: 2017016402032881 **Roll no.: 321**

1. **Name of the Student**

   Nagarajan Nambirajan

2. **Title of the Project**

   Hospital Management System

3. **Name of the Guide**

   Prof Priya Maurya & Prof krunali Petkar

4. **Teaching experience of the Guide 5 years**

5. **Is this your first submission?** **Yes** ☑ **No** ☐

Signature of the Student **Signature of the Guide**

Date: …….. **Date: ……..**

Signature of the Coordinator

Date: ……

ACKNOWLEDGEMENT

Before we get into thick of things we would like to add a few heartfelt words for the people who were part of this project in numerous ways, people who gave unending support right from the stage the project idea was conceived. A project report is such a comprehensive coverage, it would not have been materialized without they help of many.

The four things that go on to make a successful endeavour are dedication, hard word patience and correct guidance. Able and timely guidance not only helps in making an effort fruitful but also transforms the whole process of learning and implementing into an enjoyable experience.

In particular, I would like to thanks our principal **"Dr.(Mrs.)Trishla Mehta"** for her blessing and for being a constant source of inspiration to us and also grateful thank to our co-ordinator **"Prof.Mukesh Sharma"**. With immense gratitude, I would to give a very special honour and respect to our teacher, **"Prof.Krunali Petkar & Prof Priya Maurya"**who kept taking interest in checking the minute details of project work and guidance us throughout the same.

A sincere quote of thanks to the non-caching for providing us book with all the information we needed for this project, without which the successful completion of this project would not have been possible. I appreciate the outstanding cooperation by the non-teaching staff, especially for the long lab timing that could receive.

Last but not least I wish to avail myself of this opportunity. Express a sense of gratitude and love to my friend and my parents for their manual support, strength and help for everything.

DECLARATION

I hereby declare that the project entitled, "**HOSPITAL MANAGEMEN SYSTEM**" done at **SMT.PARMESHWARIDEVI DURGADUTT TIBREWALA LIONS JUHU COLLEGE OF ARTS, COMMERECE & SCIENCE**. of Information Technology, for the Academic year 2019- 2020, has not been in any case duplicated to submit to any other universities for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of **BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum.

NAGARAJAN NAMBIRAJAN

ABSTRACT

The purpose of the project entitled as “HOSPITAL MANAGEMENT SYSTEM” is to computerize the Front Office Management of Hospital to develop software which is user friendly simple, fast, and cost – effective. It deals with the collection of patient’s information, diagnosis details, etc. Traditionally, it was done manually. The main function of the system is register and store patient details and doctor details and retrieve these details as and when required, and also to manipulate these details meaningfully System input contains patient details, diagnosis details, while system output is to get these details on to the screen. The Hospital Management System can be entered using a username and password. It is accessible either by an administrator or receptionist. Only they can add data into the database. The data can be retrieved easily. The data are well protected for personal use and makes the data processing very fast.

TABLE OF CONTENTS

List Of Tables

Pg.No

List of Figures

Pg. No

CHAPTER 1
INTRODUCTION

INTRODUCTION

The project Hospital Management system includes registration of patients, storing their details into the system, and also computerized billing in the pharmacy, and labs. The software has the facility to give a unique id for every patient and stores the details of every patient and the staff automatically. It includes a search facility to know the current status of each room. User can search availability of a doctor and the details of a patient using the id.

The Hospital Management System can be entered using a username and password. It is accessible either by an administrator or receptionist. Only they can add data into the database. The data can be retrieved easily. The interface is very user-friendly. The data are well protected for personal use and makes the data processing very fast.

1.1 BACKGROUND

A Hospital is a place where Patients come up for general diseases. Hospitals provide facilities like:-

- Consultation by Doctors on Diseases.
- Diagnosis for diseases.
- Providing treatment facility.
- Facility for admitting Patients (providing beds, nursing, medicines etc.)
- Immunization for Patients/Children.

Various operational works that are done in a Hospital are:-

- Recording information about the Patients that come.
- Generating bills.
- Recording information related to diagnosis given to Patients.
- Keeping record of the Immunization provided to children/patients.
- Keeping information about various diseases and medicines available to cure them.

These are the various jobs that need to be done in a Hospital by the operational staff and Doctors. All these works are done on papers.

The work is done as follows:-

- Information about Patients is done by just writing the Patients name, age and gender. Whenever the Patient comes up his information is stored freshly.
- Bills are generated by recording price for each facility provided to

Patient on a separate sheet and at last they all are summed up.

- Diagnosis information to patients is generally recorded on the document, which contains Patient information. It is destroyedafter some time period to decrease the paper load in the office.
- Immunization records of children are maintained in pre-formatted sheets, which are kept in a file.
- Information about various diseases is not kept as any document. Doctors themselves do this job by remembering various medicines.

All this work is done manually by the receptionist and other operational staff and lot of papers are needed to be handled and taken care of. Doctors have to remember various medicines available for diagnosis and sometimes miss better alternatives as they can't remember them at that time.

1.2 OBJECTIVES

The objective of " Hospital Management System" is to simply track the knowledge of all the staffs, patients, treatment provided, and prescription and also to produce periodic reports for analysis. The main goal of the software is to make a decent management tool. The main purpose of this software is to cut back the time taken through manual system so as to take care of all the records. This project is helpful to cut back the time and quality of maintaining the records. It also h

elps in correct maintenance of patient and patient details. This project has GUI primarily based software system that can facilitate in storing, updating, and retrieving the information through varied user- friendly menu driven modules.

1.3.1Purpose Of the System

Hospital management system (HMS) is an element of medical informatics solutions that focuses mainly on the administration needs of hospitals. An HMS is a computer or web-based application that manages the functioning of an entire hospital. This customizable integrated system is designed to manage all the operations of the hospital, such as administration, patient details, medical history of the patient, appointment booking, inventory management, billing, bed management, drug management, revenue management, Electronic Medical record (EMR)and so on.

Hospital Management Systems of multiple branches of the same hospital have to be integrated so that if a patient consults a doctor in a different branch, their medical history can be retrieved easily. HMS should provide secure storage of data and a cloud-based solution to access it from anywhere at any time.

HMS is essential for all healthcare establishments, be it hospitals, nursing homes, health clinics, rehabilitation centres, dispensaries, or clinics. The main goal is to computerise all the details regarding the patient and the hospital. The installation of this healthcare software results in improvement in administrative functions and hence better patient care, which is the prime focus of any healthcare unit.

Benefits of implementing a hospital management system:

- **Appointment booking**
  - Helps patients cut the long queue and saves their time
  - Is equipped with features like automated email and text message reminders
- **Role-Based Access Control**
  - Allows employees to access only the necessary information to effectively perform their job duties
  - Increases data security and integrity
- **Overall cost reduction**
  - Cuts down paper costs as all the data are computerised
  - No separate costs for setting up physical servers
- **Data accuracy**
  - Removes human errors
  - Alerts when there's a shortage of stock
- **Data security**
  - Helps to keep patients records private
  - Restricts access through role-based access control
- **Revenue management**
  - Makes daily auditing simple
  - Helps with statistics and other financial aspects

1.3.2 Scope of the Project:-

1) Information about Patients is done by just writing the Patients name, age and gender. Whenever the Patient comes up his information is stored freshly.
2) Bills are generated by recording price for each facility provided to Patient on a separate sheet and at last they all are summed up.
3) Diagnosis information to patients is generally recorded on the document, which contains Patient information. It is destroyed after some time period to decrease the paper load in the office.
4) Immunization records of children are maintained in pre-formatted sheets, which are kept in a file.
5) Information about various diseases is not kept as any document. Doctors themselves do this job by remembering various medicines.

All this work is done manually by the receptionist and other operational staff and lot ofpapers are needed to be handled and taken care of. Doctors have to remember various medicines available for diagnosis and sometimes miss better alternatives as they can't remember them at that time.

1.3.3 Applicability

As long as each stage implementation needs to be accurate and explicit, the clinic management system provides certain automation of many vital daily processes. The hospital system software covers the services that unify and simplify the work of healthcare professionals as well as their interactions with patients. There is always the wide choice of features that can be included in the system. Moreover, the most important thing they are created to streamline various procedures that meet the needs of all the users. The hospital management system feature list is concentrated on providing the smooth experience of patients, staff and hospital authorities. It might seem that their expectations differ, they still are covered by components of the hospital information system. Quality and security still remain the main criteria of the medical industry. It is also known for the constant and rapid changes to improve the efficiency of medical services and satisfaction of the patients. Hospital management has greatly changed over the last decades. Business expertise, modern technologies, connected devices, mobile apps, and knowledge of healthcare are key elements for the implementation of hospital management system project. The number of healthcare providers has increased and the patients have a wide choice of medical specialists. The interactions between the hospital and the patient can be simplified for the convenience of both sides. Each institution has the opportunity to create the efficient, clear and fast delivering healthcare model.

1.4 Existing System:

The objective of the project is to automate maximum manual processes in the current system. The current system is totally manual.

They currently work with lot of paper work. They daily note down the details of the patient who has visited them and the prescription that they gave to the patients. They also have they records of medicines which is available in the hospital chemist. The bill is also generated on the basis of the seriousness of the patient health all this work is done manually. The prescription is also given in a written paper and made an entry in the record book. The bill is generated based two category they are

1. IPD patients
2. OPD patients

If the patient is an IPD patient then the entry is done in the general register and if the patient is an OPD patient then the entry is done in the scheduled register. So everything is done manually that's why it's so time consuming and error free.

Drawback of Existing System

- Manual work
- Time consuming
- Difficult for data retrieval
- More errors
- Physical storage space required is more
- Wastage of money on stationary
- Improper maintenance of data
- Lack of co-ordination employees

1.5 Proposed System

To overcome all these difficulties, we have decided to prepare and present "Healthcare". This system consists of facilities to provide immediate data entries and data retrieval techniques. This system is implemented to save the time as well as it provides perfection in work.

Proposed system will consist of following modules:

- IPD Registration
- OPD Billing
- Billing
- Database
- Admit Patients

Admit Patients: The administrator take the admission of the patients in OPD.

OPD Registration: Where the patient has to be admitted is decided here seeing the seriousness of the patient.
Database: The module keeps all the data of the patients doctors in its own database safely. It can be referred anytime.

Billing: The bill is generated once the patient is discharged.

IPD Patients: Details of the ICU patients.

Advantages of Proposed System:

To cover the Drawbacks of Existing System following features are added in "Hospital Management System" project

- Everything will be computerized so the time is saved.

- Proper maintenance of data for long period of time.
- Human error will be less so long period of time.
- Less chance of corruption.
- Space will be saved as there will b no stronger of documents and files.

Good co-ordination between employees

1.6 FEASIBILITY STUDY

The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are:

- **Economic Feasibility**

  This study is carried out to check the economic impact will have on the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customised products have to be purchased.

- **Technical Feasibility**

  This study is carried out to check the technical feasibility, that is,the technical requirements of the system. Any system developed must not have a high demand on the available available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes for the implementing this system.

- **Operational Feasibility**

  The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

CHAPTER 2
SURVEY OF TECHNOLOGY

2.1 INTRODUCTION ANDROID

Android is an open source and Linux-based Operating System for mobile devices such as smartphones and tablet computers. Android was developed by the Open Handset Alliance, led by Google, and other companies.

Android offers a unified approach to application development for mobile devices which means developers need only develop for Android, and their applications should be able to run on different devices powered by Android.

Components:

There are following four main components that can be used within an Android application −

Sr.No	Components & Description
1	Activities They dictate the UI and handle the user interaction to the smart phone screen.
2	Services They handle background processing associated with an application.
3	Broadcast Receivers They handle communication between Android OS and applications.
4	Content Providers They handle data and database management issues.

Additional Components:

There are additional components which will be used in the construction of above mentioned entities, their logic, and wiring between them. These components are –

S.No	Components & Description
1	Fragments Represents a portion of user interface in an Activity.
2	Views UI elements that are drawn on-screen including buttons, lists forms etc.
3	Layouts View hierarchies that control screen format and appearance of the views.
4	Intents Messages wiring components together.
5	Resources External elements, such as strings, constants and drawable pictures.
6	Manifest Configuration file for the application.

Sr.No.	Folder, File & Description
1	Java This contains the .java source files for your project. By default, it includes an MainActivity.java source file having an activity class that runs when your app is launched using the app icon.
2	res/layout This is a directory for files that define your app's user interface.
3	res/values
	This is a directory for other various XML files that contain a collection of resources, such as strings and colours definitions.
4	AndroidManifest.xml This is the manifest file which describes the fundamental characteristics of the app and defines each of its components.
5	Build.gradle This is an auto generated file which contains compileSdkVersion, buildToolsVersion, applicationId, minSdkVersion, targetSdkVersion, versionCode and versionName

DEVELOPMENT TOOLS

Android is a powerful operating system competing with Apple 4GS and supports great features. Few of them are listed below :-

Sr.No.	Feature & Description
1	Beautiful UI Android OS basic screen provides a beautiful and intuitive user interface.
2	Connectivity GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC and WiMAX.
3	Storage SQLite, a lightweight relational database, is used for data storage purposes.
4	Media support H.263, H.264, MPEG-4 SP, AMR, AMR-WB, AAC, HE-AAC, AAC 5.1, MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF, and BMP.
5	Messaging SMS and MMS
6	Web browser Based on the open-source WebKit layout engine, coupled with Chrome's V8 JavaScript engine supporting HTML5 and CSS3.
7	Multi-touch Android has native support for multi-touch which was initially made available in handsets such as the HTC Hero.
8	Multi-tasking User can jump from one task to another and same time various applications can run simultaneously.

9	Resizable widgets Widgets are resizable, so users can expand them to show more content or shrink them to save space.
10	Multi-Language Supports single direction and bi-directional text.
11	GCM Google Cloud Messaging (GCM) is a service that lets developers send short message data to their users on Android devices, without needing a proprietary sync solution.
12	Wi-Fi Direct A technology that lets apps discover and pair directly, over a high-bandwidth peer-to-peer connection.
13	Android Beam A popular NFC-based technology that lets users instantly share, just by touching two NFC-enabled phones together.

About ANDROID STUDIO:

Android Studio is Android's official IDE. It is purpose-built for Android to accelerate your development and help you build the highest-quality apps for every Android device.

Android studio

Android Studio is the official integrated development environment (IDE) for Android application development. It is based on the IntelliJ IDEA, a Java integrated development environment for software, and incorporates its code editing and developer tools.

IDE for Android application development. Android Studio and the Software Development Kit can be downloaded directly from Google.

Android studio features:

- Kotlin programming language support
- Android profiler: memory, CPU, network
- Java 8 language features built-in
- Faster build times
- Device file explorer
- Android Instant Apps support
- Adaptive icon wizard
- XML and downloadable fonts
- Android Things support
- Layout editor improvements
- APK profiling and debugging
- Layout inspector improvements
- Improved Gradle sync speed
- AAPT2 is now enabled by default
- Firebase app indexing assistant
- App links assistant

This are some new features of Android Studio.

2.2 WHY MYSQL AS BACK END?

Features of MySQL:-

Relational Database System: Like almost all other database systems on the market, MySQL is a relational database system.

Client/Server Architecture: MySQL is a client/server system. There is a database server (MySQL) and arbitrarily many clients (application programs), which communicate with the server; that is, they query data, save changes, etc. The clients can run on the same computer as the server or on another computer (communication via a local network or the Internet).

SQL compatibility: MySQL supports as its database language -- as its name suggests – SQL (Structured Query Language). SQL is a standardized language for querying and updating data and for the administration of a database. There are several SQL dialects (about as many as there are database systems). MySQL adheres to the current SQL standard (at the moment SQL: 2003), although with significant restrictions and a large number of extensions.

MySQL Features:

- **Relational Database Management System (RDBMS):** MySQL is a relational database management system.
- **Easy to use:** MySQL is easy to use. You have to get only the basic knowledge of SQL. You can build and interact with MySQL with only a few simple SQL statements.

- **It is secure:** MySQL consist of a solid data security layer that protects sensitive data from intruders. Passwords are encrypted in MySQL.
- **Client/ Server Architecture:** MySQL follows a client /server architecture. There is a database server (MySQL) and arbitrarily many clients (application programs), which communicate with the server; that is, they query data, save changes, etc.
- **Free to download:** MySQL is free to use and you can download it from MySQL official website.
- **It is scalable:** MySQL can handle almost any amount of data, up to as much as 50 million rows or more. The default file size limit is about 4 GB. However, you can increase this number to a theoretical limit of 8 TB of data.
- **Compatibale on many operating systems:** MySQL is compatible to run on many operating systems, like Novell NetWare, Windows* Linux*, many varieties of UNIX* . MySQL also provides a facility that the clients can run on the same computer as the server or on another computer
- **Allows roll-back:** MySQL allows transactions to be rolled back, commit and crash recovery.
- **High Performance:** MySQL is faster, more reliable and cheaper because of its unique storage engine architecture.
- **High Flexibility:** MySQL supports a large number of embedded applications which makes MySQL very flexible.
- **High Productivity:** MySQL uses Triggers, Stored procedures and views which allows the developer to give a higher productivity.

Disadvantages / Drawback of MySQL:

Following are the few disadvantages of MySQL:

- MySQL version less than 5.0 doesn't support ROLE, COMMIT and stored procedure.
- MySQL does not support a very large database size as efficiently.
- MySQL doesn't handle transactions very efficiently and it is prone to data corruption.
- MySQL is accused that it doesn't have a good developing and debugging tool compared to paid databases.
- MySQL doesn't support SQL check constraints.

Chapter 3

REQUIREMENT ANALYSIS AND PLANNING

3.1 Planning and scheduling

What is SDLC?

SDLC consists of a detailed plan which explains how to plan, build, and maintain specific software. Every phase of the SDLC lifecycle has its own process and deliverables that feed into the next phase.

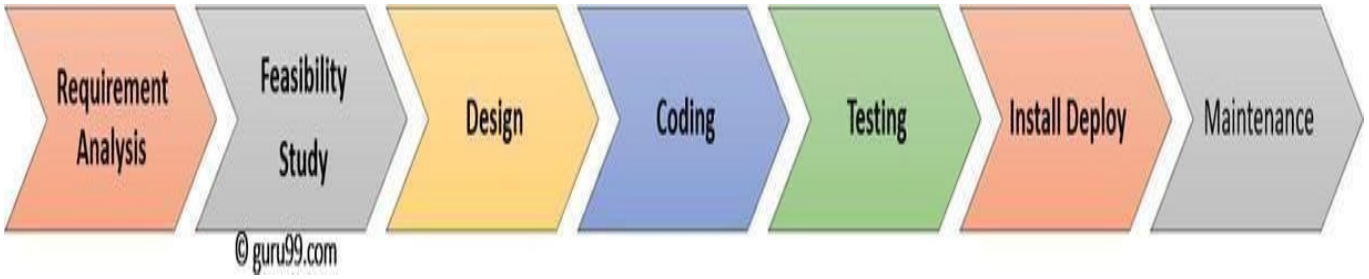
Why SDLC?

Here, are prime reasons why SDLC is important for developing a software system.

