CHAPTER 1

INTRODUCTION

In this present world of automation and computerization, still there are many hospitals and health care centers that use manual methods for managing the records of hospital. The process of manually generating the records requires considerable time and efforts. This project on health plus is a constraint satisfaction problem, where we find a solution satisfying the given constraints.

The adoption of this technology in healthcare has started and we can expect to have commercial solutions in the market in near future. Most of the healthcare use cases for blockchain are intended to provide secure and integrated care to the patients. A Management System for a Health Care Facility. The system includes Registration of patients, making appointments, Storing patient records, Billing in the pharmacy & Pharmacy stock controlling.

In this project we are storing the details of the user who are logged in, details of the patients, details of the doctor, availability of the rooms and also the billing details of the patients. The resulted details is generated based on the inputs given by the user. The details of the patients and the staff is stored in the database for later use and analysis. The front end for this project is implemented using swings. The main scope of the project is to reduce the time effort, automation of the system, user friendly etc.

1.1 COURSE OBJECTIVES

1. To be able to demonstrate the concepts of OOPS like inheritance, polymorphism, abstraction, etc and Java Concepts (like exceptions, etc) for the chosen problem statement.
2. To be able to design a ER diagram for the chosen problem statement.
3. To be able to design a relational database model and schema with all constraints defined.
4. To be able to design interactive GUI using concepts of java swings.
5. To be able to connect the front end with the backend database using java.
6. To be able to apply the software engineering concepts on the designed project like use case diagram, sequence diagram, class diagram etc.

1.2 PROBLEM STATEMENT

The main objective of this system is to automate the existing manual system by using computer based system for easy accessing and manipulation. With this automation, the management can easily record accurate details of the patient, pharmacy stock, details of the doctor etc. When a patient wants to consult a doctor he or she first have to take appointment. After taking an appointment, it checks for the availability of the doctor. If the doctor is available, patient has to update his details. The patient details is stored in the database. Here the billing is done through cash or through net banking or through using cryptocurrency. This system also controls pharmacy stock.

The flow of events:

- The user has to either login or sign to their accounts.
- If the user is the patient then he has to enter his/her details such as name, date of birth, age, address etc.
- These details will be stored in database which is implemented using jdbc.
- The constraint satisfying healthplus is generated automatically.

1.3 OUTCOMES

The java application for this project is created using swings for front end and jdbc for backend. The sequence of actions performed in this is as follows.

- Firstly, when the user runs the application, Login page appear which has two options- login or create new account.

1. Login – the application checks if the login details entered by the user against the data in database. If the details are correct, home page gets displayed, otherwise error message is displayed.
2. New account – the details entered by the user is stored in the database and a new account is created.

- The Home page has sub field options like – patient, staff, room booking, taking appointments etc.
- Patient: the user can add details about the patient like name, age, weight, gender, address etc. They can also update, delete and display the details they have in their account only.
- Doctor: user can either add, delete, update or display the details of their account. The details that are maintained are doctor name, department, id.

- Room: It includes the details of the room such as room number , room type, status . All these details are stored in database.
- Bill Generation : It includes the details bill number, patient id, patient type, doctor charge, medicine charge etc.

The general outcomes in terms of knowledge, advantages, scope are as follows:

1. Able to implement and understand the OOPs concepts.
2. Able to design an attractive and interactive GUI using swings.
3. Good understanding of database, Software engineering concepts.
4. Reducing the time required in generation.
5. Considered efficient automatic timetable was generated with effective GUI.

CHAPTER 2

JAVA FEATURES AND OOPs CONCEPTS

2.1 FEATURES

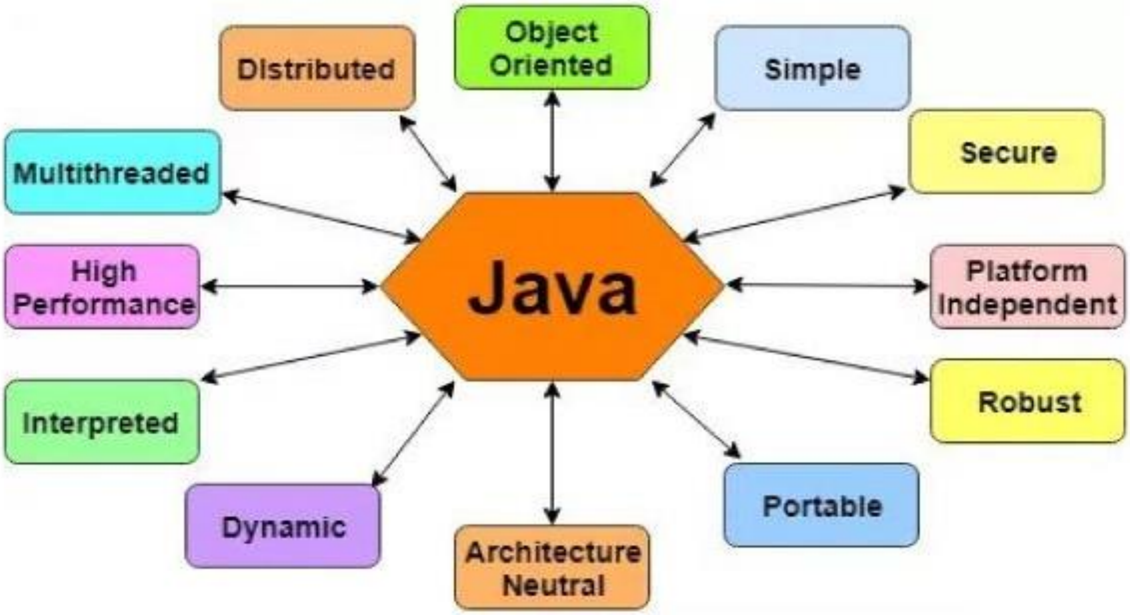


Fig. 1 Java Features

Simple:

It contains many features of C ,C ++ and all the complexities like pointers, storage classes, goto statements are removed. It doesn’t support multiple inheritance.

Object Oriented:

It is a purely an object oriented language, because all the codes are written using classes and objects.

Distributed:

For creating applications on network, java has the ability to share the data and program over the local area network, that is why java is a distributed language.

Multithreaded:

Java code can be divided into multiple threads, where each thread perform a different tasks and all the threads are executed in parallel.

Dynamic:

JVM has lot of run time information about the code and all the objects. Libraries are linked dynamically during runtime.

Architectural Neutral:

The program written on one platform can run on any other platform without having to recompile.

It follows: Write Once, Run Anywhere approach. Java code is basically written in java language with .java extention. This code is compiled into a bytecode with .class extention. Bytecode can be executed on any JVM installed machine with any OS.

Portable:

Java code is compiled into bytecode, instead of native machine level instruction and JVM executes the bytecodes on any machine.

