CHAPTER –1 INTRODUCTION

1 INTRODUCTION

1.1 Project Introduction

There are number of ways in which computer affect human lives which are uncountable. Computer process the function with much faster speed and greater accuracy. So among the various information system of data processing my project "PATIENT BILLING SYSTEM" one of the data processing system, which mainly concerns with the customer billing record of the hospital made the entire performed are computerized.

The main objective of the project is to hail a fully computerized system for main fining patient detail, patient bills, and ward details.

The project has the following features:

- Effectiveness
- Easy to use
- Consistency
- Simplicity
- Accuracy

The purpose of the Patient Billing System (PBS) is to capture necessary data elements from the agencies' internal bills/personnel systems, validate the information before bills are processed, and post the validated bills and personnel information. This new system is being developed to support the client server efforts. The main aim of the project is to create automated software which is purely used for billing operation in a hospital. This application would be facilitating the particular authorities of a hospital to generate and update the bills of a patient and store them. This will later be retrieved by the administrator at the time of discharge.

Developing a local patient billing software system would benefit the hospital management. This system basically works for the hospital management, helping them to generate bills and preserve patient's details in a well-organized approach. Since it is Software driven the quality of services can be enhanced considerably

CHAPTER -2 SYSTEM STUDY

2 SYSTEM STDUY

2.1 Existing System

- The existing system needs to save patients information in the form of excels sheets.
- The manual system gives us very less security for saving data; some data may be lost due to mismanagement.
- Searching of particular room vacancy information is very critical where it takes lot of time.
- This system is not having the facility of tracking transfers of patients from the general ward to the ICU and vice versa.

Limitations in existing system

- The current system is not completely complete computerized and manual system in entering students and staff data and handling it.
- There is no centralized database maintenance.
- There is no easy access to the particular employee record.
- The administrator and employee cannot easily navigate through the database.

2.2 Purposed System With Objectives

The purpose of this application is to provide complete convenience to the management of the hospital, in order to make Billing, an effortless task to perform. This project helps the administrator to overcome the difficulty in tracking and maintaining records of the patients. It saves his time because the tasks are sub-divided and assigned to respective authorities. So, every authority has only some degree of job to accomplish and hence avoids overload.

Objective

The main aim of the project is to develop a feature-rich, practical Patient Billing Software (PBS) for a hospital. The authorities of a hospital can login into their respective accounts and then update the bills consequently.

- PBS will provide a repository for detailed patient's bills and personnel data.
- Have a standard input format that is not patient specific.
- Validate, through edits, the information reported is properly authorized.
- Generate proper & valid transactions to initiate the payment process.

2.3 Feasibility Study

A feasibility study is defined as an evaluation or analysis of the potential impact of proposed project or program. A feasibility study is conducted to assist decision-makers in determining whether or not to implement a particular project or program. The feasibility study is based on extensive research on both the current practices and the proposed project and its impaction the school foodservice operation. The feasibility study will contain extensive data related to financial and operational impact and will include advantages and disadvantages of both the current situation and the proposed plan.

The feasibility study is conducted to assist the decision-makers in making the decision that will be in the best interest of the school foodservice operation. The extensive research, conducted in anon-biased manner, will provide data upon which to base a decision.

Economic Feasibility

This involves questions such as whether the firm can afford to build the system, whether its benefits should substantially exceed its costs, and whether the project has higher priority and profits than other projects that might use the same resources. This also includes whether the project is in the condition to fulfill all the eligibility criteria and the responsibility of both sides in case there are two parties involved in performing any project.

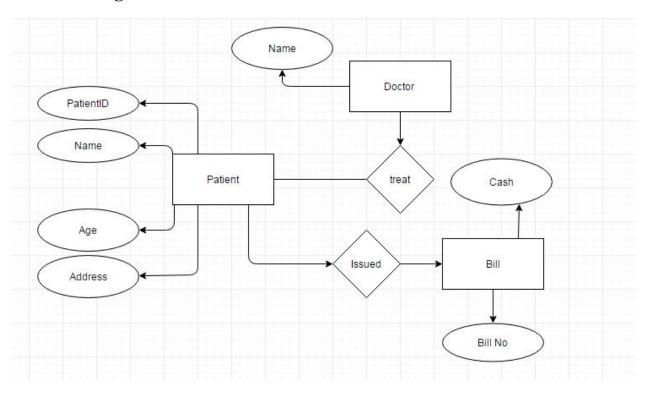
Technical Feasibility

This involves questions such as whether the technology needed for the system exists, how difficult it will be to build, and whether the firm has enough experience using that technology. The assessment is based on an outline design of system requirements in terms of Input, Output, Fields, Programs, and Procedures. This can be qualified in terms of volumes of data, trends, frequency of updating, etc. In order to give an introduction to the technical system.