- It offers a basis for project planning, scheduling, and estimating
- Provides a framework for a standard set of activities and deliverables
- It is a mechanism for project tracking and control
- Increases visibility of project planning to all involved stakeholders of the development process
- Increased and enhance development speed
- Improved client relations
- Helps you to decrease project risk and project management plan overhead

SDLC Phases

The entire SDLC process divided into the following stages:



- Phase 1: Requirement collection and analysis
- Phase 2: Feasibility study:
- Phase 3: Design:
- Phase 4: Coding:
- Phase 5: Testing:
- Phase 6: Installation/Deployment:
- Phase 7: Maintenance:

Phase 1: Requirement collection and analysis:

The requirement is the first stage in the SDLC process. It is conducted by the

senior team members with inputs from all the stakeholders and domain experts in the industry. Planning for the quality assurance requirements and recognization of the risks involved is also done at this stage.
This stage gives a clearer picture of the scope of the entire project and the anticipated issues, opportunities, and directives which triggered the project. Requirements Gathering stage need teams to get detailed and precise requirements. This helps companies to finalize the necessary timeline to finish the work of that system.

Phase 2: Feasibility study:

Once the requirement analysis phase is completed the next step is to define and document software needs. This process conducted with the help of 'Software Requirement Specification' document also known as 'SRS' document. It includes everything which should be designed and developed during the project life cycle.

There are mainly five types of feasibilities checks:

- **Economic:** Can we complete the project within the budget or not?
- **Legal:** Can we handle this project as cyber law and other regulatory framework/compliances.
- **Operation feasibility:** Can we create operations which is expected by the client?
- **Technical:** Need to check whether the current computer system can support the software
- **Schedule:** Decide that the project can be completed within the given schedule or not.

Phase 3: Design:

In this third phase, the system and software design documents are prepared as per the requirement specification document. This helps define overall system architecture.

This design phase serves as input for the next phase of the model.

There are two kinds of design documents developed in this phase:

High-Level Design (HLD)

- Brief description and name of each module
- An outline about the functionality of every module
- Interface relationship and dependencies between modules
- Database tables identified along with their key elements
- Complete architecture diagrams along with technologydetails
  Low-Level Design(LLD)
- Complete input and outputs for every module

Phase 4: Coding:

Once the system design phase is over, the next phase is coding. In this phase, developers start build the entire system by writing code using the chosen programming language. In the coding phase, tasks are divided into units or modules and assigned to the various developers. It is the longest phase of the Software Development Life Cycle process.
In this phase, Developer needs to follow certain predefined coding guidelines. They also need to use programming tools like compiler, interpreters, debugger to generate and implement the code.

Phase 5: Testing:

Once the software is complete, and it is deployed in the testing environment. The testing team starts testing the functionality of the entire system. This is done to verify that the entire application works according to the customer requirement. During this phase, QA and testing team may find some bugs/defects which they communicate to developers. The development team fixes the bug and send back to QA for a re-test. This process continues until the software is bug-free, stable, and working according to the business needs of that system.

Phase 6: Installation/Deployment:

Once the software testing phase is over and no bugs or errors left in the system then the final deployment process starts. Based on the feedback given by the project manager, the final software is released and checked for deployment issues if any.

Phase 7: Maintenance:

Once the system is deployed, and customers start using the developed system, following 3 activities occur

- Bug fixing - bugs are reported because of some scenarios which are not tested at all
- Upgrade - Upgrading the application to the newer versions of the Software
- Enhancement - Adding some new features into the existing software

The main focus of this SDLC phase is to ensure that needs continue to be met and that the system continues to perform as per the specification mentioned in the first phase.

3.2 Problem Defination

Problems with conventional system

1. **Lack of immediate retrievals: -**The information is very difficult to retrieve and to find particular information like- E.g. - To find out about the patient's history, the user has to go through various registers. This results in inconvenience and wastage of time.

2. **Lack of immediate information storage: -** The information generated by various transactions takes time and efforts to be stored at right place.

3. **Lack of prompt updating: -** Various changes to information like patient details or immunization details of child are difficult to make as paper work is involved.

4. **Error prone manual calculation: -** Manual calculations are error prone and take a lot of time this may result in incorrect information. For example calculation of patient's bill based on various treatments.

5. **Preparation of accurate and prompt reports: -** This becomes a difficult task as information is difficult to collect from various registers.

3.3 Alternative Solutions

1. Improved Manual System:-

One of the alternative solutions is the improvement of the manual system. Anything, which can be done by using automated methods, can be done manually. But the question arises how to perform thing manually in a sound manner. Following are some suggestions, which can be useful in the manual system.

A more sophisticate register maintenance for various Patient Information, Doctor diary, Immunization Details and a good system for writing bill amount employees and stock availed for the customers can be maintained at central place.

Adequate staff may be maintained so that updations are made at the very moment at the same time. Proper person for proper work should be made responsible so that a better efficiency could be achieved. This needs a lot of work force.

2. Batch System:-

Another alternative solution can be used of computer based batch system for maintaining the information regarding purchase details, customers and employees. A batch system refers to a system in which data is processed in a periodical basis.

The batch system is able to achieve most of the goals and sub goals. But a batch system data is processed in sequential basis. Therefore batch system is not suggested.

3. Online System:-

This system **(HMS)** provides online storage/ updations and retrieval facility. This system promises very less or no paper work and also provides help to Doctor and operational staff.

In this system everything is stored electronically so very less amount of paper work is required and information can be retrieved very easily without searching here and

3.4 Cost analysis

Economic justification is generally the “Bottom Line” consideration for most systems. Economic justification includes a broad range of concerns that includes cost benefit analysis. In this we weight the cost and the benefits associated with the candidate system and if it suits the basic purpose of the organization i.e. profit making, the project is making to the analysis and design phase.

The financial and the economic questions during the preliminary investigation are verified to estimate the following:

- The cost to conduct a full system investigation.
- The cost of hardware and software for the class of application being considered.
- The benefits in the form of reduced cost.
- The proposed system will give the minute information, as a result the performance is improved which in turn may be expected to provide increased profits.

This feasibility checks whether the system can be developed with the available funds. The **Hospital Management System** does not require enormous amount of money to be developed. This can be done economically if planned judicially, so it is economically feasible. The cost of project depends upon the number of man- hours required.

3.5 System Requirements

Hardware Requirement & Software Requirement:

Hardware Requirements/Configuration (PC)

SYSTEM	Laptop or Desktop
Processor	(Computer) Pentium Quad Core
Processor Speed	1.5 GHz or more
RAM	4 GB or More
Hard Disk	20 GB or More

Hardware Requirements/Configuration (smartphones)

SYSTEM	Android Smartphones
Processor	Quad Core
Processor Speed	1.5 GHz or more
RAM	1 GB

Software requirements/Configuration

Operating System	Android os
BackEnd	MySQL 5.0
Android version	Android 4.3 or higher
IDE	Android Studio 1.5
Language	Android

3.6 Conceptual model

SPIRAL MODEL

The spiral model combines the idea of iterative development with the systematic, controlled aspects of the waterfall model. This Spiral model is a combination of iterative development process model and sequential linear development model i.e. the waterfall model with a very high emphasis on risk analysis. It allows incremental releases of the product or incremental refinement through each iteration around the spiral.

Spiral Model - Design

The spiral model has four phases. A software project repeatedly passes through these phases in iterations called Spirals.

Identification

This phase starts with gathering the business requirements in the baseline spiral. In the subsequent spirals as the product matures, identification of system requirements, subsystem requirements and unit requirements are all done in this phase.

This phase also includes understanding the system requirements by continuous communication between the customer and the system analyst. At the end of the spiral, the product is deployed in the identified market.

Design

The Design phase starts with the conceptual design in the baseline spiral and involves architectural design, logical design of modules, physical product design and the final design in the subsequent spirals.

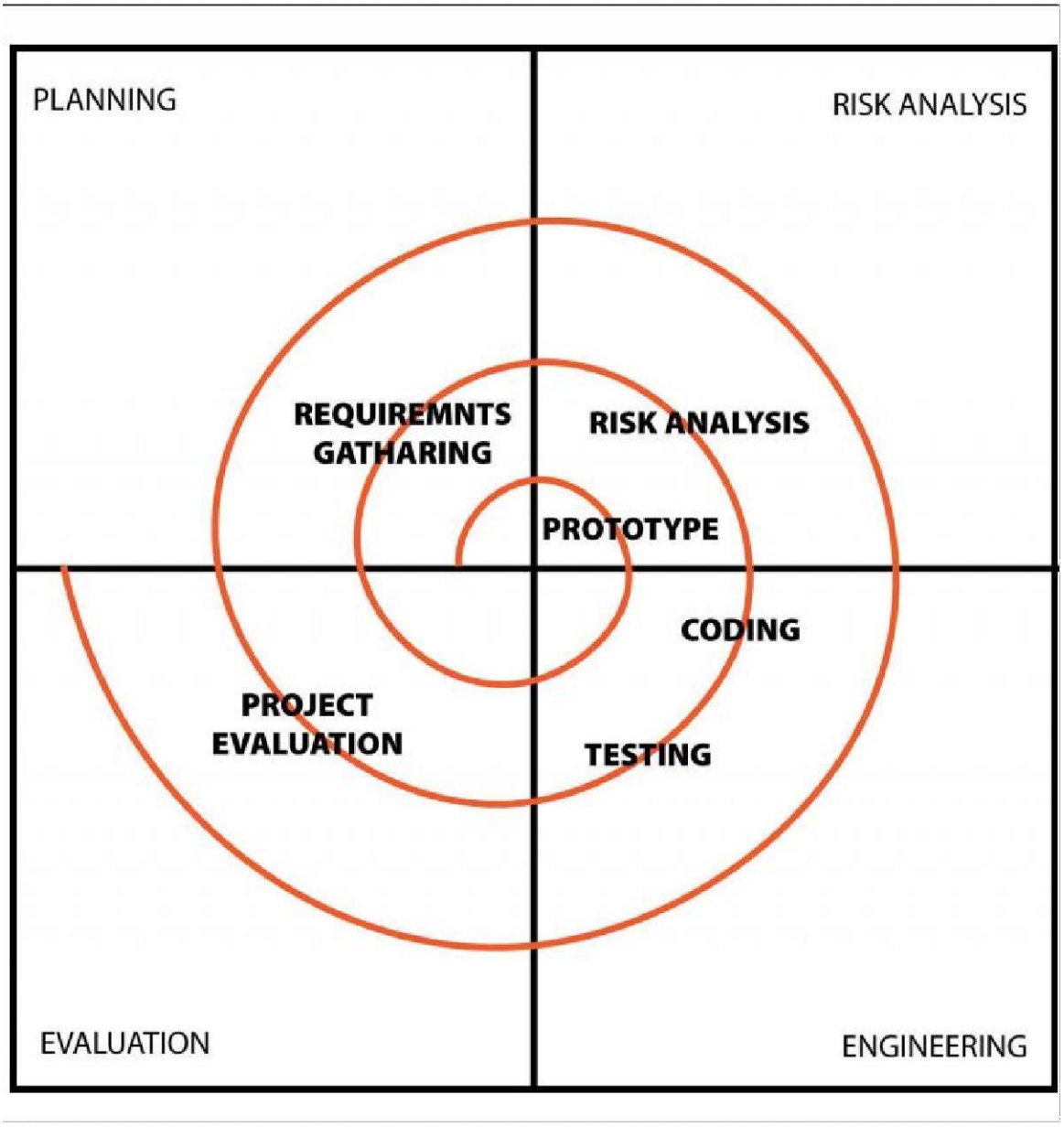
Construct or Build

The Construct phase refers to production of the actual software product at every spiral. In the baseline spiral, when the product is just thought of and the design is being developed a POC (Proof of Concept) is developed in this phase to get customer feedback.

Then in the subsequent spirals with higher clarity on requirements and design details a working model of the software called build is produced with a version number. These builds are sent to the customer for feedback.

Evaluation and Risk Analysis

Risk Analysis includes identifying, estimating and monitoring the technical feasibility and management risks, such as schedule slippage and cost overrun. After testing the build, at the end of first iteration, the customer evaluates the software and provides feedback



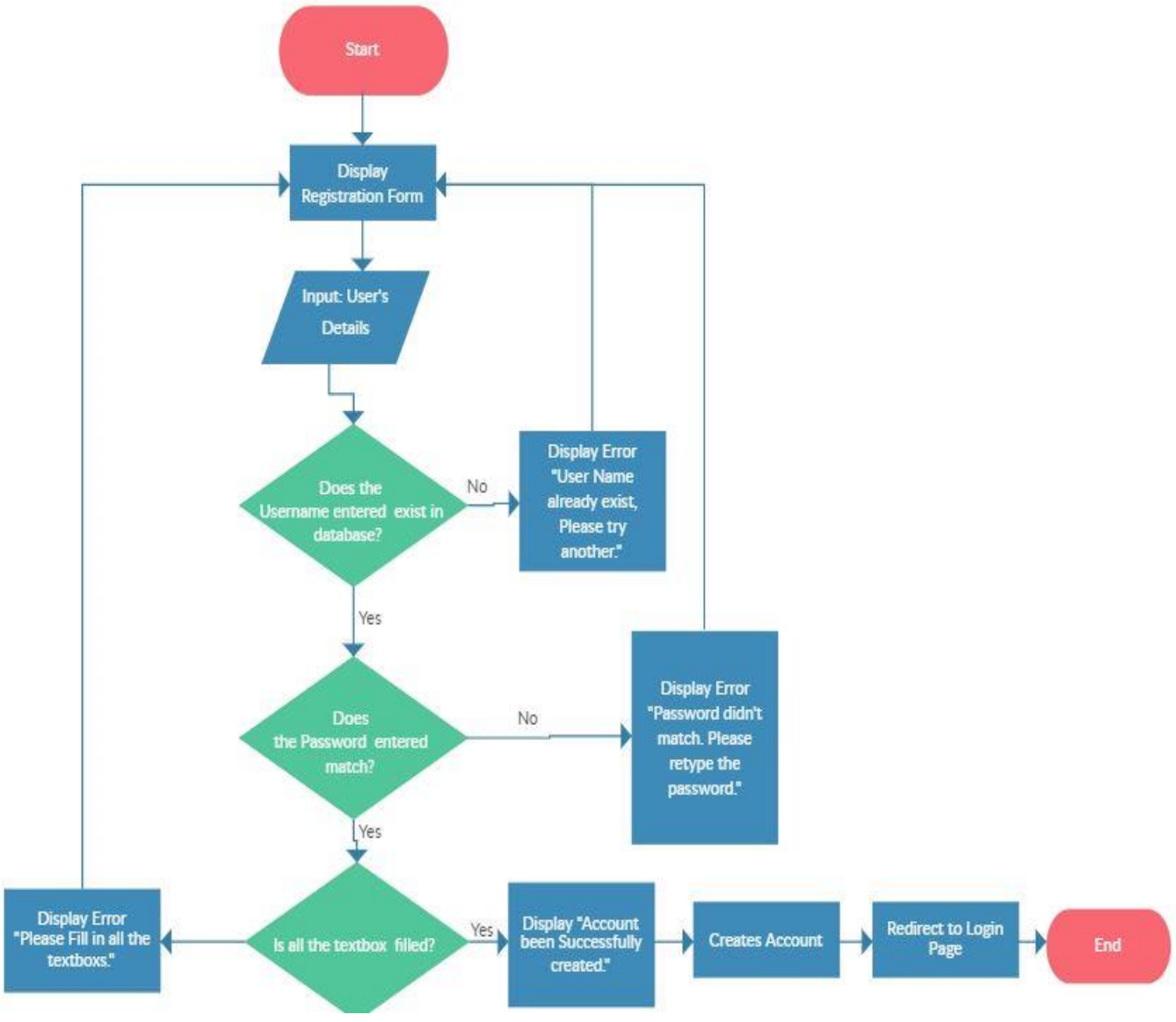
Spiral model

Chapter 4

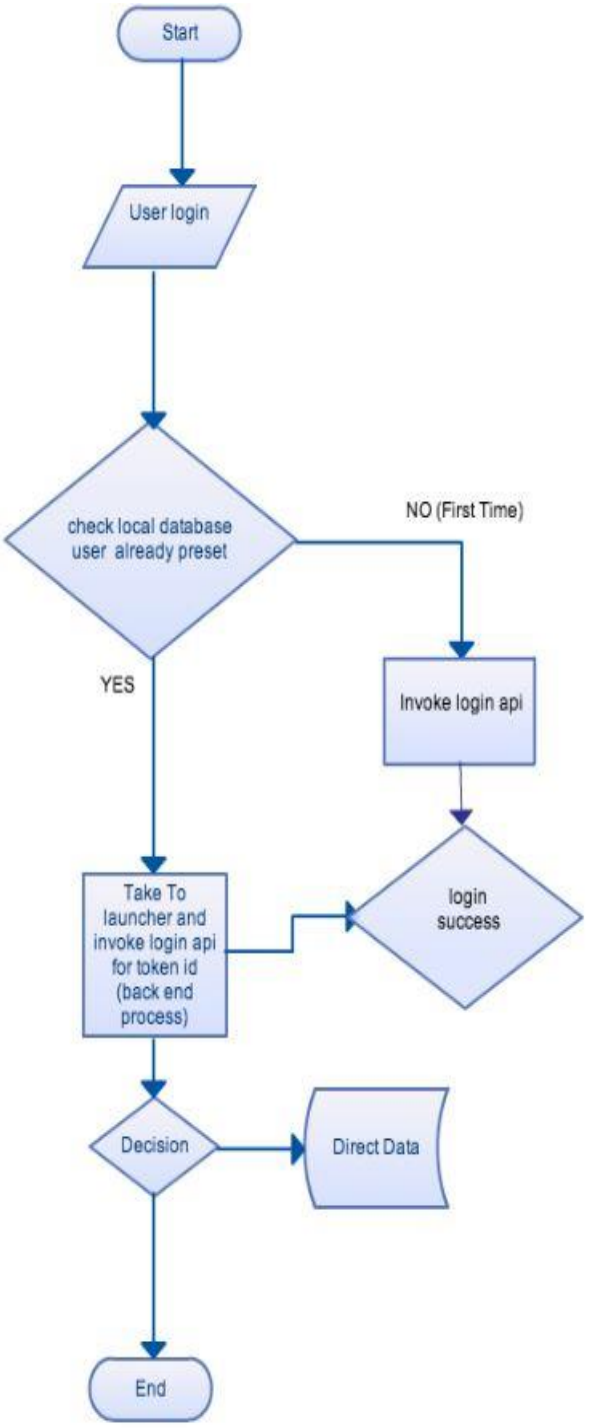
System Designing

4.1 Basic modules

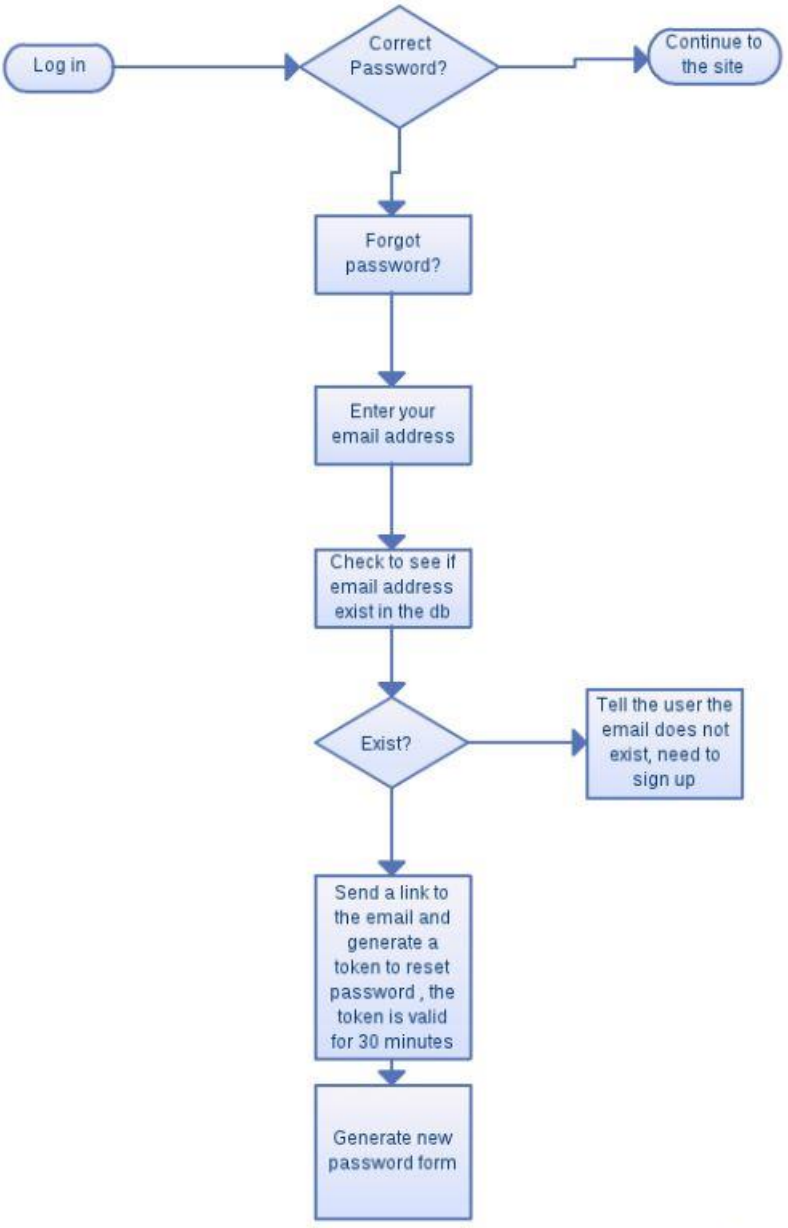
1. **Registration or signup module :-** The register module provides a conceptual framework for entering data on those patients in a way that: eases data entry & accuracy by matching the OpenMRS entry to the data source (usually paper files created at point of care), ties easily back to individual patient records to connect registers to patient data, and collects data elements to enable better supervision of treatment programs.