High Performance:

Java programs are faster than scripts/programs written in interpreted language but slower than C and C++ that compiled to native machine language.

Robust:

It is able give some response in any kind of context like exception handling and typechecking.

Secure:

Java provides security through encapsulation, access specifiers. It checks the code while transferring from one machine to another, whether it is affected by virus or not. It doesn't execute the code if it is affected by virus.

2.2 OOPS CONCEPTS

The OOPs concepts are as follows:

- Class
- Object
- Inhertance
- Polymorphism
- Encapsulation

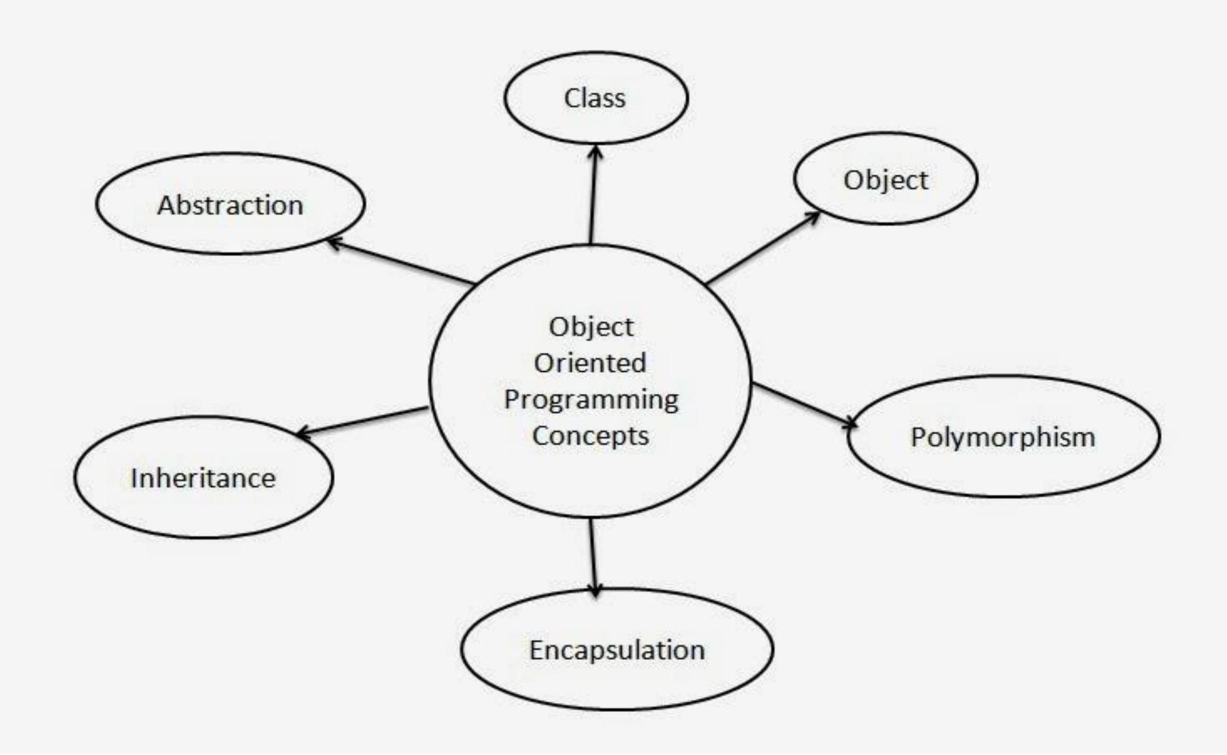


Fig.2 OOPs Concepts

- **Class**

A class is a blueprint or prototype from which objects are creaed. It is a template for an object and describes an objects behavior. It defines a new data type, whose values are objects.

Syntax:

```
Class classname{
Type instance-variable-1;
                ….
Type instance-variable-n;
Type method-name-1(parameter-list){…..}
                ……
Type method-name-n(parameter-list){…..}
}
```

- **Object**

It is an real world entity. It has state and behavior. An object is a runtime entity. It is an instance of a class.

An object creation has three parts:

- **Declaration** : creating a variable of class type.

Ex: course window;

- **Instantiation**: new keyword in the java that creates the object.
- **Initialization**: constructor of the class initializes the new object.

Syntax: class_name object_name = new constructor();

- **Inheritance**

It is a mechanism in which one object acquires all the properties and behavior of a parent object. The main advantage is reusability an to achieve runtime polymorphism.

Syntax:

```
Class Subclass-name extends Superclass-name
{
//methods and fields
}
```

Types of inheritance:

1. simple
2. multilevel
3. hierarchical
4. hybrid

- **Polymorphism:**

Polymorphism is the ability to take more than one form. Here the member function of a class behaves differently for different inputs. It is achieved through overloading and overriding.

Overloading:

1. by changing the number of arguments.
2. By changing the data type.

Overriding:

If a subclass has the same methods as declared in the parent class, it is known as method overriding.

- **Encapsulation**

It is a mechanism of wrapping the data and methods together as a single unit. The variables will be hidden using access specifiers like private, protected public. This concept is known as data hiding.

CHAPTER 3

REQUIREMENTS AND DESIGN

3.1 HARDWARE SPECIFICATIONS

Processor	:	Intel Core Processor
Speed	:	1.8 GHz
RAM	:	8GB

3.2 SOFTWARE SPECIFICATIONS

Operating System	:	WINDOWS 10
Programming Language	:	Java with JDK
IDE	:	Eclipse/Netbeans
Database	:	MYSQL

3.3 CLASS DIAGRAM

It is a type of UML diagram, which represents the structure of the system. It shows the classes, attributes, operations and the relationships among the objects. It is used for data modelling. In object-oriented modelling, class diagram is the main building block.