CHAPTER -3 SYSTEM ANALYSIS

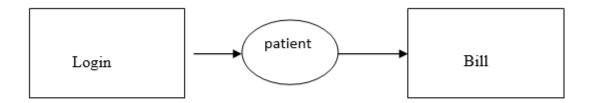
3 SYSTEM ANALYSIS

3.1 E-R Diagram

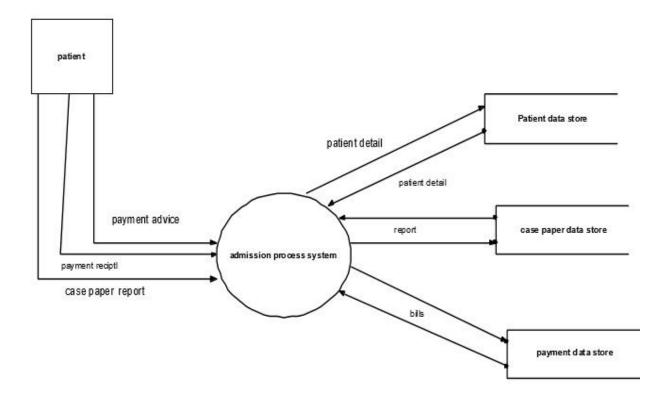


3.2 Data Flow Diagram

Zero Level DFD:



First Level DFD:



3.3 Hardware and software requirement specification

The following are the hardware and software environment on which the present system is developed and is being tested and likely to be implemented on. The system developed on a web based architecture with following configuration.

Hardware Requirements:

Processor : Intel Pentium-IV 2.5GHz or above

RAM : 1GB or above

Hard Disk : 40GB

Software Requirements:

Language : Java

Frame work : Swings

Back End : Oracle 10g

CHAPTER – 4 SYSTEM DESIGN

4 SYSTEM DESIGN

4.1 File / Database Design

Login→

Column Name	Data Type	Width
ID	number	10
Username	varchar	50
Password	varchar	50

info→

Column Name	Data Type	Width
FName	varchar	50
MName	varchar	50
LName	varchar	50
Age	varchar	50
Address	varchar	50
PatientID	varchar	50
Status	varchar	50
Gender	varchar	50

Confine→

Column Name	Data Type	Width
PatientId	Varchar	50
Room	Varchar	50
Doctor	Varchar	50
Illness	Varchar	50
Medicine	Varchar	50
Quantity	Varchar	50

mede→

Column Name	Data Type	Width
Item	varchar	50
Mede	varchar	50
Price	varchar	50

billing→

Column Name	Data Type	Width	
Fname	Varchar	50	
RoomNumber	Varchar	50	
RoomPrice	Varchar	50	
Mede	Varchar	50	
Medecost	Varchar	50	
TotalPayment	Varchar	50	

4.2 SCREEN DESIGN

Login:



Fig 1: Login Page

Mainframe Page:

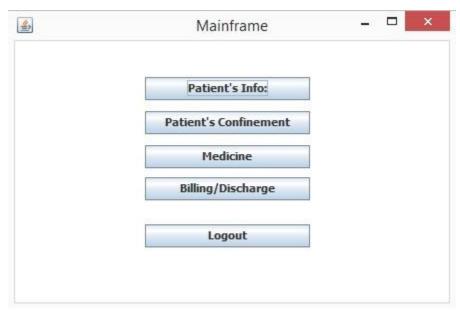


Fig 2: Main Page

Patient's Information:

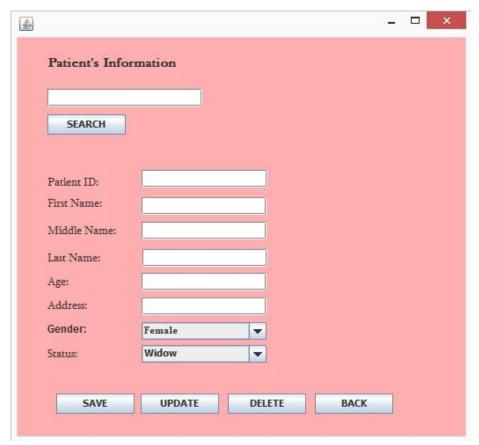


Fig 3: Patient Information Page

Medicine Information:



Fig 4: Medicine Page

Patient Confinement:

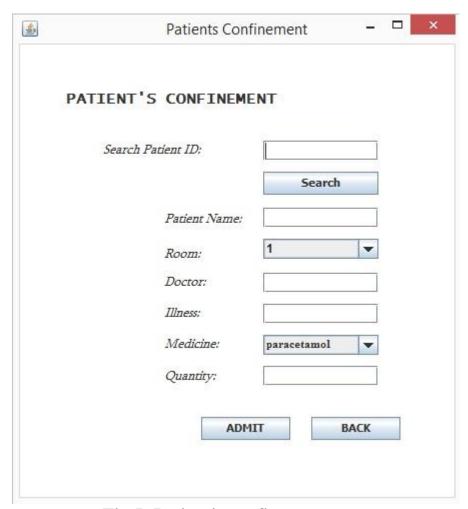


Fig 5: Patient's confinement page

Billing:

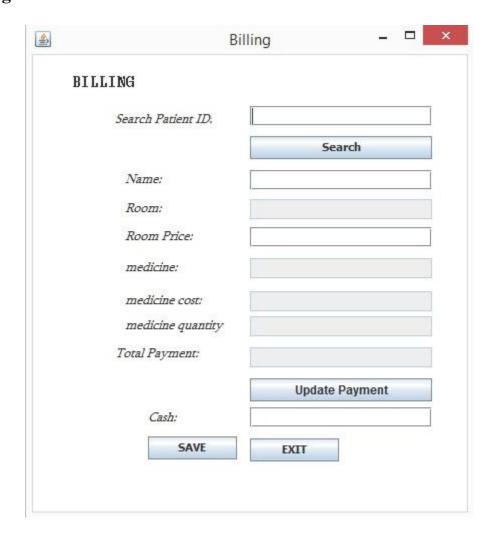


Fig 6: Billing Page

CHAPTER - 5
CODING

5 Coding

Login.java

```
import java.awt.BorderLayout;
import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JTextField;
import java.awt.Font;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import java.awt.Color;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.awt.event.ActionEvent;
import javax.swing.JPasswordField;
import javax.swing.JTextPane;
public class Login extends JFrame
private JPanel contentPane;
private JTextField textField;
private JPasswordField passwordField;
/*** Launch the application. */
public static void main(String[] args) {
EventQueue.invokeLater(new Runnable() {
public void run() {
try {
Login frame = new Login();
frame.setVisible(true);
} catch (Exception e) {
e.printStackTrace();
```

```
}
});
}
/*** Create the frame.*/
public Login() {
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setBounds(100, 100, 450, 300);
contentPane = new JPanel();
contentPane.setBackground(Color.PINK);
contentPane.setForeground(Color.BLACK);
contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
setContentPane(contentPane);
contentPane.setLayout(null);
setTitle("Patient Billing");
JLabel lblUsername = new JLabel("Username:");
lblUsername.setFont(new Font("Sylfaen", Font.BOLD, 13));
lblUsername.setBounds(83, 74, 89, 21);
contentPane.add(lblUsername);
textField = new JTextField();
textField.setBounds(182, 73, 126, 22);
contentPane.add(textField);
textField.setColumns(10);
JLabel lblPassword = new JLabel("Password:");
lblPassword.setFont(new Font("Sylfaen", Font.BOLD, 13));
lblPassword.setBounds(83, 122, 89, 21);
contentPane.add(lblPassword);
JButton btnLogIn = new JButton("LOG IN");
btnLogIn.setFont(new Font("Tahoma", Font.BOLD, 11));
btnLogIn.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent arg0) {
Connection con = null;
try{
Class.forName("oracle.jdbc.driver.OracleDriver");
              con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","scott"
       ,"tiger");
}
       catch(ClassNotFoundException e){
       System.err.println("Failed to Load Driver");
```

```
e.printStackTrace();
}catch (SQLException e){
System.err.println("Unable to Connect");
e.printStackTrace();}
try {
Statement statement = con.createStatement();
String qry = "SELECT * FROM login WHERE username = "" + textField.getText()+ "' AND
password = "" + String.valueOf(passwordField.getPassword()) + "" ";
ResultSet rs = statement.executeQuery(gry);
int count = 0;
while (rs.next()) {
count++;
//or count = count + 1;
}
statement.close();
//rs.close();
if (count > 0) {
JOptionPane.showMessageDialog(null, "Welcome to CUK Hospital!", "",
JOptionPane.INFORMATION_MESSAGE);
Mainframe Mainframe = new Mainframe();
Mainframe.setVisible(true);
dispose();
\} else if (count == 0) {
JOptionPane.showMessageDialog(null, "Invalid keyword!", "Error",
JOptionPane.ERROR_MESSAGE);
} catch (SQLException e) {
// TODO Auto-generated catch block
e.printStackTrace();
}
}
});
btnLogIn.setBounds(104, 189, 89, 23);
       contentPane.add(btnLogIn);
       JButton btnCancel = new JButton("CANCEL");
btnCancel.setFont(new Font("Tahoma", Font.BOLD, 11));
btnCancel.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent arg0) {
System.exit(0);
```

```
}
});
btnCancel.setBounds(219, 189, 89, 23);
contentPane.add(btnCancel);
passwordField = new JPasswordField();
passwordField.setBounds(182, 116, 126, 27);
contentPane.add(passwordField);
}
```