2. **Login module:-** This module displays a username and password login form. It also displays a link to retrieve a forgotten password. If user registration is enabled (in the Global Configuration settings), another link will be shown to enable self-registration for users.



3. **Forgot password module:-** The Forgotten Password module is a configurable feature. After enabling this feature, users see the Forgotten Password option on the user login web page. The Forgotten Password module uses challenge-response authentication to let users recover their passwords.



4.2 Gantt Chart

A Gantt chart, commonly used in project management, is one of the most popular and useful ways of showing activities (tasks or events) displayed against time.

On the left of the chart is a list of the activities and along the top is a suitable time scale. Each activity is represented by a bar, the position and length of reflects the start date, duration and end date of the activity.

This allows you to see at a glance: What the

various activities are?

When each activity begins and ends?
How long each activity is scheduled to last?

Where activities overlap with other activities, and by how much? The

start and end date of the whole project.

Gantt chart advantages:

Allows for efficient organization – In order for a Gantt chart to be successful, you first need to identify project elements or tasks. If you are using this type of chart you are essentially forced to focus on what truly needs to be done, thus making you somewhat more organized and encouraging a potentially higher chance of success.

Helps establish timeframes – Because many project elements often depend on other tasks, it can be tough to deduce how long one task should take and when to start and finish it by. Gantt charts use bars to indicate how long a task should take and what this does is give you a better perspective of the total project, and timeframe as a whole. Just be sure to consider time factors outside of the project such as holidays.

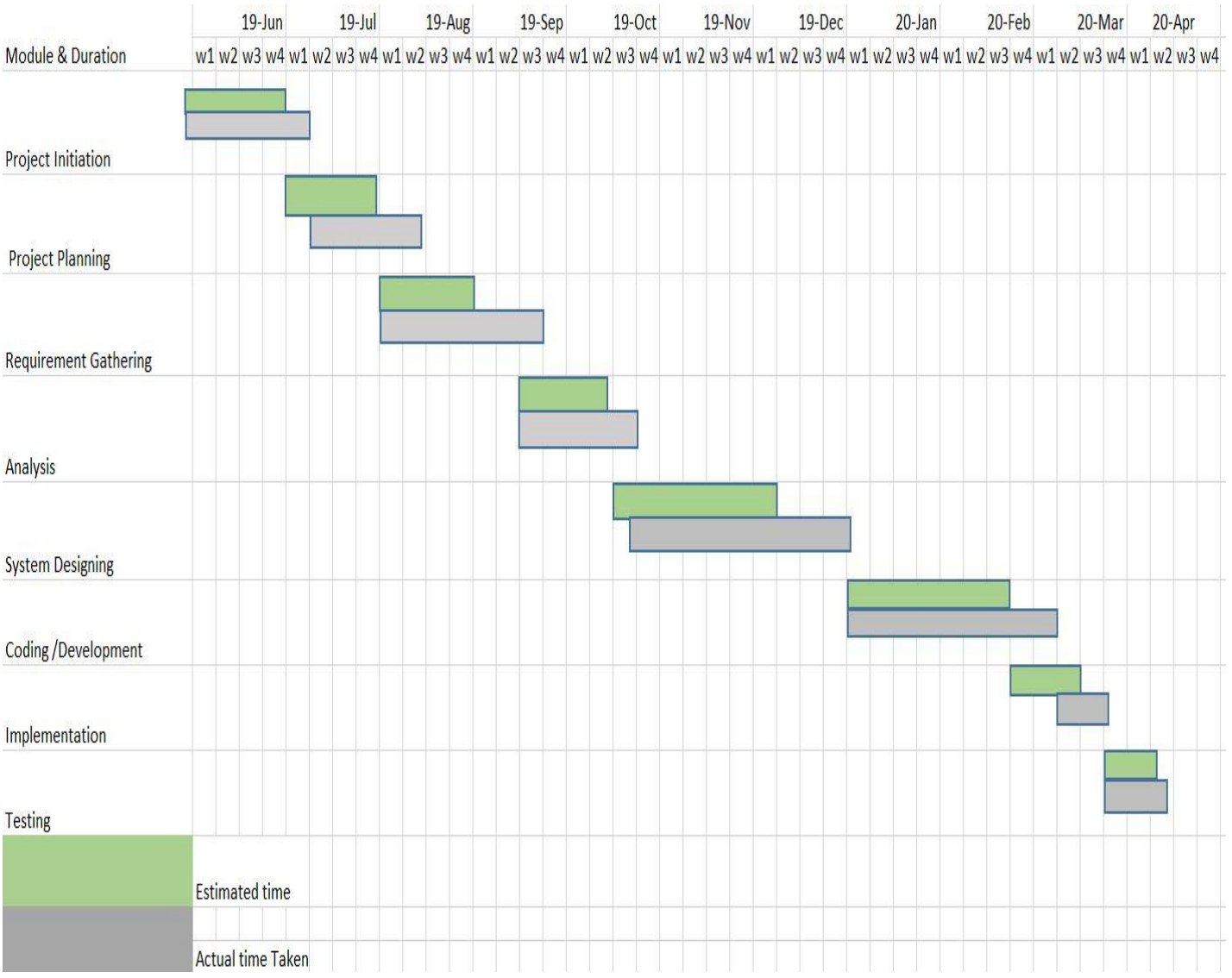
Highly visual – Gantt charts are visual, and give you an excellent way to instantly see and comprehend all of the different elements in once place, thus bringing thoughts and ideas together. Beyond that the visuals provide customers with an easy to see chart of what needs to be done next.

Gantt chart disadvantages:

Potentially overly complex – If you‟ve ever worked on a complex project, and looked at the Gantt chart, you know that these charts can be large and hard to read. For big projects businesses may need to hire specific managers to look after the details of the project, something which could be costly for small businesses that don't have an in house project manager.

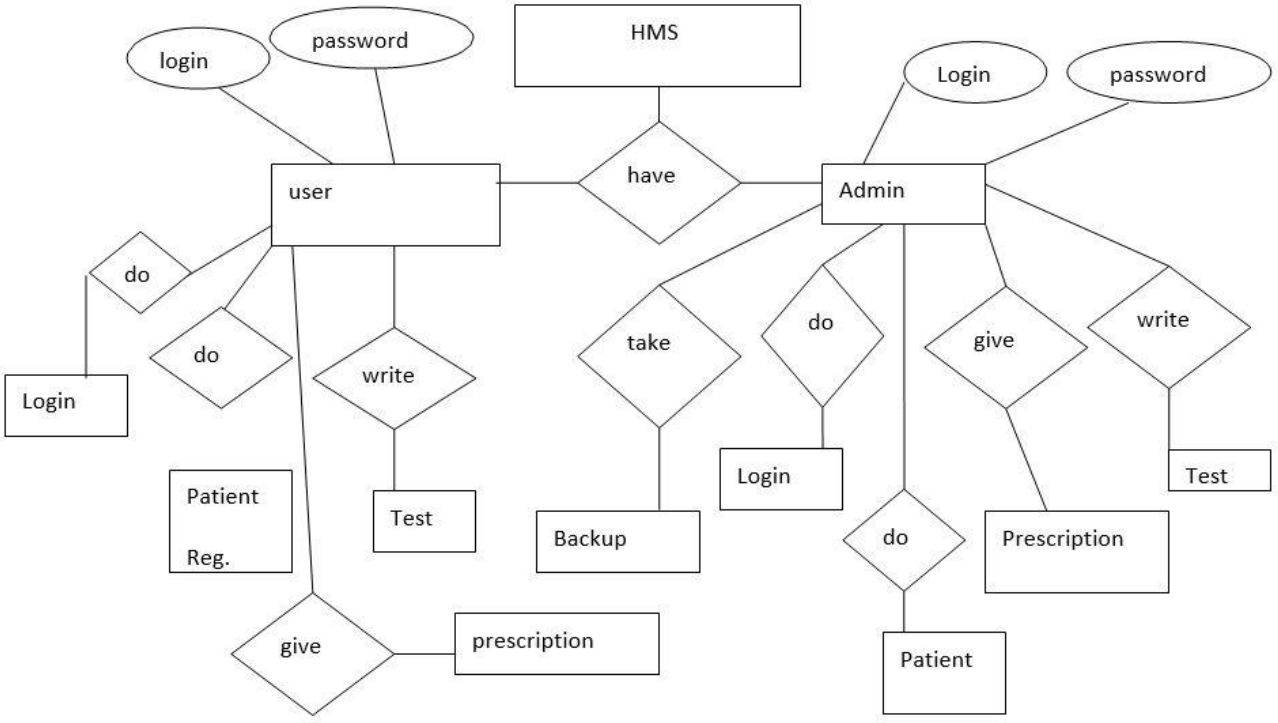
Need to be updated – Gantt charts are developed early in the planning stages of a project, there is a good chance that the project will change, thus the chart will need to be updated. Also, as tasks are completed or reviewed the chart will need to be updated to reflect these changes too. Any amendments take time, especially if there are dependent tasks that need to also be revised. It is a pretty sure thing that most people involved in the project probably don‟t have the time to do this.

Don't show the whole picture – Gantt charts show what tasks need to be done and the time they should take. They don't show how much work each task will involve or how many people/resources each task will require. This can give some people an incomplete picture or 20 the wrong idea about an individual task, which can cause issues as the project gets underway.



4.3 ER diagram

An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database. Lets have a look at a simple ER diagram to understand this concept.

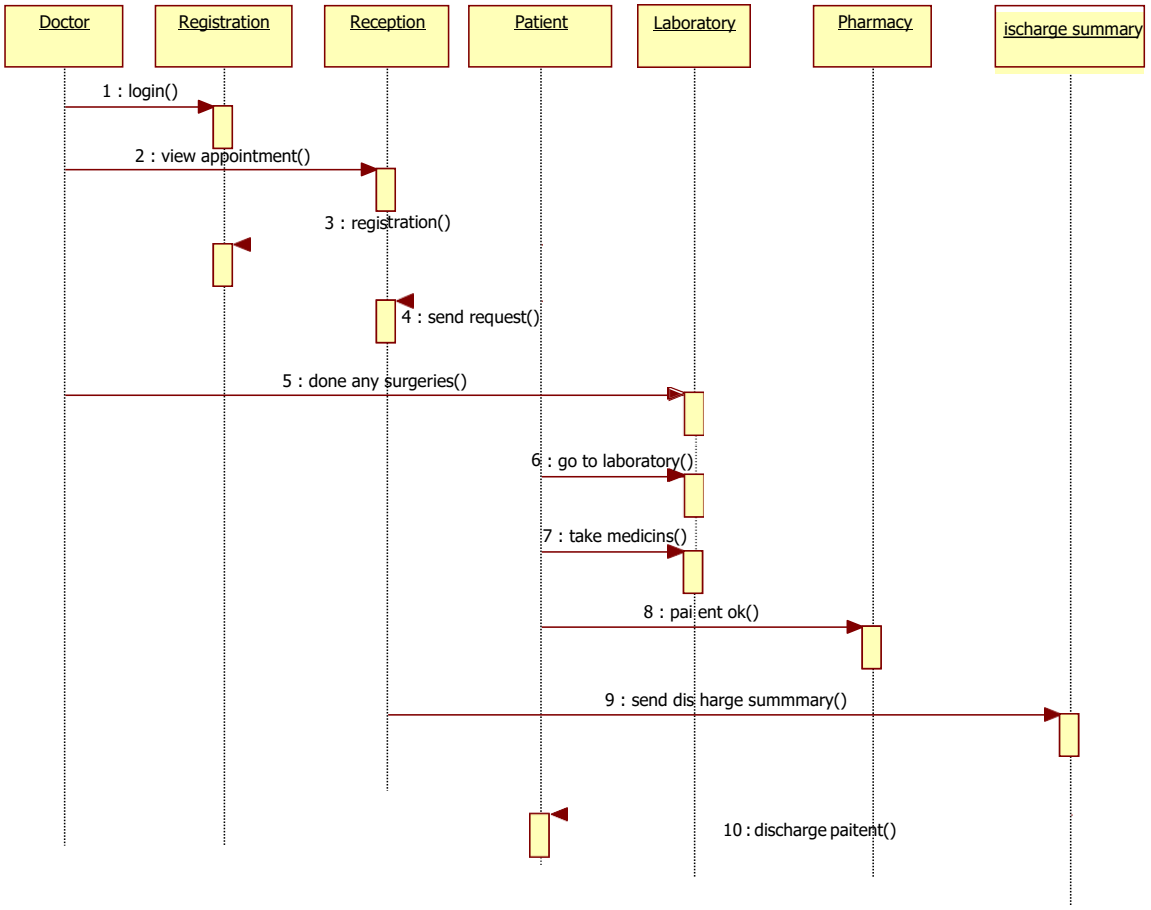


4.4 UML Diagrams

A diagram is the graphical presentation of a set of elements, most often rendered as a connected graph of vertices and arcs . you draw diagram to visualize a system from different perspective, so a diagram is a projection into a system. For all but most trivial systems, a diagram represents an elided view of the lements that make up a system. The same element may appear in all diagrams, only a few diagrams , or in no diagrams at all. In theory, a diagram may contain any combination of things and relationships

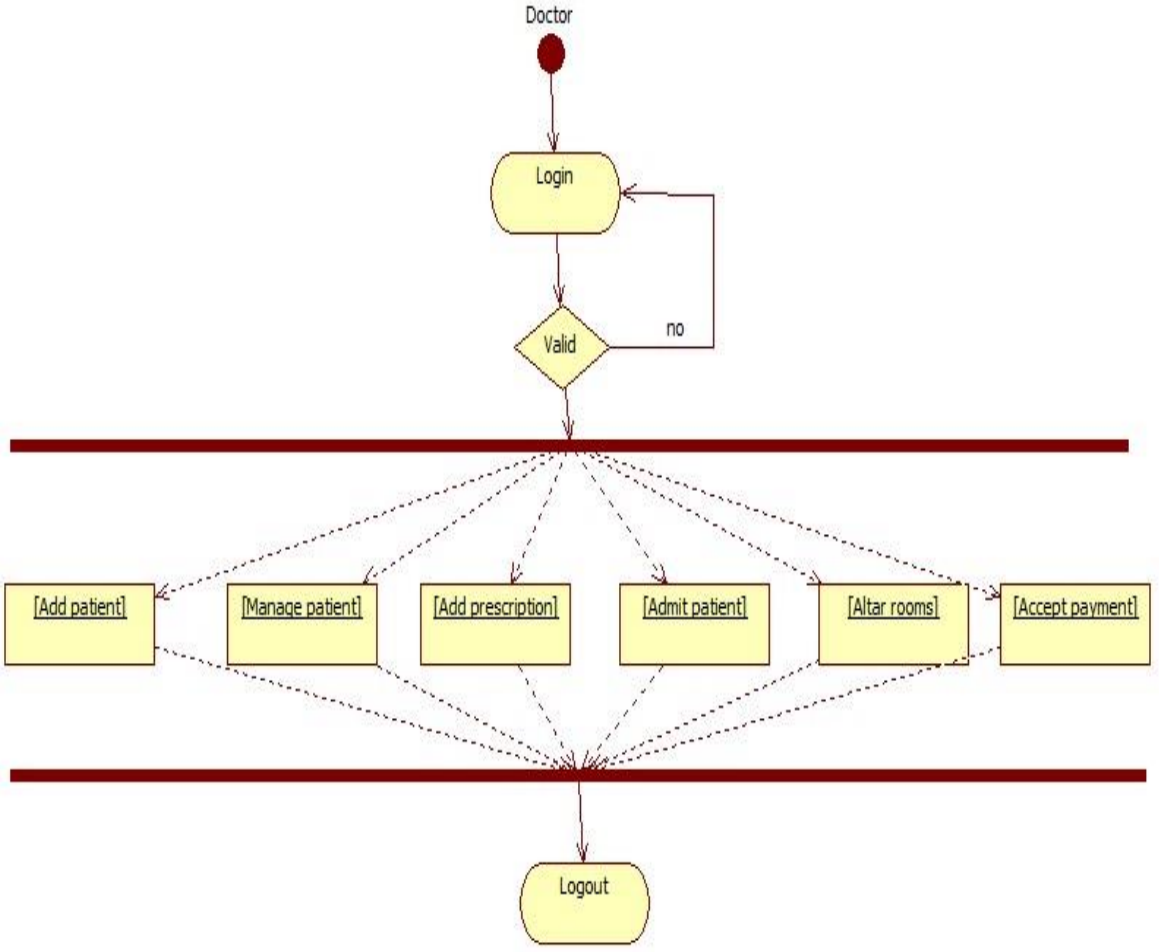
4.4.1 Sequence diagram

A **Sequence Diagram** is an interaction diagram that emphasis the time ordering of messages; a collaboration diagram is an interaction diagram that emphasizes the structural organization of the objects that send and receive messages. Sequence diagrams and collaboration diagrams are isomorphic, meaning that you can take one and transform it into the other.