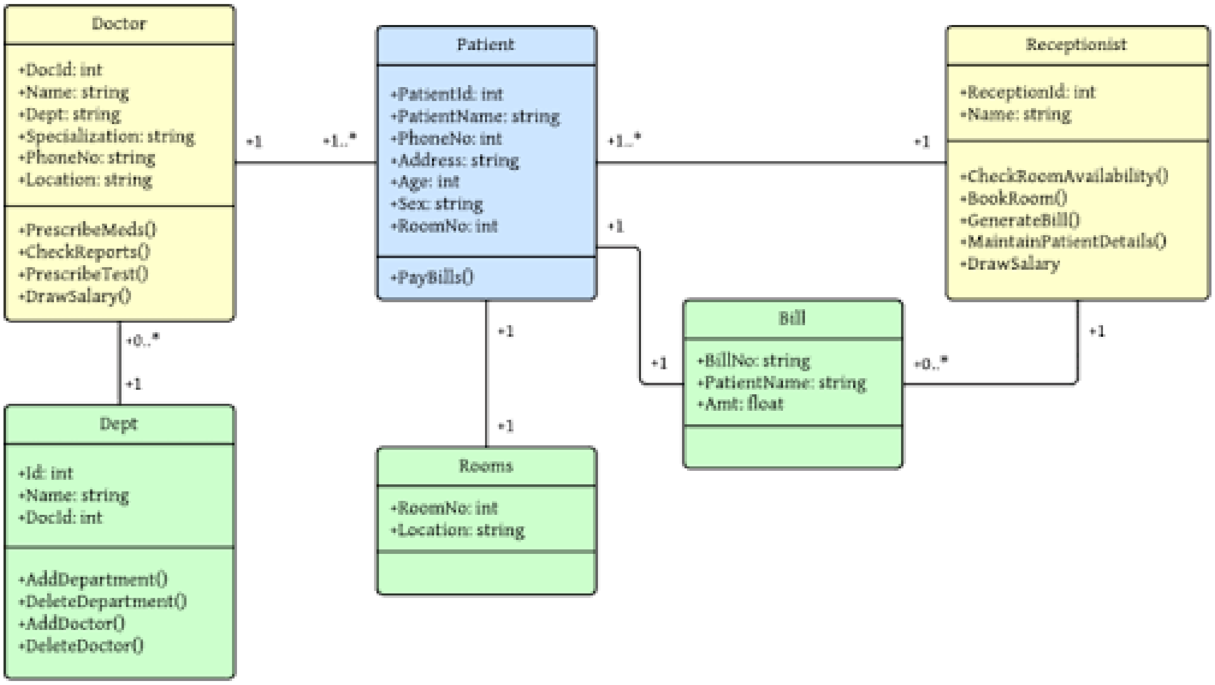


Fig. 3 Class Diagram

3.4 DATA FLOW DIAGRAM

It is a way of representing the flow of data of a system of process. There is no control flow, loops,rules. The inputs or output information is also provided by the data flow diagram. The role of data flow diagram is represented by the activity diagram in UML.

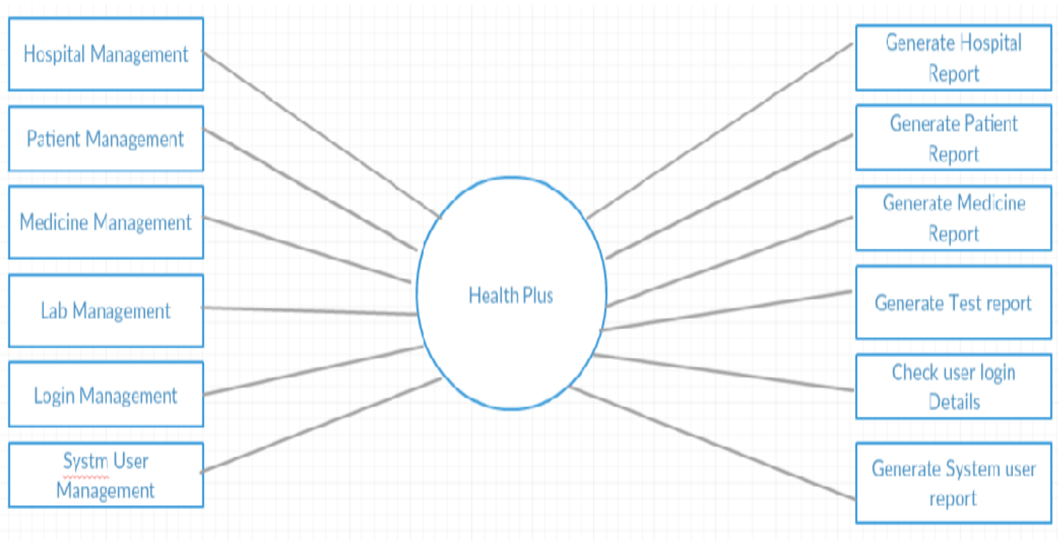


Fig. 4 Data flow diagram 1

3.5 USE CASE DIAGRAM

It is the simplest way of representing about how the users interact with the system. It shows the relationship between different use cases and user, in which they are involved. Either ellipses or circles are used to represent the use cases. It is actually a blueprint for the system/process. The standard form is defined by the unified modelling language. The various uses of the use case diagram are as follows:

1. The contents of the system can be specified.
2. The requirements of the system is captured.
3. System architecture can be validated.
4. Test cases are generated.