Mainframe.java

```
import java.awt.BorderLayout;
import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.Color;
import java.awt.Font;
import java.awt.BorderLayout;
import java.awt.EventQueue;
public class Mainframe extends JFrame {
private JPanel contentPane;
/*** Launch the application. */
public static void main(String[] args) {
EventQueue.invokeLater(new Runnable() {
public void run() {
try {
Mainframe frame = new Mainframe();
frame.setVisible(true);
} catch (Exception e) {
e.printStackTrace();
}
});
```

```
}
/*** Create the frame.*/
public Mainframe() {
setDefaultCloseOperation(JFrame.EXIT ON CLOSE);
setBounds(100, 100, 450, 300);
contentPane = new JPanel();
contentPane.setBackground(Color.WHITE);
contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
setContentPane(contentPane);
contentPane.setLayout(null);
setTitle("Mainframe");
JButton btnPatientsInfo = new JButton("Patient's Info:");
btnPatientsInfo.setFont(new Font("Tahoma", Font.BOLD, 11));
btnPatientsInfo.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent arg0) {
PatientINFO PatientINFO = new PatientINFO();
PatientINFO.setVisible(true);
});
btnPatientsInfo.setBounds(130, 36, 165, 23);
contentPane.add(btnPatientsInfo);
JButton btnNewButton = new JButton("Patient's Confinement");
btnNewButton.setFont(new Font("Tahoma", Font.BOLD, 11));
btnNewButton.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent arg0) {
CONFINEMENT CONFINEMENT = new CONFINEMENT():
CONFINEMENT.setVisible(true);
});
btnNewButton.setBounds(130, 70, 165, 23);
contentPane.add(btnNewButton);
JButton btnNewButton 1 = new JButton("Billing/Discharge");
btnNewButton 1.setFont(new Font("Tahoma", Font.BOLD, 11));
btnNewButton 1.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent arg0) {
Billing Billing = new Billing();
Billing.setVisible(true);
});
btnNewButton 1.setBounds(130, 136, 165, 23);
contentPane.add(btnNewButton_1);
```

```
JButton btnNewButton_2 = new JButton("Logout");
btnNewButton 2.setFont(new Font("Tahoma", Font.BOLD, 11));
btnNewButton 2.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent arg0) {
dispose();
Login Login = new Login();
Login.setVisible(true);
}
});
btnNewButton_2.setBounds(130, 183, 165, 23);
contentPane.add(btnNewButton_2);
JButton btnNewButton 3 = new JButton("Medicine");
btnNewButton_3.setFont(new Font("Tahoma", Font.BOLD, 11));
btnNewButton_3.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent arg0) {
MedicineINFO MedicineINFO = new MedicineINFO();
MedicineINFO.setVisible(true);
});
btnNewButton_3.setBounds(130, 104, 165, 23);
contentPane.add(btnNewButton 3);
```