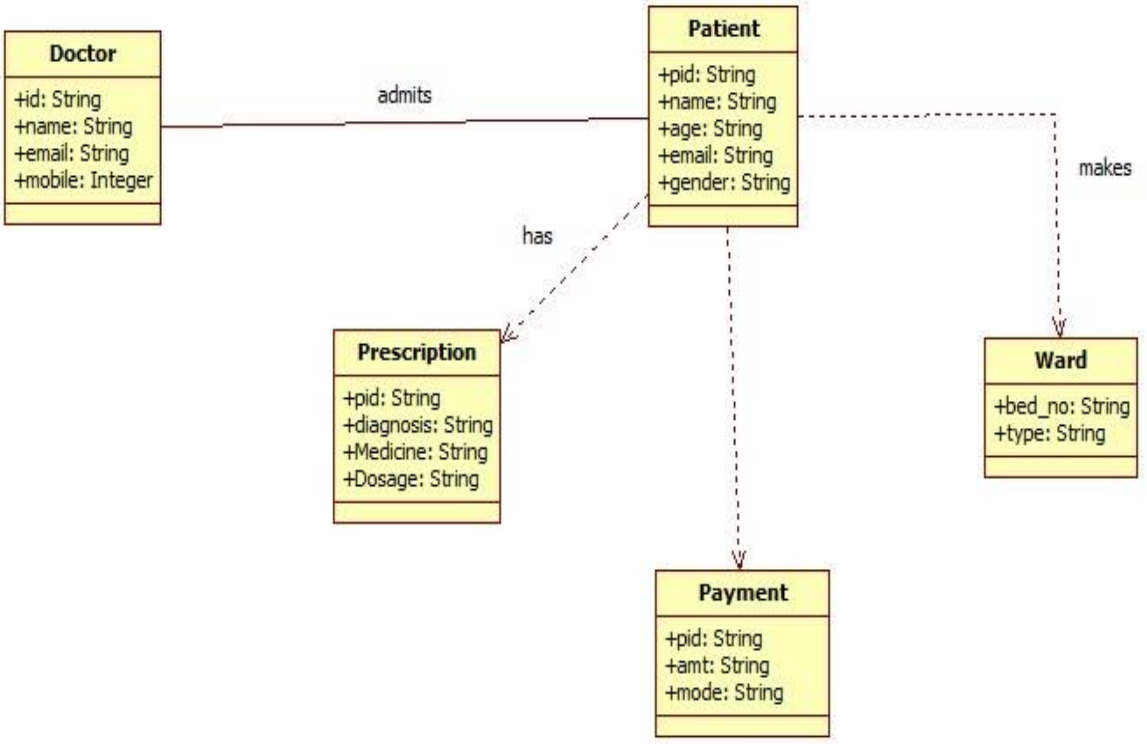
4.4.2 Activity Diagram

We use Activity Diagrams to illustrate the flow of control in a system and refer to the steps involved in the execution of a use case. We model sequential and concurrent activities using activity diagrams. So, we basically depict workflows visually using an activity diagram. An activity diagram focuses on condition of flow and the sequence in which it happens. We describe or depict what causes a particular event using an activity diagram.



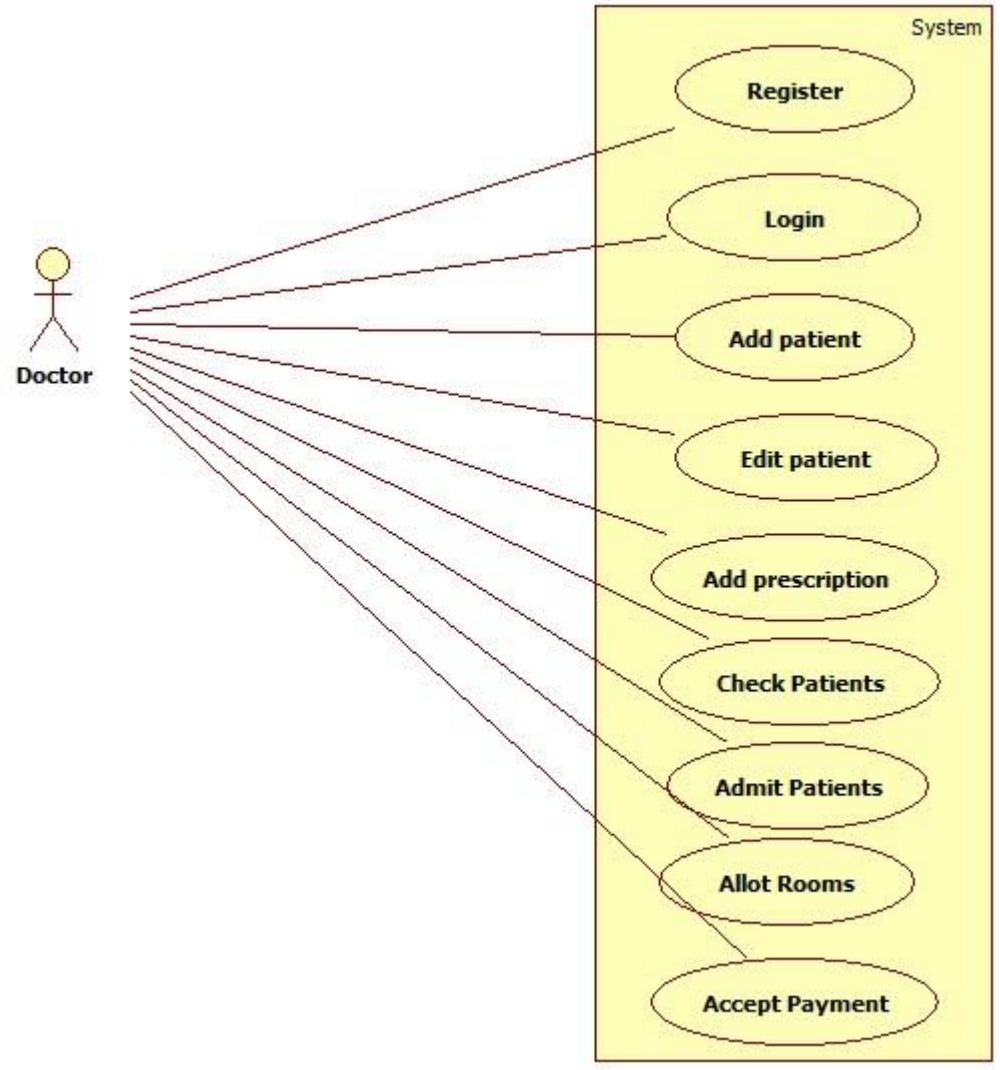
4.4.3 Class Diagram

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object- oriented languages



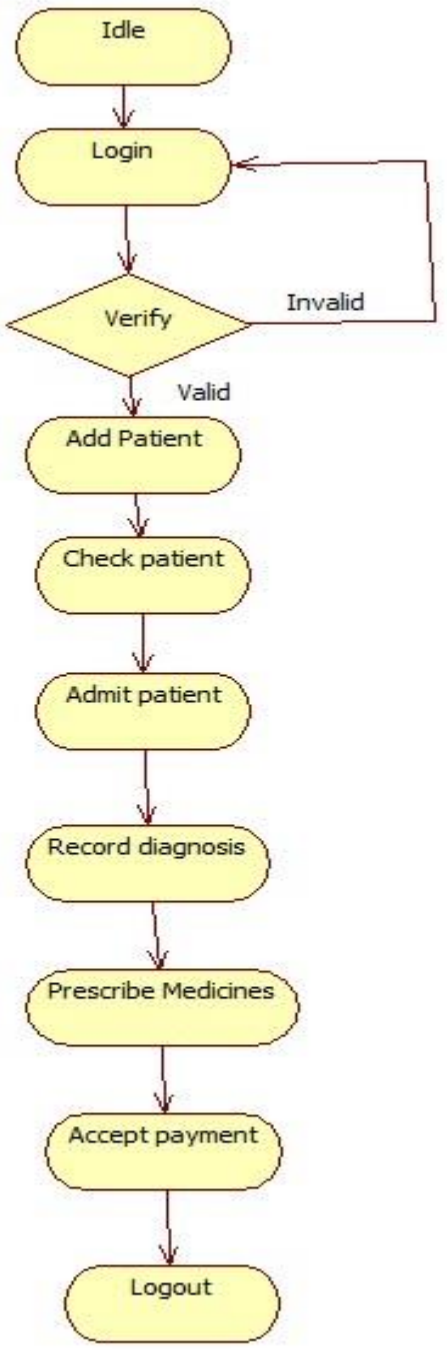
4.4.4. Use Case Diagram

A UML use case diagram is the primary form of system/software requirements for a new software program underdeveloped. Use cases specify the expected behavior (what), and not the exact method of making it happen (how). Use cases once specified can be denoted both textual and visual representation (i.e. use case diagram). A key concept of use case modeling is that it helps us design a system from the end user's perspective. It is an effective technique for communicating system behavior in the user's terms by specifying all externally visible system behavior



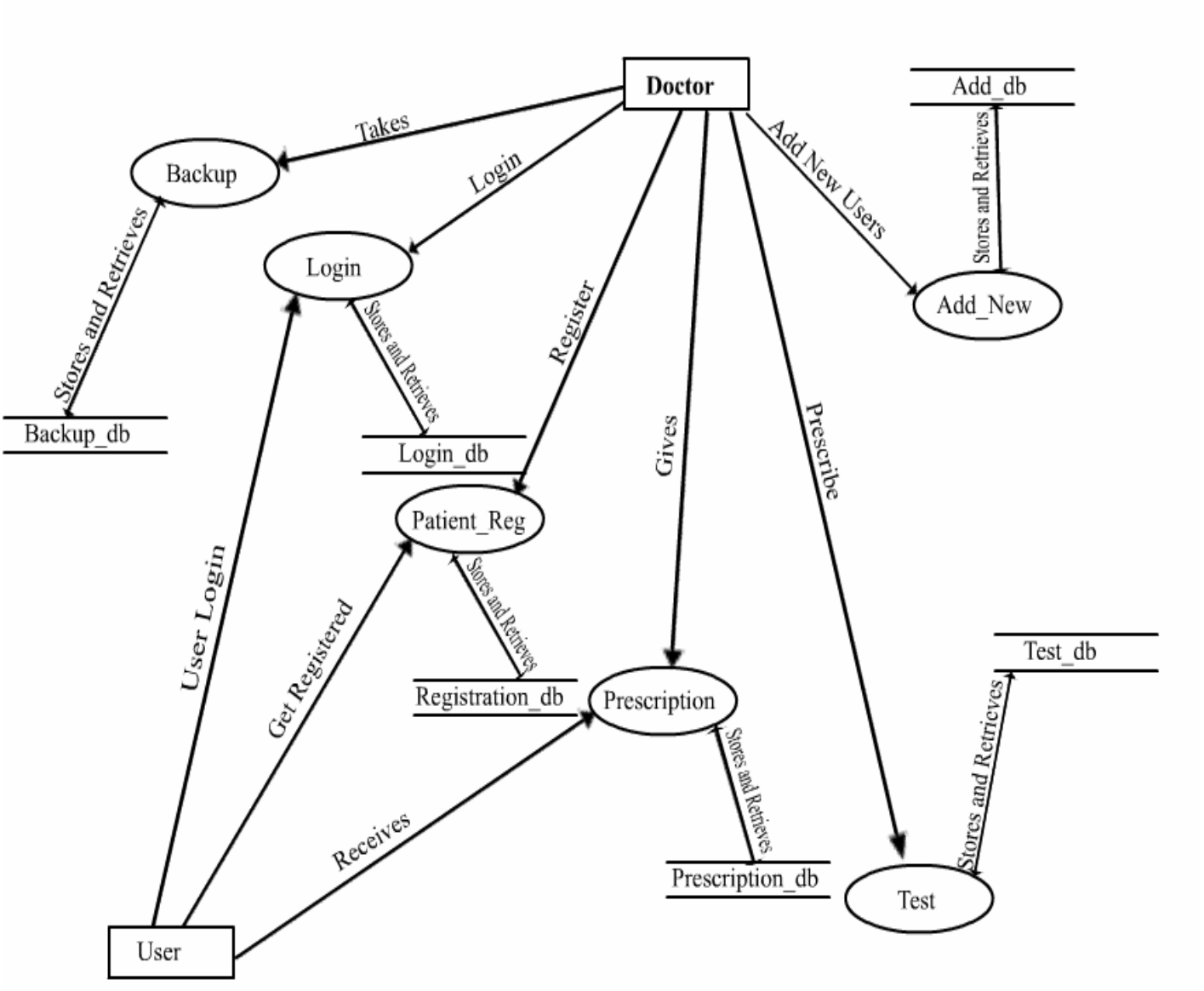
4.4.5 State Transistion Diagram

A **state diagram** is used to represent the condition of the system or part of the system at finite instances of time. It's a **behavioral** diagram and it represents the behavior using finite state transitions. State diagrams are also referred to as **State machines** and **State-chart Diagrams**. These terms are often used interchangeably. So simply, a state diagram is used to model the dynamic behavior of a class in response to time and changing external stimuli. We can say that each and every class has a state but we don't model every class using State diagrams. We prefer to model the states with three or more states.

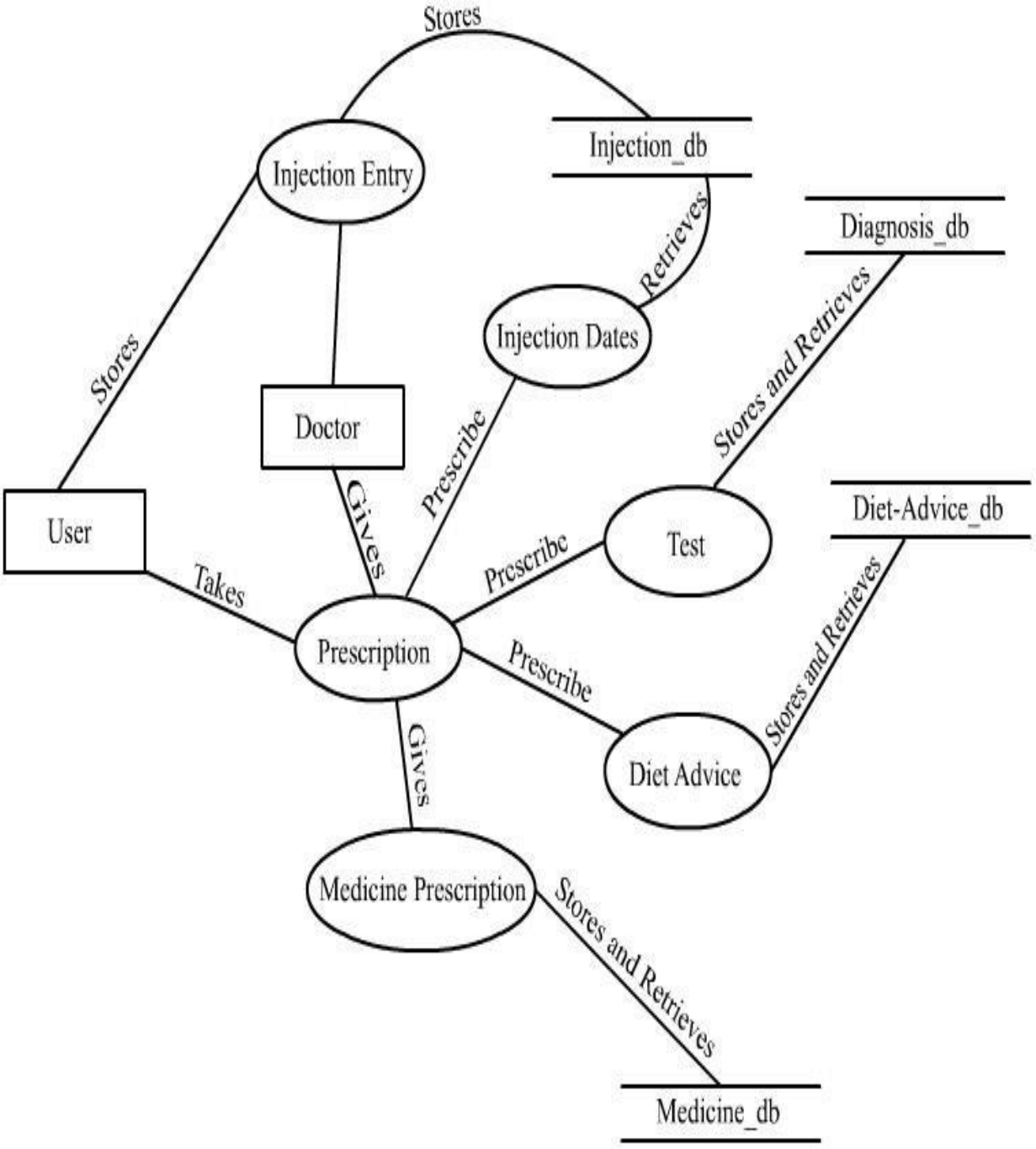


4.4.6 Data Flow Diagram

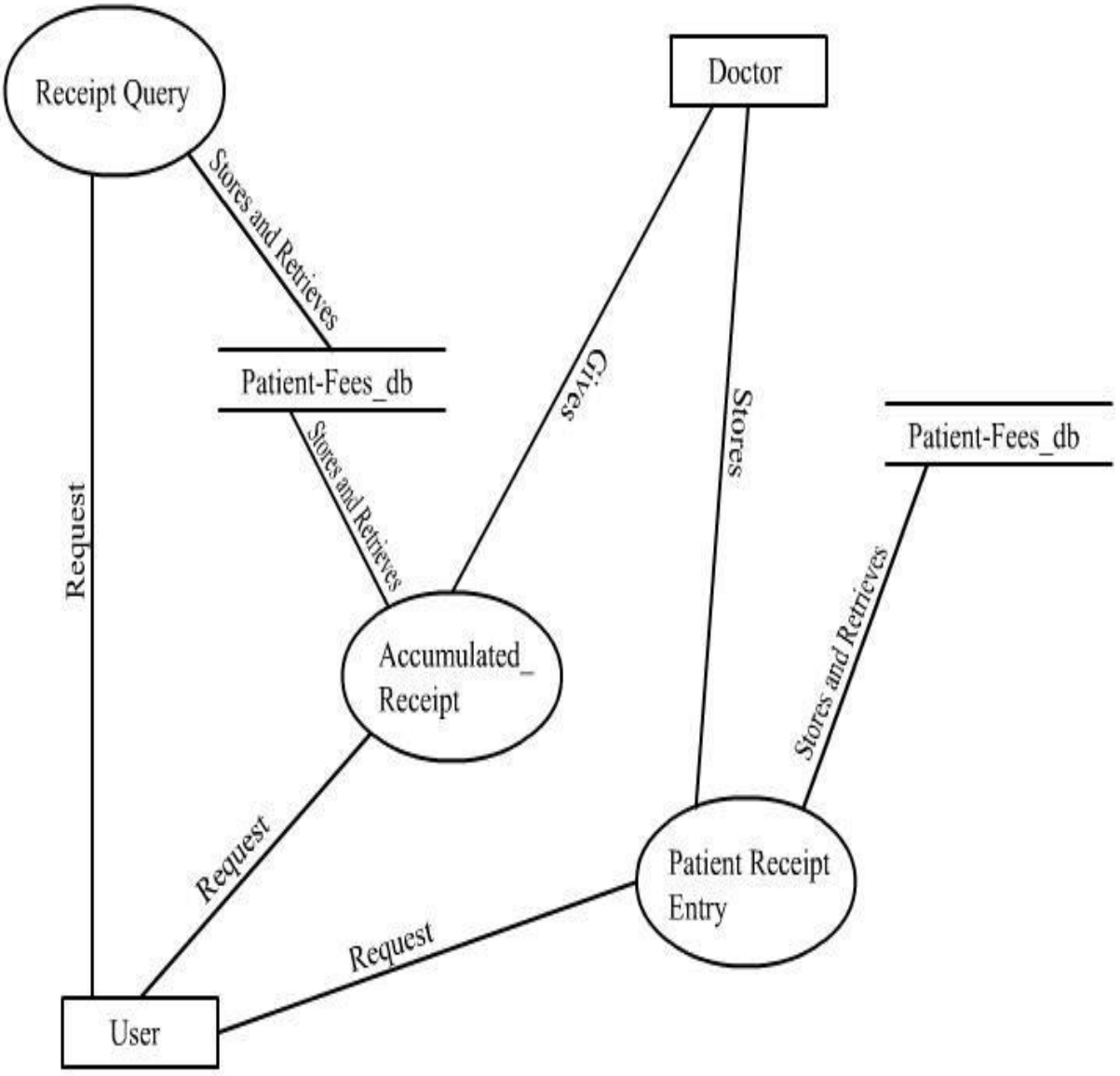
A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually "say" things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO