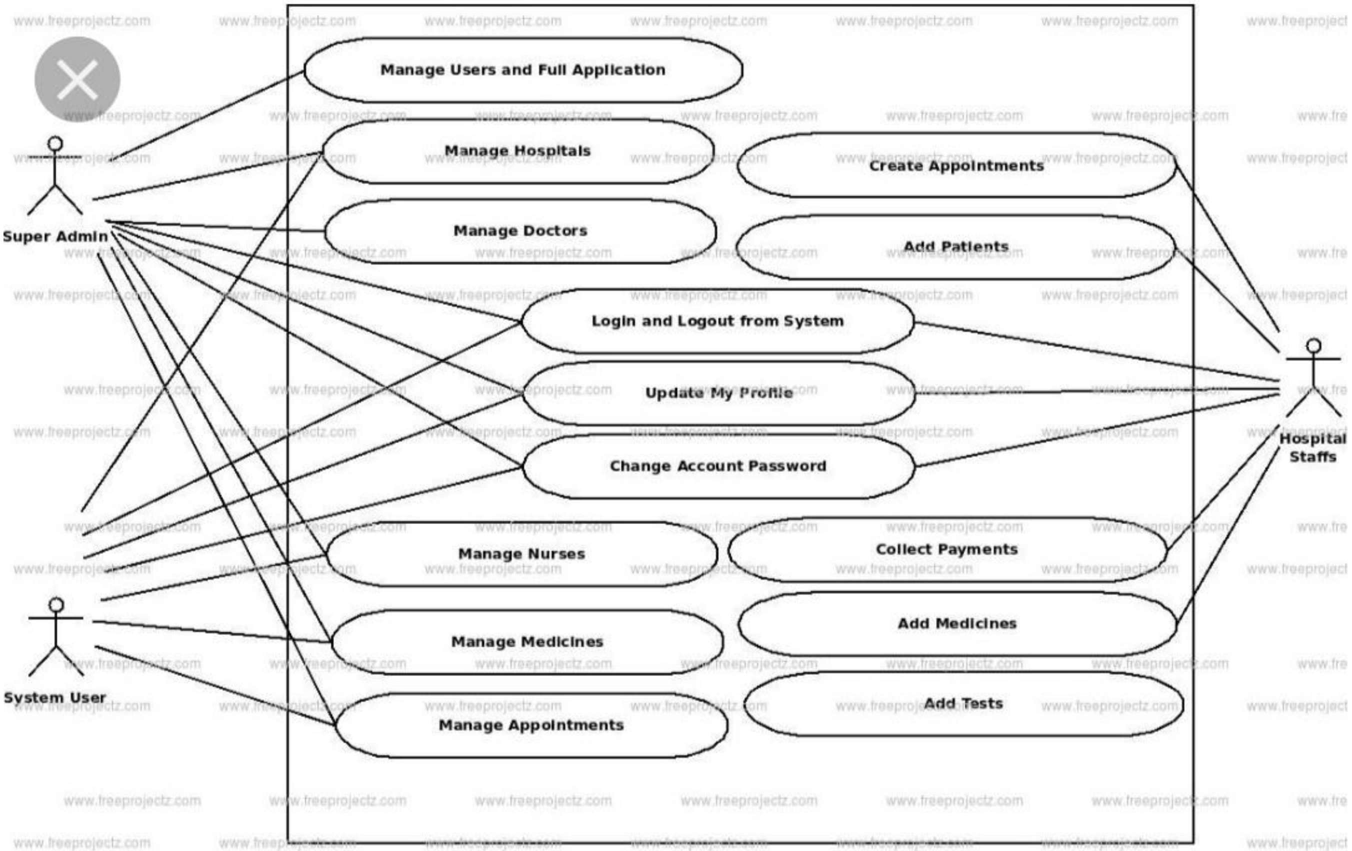


Fig. 5 Use case diagram

3.5 SEQUENCE DIAGRAM

The object interactions is shown by the sequence diagram, which is arranged in time sequence. It shows the classes and objects involved in the system and the sequence of message exchanged between them. They are sometimes called as Event diagrams or scenarios.