PatientINFO.java

```
import java.awt.BorderLayout;
import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.table.DefaultTableModel;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JRadioButton;
import javax.swing.JTextField;
import javax.swing.JComboBox;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JButton;
import java.awt.Color;
import java.awt.Font;
```

```
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.sql.Statement;
import java.util. Vector;
import java.awt.event.ActionEvent;
public class PatientINFO extends JFrame {
private JPanel contentPane;
private JTextField age;
private JTextField address;
private JTextField fname;
private JTextField mname;
private JTextField lname;
private JTextField patientid;
private JComboBox combogender;
private JTextField textField;
long idglobal;
String wherestring="";
/*** Launch the application. */
public static void main(String[] args) {
EventQueue.invokeLater(new Runnable() {
public void run() {
try {
PatientINFO frame = new PatientINFO();
frame.setVisible(true);
} catch (Exception e) {
e.printStackTrace();
}
});
}
/** * Create the frame. */
public PatientINFO() {
setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
setBounds(100, 100, 525, 491);
contentPane = new JPanel();
contentPane.setBackground(Color.PINK);
contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
```

```
setContentPane(contentPane);
contentPane.setLayout(null);
JLabel lblPatientsInformation = new JLabel("Patient's Information");
lblPatientsInformation.setFont(new Font("Baskerville Old Face", Font.BOLD, 15));
lblPatientsInformation.setBounds(34, 11, 250, 36);
contentPane.add(lblPatientsInformation);
JLabel lblName = new JLabel("First Name:");
lblName.setFont(new Font("Sylfaen", Font.PLAIN, 13));
lblName.setBounds(34, 182, 72, 14);
contentPane.add(lblName);
JLabel lblAge = new JLabel("Age:");
lblAge.setFont(new Font("Sylfaen", Font.PLAIN, 13));
lblAge.setBounds(34, 271, 46, 14);
contentPane.add(lblAge);
JLabel lblAddress = new JLabel("Address:");
lblAddress.setFont(new Font("Sylfaen", Font.PLAIN, 13));
lblAddress.setBounds(34, 298, 46, 14);
contentPane.add(lblAddress);
JLabel lblSex = new JLabel("Gender:");
lblSex.setFont(new Font("Sylfaen", Font.PLAIN, 13));
lblSex.setBounds(65, 182, 46, -10);
contentPane.add(lblSex);
age = new JTextField();
age.setBounds(141, 267, 142, 20);
contentPane.add(age);
age.setColumns(10);
address = new JTextField();
address.setBounds(141, 295, 142, 20);
contentPane.add(address);
address.setColumns(10);
fname = new JTextField();
fname.setBounds(141, 181, 142, 20);
contentPane.add(fname);
fname.setColumns(10);
JLabel lblMiddleName = new JLabel("Middle Name:");
lblMiddleName.setFont(new Font("Sylfaen", Font.PLAIN, 13));
lblMiddleName.setBounds(34, 213, 86, 14);
```

```
contentPane.add(lblMiddleName);
mname = new JTextField();
mname.setBounds(141, 209, 142, 20);
contentPane.add(mname);
mname.setColumns(10);
JLabel lblLastName = new JLabel("Last Name:");
lblLastName.setFont(new Font("Sylfaen", Font.PLAIN, 13));
lblLastName.setBounds(34, 244, 79, 14);
contentPane.add(lblLastName);
lname = new JTextField();
lname.setBounds(142, 240, 142, 20);
contentPane.add(lname);
lname.setColumns(10);
JLabel lblStatus = new JLabel("Status:");
lblStatus.setFont(new Font("Sylfaen", Font.PLAIN, 13));
lblStatus.setBounds(34, 353, 46, 14);
contentPane.add(lblStatus);
final JComboBox combostatus = new JComboBox();
combostatus.setFont(new Font("Tahoma", Font.BOLD, 11));
combostatus.setModel(
new DefaultComboBoxModel(new String[] {"Single", "Married", "Widow", "Divorced"}));
combostatus.setSelectedIndex(2);
combostatus.setBounds(141, 349, 142, 20);
contentPane.add(combostatus);
JButton btnSave = new JButton("SAVE");
btnSave.addActionListener(new ActionListener() {
private Connection con;