DFD level 0


Stores
Injection Entry
Injection_db
Retrieves
Diagnosis_db
Stores
Injection Dates
Stores and Retrieves
Doctor
Prescribe
Gives
User
Test
Diet-Advice_db
Takes
Prescribe
Prescription
Prescribe
Stores and Retrieves
Gives
Diet Advice
Medicine Prescription
Stores and Retrieves
Medicine_db

DFD Level 1


Receipt Query
Doctor
Stores and Retrieves
Patient-Fees_db
Gives
Stores
Patient-Fees_db
Request
Stores and Retrieves
Accumulated_
Receipt
Stores and Retrieves
Request
Patient Receipt
Entry
Request
User

DFD level 2

4.5 Data Base Table

Login_tbl			
Column Name	**Datatype**	**Length**	**nullable**
userid	Varchar	100	No
password	varchar	20	No
mobile	Varchar	10	No

Patient_tbl			
Column Name	**Datatype**	**Length**	**nullable**
pid	varchar	100	no
name	varchar	100	no
address	varchar	500	no
mobile	varchar	10	no
age	int	4	no
email	varchar	10	no
gender	varchar	10	No

Patient_diagnosis_tbl			
Column Name	**Datatype**	**Length**	**nullable**
pid	varchar	20	no
Date	date	8	no
diagnosis	varchar	500	no

Patient_Medicines_tbl			
Column Name	**Datatype**	**Length**	**nullable**
pid	varchar	20	no
date	date	8	no
mname	varchar	100	no
morning	varchar	10	no
afternoon	varchar	10	no
evening	varchar	10	no

bed_tbl			
Column Name	**Datatype**	**Length**	**nullable**
wardtype	varchar	20	no
bedno	int	8	no
cost	Int	8	no
status	Varchar	20	no

4.6 Database Design

Creating Database:-

```
mysql> create database HealthCare_db;
Query OK, 1 row affected (0.02 sec)

mysql> use Healthcare_db;
Database changed
mysql>
mysql>
mysql> create table login(userid varchar(100), password varchar(20));
Query OK, 0 rows affected (0.04 sec)

mysql>
mysql>
mysql> create table users(fname varchar(50),lname varchar(50), emailid varchar(100), mobile varchar(10),secret_question
varchar(500), answer varchar(500));
Query OK, 0 rows affected (0.02 sec)

mysql>
mysql>
mysql> create table patient(pid varchar(20),name varchar(100),address varchar(500),mobile varchar(10), age int, emailid
varchar(100),gender varchar(10));
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql>
mysql> create table patient_diagnosis(pid varchar(20),date date, diagnosis varchar(500));
Query OK, 0 rows affected (0.02 sec)

mysql>
```

```
mysql>
mysql>
mysql> create table patient_medicines(pid varchar(20),date date,mname varchar(100),morning varchar(10), afternoon varcha
r(10),evening varchar(10));
Query OK, 0 rows affected (0.02 sec)

mysql>
mysql> create table ipdpatient(pid varchar(20),date date,time time ,remarks varchar(500), bp varchar(20), temp varchar(1
0), pulse varchar(10));
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> create table ipdpatientmedicines(pid varchar(20),date date, time time,mname varchar(100),dose varchar(10));
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> create table beds(wardtype varchar(20),bedno int,cost int,status varchar(20));
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> create table patient_bed(pid varchar(20),start_date date,end_date date,remarks varchar(500),m_status varchar(10),
 pno varchar(20), company varchar(100),wardtype varchar(20),bedno int);
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> create table payment(pid varchar(20),date date, amt int,mode varchar(20));
Query OK, 0 rows affected (0.01 sec)

mysql>
```

Tables:-

```
mysql> show tables;
+-------------------------+
| Tables_in_healthcare_db |
+-------------------------+
| beds                    |
| ipdpatient              |
| ipdpatientmedicines     |
| login                   |
| patient                 |
| patient_bed             |
| patient_diagnosis       |
| patient_medicines       |
| payment                 |
| users                   |
+-------------------------+
10 rows in set (0.01 sec)
```

Login:-

```
mysql> desc login;
+----------+--------------+------+-----+---------+-------+
| Field    | Type         | Null | Key | Default | Extra |
+----------+--------------+------+-----+---------+-------+
| userid   | varchar(100) | YES  |     | NULL    |       |
| password | varchar(20)  | YES  |     | NULL    |       |
+----------+--------------+------+-----+---------+-------+
2 rows in set (0.01 sec)

mysql> desc users;
+-----------------+--------------+------+-----+---------+-------+
| Field           | Type         | Null | Key | Default | Extra |
+-----------------+--------------+------+-----+---------+-------+
| fname           | varchar(50)  | YES  |     | NULL    |       |
| lname           | varchar(50)  | YES  |     | NULL    |       |
| emailid         | varchar(100) | YES  |     | NULL    |       |
| mobile          | varchar(10)  | YES  |     | NULL    |       |
| secret_question | varchar(500) | YES  |     | NULL    |       |
| answer          | varchar(500) | YES  |     | NULL    |       |
+-----------------+--------------+------+-----+---------+-------+
6 rows in set (0.01 sec)
```

Patient:-

```
mysql>
mysql>
mysql> desc patient;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| pid     | varchar(20) | YES  |     | NULL    |       |
| name    | varchar(100)| YES  |     | NULL    |       |
| address | varchar(500)| YES  |     | NULL    |       |
| mobile  | varchar(10) | YES  |     | NULL    |       |
| age     | int(11)     | YES  |     | NULL    |       |
| emailid | varchar(100)| YES  |     | NULL    |       |
| gender  | varchar(10) | YES  |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
7 rows in set (0.01 sec)
```

Patient Diagnosis:-

```
mysql>
mysql> desc patient_diagnosis;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| pid       | varchar(20) | YES  |     | NULL    |       |
| date      | date        | YES  |     | NULL    |       |
| diagnosis | varchar(500)| YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

4.7 Security issues

This document explains how Symfony security issues are handled by the Symfony core team (Symfony being the code hosted on the main symfony/symfony Git repository).

Reporting a Security Issue

If you think that you have found a security issue in Symfony, don't use the bug tracker and don't publish it publicly. Instead, all security issues must be sent to **security [at] symfony.com**. Emails sent to this address are forwarded to the Symfony core team private mailing-list.

Resolving Process

For each report, we first try to confirm the vulnerability. When it is confirmed, the core team works on a solution following these steps:

1. Send an acknowledgement to the reporter;
2. Work on a patch;
3. Get a CVE identifier from mitre.org;
4. Write a security announcement for the official Symfony blog about the vulnerability. This post should contain the following information:
   - a title that always include the "Security release" string;
   - a description of the vulnerability;
   - the affected versions;
   - the possible exploits;
   - how to patch/upgrade/workaround affected applications;
   - the CVE identifier;
   - credits.
5. Send the patch and the announcement to the reporter for review;
6. Apply the patch to all maintained versions of Symfony;
7. Package new versions for all affected versions;
8. Publish the post on the official Symfony blog (it must also be added to the "Security Advisories" category);
9. Update the public security advisories database maintained by the FriendsOfPHP organization and which is used by the check:security command.

Chapter 5
Implementation and Testing

5.1 Implementation Approches

1. **Defining requirements:** Once the requirement analysis is done, the next stage is to surely document the software specifications and get them approved by the project stakeholders. This can be accomplished through the “SRS”- Software Requirement Specification document, which embraces all the product elements to be created and developed during the project life cycle.

2. **Designing:** In this stage, the requirements gathered in the SRS document is used as information to obtain the software architecture. Moreover, this phase also consists of storyboarding or wireframing software that is needed for functionality. Through this, the developers then create either rough working models, or illustrates how the software will work, how it will look, how usage flows will move from screen to screen, and more.

3. **Implementation or Coding:** In this stage of SDLC, the exact development begins, and the programming is built. The execution of design begins concerning script code. Developers have to follow the coding guidelines defined by their management, and programming tools like compilers, interpreters, debuggers, etc. are used to generate and implement the code

4. **Testing:** After the code is generated, it is tested against the specifications to ensure that the products are solving the needs directed and inferred during the requirements stage. During this phase, several testing like unit testing, integration testing, system testing, and approval testing are done.

5. **Deployment:** Once the software is approved, and no bugs or errors are asserted, then it is deployed. Later, based on the assessment, the software is delivered as it is or with suggested augmentation in the object segment. After the software is deployed, then its sustenance begins.

6. **Maintenance:** Once the client starts using the developed software, then the real issues start coming up. In this stage, the team is required to fix the issues, roll out new features and refine the functionalities as required. The method where the care is taken for the finished product is thus known as maintenance.

5.2 Program list

Program	Description
1. Signup.java	In this program user i.e a Doctor can register himself into the system.
2. Login.java	In this program the user can does login with the help of valid userID and password
3. Addpatient.java	In this program the user which means a doctor can add patient to the system
4. EditPatient.java	In this program the user which means a doctor can edit patient details like Name, Age, Gender, ETC to the system
5. patientDiagnosis.java	In this program the user which means a doctor can Diagnos patient's problem related to health and add that details to the system
6. Addward.java	In this program the user which means a doctor can add patient to the system
7. Payment.java	In this program the user which means a doctor can check and verify the payments of the patients
8. OTP_alert.xml	In this program the user which means a doctor email id will be verified for login purpose
9. Mainpage.java	In this program the user which means a doctor can be able to check the main page for selection of the function

5.3 Code Details and Code Efficiency:

Coding:-

Coding, in simpler terms, is the language used by computers to understand our commands and, therefore, process our requests. Programming is a list of codes arranged in a sequence that results in the completion of work. Take, for example, the following analogy - you click on a video app on your smartphones, and it plays a video.A program is what brings about the completion of the task 'playing the said video.'

The program is made up of a series of smaller tasks that direct your smartphone to do the above task and bring it to completion. Each smaller task is written in code, i.e., the computer language, and that is what coding is all about.

How does coding work?

Computers and artificial intelligence are built up of, mainly, transistors; and these transistors act as the 'brain' of the computer. Hence, the computer only understands the language of 'on' and 'off,' guided by the transistor switches. The on and off are represented by 1 and 0, respectively, in a binary system. Therefore, your computer and every other gadget run on an infinite sequence of binary codes

These binary codes form the machine code, with each number directing the machine (your computer) to change a sequence in its memory.

Programming languages make the binary code language of the computers more manageable by translating our commands into binary code.

Coding means using the programming language to get the computer to behave as desired.

Coding vs Programming: Head to Head Comparison

An essential distinction between coding and programming is that programming is the higher level of coding that assembles a set of instructions (codes) to allow your computer to carry out the task.

Definition	Coding- Writing codes to translate one language to another.	Programming- The process of assembling a set of instructions in machine language that your machine can execute.
Aim	Coding is aimed at enabling communication between humans and computers.	Programming is built for translating human thoughts into machine commands that can be followed by your computer to bring about a function.
Skill-set required	Basic	Complex
Procedure	Coding involves writing a certain line of code to send out a message to the computer.	Programming analyses and creates different sets of commands or instructions that help the machine to understand the steps involved and produce an output.
Summary	Coding is converting human language into the binary language of computers.	Programming is using the codes to create a set of instructions that helps the computer complete a task.

What are the advantages of coding?

1. Understanding the world around you

Coding helps you understand the ABC of how the technology works- it helps one become apter in using technology around them.

2. Problem-solving

Programming is essentially about creating solutions to the problems faced while transmitting data. Hence, coding helps you become more apt and creative in problem-solving.

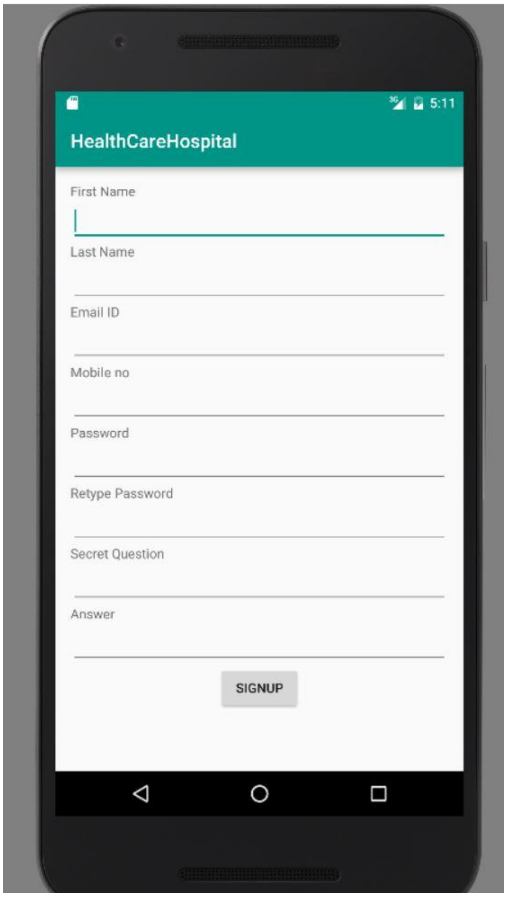
3. A broad range of job opportunities

With the increasing demand for technology, software engineering and programming are the fastest-growing job opportunities all over the world.

4. Be the master of your technology

Be it in the personal or professional front, knowing coding and programming is an empowering skill as problems can be tackled without asking for help from webmasters and IT employees.

Code for Signup:-




```
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.DialogInterface;

import android.os.AsyncTask;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.text.Html;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.view.ViewParent;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.*;
import javax.mail.*;
import javax.mail.internet.*;
```

```
public class Signup extends AppCompatActivity {

    String fname, lname, emailid, mobile, password, confirmpassword, question, answer;
    EditText
txt_fname,txt_lname,txt_emailid,txt_mobile,txt_password,txt_confirmpassword,
txt_question, txt_answer;
    ProgressDialog pd;
    int i;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_signup);


        txt_fname = (EditText) findViewById(R.id.txt_fname);
        txt_lname = (EditText) findViewById(R.id.txt_lname);
        txt_emailid = (EditText) findViewById(R.id.txt_emailid);
        txt_mobile = (EditText) findViewById(R.id.txt_mobile);
        txt_password = (EditText) findViewById(R.id.txt_password);
        txt_confirmpassword = (EditText) findViewById(R.id.txt_confirmpassword);
        txt_question = (EditText) findViewById(R.id.txt_question);
        txt_answer = (EditText) findViewById(R.id.txt_answer);

        txt_confirmpassword.setOnFocusChangeListener(new
View.OnFocusChangeListener() {

            public void onFocusChange(View v, boolean hasFocus) {
                if(!hasFocus)
                {

if(!txt_confirmpassword.getText().toString().equals(txt_password.getText().toString()))
                    {
                        txt_confirmpassword.setError("Password and Confirm password do not
match");
                        txt_confirmpassword.setText("");
                    }
                }
            }
        });

        Button b1 = (Button) findViewById(R.id.btn_register);

        b1.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {

                fname = txt_fname.getText().toString();
                lname = txt_lname.getText().toString();
                emailid = txt_emailid.getText().toString();
                mobile = txt_mobile.getText().toString();
                password = txt_password.getText().toString();
```

```
            confirmpassword = txt_confirmpassword.getText().toString();
            question = txt_question.getText().toString();
            answer = txt_answer.getText().toString();


            if (fname.equals("") == false) {
                if (lname.equals("") == false) {
                    if (emailid.equals("") == false &&
android.util.Patterns.EMAIL_ADDRESS.matcher(emailid).matches()) {
                        if (mobile.equals("") == false) {
                            if (password.equals("") == false) {
                                if (confirmpassword.equals("") == false) {
                                    if (question.equals("") == false) {
                                        if (answer.equals("") == false) {

                                            DB_Conn obj = new DB_Conn();
                                            obj.execute(fname, lname, emailid, mobile, password,
question, answer);
                                            pd = new ProgressDialog(Signup.this);
                                            pd.setMessage("Sending OTP on email..Please wait..."); //
Setting Message
                                            pd.setTitle("Registration"); // Setting Title
                                            pd.setProgressStyle(ProgressDialog.STYLE_SPINNER); //
Progress Dialog Style Spinner
                                            pd.show(); // Display Progress Dialog
                                            pd.setCancelable(false);
                                        } else {
                                            txt_answer.setError("Please enter password");
                                        }
                                    } else {
                                        txt_question.setError("Please enter password");
                                    }
                                } else {
                                    txt_confirmpassword.setError("Please enter password");
                                }
                            } else {
                                txt_password.setError("Please enter password");
                            }
                        } else {
                            txt_mobile.setError("Please enter mobile no");
                        }
                    } else {
                        txt_emailid.setError("Either emailid is blank or Invalid Format");
                    }
                } else {
                    txt_lname.setError("Please enter last name");
                }
            } else {
                txt_fname.setError("Please enter first name");
            }

        }
```

```
        });
    }

        class DB_Conn extends AsyncTask<String, Void, String> {


            String otp = "",otpval="1111";
            String ar[] = new String[7];
            AlertDialog alertDialog;

            @Override
            public String doInBackground(String... arg) //compulsory to implement
            {
                String result = "";
                ar[0] = arg[0];
                ar[1] = arg[1];
                ar[2] = arg[2];
                ar[3] = arg[3];
                ar[4] = arg[4];
                ar[5] = arg[5];
                ar[6] = arg[6];
                try {
                    Connection con = DB_Connection.get_DBConnection();
                    PreparedStatement pst = con.prepareStatement("select * from patient where
emailid=?");
                    pst.setString(1, arg[2]);
                    ResultSet rs = pst.executeQuery();

                    if (rs.next() == false) {

                        Random r = new Random();
                        otp = r.nextInt(9) + "" + r.nextInt(9) + "" + r.nextInt(9) + "" + r.nextInt(9);
                        Properties p=new Properties();
                        p.put("mail.smtp.starttls.enable","true");//here smtp donot get start security
gets started
                        p.put("mail.smtp.auth","true");
                        p.put("mail.smtp.host","smtp.gmail.com");
                        p.put("mail.smtp.port","587");