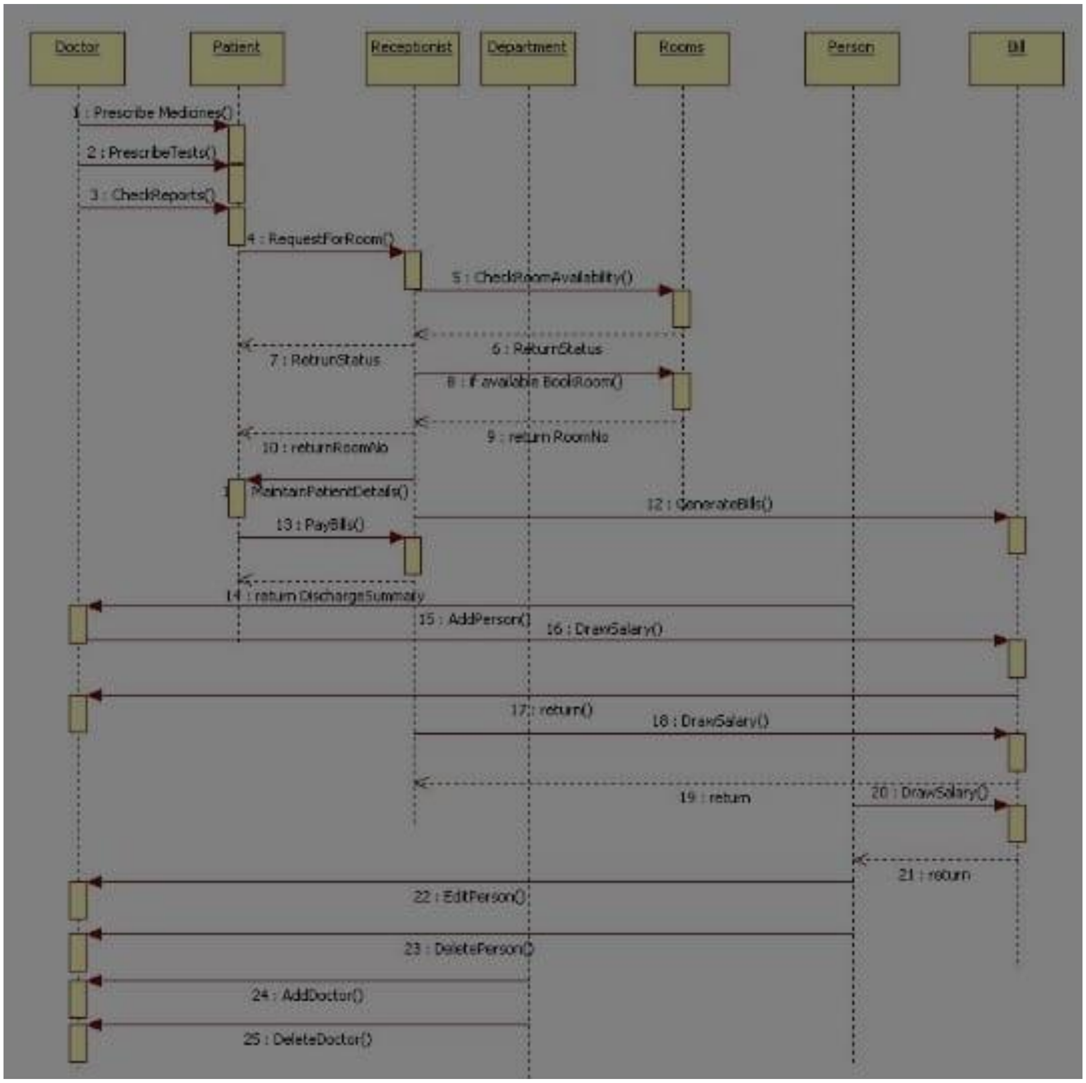


Fig. 6 Sequence diagram

CHAPTER 4

IMPLEMENTATION

4.1 ER DIAGRAM

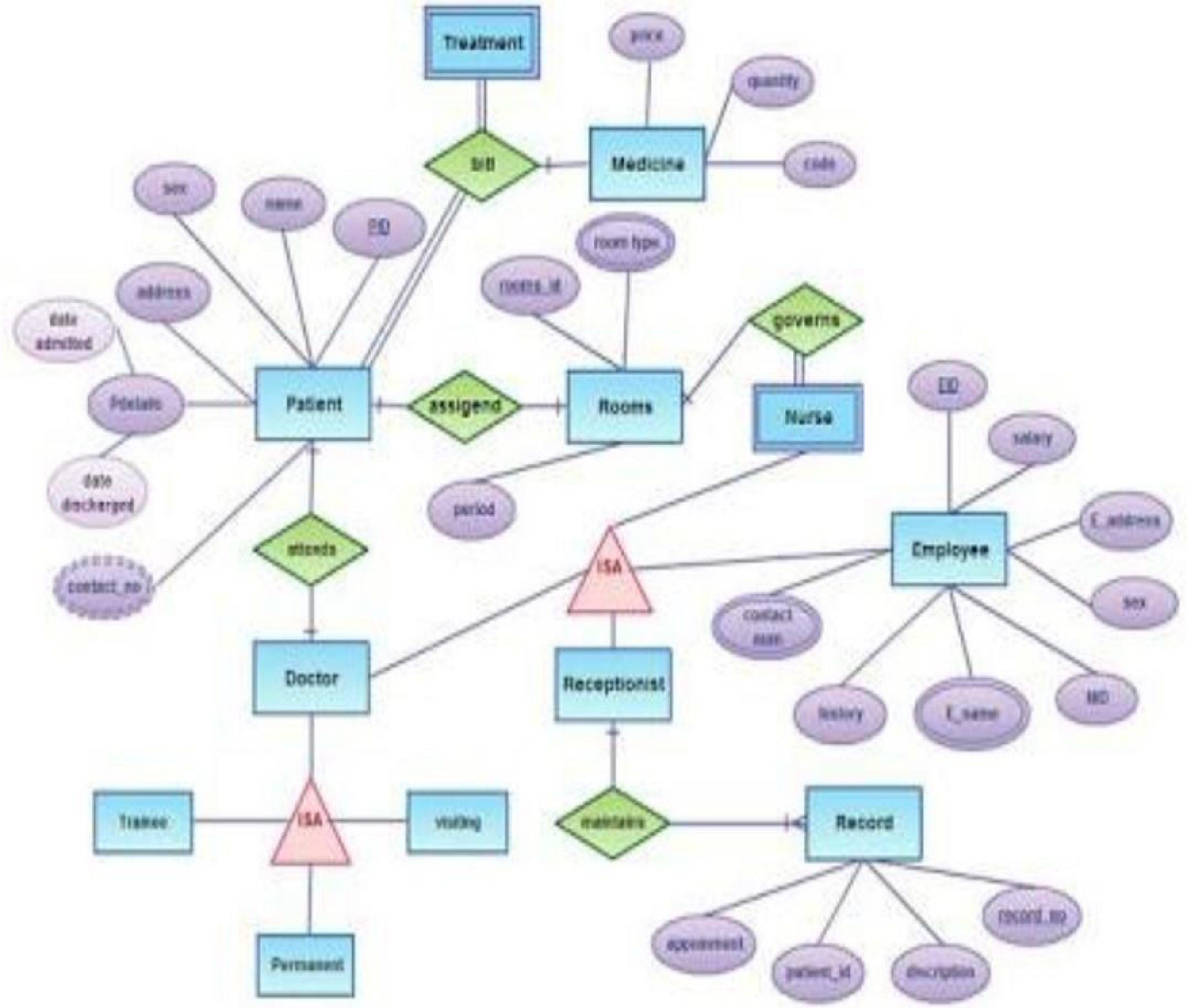


Fig. 7 ER Diagram

4.2 TABLE CREATION

The relations/tables are created using CREATE command. There are five tables.

1. **Create table command for login**:

```
CREATE TABLE login(id int PRIMARY KEY AUTO_INCREMENT, username varchar(15) not null unique, password varchar(15) not null unique);
```

2. **Create table command for doctor**:

```
CREATE TABLE doctor(did varchar(5)primary key,dname varchar(10),department varchar(10),pid int, CONSTRAINT fk1 FOREIGN KEY(pid) REFERENCES login(id) ON DELETE CASCADE ON UPDATE CASCADE);
```

3. **Create table command for patient**:

```
CREATE TABLE patient(pid varchar(5) PRIMARY KEY, patient_name varchar(15) NOT NULL, age int, weight int, gender varchar(5),address varchar(10), phno int, disease varchar(16), did int, CONSTRAINT fk2 FOREIGN KEY(did) REFERENCES login(id) ON DELETE CASCADE ON UPDATE CASCADE);
```

4. **Create table command for bill**:

```
CREATE TABLE bill(bill_no I nt, pid int, p_type int, d_charge int, m_charge int, r_charge int, no_of_days int, bill int,CONSTRAINT fk3 FOREIGN KEY(did) REFERENCES login(id) ON DELETE CASCADE ON UPDATE CASCADE);
```

5. **Create table command for room**:

```
CREATE TABLE tea_cou(r_no varchar(8), r_type varchar(10),status varchar(8),pid varchar(5),CONSTRAINT fk5 FOREIGN KEY(pid) REFERENCES patient(pid) ON DELETE CASCADE ON UPDATE CASCADE);
```

4.3 EXCEPTION HANDLING

An exception is a run-time error, that interrupts the execution of the program. This disruption is known as error or exception.

Basically there are two types of errors:

1. Compile time errors- it is further divided into two types:

a. Syntax error

b. Semantic Errors

2. Runtime errors.

The runtime errors are called as Exceptions. Exception is basically an event. The exceptions are handled by the try & catch block. Exception Handler is generally a piece of code to handle the errors/exceptions.

Try Catch block:

It is an inbuilt handling where,

1. The try block contains the normal code and
2. The catch block contains the exception handling.

Syntax :

```
Try{

//statements causing exception

}

Catch(exception_type object){

//code to handle error

}
```

Types of exceptions in java:

1. IO Exception

It is raised when there is an interruption in input-output operation.

2. SQL Exception

It arises, when there is no proper connection with the database/sql.

3. ClassNotFound Exception

When a class which is not definied is being accessed, exception occurs.

4. Runtime Exception

These are the exceptions that occur during runtime.

a. Arithmetic Exception

If in an arithmetic operation, if there is any exceptional condition, then the exception is thrown.

b. NullPointer Exception

This exception arises, when we try to refer a null object.

c. NumberFormat Exception

It arises, when the method is not able convert the string to numeric.

d. IndexOutOfBound Exception

Thisexception occurs, if the index is either greater or negative.

1.ArrayIndexOutOfBound Exception

2. String IndexOutOfBound Exception

Ex :SQL Exception used in this project.

```
try{

   :

   :

if(rs.next()){

JOptionPane.showMessageDialog(null,"You Logined Sucessfully....");

details.main(null);

id=rs.getInt(1);
}
            :
            :

}catch(Exception e1){
  System.out.println(e1); }
```

4.4 SWINGS

It is used for building the Graphical user interface using java as a platform for coding. There are many set of components present in swing framework, which provides functionalities. The window based applications are created using java swings. Swing components are platform independent, lightweight, etc. It is a part of JFC(Java Foundation class).