public void actionPerformed(ActionEvent arg0) {
Connection con = null;
try{
Class.forName ("oracle.jdbc.driver.OracleDriver");
con = DriverManager.getConnection("idbc:oracle:thin:@ localhost:1521:xe", "scott", "tiger");
}catch(ClassNotFoundException e){
System.err.println("Failed to Load Driver");
e.printStackTrace();
}catch (SQLException e){
System.err.println("Unable to Connect");
e.printStackTrace();}
String query = "INSERT into info (fname, mname, lname, age, address, patientid, status, gender)
VALUES (?,?,?,?,?,?,?)";
Statement statement2;
try {
```

```
statement2 = con.createStatement();
String qry2 = "SELECT * FROM info WHERE patientid = "" + patientid.getText() + "" ";
ResultSet rs = statement2.executeQuery(gry2);
int count = 0;
while (rs.next()) {
count++;
if (count == 0) {
                                if (!fname.getText().equals("") \&\&!age.getText().equals("") \&@!age.getText().equals("") \&@!age.getText().equals(
                 !address.getText().equals("")&&!mname.getText().equals("")&&!lname.getText().equals
                ("")&& !patientid.getText().equals(""))
try {
Statement statement = con.createStatement();
PreparedStatement preparedStmt = con.prepareStatement(query);
preparedStmt.setString(1, fname.getText());
preparedStmt.setString(2, mname.getText());
preparedStmt.setString(3, lname.getText());
preparedStmt.setString(4, age.getText());
preparedStmt.setString(5, address.getText());
preparedStmt.setString(6, patientid.getText());
preparedStmt.setString(7, combostatus.getSelectedItem().toString());
preparedStmt.setString(8, combogender.getSelectedItem().toString());
preparedStmt.executeUpdate();
JOptionPane.showMessageDialog(null, "Patient Added", "Title",
JOptionPane.INFORMATION MESSAGE);
} catch (SQLException e) {
// TODO Auto-generated catch block
e.printStackTrace();
else {JOptionPane.showMessageDialog(null, "Fields should not be empty.", "Title",
                JOptionPane.ERROR_MESSAGE);
} else {JOptionPane.showMessageDialog(null, "Patient ID already exist!", "Title",
JOptionPane.INFORMATION MESSAGE):
} catch (SQLException e1) {
// TODO Auto-generated catch block
e1.printStackTrace();
}
});
btnSave.setFont(new Font("Tahoma", Font.BOLD, 11));
btnSave.setBounds(44, 404, 89, 23);
contentPane.add(btnSave);
JButton btnUpdate = new JButton("UPDATE");
```

```
btnUpdate.setFont(new Font("Tahoma", Font.BOLD, 11));
btnUpdate.addActionListener(new ActionListener() {
private Connection con;
private String query;
public void actionPerformed(ActionEvent arg0) {
try{Class.forName ("oracle.jdbc.driver.OracleDriver");
con = DriverManager.getConnection("jdbc:oracle:thin:@ localhost:1521:xe", "scott", "tiger");
}catch(ClassNotFoundException e){
System.err.println("Failed to Load Driver");
e.printStackTrace();
}catch (SQLException e){
System.err.println("Unable to Connect");
e.printStackTrace();}
query = "UPDATE info SET fname = ?, mname = ?, lname = ?, address = ?, patientid = ?,
status =?, gender=? WHERE patientid = ""+ wherestring +""";
if (!fname.getText().equals("") && !mname.getText().equals("") &&
!lname.getText().equals("") && !age.getText().equals("") && !address.getText().equals("") &&
!patientid.getText().equals(""))
try {
PreparedStatement
                     preparedStmt = con.prepareStatement(query);
preparedStmt.setString(1, fname.getText());
preparedStmt.setString(2, mname.getText());
preparedStmt.setString(3, lname.getText());
preparedStmt.setString(4, age.getText());
preparedStmt.setString(5, address.getText());
preparedStmt.setString(6, patientid.getText());
preparedStmt.setString(7, combostatus.getSelectedItem().toString());
preparedStmt.setString(8, combogender.getSelectedItem().toString());
JOptionPane.showMessageDialog(null, "updated", "Title",
       JOptionPane.INFORMATION_MESSAGE);