                        Session s= Session.getDefaultInstance(p,new Authenticator()
                        {
                            protected PasswordAuthentication getPasswordAuthentication()
                            {
                                return new
PasswordAuthentication(DB_Connection.SENDERS_EMAILID,DB_Connection.SENDER
S_PASSWORD);
                            }
                        });
```

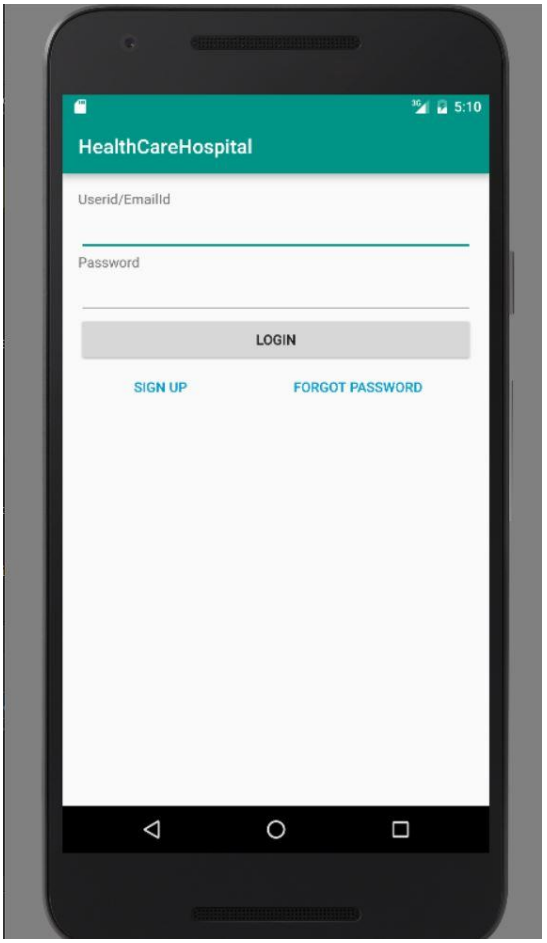
```
                MimeMessage msg=new MimeMessage(s);//multipurpose internet mail
extension mime
                msg.setFrom(new InternetAddress(DB_Connection.SENDERS_EMAILID));
                msg.addRecipient(Message.RecipientType.TO,new
InternetAddress(emailid));//here type recipient email id
                msg.setSubject("OTP for registration");
                String m="Greeting,\n Your OTP for account activation is "+otp;
                //msg.setText(m,"UTF-8","html");
                msg.setContent(m, "text/html; charset=utf-8");
                Transport.send(msg);


                result = "success";
             } else {
                result = "failed";
             }

          } catch (Exception e) {
             e.printStackTrace();
          }
          return result;
       }

       @Override
       public void onProgressUpdate(Void... arg0) //optional
       {
          /*AlertDialog.Builder alert = new AlertDialog.Builder(Signup.this);

          LayoutInflater li = LayoutInflater.from(Signup.this);
          final View promptsView = li.inflate(R.layout.otp_alert,null);
          alert.setView(promptsView);
          alert.setPositiveButton("Validate", new DialogInterface.OnClickListener() {
             public void onClick(DialogInterface obj, int x) {
                otpval = ((EditText)
promptsView.findViewById(R.id.txt_otp)).getText().toString();

             }
          });

          AlertDialog alertDialog = alert.create();
          alertDialog.show();*/

       }

       @Override
       public void onPostExecute(String result) //optional
       {

          pd.dismiss();
          final AlertDialog.Builder alert = new AlertDialog.Builder(Signup.this);

          if (result.equals("success")) {
```

```
                i=1;
                alert.setTitle("OTP Validation");
                alert.setMessage(Html.fromHtml("An OTP has been sent to your emailid.<br/>
Please check and enter the same.<font color=red> (max 3 attempts)</font>"));
                LayoutInflater li = LayoutInflater.from(Signup.this);
                final View promptsView = li.inflate(R.layout.otp_alert,null);
                alert.setView(promptsView);
                alert.setPositiveButton("Validate", new DialogInterface.OnClickListener() {
                   public void onClick(DialogInterface obj, int x) {
                      String
otpval=((EditText)promptsView.findViewById(R.id.txt_otp)).getText().toString();
                      if(otpval.equals(otp)) {
                         Runnable runnable = new Runnable() {
                            @Override
                            public void run() {
                               insert();


                            }
                         };
                         Thread t=new Thread(runnable);
                         t.start();
                         Toast.makeText(Signup.this,"Registration
successful.",Toast.LENGTH_LONG).show();




                      }
                      else {

                         if(i==3)
                         {
                            Toast.makeText(Signup.this, "Registration failed.",
Toast.LENGTH_LONG).show();
                            txt_fname.setText("");
                            txt_lname.setText("");
                            txt_emailid.setText("");
                            txt_mobile.setText("");
                            txt_password.setText("");
                            txt_confirmpassword.setText("");
                            txt_question.setText("");
                            txt_answer.setText("");


                         }
                         else {
                            i++;
                            alert.setMessage(Html.fromHtml("Incorrect OTP
entered.<br/>Please try again. <font color=red>("+(4-i)+" attempts left)</font>"));
                            ((EditText) promptsView.findViewById(R.id.txt_otp)).setText("");
                            ((ViewGroup)
```

```
promptsView.getParent()).removeView(promptsView);

                        alert.show();
                    }
                }
            }
        });


    } else if (result.equals("failed")) {
        alert.setTitle("Error");
        alert.setMessage("This emailid is already registered. Try a different one");
        alert.setPositiveButton("OK", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface obj, int x) {

            }
        });
    }

    alertDialog = alert.create();
    alertDialog.show();


}

@Override
public void onPreExecute() //optional
{
    // do something before start
}

public void insert()
{
    try {

        Connection con = DB_Connection.get_DBConnection();

        PreparedStatement pst = con.prepareStatement("insert into patient
values(?,?,?,?,?,?,?)");
        pst.setString(1, ar[0]);
        pst.setString(2, ar[1]);
        pst.setString(3, ar[2]);
        pst.setString(4, ar[3]);
        pst.setString(5, ar[4]);
        pst.setString(6, ar[5]);
        pst.setString(7, ar[6]);
        pst.executeUpdate();
        con.close();
    }
    catch(Exception e)
    {
        e.printStackTrace();
```

```
            }
        }

    }

}
```

Code for login:-



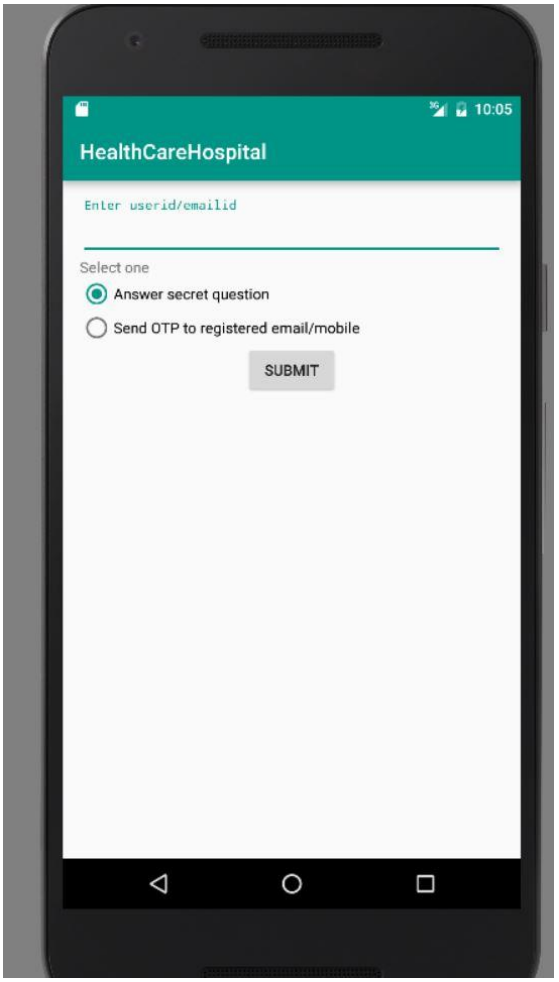

```
import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.*;
import java.sql.*;
```

```
public class Login extends AppCompatActivity implements View.OnClickListener {

    SharedPreferences sp;
    String u,p1;

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

        Button btn_login=(Button)findViewById(R.id.button);
        btn_login.setOnClickListener(this);
    }
    public void onClick(View v)
    {
        EditText txt_userid = (EditText) findViewById(R.id.editText);
        EditText txt_password = (EditText) findViewById(R.id.editText2);

         u = txt_userid.getText().toString();
         p1 = txt_password.getText().toString();

        if(v.getId()==R.id.button)
        {
            if(u.equals("")==false ) {
                if (android.util.Patterns.EMAIL_ADDRESS.matcher(u).matches()) {
                    if (p1.equals("") == false) {


                        DB_Conn obj = new DB_Conn();
                        obj.execute(u, p1);

                    } else {
                        txt_password.setError("Please enter password");
                    }
                } else {
                    txt_userid.setError("Invalid Emailid Format");
                }
            }
            else {
                txt_userid.setError("Please enter userid");
            }
        }


    }
    class DB_Conn extends AsyncTask<String,Void,String>
    {


        @Override
        public String doInBackground(String...arg) //compulsory to implement
        {
```

```
        String r="";
        try {

            Connection con=DB_Connection.get_DBConnection();

            PreparedStatement pst = con.prepareStatement("select * from login where emailid=? and
password=?");
            pst.setString(1, arg[0]);
            pst.setString(2, arg[1]);
            ResultSet rs = pst.executeQuery();
            if (rs.next() == true) {

                  r = "success";

            }
            else
            {
               r="failure";
            }
            con.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
        return r;
    }
    @Override
    public void onProgressUpdate(Void...arg0) //optional
    {

    }
    @Override
    public void onPostExecute(String result) //optional
    {
       //  do something after execution
       if(result.equals("success"))
       {
          sp = getSharedPreferences("pf", Context.MODE_PRIVATE);
          SharedPreferences.Editor ed = sp.edit();
          ed.putString("userid", u);
          ed.putBoolean("loggedin", true);

          ed.commit();

          Intent i=new Intent(Login.this,MainActivity_Admin.class);
          startActivity(i);
          Login.this.finish();
       }

       else
       {
          AlertDialog.Builder alert = new AlertDialog.Builder(Login.this);
          alert.setTitle("Error");
```

```
            alert.setMessage("Invalid credentials");
            alert.setPositiveButton("OK", new DialogInterface.OnClickListener() {
               public void onClick(DialogInterface obj, int x) {


               }
            });
            AlertDialog alertDialog = alert.create();
            alertDialog.show();
         }
      }


      @Override
      public void onPreExecute() //optional
      {
         // do something before start
      }

   }
}
```

Code for forgotpassword:-



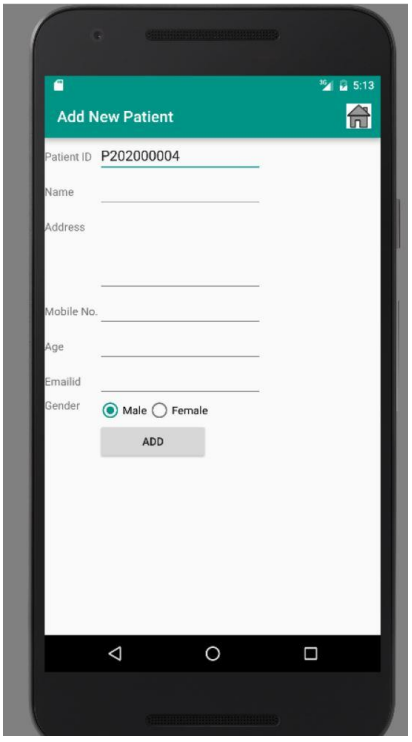
```
  import android.app.AlertDialog;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentTransaction;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.telephony.SmsManager;
import android.text.Layout;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.EditText;
import android.widget.FrameLayout;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.Properties;
import java.util.Random;
import javax.mail.Authenticator;
import javax.mail.Message;
import javax.mail.PasswordAuthentication;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;

public class ForgotPassword extends AppCompatActivity implements View.OnClickListener {

    public String u,r;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_forgot_password);
    }

    public void onClick(View v) {
        EditText txt_userid = (EditText) findViewById(R.id.editText);
        RadioButton radio_selected = (RadioButton) findViewById(((RadioGroup)
findViewById(R.id.rg)).getCheckedRadioButtonId());


        u = txt_userid.getText().toString();
        r = radio_selected.getText().toString();

        if (u.equals("") == false && android.util.Patterns.EMAIL_ADDRESS.matcher(u).matches()) {

            DB_Conn obj = new DB_Conn();
            obj.execute(u, r);
```

```
        } else {
            txt_userid.setError("No value or Invalid Emailid Format entered");

        }

    }

    class DB_Conn extends AsyncTask<String, Void, String> {

        String ar[] = new String[7];
        String r = "", otp;

        @Override
        public String doInBackground(String... arg) //compulsory to implement
        {

            ar[0] = arg[0];
            ar[1] = arg[1];

            try {

                Connection con = DB_Connection.get_DBConnection();

                PreparedStatement pst = con.prepareStatement("select * from login where userid=?");
                pst.setString(1, arg[0]);

                ResultSet rs = pst.executeQuery();
                if (rs.next() == true) {
                    if(ar[1].equalsIgnoreCase("OTP on email")) {

                        Random r1 = new Random();
                        otp = r1.nextInt(9) + "" + r1.nextInt(9) + "" + r1.nextInt(9) + "" + r1.nextInt(9);
                        Properties p = new Properties();
                        p.put("mail.smtp.starttls.enable", "true");//here smtp donot get start security gets
started
                        p.put("mail.smtp.auth", "true");
                        p.put("mail.smtp.host", "smtp.gmail.com");
                        p.put("mail.smtp.port", "587");

                        Session s = Session.getDefaultInstance(p, new Authenticator() {
                            protected PasswordAuthentication getPasswordAuthentication() {
                                return new PasswordAuthentication(DB_Connection.SENDERS_EMAILID,
DB_Connection.SENDERS_PASSWORD);
                            }
                        });


                        MimeMessage msg = new MimeMessage(s);//multipurpose internet mail extension
mime
                        msg.setFrom(new InternetAddress(DB_Connection.SENDERS_EMAILID));
                        msg.addRecipient(Message.RecipientType.TO, new InternetAddress(ar[0]));//here
type recipient email id
```

```
                msg.setSubject("OTP for registration");
                String m = "Greeting,\n Your OTP for account activation is " + otp;
                //msg.setText(m,"UTF-8","html");
                msg.setContent(m, "text/html; charset=utf-8");
                Transport.send(msg);
                r = "success";
            }
            else {

                SmsManager smsManager = SmsManager.getDefault();
                smsManager.sendTextMessage(rs.getString("mobile"), null, "sms message", null,
null);
                r = "success";
            }

        } else {
            r = "failure";
        }
        con.close();
        return r;


    } catch (Exception e) {
        e.printStackTrace();
    }
    return r;
}

@Override
public void onProgressUpdate(Void... arg0) //optional
{

}

@Override
public void onPostExecute(String result) //optional
{
    //  do something after execution

    if (result.equals("success")) {

        final AlertDialog.Builder alert = new AlertDialog.Builder(ForgotPassword.this);
        alert.setTitle("OTP Validation");
        alert.setMessage("An OTP has been sent to your emailid/mobile. Please check and enter
the same");
        LayoutInflater li = LayoutInflater.from(ForgotPassword.this);
        final View promptsView = li.inflate(R.layout.otp_alert,null);
        alert.setView(promptsView);
        alert.setPositiveButton("Validate", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface obj, int x) {
                String otpval = ((EditText)
promptsView.findViewById(R.id.txt_otp)).getText().toString();
```

```
                    if (otpval.equals(otp)) {
                        FragmentManager fragmentManager = getSupportFragmentManager();
                        FragmentTransaction fragmentTransaction =
fragmentManager.beginTransaction();


                        fragmentTransaction.replace(R.id.fm, new ForgotPassword1());

                    } else {

                        //((TextView) promptsView.findViewById(R.id.tv_msg)).setText("Incorrect OTP
value. Please try again.");
                        alert.setMessage("Incorrect OTP entered.Please try again.");
                        ((EditText) promptsView.findViewById(R.id.txt_otp)).setText("");
                        ((ViewGroup) promptsView.getParent()).removeView(promptsView);

                        alert.show();
                    }

                }
            });
            AlertDialog alertDialog = alert.create();
            alertDialog.show();

        } else {
            AlertDialog.Builder alert = new AlertDialog.Builder(ForgotPassword.this);
            alert.setTitle("Error");
            alert.setMessage("UserId does not exist");
            alert.setPositiveButton("OK", new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface obj, int x) {


                }
            });
            AlertDialog alertDialog = alert.create();
            alertDialog.show();
        }
    }


    @Override
    public void onPreExecute() //optional
    {
        // do something before start
    }
  }
}
```

Code for AddPatient:-




```
import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.Calendar;

public class AddPatient extends AppCompatActivity implements View.OnClickListener {

    EditText txt_pid,txt_pname,txt_emailid,txt_address,txt_mobile,txt_age;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_add_patient);
        DB_Conn1 obj1 = new DB_Conn1();
        obj1.execute();
```

```
        Button b1=(Button)findViewById(R.id.button4);
        b1.setOnClickListener(this);
    }


    public void onClick(View v)
    {
        txt_pid = (EditText) findViewById(R.id.txt_pid);
        txt_pname = (EditText) findViewById(R.id.txt_pname);
        txt_address = (EditText) findViewById(R.id.txt_address);
        txt_mobile = (EditText) findViewById(R.id.txt_mobile);
        txt_emailid = (EditText) findViewById(R.id.txt_emailid);
        txt_age = (EditText) findViewById(R.id.txt_age);

        String pid=txt_pid.getText().toString();
        String pname=txt_pname.getText().toString();
        String address=txt_address.getText().toString();
        String mobile=txt_mobile.getText().toString();
        String age=txt_age.getText().toString();
        String emailid=txt_emailid.getText().toString();

        if (pname.equals("") == false) {
            if (address.equals("") == false) {
                if (mobile.equals("") == false) {
                    if (age.equals("") == false) {
                        if (emailid.equals("") == false &&
android.util.Patterns.EMAIL_ADDRESS.matcher(emailid).matches()) {

                                String
gender=((RadioButton)findViewById(((RadioGroup)findViewById(R.id.rg)).getCheckedRadioBu
ttonId())).getText().toString();
                                DB_Conn obj = new DB_Conn();
                                obj.execute(pid, pname, address, mobile,age,emailid,gender);
                            }
                        else {
                                txt_emailid.setError("Either emailid is blank or Invalid Format");
                        }
                    } else {
                            txt_age.setError("Value is required");
                    }
                } else {
                            txt_mobile.setError("Please enter mobile no");
                }
            } else {
                txt_address.setError("Please enter patient address");
            }
        } else {
            txt_pname.setError("Please enter patient name");
        }


    }
```

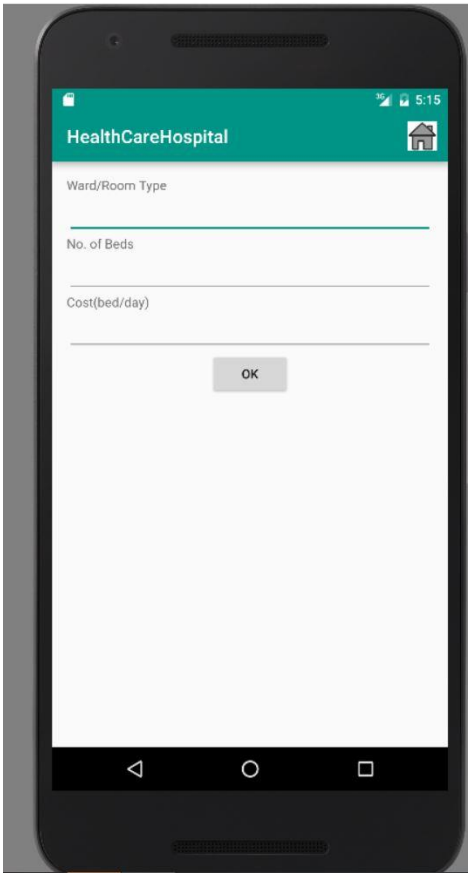
```
class DB_Conn extends AsyncTask<String,Void,String>
{