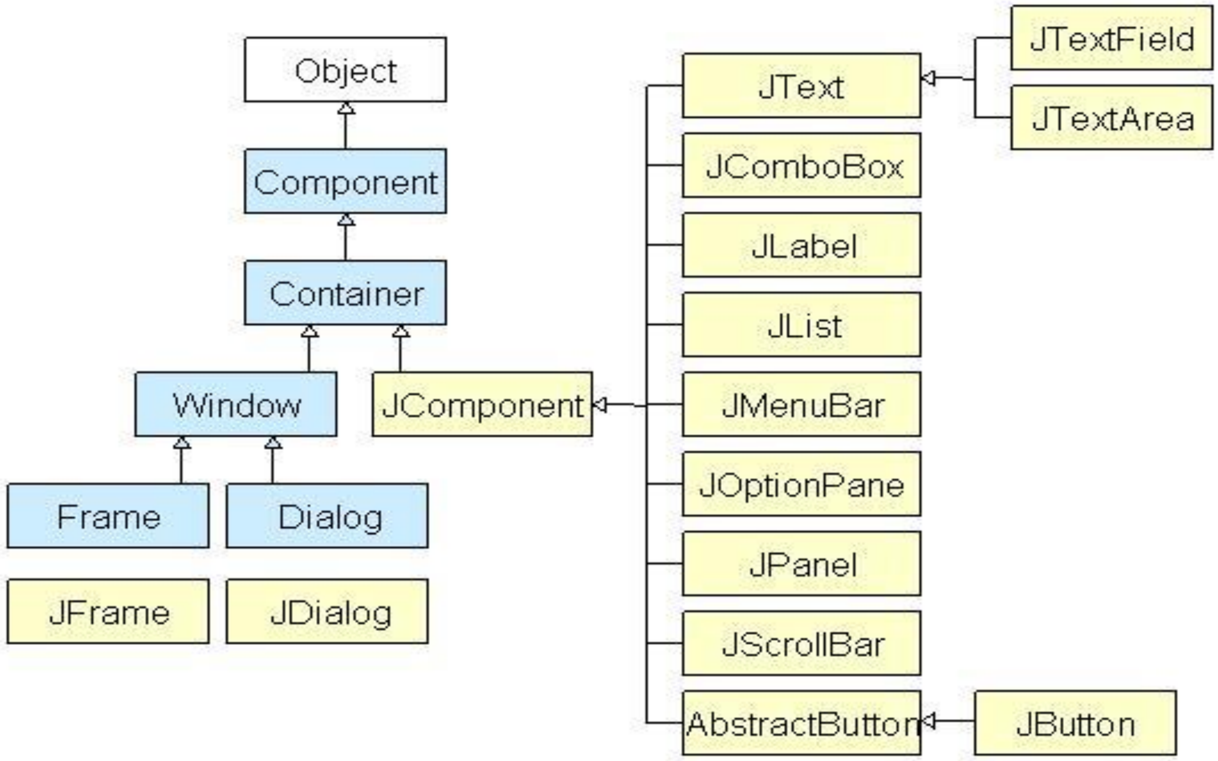


Fig. 8 Swing Components

4.5 JDBC

JDBC stands for Java Database Connectivity, which is an API for java. It is used for java database connection. The database can be queried using methods provided by jdbc. Two main queries used in this project are select and insert.

1. Select query:

Ex:
```
Class.forName("com.mysql.jdbc.Driver");
Connection con=DriverManager.getConnection ("jdbc:mysql://localhost:3306/timetable","root","root");
Statement str =con.createStatement();
String query="select * from login where username ='"+ t1.getText()+ "' and password ='"+p1.~~getText~~().toString()+"'";

ResultSet rs=str.executeQuery(query);
```

2. Insert query

Ex:
```
Class.forName("com.mysql.jdbc.Driver");
Connection con=DriverManager.getConnection ("jdbc:mysql://localhost:3306/timetable","root","root");

String query = "insert into course values(?,?,?,?,?,?,?)";

PreparedStatement s = con.prepareStatement(query);

String s1 = t1.getText();

String s2 = t2.getText();

int s3 = Integer.parseInt(t3.getText());

int s4 = Integer.parseInt(t4.getText());

String s5 = t5.getText();

String s6 = t6.getText();
```

```
s.setString(1, s1);

s.setString(2, s2);

s.setInt(3, s3);

s.setInt(4, s4);   …………

s.executeUpdate();
```