preparedStmt.executeUpdate();
} catch (SQLException e) {
// TODO Auto-generated catch block
e.printStackTrace();
}else {
JOptionPane.showMessageDialog(null, "Empty Fields", "Title",
JOptionPane.INFORMATION_MESSAGE);
}
});
btnUpdate.setBounds(140, 404, 89, 23);
contentPane.add(btnUpdate);
JButton btnDelete = new JButton("DELETE");
btnDelete.setFont(new Font("Tahoma", Font.BOLD, 11));
```

```
btnDelete.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent arg0) {
Connection con = null;
try{Class.forName ("oracle.jdbc.driver.OracleDriver");
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","scott","tiger");
}catch(ClassNotFoundException e){
System.err.println("Failed to Load Driver");
e.printStackTrace();
}catch (SQLException e){
System.err.println("Unable to Connect");
e.printStackTrace();}
String query = "DELETE FROM info WHERE patientid = ""+ wherestring +""";
PreparedStatement preparedStmt = con.prepareStatement(query);
preparedStmt.executeUpdate();
JOptionPane.showMessageDialog(null, "Deleted", "Title",
JOptionPane.INFORMATION MESSAGE);
fname.setText("");
age.setText("");
address.setText("");
mname.setText("");
lname.setText("");
patientid.setText("");
} catch (SQLException e1) {
// TODO Auto-generated catch block
e1.printStackTrace();
}
});
btnDelete.setBounds(239, 404, 89, 23);
contentPane.add(btnDelete);
JButton btnSearch = new JButton("SEARCH");
btnSearch.addActionListener(new ActionListener() {
private Connection con;
public void actionPerformed(ActionEvent arg0) {
try{Class.forName ("oracle.jdbc.driver.OracleDriver");
con = DriverManager.getConnection("jdbc:oracle:thin:@ localhost:1521:xe","scott","tiger");
}catch(ClassNotFoundException e){
System.err.println("Failed to Load Driver");
e.printStackTrace();
}catch (SQLException e){
System.err.println("Unable to Connect");
e.printStackTrace();}
Statement statement2;
statement2 = con.createStatement();
```

```
String qry2 = "SELECT * FROM info WHERE patientid = "" + textField.getText() + "" ";
ResultSet rs = statement2.executeQuery(qry2);
while (rs.next()) {
wherestring = rs.getString("patientid");
patientid.setText(wherestring);
fname.setText(rs.getString("fname"));
age.setText(rs.getString("age"));
address.setText(rs.getString("address"));
mname.setText(rs.getString("mname"));
lname.setText(rs.getString("lname"));
combogender.setSelectedItem(rs.getString("gender"));
combostatus.setSelectedItem(rs.getString("status"));
idglobal = rs.getLong("ID");
} catch (SQLException e) {
// TODO Auto-generated catch block
e.printStackTrace();
}
});
btnSearch.setFont(new Font("Tahoma", Font.BOLD, 11));
btnSearch.setBounds(34, 87, 89, 23);
contentPane.add(btnSearch);
JButton btnBack = new JButton("BACK");
btnBack.setFont(new Font("Tahoma", Font.BOLD, 11));
btnBack.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent arg0) {
dispose();
Mainframe Mainframe = new Mainframe();
Mainframe.setVisible(true);
}
});
btnBack.setBounds(338, 404, 89, 23);
contentPane.add(btnBack);
JLabel lblNewLabel = new JLabel("Patient ID:");
lblNewLabel.setFont(new Font("Sylfaen", Font.PLAIN, 13));
lblNewLabel.setBounds(34, 157, 89, 17);
contentPane.add(lblNewLabel);
patientid = new JTextField();
patientid.setBounds(141, 150, 143, 20);
```

```
contentPane.add(patientid);
patientid.setColumns(10);
JLabel lblGender = new JLabel("Gender:");
lblGender.setBounds(34, 323, 46, 14);
contentPane.add(lblGender);
combogender = new JComboBox();
combogender.setFont(new Font("Times New Roman", Font.BOLD, 11));
combogender.setModel(new DefaultComboBoxModel(new String[] {"Male", "Female"}));
combogender.setSelectedIndex(1);
combogender.setBounds(141, 323, 142, 20);
contentPane.add(combogender);
textField = new JTextField();
textField.setBounds(34, 58, 176, 20);
contentPane.add(textField);
textField.setColumns(10);
}
```