   String ar[] = new String[7];
   @Override
   public String doInBackground(String...arg) //compulsory to implement
   {
      String r="";
      ar[0] = arg[0];
      ar[1] = arg[1];
      ar[2] = arg[2];
      ar[3] = arg[3];
      ar[4] = arg[4];
      ar[5] = arg[5];
      ar[6] = arg[6];
      try {

         Connection con=DB_Connection.get_DBConnection();

         PreparedStatement pst = con.prepareStatement("insert into patient values(?,?,?,?,?,?,?)");
         pst.setString(1, arg[0]);
         pst.setString(2, arg[1]);
         pst.setString(3, arg[2]);
         pst.setString(4, arg[3]);
         pst.setString(5, arg[4]);
         pst.setString(6, arg[5]);
         pst.setString(7, arg[6]);
         pst.execute();
         con.close();
         r = "success";



      } catch (Exception e) {
         e.printStackTrace();
      }
      return r;
   }
   @Override
   public void onProgressUpdate(Void...arg0) //optional
   {

   }
   @Override
   public void onPostExecute(String result) //optional
   {
      //  do something after execution
      if(result.equals("success"))
      {
         /*sp = getSharedPreferences("pf", Context.MODE_PRIVATE);
         SharedPreferences.Editor ed = sp.edit();
```

```
            ed.putString("userid", u);
            ed.putBoolean("loggedin", true);

            ed.commit();*/
            AlertDialog.Builder alert = new AlertDialog.Builder(AddPatient.this);
            alert.setTitle("Success");
            alert.setMessage("Patient added successfully");
            alert.setPositiveButton("OK", new DialogInterface.OnClickListener() {
               public void onClick(DialogInterface obj, int x) {
                  txt_pid.setText("");
                  txt_pname.setText("");
                  txt_address.setText("");
                  txt_mobile.setText("");
                  txt_emailid.setText("");
                  txt_age.setText("");
                  DB_Conn1 obj1 = new DB_Conn1();
                  obj1.execute();

               }
            });
            AlertDialog alertDialog = alert.create();
            alertDialog.show();

         }

         else
         {
            AlertDialog.Builder alert = new AlertDialog.Builder(AddPatient.this);
            alert.setTitle("Error");
            alert.setMessage("Error occured");
            alert.setPositiveButton("OK", new DialogInterface.OnClickListener() {
               public void onClick(DialogInterface obj, int x) {



               }
            });
            AlertDialog alertDialog = alert.create();
            alertDialog.show();
         }
      }


      @Override
      public void onPreExecute() //optional
      {
         // do something before start
      }

   }

   class DB_Conn1 extends AsyncTask<Void,Void,String>
```

```
{
   String pid="";

   @Override
   public String doInBackground(Void...arg) //compulsory to implement
   {

      try {

         Connection con=DB_Connection.get_DBConnection();

         Calendar cl=Calendar.getInstance();

         PreparedStatement pst=con.prepareStatement("select max(pid) from patient");
         ResultSet rs=pst.executeQuery();
         rs.next();
         String v=rs.getString(1);
         if(v!=null)
         {
            if(String.valueOf(cl.get(Calendar.YEAR)).equals(v.substring(1,5)))
            {
               int v1=Integer.parseInt(v.substring(v.length()-5));
               v1=v1+1;
               pid="P"+cl.get(Calendar.YEAR)+String.format("%05d",v1);
            }
            else
            {
               pid="P"+cl.get(Calendar.YEAR)+"00001";
            }
         }
         else
         {

            pid="P"+cl.get(Calendar.YEAR)+"00001";
         }
         pst.close();



      } catch (Exception e) {
         e.printStackTrace();
      }
      return null;

   }
   @Override
   public void onProgressUpdate(Void...arg0) //optional
   {

   }
   @Override
   public void onPostExecute(String result) //optional
```

```
        {
           //  do something after execution
           ((EditText)findViewById(R.id.txt_pid)).setText(pid);
        }


        @Override
        public void onPreExecute() //optional
        {
           // do something before start
        }

    }

}
```

Code for AddWard:-




```
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.RadioButton;
import android.widget.RadioGroup;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

public class AddWard extends AppCompatActivity implements View.OnClickListener {

    EditText txt_wardtype,txt_beds,txt_cost;
    String wardtype,beds,cost,r1,r2="";

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

```
        Button b1=(Button)findViewById(R.id.button7);
        b1.setOnClickListener(this);

        txt_wardtype = (EditText) findViewById(R.id.txt_wardtype);
        txt_beds = (EditText) findViewById(R.id.txt_beds);
        txt_cost = (EditText) findViewById(R.id.txt_cost);
    }

    public void onClick(View v) {

        wardtype=txt_wardtype.getText().toString();
        beds=txt_beds.getText().toString();
        cost=txt_cost.getText().toString();

        if (wardtype.equals("") == false) {
            if (beds.equals("") == false && Integer.parseInt(beds)>0) {
                if (cost.equals("") == false && Integer.parseInt(cost)>0) {



                        DB_Conn obj = new DB_Conn();
                        obj.execute("not verified");

                } else {
                    txt_cost.setError("No value or Invalid value");
                }
            } else {
                txt_beds.setError("No value or Invalid value");
            }
        } else {
            txt_wardtype.setError("Please enter ward type");
        }

    }

    class DB_Conn extends AsyncTask<String,Void,String>
    {


        @Override
        public String doInBackground(String...arg) //compulsory to implement
        {

            try {
                int bedno;
                Connection con=DB_Connection.get_DBConnection();

                PreparedStatement pst=con.prepareStatement("select max(bedno) from beds where
wardtype=?");
                pst.setString(1, wardtype);
                ResultSet rs=pst.executeQuery();
```

```
        rs.next();
        String v=rs.getString(1);
        if(v!=null)
        {

           bedno=Integer.parseInt(v)+1;
           r1="exists";
        }
        else
        {

           bedno=1;
           r1="new";
        }
        pst.close();

        if(arg[0].equals("verified")) {
           pst = con.prepareStatement("insert into beds values(?,?,?,?)");

           for (int i = 1; i <=Integer.parseInt(beds); i++) {
              pst.setString(1, wardtype);
              pst.setInt(2, bedno);
              pst.setInt(3, Integer.parseInt(cost));
              pst.setString(4, "vacant");
              pst.execute();
              bedno++;
           }
           r1 = "success";
        }


        con.close();




     } catch (Exception e) {
        e.printStackTrace();
     }
     return r1;
  }
  @Override
  public void onProgressUpdate(Void...arg0) //optional
  {

  }
  @Override
  public void onPostExecute(String result) //optional
  {
     //  do something after execution
     if(result.equals("success"))
     {
```

```
            /*sp = getSharedPreferences("pf", Context.MODE_PRIVATE);
            SharedPreferences.Editor ed = sp.edit();
            ed.putString("userid", u);
            ed.putBoolean("loggedin", true);

            ed.commit();*/
            AlertDialog.Builder alert = new AlertDialog.Builder(AddWard.this);
            alert.setTitle("Success");
            alert.setMessage("Record added successfully");
            alert.setPositiveButton("OK", new DialogInterface.OnClickListener() {
               public void onClick(DialogInterface obj, int x) {
                  txt_wardtype.setText("");
                  txt_beds.setText("");
                  txt_cost.setText("");


               }
            });
            AlertDialog alertDialog = alert.create();
            alertDialog.show();

         }
         else if(result.equals("new"))
         {
            /*sp = getSharedPreferences("pf", Context.MODE_PRIVATE);
            SharedPreferences.Editor ed = sp.edit();
            ed.putString("userid", u);
            ed.putBoolean("loggedin", true);

            ed.commit();*/
            AlertDialog.Builder alert = new AlertDialog.Builder(AddWard.this);
            alert.setTitle("Success");
            alert.setMessage("You are trying to add a new ward. You are sure you want to
continue?");
            alert.setPositiveButton("YES", new DialogInterface.OnClickListener() {
               public void onClick(DialogInterface obj, int x) {
                  r2 = "verified";
                  DB_Conn obj1 = new DB_Conn();
                  obj1.execute("verified");

               }
            });
            alert.setNegativeButton("NO", new DialogInterface.OnClickListener() {
               public void onClick(DialogInterface obj, int x) {

                  r2 = "";

               }

            });
            AlertDialog alertDialog = alert.create();
            alertDialog.show();
```

```
            }
            else if(result.equals("exists"))
            {
               /*sp = getSharedPreferences("pf", Context.MODE_PRIVATE);
               SharedPreferences.Editor ed = sp.edit();
               ed.putString("userid", u);
               ed.putBoolean("loggedin", true);

               ed.commit();*/
               AlertDialog.Builder alert = new AlertDialog.Builder(AddWard.this);
               alert.setTitle("Success");
               alert.setMessage("This ward/room already exists. Are you sure you want to increase
beds?");
               alert.setPositiveButton("YES", new DialogInterface.OnClickListener() {
                  public void onClick(DialogInterface obj, int x) {
                     r2 = "verified";
                     DB_Conn obj1 = new DB_Conn();
                     obj1.execute("verified");

                  }

               });
               alert.setNegativeButton("NO", new DialogInterface.OnClickListener() {
                  public void onClick(DialogInterface obj, int x) {

                     r2 = "";

                  }

               });
               AlertDialog alertDialog = alert.create();
               alertDialog.show();

            }

         }


         @Override
         public void onPreExecute() //optional
         {
            // do something before start
         }

      }

   }
```

DB_Connection.java

```
import java.sql.*;

public class DB_Connection
{
   public static String SENDERS_EMAILID="mercynadar95@gmail.com";
   public static String SENDERS_PASSWORD="gootojesus";
   public static Connection get_DBConnection()
   {
      Connection conn=null;
      try
      {
         Class.forName("com.mysql.jdbc.Driver");
         conn = DriverManager.getConnection("jdbc:mysql://10.0.2.2:3306/hospital_db", "root",
"abc");

      }
      catch(Exception e)
      {
         e.printStackTrace();
      }
      return conn;
   }

}
```

5.4 Future scope of the application:

Some of the future enhancements that can be done to this system are:

- As the technology emerges, it is possible to upgrade the system and can be adaptable to desired environment.
- Because it is based on object-oriented design, any further changes can be easily adaptable.
- Based on the future security issues, security and be improved using emerging technologies.
- The next Version of the Software Will be More Updated Version.
- It will be more user-friendly.
- It will provide user with the more high performance and more stability than this.

5.5 Utilities:

- Notepad: To notedown important things
- Word: To save word format.
- Backup: To take backup of database for particular day.
- Restore: To restore backup again into database.
- Environment: To change look and feel of system.

5.6 Testing Phase:

Test Case Design:

Software Testing Techniques help you design better test cases. Since exhaustive testing is not possible; Manual Testing Techniques help reduce the number of test cases to be executed while increasing test coverage. They help identify test conditions that are otherwise difficult to recognize.

A TEST CASE is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly.

The process of developing test cases can also help find problems in the requirements or design of an application.

A test plan is neither not related to the details of testing units nor it specifies the test cases to be used for testing units. Thus, test case specification is done in order to test each unit separately. Depending on the testing method specified in a test plan, the features of the unit to be tested are determined. The overall approach stated in the test plan is refined into two parts: specific test methods and the evaluation criteria. Based on these test methods and the criteria, the test cases to test the unit are specified.

For each unit being tested, these test case specifications describe the test cases, required inputs for test cases, test conditions, and the expected outputs from the test cases. Generally, it is required to specify the test cases before using them for testing. This is because the effectiveness of testing depends to a great extent on the nature of test cases.

Test case specifications are written in the form of a document. This is because the quality of test cases is evaluated by performing a test case review, which requires a formal document. The review of test case document ensures that test cases satisfy the chosen criteria and conform to the policy specified in the test plan. Another benefit of specifying test cases in a formal document is that it helps testers to select an effective set of test cases.

Types of testing used:

Unit Testing:

Unit testing is designed to ensure that the purpose for which it was designed is fulfilled . Each and every module was tested individuall y with the test data and error messages were displayed for incorrect and sufficient for entry works. All validations was tested to correctness. Test data were fed in and results was checked for the maintenance module , to ensure that all tables created contained nothing but valid data.

Integration Testing

In integration testing a system consisting of different modules is tested for problems arising from component interaction. Integration testing should be developed from the system specification . Firstl y, a minimum configuration must be integrated and tested. In m y project I have done integration testing in a bottom up fashion i .e . in this project I have started construction and testing with atomic modules . After unit testing the modules are integrated one by one and then tested the syste m for problems arising from component interaction.

White Box Testing:

In white box testing knowing the internal working of the base, test can be conducted to ensure that internal operations are performed according to specification and all internal componen ts have been adequately exercised. In white box testing logical path through the software are tested by providing test cases that exercise specific set of conditions and loops. Using white -box testing software developer can derive test case that Guarantee that all independent paths within a module have been exercised at least once.

1. Exercise all logical decisions on their true and false side .
2. Exercise all loops at their boundaries and within their operational bound .
3. Exercise internal data structure to ensure their validity.

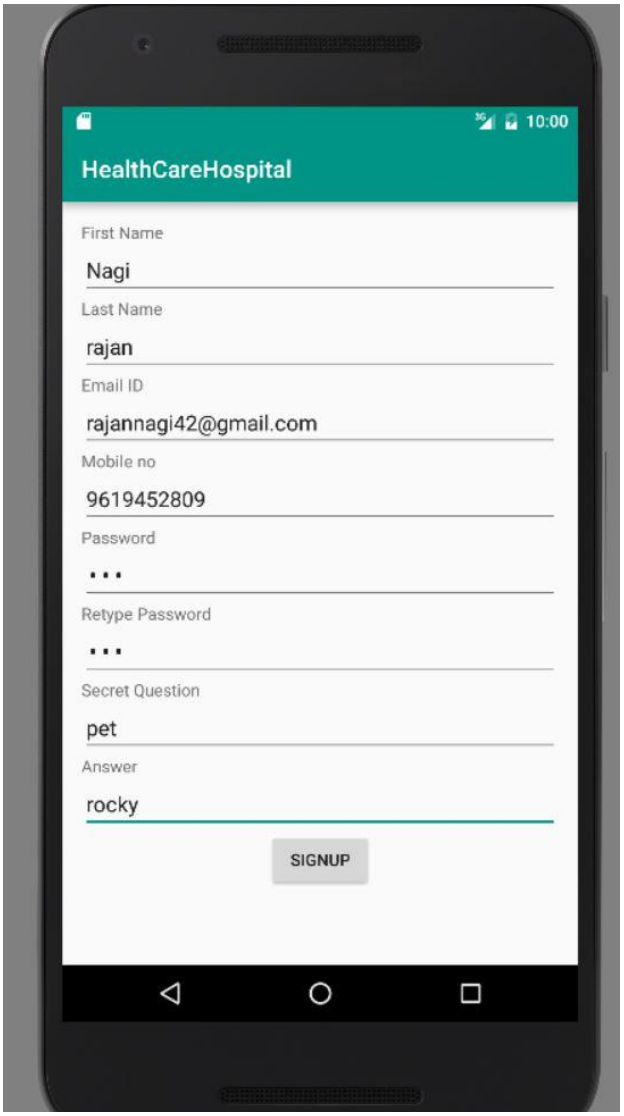
Chapter 6

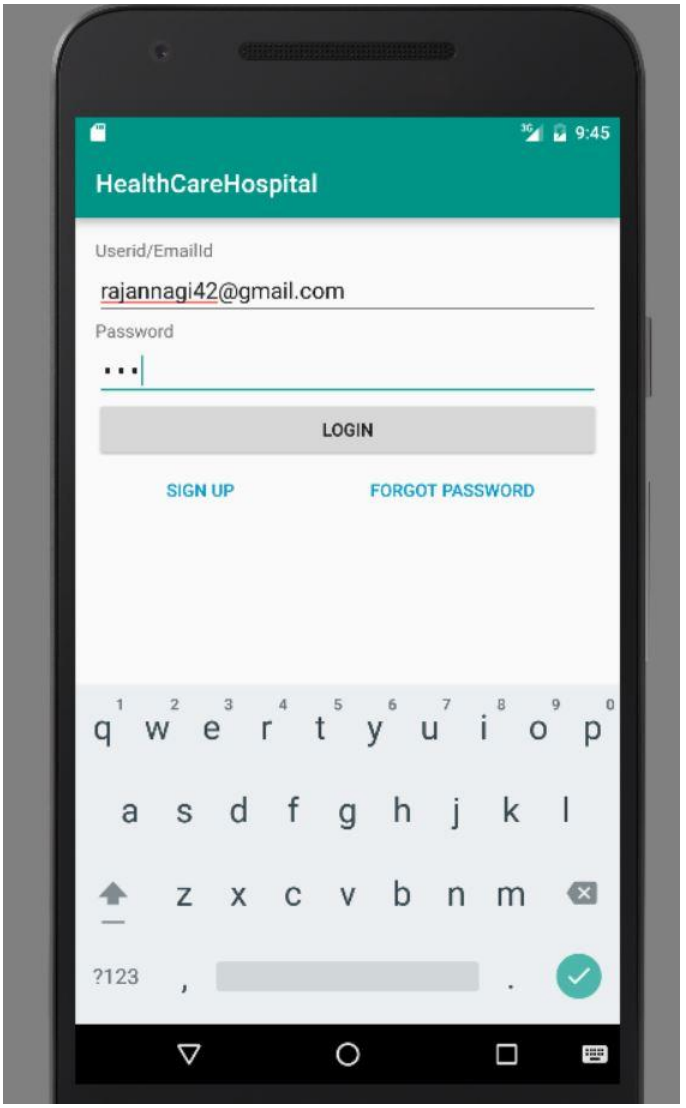
RESULT AND DECISION

6.1 Test Conditions and Test Cases:

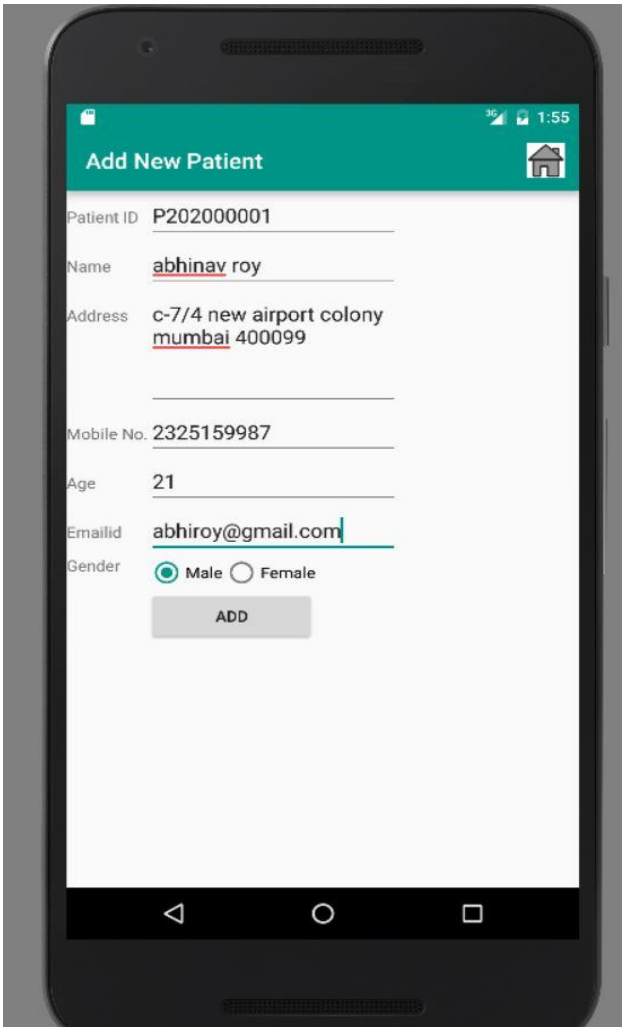
Test case no.	Test for	Input Values	Expected Result	Actual Result	Remark
1	Signup page	Blank	Unsuccessful	Unsuccessful	Pass
2	Signup page	xyz@gmailcom	Unsuccessful	successful	Fail
3	Login page	Blank	Unsuccessful	Unsuccessful	Pass
4	Login page	xyz%#gamil.com	Unsuccessful	Unsuccessful	Pass
5	Payment page	Blank	Unsuccessful	Successful	Fail
6	Examine patient	Blank	Unsuccessful	Unsuccessful	Pass
7	Add patient	Blank	Unsuccessful	Unsuccessful	Pass
8	Forget Password	Blank	Unsuccessful	Successful	Fail
9	Forget Password	xyz@gmail.com	Successful	Successful	Pass
10	Reset Password	Blank	Successful	Successful	Pass