CHAPTER 5

OUTPUT SNAPSHOTS

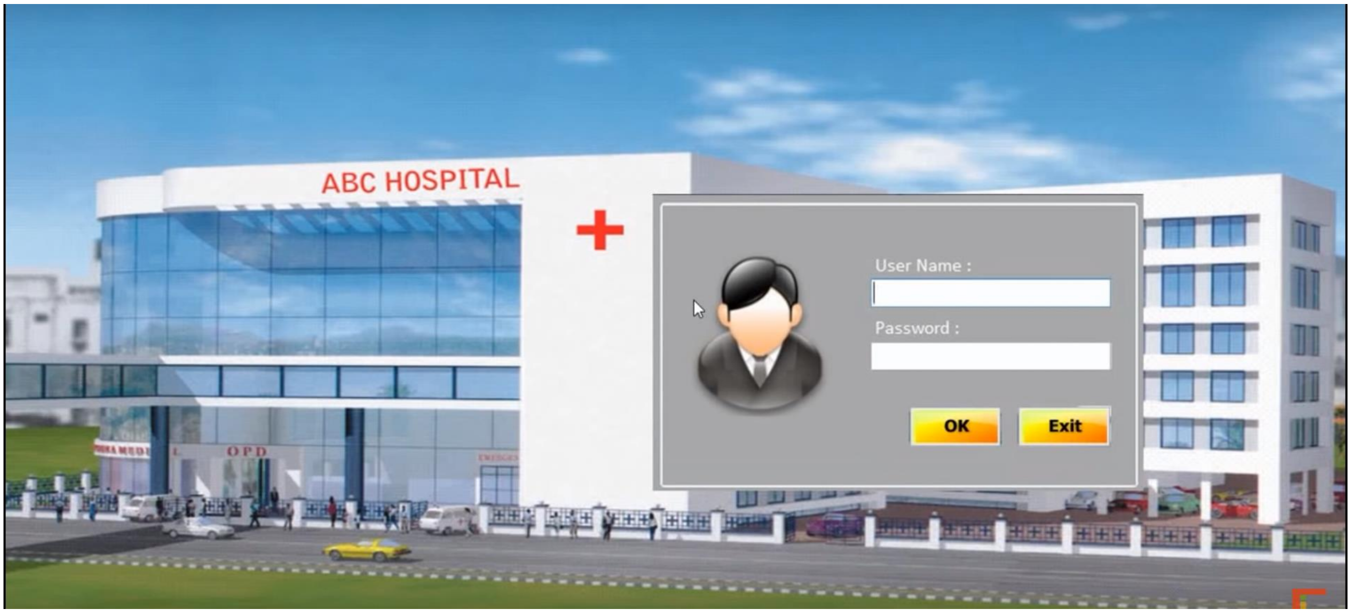


Fig. 9 Login Page

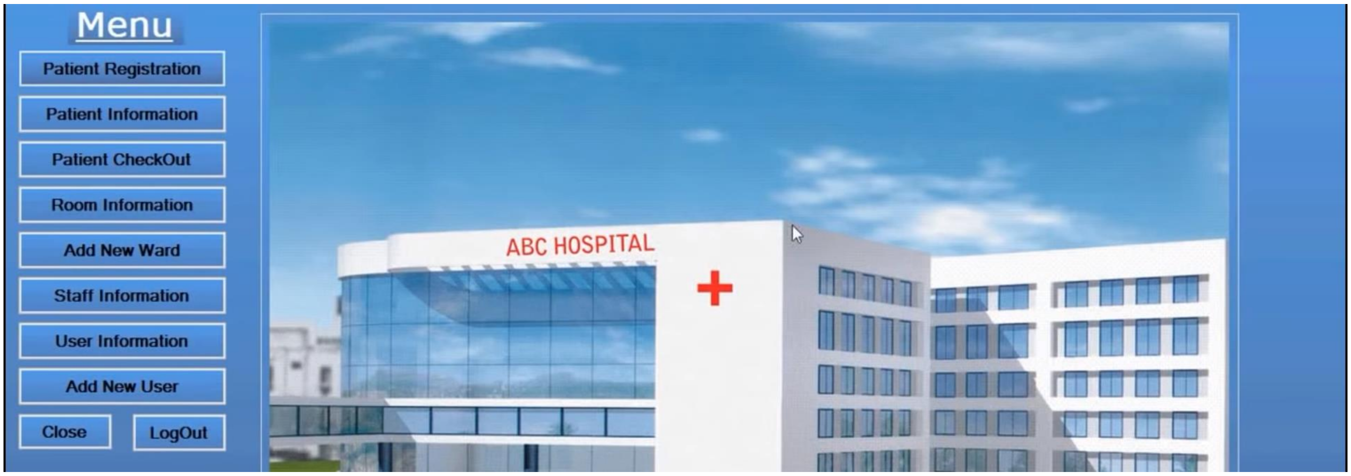
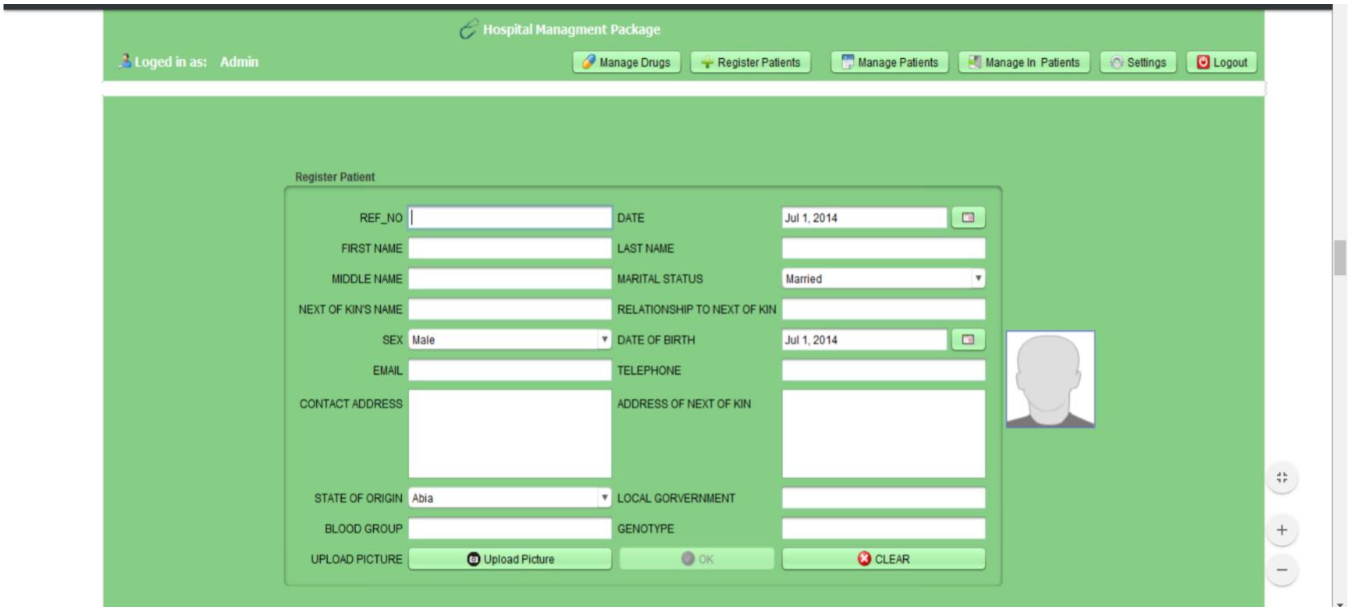