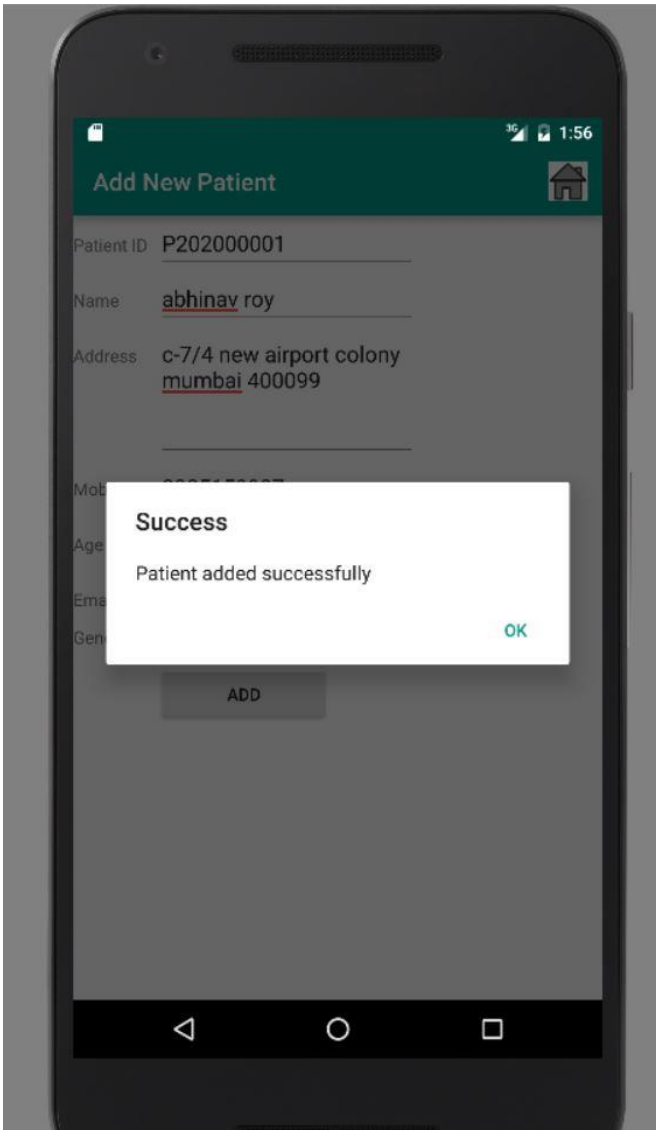
6.2 User Test Screen shots:-

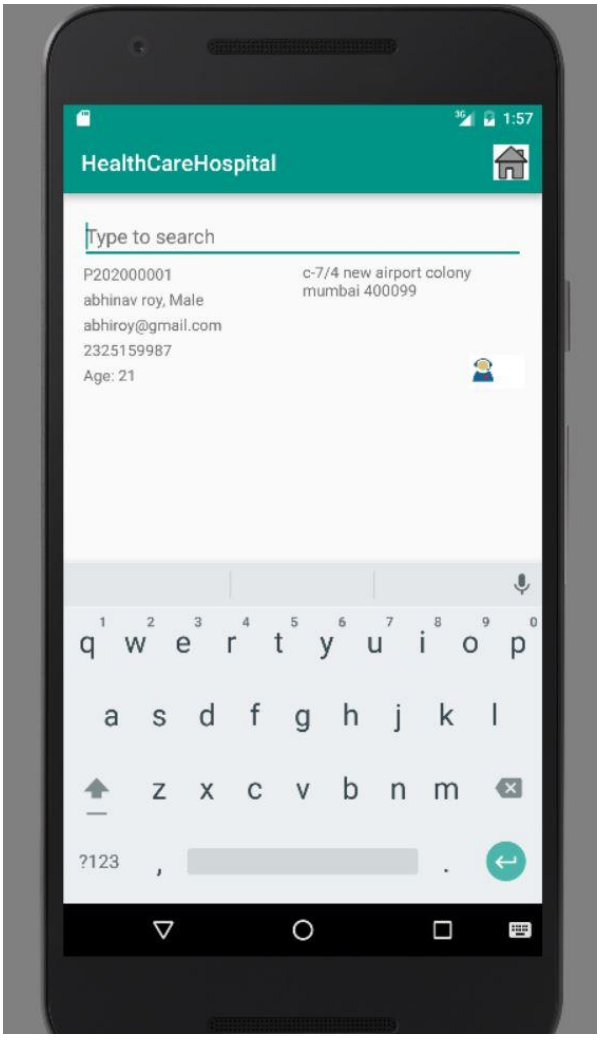


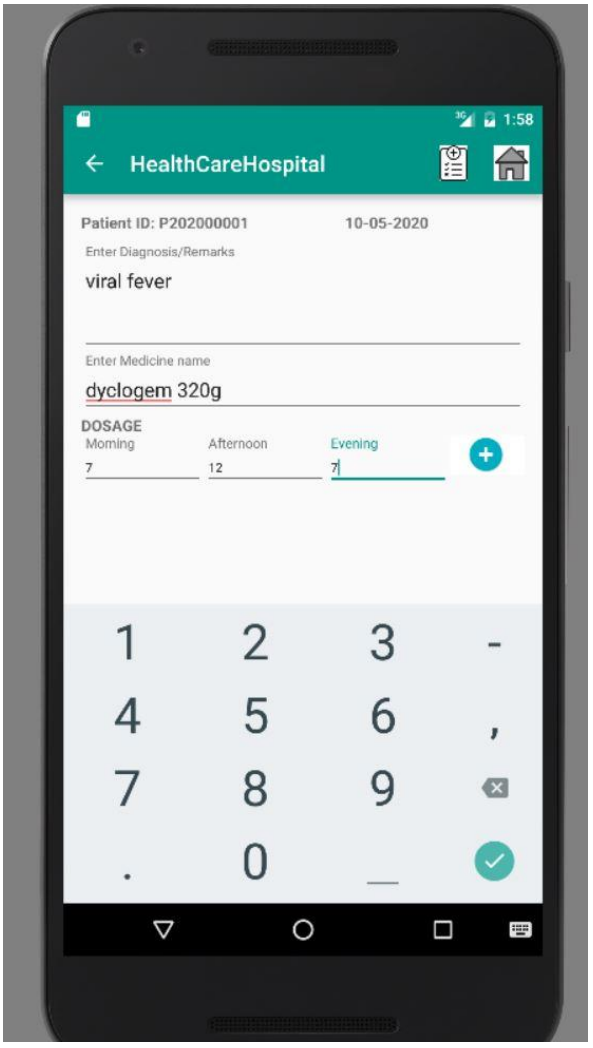


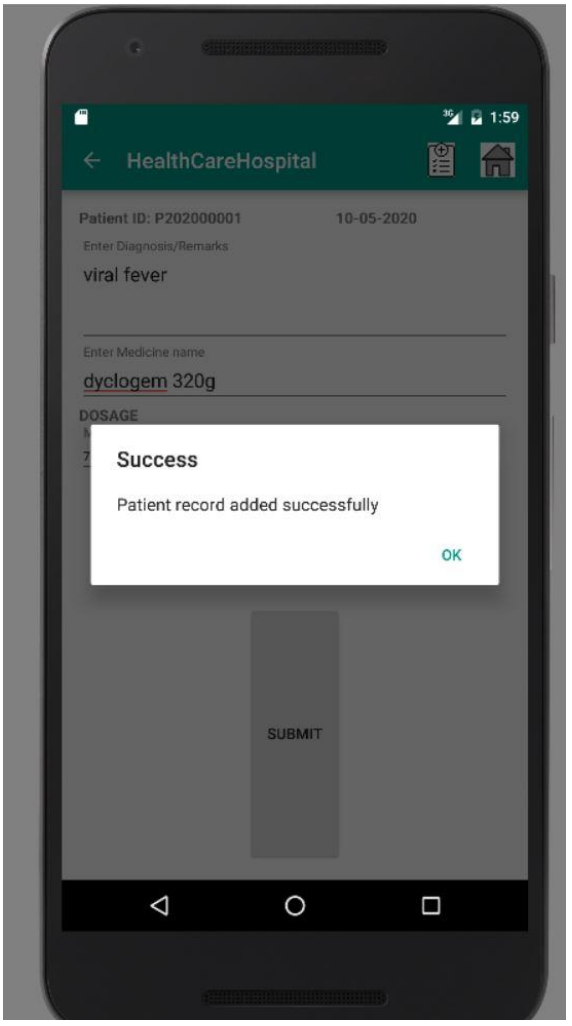

5:12
HealthCareHospital
ADD PATIENT
VIEW PATIENT
VIEW IPD PATIENT
ADD WARD/BEDS
VIEW WARDS
CHANGE PASSWORD
SIGN OUT

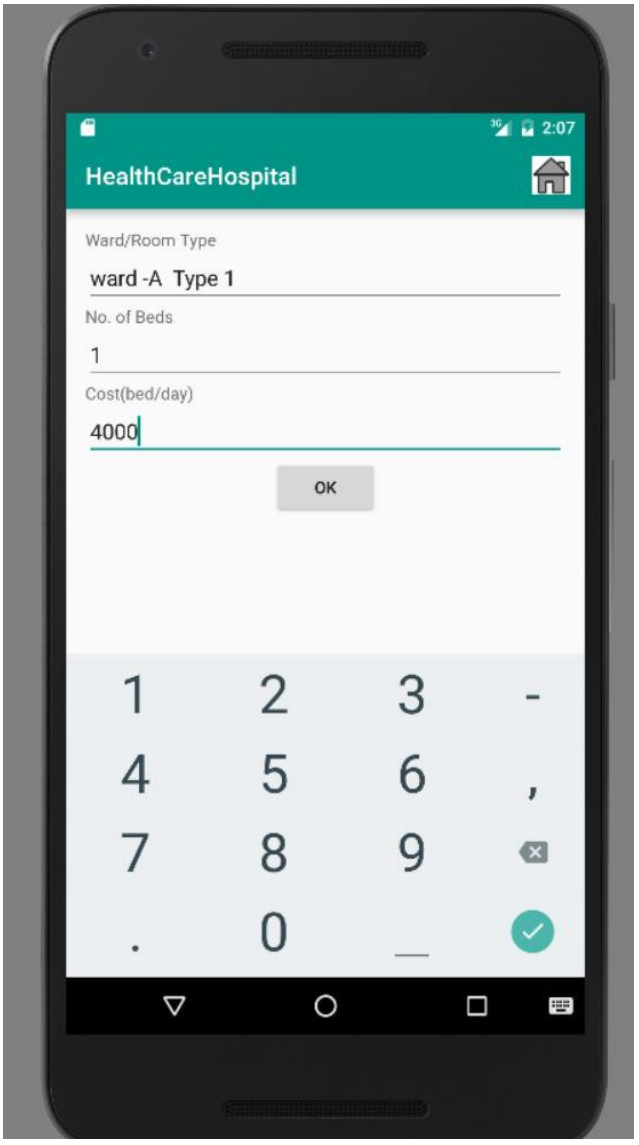

1:55
Add New Patient
Patient ID P202000001
Name abhinav roy
Address c-7/4 new airport colony mumbai 400099
Mobile No. 2325159987
Age 21
Emailid abhiroy@gmail.com
Gender Male Female
ADD

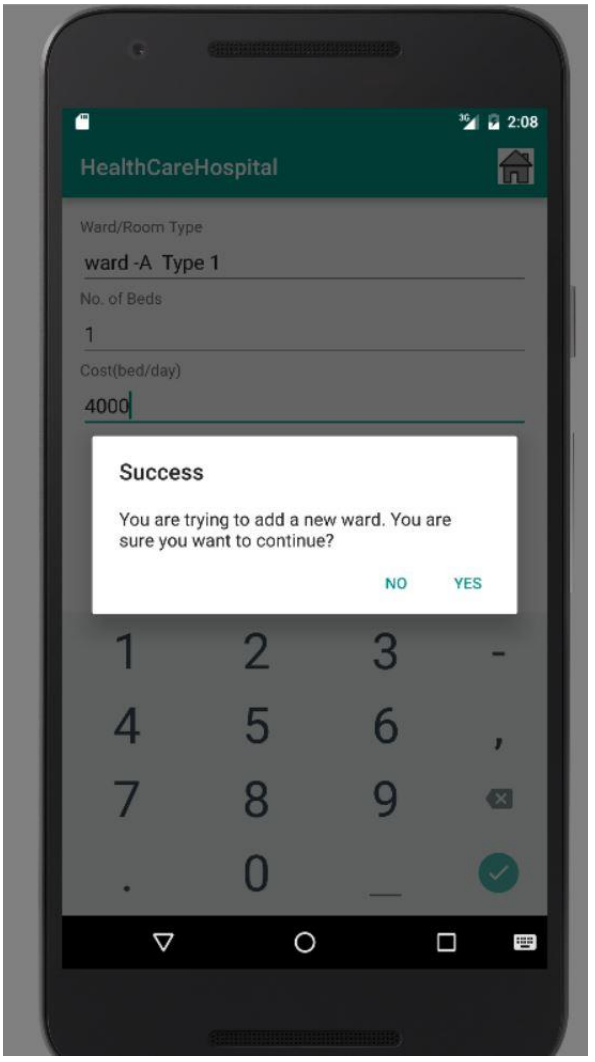

1:56
Add New Patient
Patient ID P202000001
Name abhinav roy
Address c-7/4 new airport colony
mumbai 400099
Success
Patient added successfully
OK
ADD

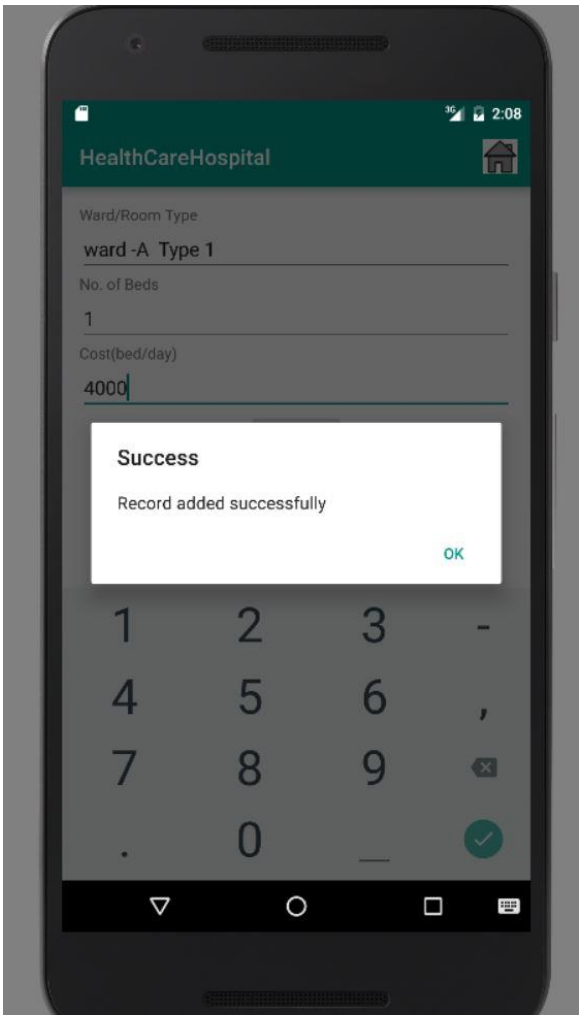

1:57
HealthCareHospital
Type to search
P202000001
abhinav roy, Male
abhiroy@gmail.com
2325159987
Age: 21
c-7/4 new airport colony
mumbai 400099


1:58
HealthCareHospital
Patient ID: P202000001
10-05-2020
Enter Diagnosis/Remarks
viral fever
Enter Medicine name
dyclogem 320g
DOSAGE
Morning
Afternoon
Evening
7
12
7


1:59
HealthCareHospital
Patient ID: P202000001
10-05-2020
Enter Diagnosis/Remarks
viral fever
Enter Medicine name
dyclogem 320g
DOSAGE
Success
Patient record added successfully
OK
SUBMIT


2:07
HealthCareHospital
Ward/Room Type
ward -A Type 1
No. of Beds
1
Cost(bed/day)
4000
OK


2:08
HealthCareHospital
Ward/Room Type
ward -A Type 1
No. of Beds
1
Cost(bed/day)
4000
Success
You are trying to add a new ward. You are sure you want to continue?
NO
YES


2:08
HealthCareHospital
Ward/Room Type
ward -A Type 1
No. of Beds
1
Cost(bed/day)
4000
Success
Record added successfully
OK


2:32
HealthCareHospital
Enter userid/emailid
rajannagi42@gmail.com
Select one
Answer secret question
Send OTP to registered email/mobile
SUBMIT

Chapter 7
Conclusion and Reference

7.1 Conclusion:

I have sincerely learnt the principles and various aspects of software engineering from this project work. This project required strong determination to achieve results, accompanied by sustained, methodical, hardwork and thus I proud to achieve such determination.

This project provides me next footstep in an academic year and help for standing at the threshold a informative and wizardry world.

The project **Hospital Management System (HMS)** is for computerizing the working in a hospital. The software takes care of all the requirements of an average hospital and is capable to provide easy and effective storage of information related to patients that come up to the hospital.

It generates test reports; provide prescription details including various tests, diet advice, and medicines prescribed to patient and doctor. It also provides injection details and billing facility on the basis of patient's status whether it is an indoor or outdoor patient.

The system also provides the facility of backup as per the requirement.

7.2 Biblography:-

SITES:

- https://www.google.co.in
- http://www.tutorialspoint.com
- http://www.w3schools.com/
- http://www.programmer2programmer.net
- http://www.codeproject.com
- http://www.slideshare.net
- http://www.stackoverflow.com
- http://www.youtube.com
- Creately.com (Diagrams)
- www.guru99.com

7.3 Reference:-

- Introduction of Software Engineering, 8th Edition
- The Complete Reference Java 2,Herbert Schildt
- Android Programming for Beginners, John Horton
- Android Apps for Absolute Beginner
- Android For programmers an APP driven approach
- Beginning Android4 Application Development