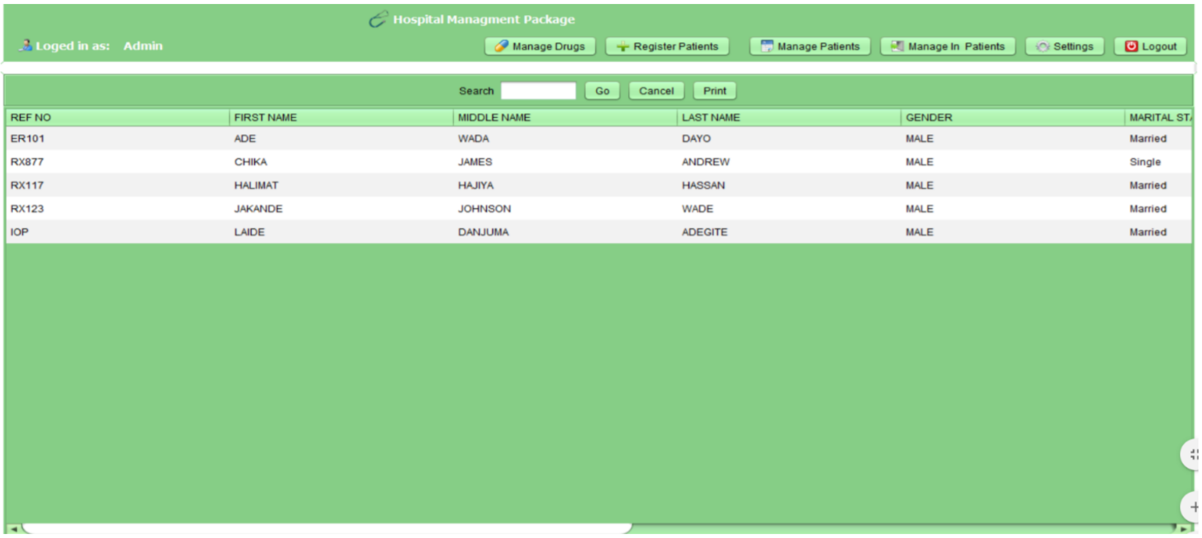
Fig. 10 Menu page



Register Patient

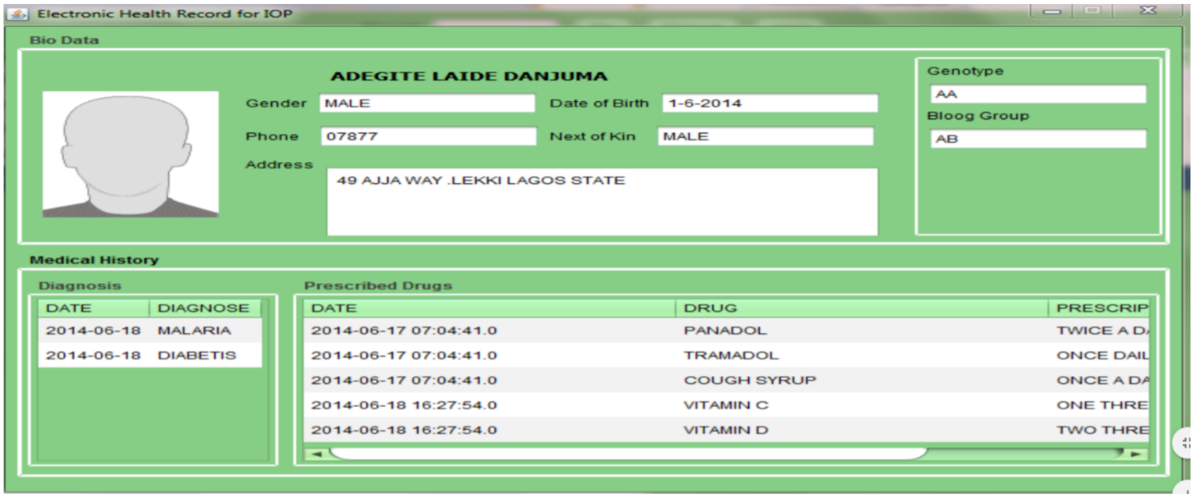
REF_NO		DATE	Jul 1, 2014
FIRST NAME		LAST NAME	
MIDDLE NAME		MARITAL STATUS	Married
NEXT OF KIN'S NAME		RELATIONSHIP TO NEXT OF KIN	
SEX	Male	DATE OF BIRTH	Jul 1, 2014
EMAIL		TELEPHONE	
CONTACT ADDRESS		ADDRESS OF NEXT OF KIN	
STATE OF ORIGIN	Abia	LOCAL GORVERNMENT	
BLOOD GROUP		GENOTYPE	
UPLOAD PICTURE			

Fig.11 Sign in Page



REF NO	FIRST NAME	MIDDLE NAME	LAST NAME	GENDER	MARITAL ST
ER101	ADE	WADA	DAYO	MALE	Married
RX877	CHIKA	JAMES	ANDREW	MALE	Single
RX117	HALIMAT	HAJIYA	HASSAN	MALE	Married
RX123	JAKANDE	JOHNSON	WADE	MALE	Married
IOP	LAIDE	DANJUMA	ADEGITE	MALE	Married

Fig.12 Patient details



Bio Data

ADEGITE LAIDE DANJUMA

Gender: MALE

Date of Birth: 1-6-2014

Phone: 07877

Next of Kin: MALE

Address: 49 AJJA WAY .LEKKI LAGOS STATE

Genotype: AA

Bloog Group: AB

Medical History

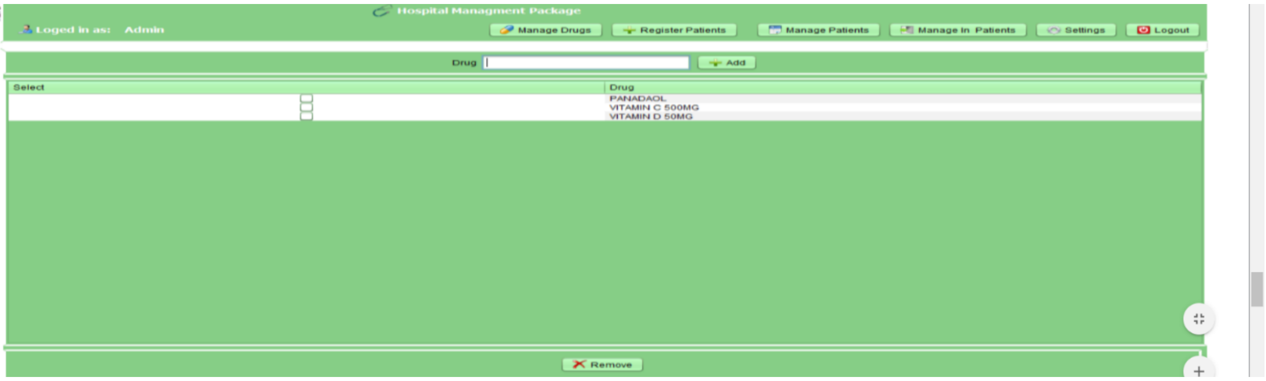
Diagnosis

DATE	DIAGNOSE
2014-06-18	MALARIA
2014-06-18	DIABETIS

Prescribed Drugs

DATE	DRUG	PRESCRIP
2014-06-17 07:04:41.0	PANADOL	TWICE A D
2014-06-17 07:04:41.0	TRAMADOL	ONCE DAIL
2014-06-17 07:04:41.0	COUGH SYRUP	ONCE A DA
2014-06-18 16:27:54.0	VITAMIN C	ONE THRE
2014-06-18 16:27:54.0	VITAMIN D	TWO THRE

Fig. 13 Doctor details



Loged in as: Admin

Drug:

Select	Drug
□	PANADAOL
□	VITAMIN C 500MG
□	VITAMIN D 50MG

Fig. 14 Pharmacy Details

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

The whole system activities are divided into patient, doctors, addmin. This health plus project was successfully implemented with interactive and effective GUI. This GUI is implemented using swings and database is designed for back end storage of data. This project has being implemented in step by step procedure using the concepts of OOPs in platform independent language Java. The health plus turns to be efficient by following most of the constraints. It was able to reduce the time required in manual generation of health records of patient. This project gave me an good understanding of all the concepts like DBMS,JAVA,OOPS etc.

6.2 FUTURE WORK

There are few features which can be integrated with this system to make it more flexible. Below list shows some future points to be consider.

- Video conferencing facility for remote areas for treatments.
- Hangout for different doctors and patients at different location.

6.3 REFERENCE

1. https://beginnersbook.com
2. www.roseindia.net
3. https://stackoverflow.com
4. www.quora.com