CHAPTER -6 SOFTWARE TESTING

SOFTWARE TESTING

The aim of any software development is to create bug-free, reliable & secured system that provide solution to users requirements. The implementation of newly designed package is an important phase in adopting successful new system. The implementation of the package involves testing, user training, acceptance and change over.

Testing is an important & critical stage in software development. It accounts for the largest percentage of technical effort in the software development process. It plays a critical role in determining quality & reliability of an application. Testing phase in development life cycle validates the code against the functional specification. The feedback from test-stage will be incorporated which thereby ensures high reliability.

System Testing is an important phase. Testing represents an interesting anomaly for the software. Thus a series of testing are performed for the proposed system before the system is ready for user acceptance testing.

A good test case is one that has a high probability of finding an as undiscovered error. A successful test is one that uncovers an as undiscovered error.

Testing Objectives:

- Testing is a process of executing a program with the intent of finding an error
- A good test case is one that has a probability of finding an as yet undiscovered error
- A successful test is one that uncovers an undiscovered error

Testing Principles:

- ➤ All tests should be traceable to end user requirements
- Tests should be planned long before testing begins
- > Testing should begin on a small scale and progress towards testing in large
- > Exhaustive testing is not possible
- To be most effective testing should be conducted by a independent third party

The primary objective for test case design is to derive a set of tests that has the highest livelihood for uncovering defects in software. To accomplish this objective two different categories of test case design techniques are used. They are

- White box testing.
- ► Black box testing.

White-box Testing:

White box testing focus on the program control structure. Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been executed.

Block-box Testing:

Black box testing is designed to validate functional requirements without regard to the internal workings of a program. Black box testing mainly focuses on the information domain of the software, deriving test cases by partitioning input and output in a manner that provides through test coverage. Incorrect and missing functions, interface errors, errors in data structures, error in functional logic are the errors falling in this category.

Name of the Test Case: Admin Login Page

Test Case #	Test Case Description	Expected Results	Pass / Fail	Actual Results
01	Click on login button	It should be open login page without missing any themes	Pass	It has opened proper login page, by clicking the login button
02	Click on login button without giving username & password	It should be ask for enter username & password	Pass	It has showing error message for "enter username & password"
03	Enter username without password	It should be ask for enter password	Pass	It has displayed error message for "enter password"
04	Enter password	It should be ask	Pass	It has displayed

	without username	for enter username		error message for "enter username"
05	Enter invalid username & password	It should be show message for invalid username & password	Pass	It has displayed error message for "please enter valid username & password"
06	Enter valid username & password	It should be redirect to other page	Pass	It has redirected to other page

Name of the Test Case: Patient's Infromantion Page

Test Case #	Test Case Description	Expected Results	Pass / Fail	Actual Results
01	Click on Searchbutton	It should display the information of the particular PatientID	Pass	It has displayed the information
02	Click on save button without giving information	It should be ask for enter the details	Pass	It has showing error message for "enter Details"
03	Enter PatientID which already exits	It should be say patientId exist	Pass	It has displayed error message for "patient exist"
04	Click on update button with information.	It should be say Added	Pass	It has displayed message for "Updated"
05	Click on delete button with patientId.	It should be show message deleted	Pass	It has displayed message for "Deleted"

Name of the Test Case: Patient's Confinement Page

Test Case #	Test Case Description	Expected Results	Pass / Fail	Actual Results
01	Click on Searchbutton	It should display the information of the particular PatientID	Pass	It has displayed the information
02	Click on admit button with all information	It should show admit message	Pass	It has displayed message for "Admitted"
03	Click on save button without all information	It should be say enter details	Pass	It has displayed error message for "enter details"

Name of the Test Case: Medicine infromation Page

Test Case #	Test Case Description	Expected Results	Pass / Fail	Actual Results
01	Click on Add medicine button	It should display the message added	Pass	It has displayed message for "Added"
02	Click on Add medicine without information	It should error message	Pass	It has displayed message for "enter details"

Name of the Test Case: Patient's Billing Page

Test Case #	Test Case Description	Expected Results	Pass / Fail	Actual Results
01	Click on Searchbutton	It should display the information of the particular PatientID	Pass	It has displayed the information
02	Click on save button with all information	It should show total cash in cash textbox	Pass	It has showing total cash
03	Click on save button without all information	It should be say enter details	Pass	It has displayed error message for "enter details"

CHAPTER - 7 CONCLUSION

SCOPE

The Scope of this document includes the design details for the functioning of different modules of the system, database details are also included that is required for the storage of information. Focuses on particular authorities in a hospital.

CHAPTER – 8 REFERENCE

Bibliography:

- www.javatpoint.com/oracle-tutorial
- www.oracle-dba-online.com/sql/oracle_sql_tutorial.htm
- https://www.tutorialspoint.com/listtutorials/oracle/1
- https://www.tutorialspoint.com/java/
- www.javatpoint.com/java-tutorial
- www.guru99.com/java-tutorial.html
- zetcode.com/tutorials/javaswingtutorial/
- beginnersbook.com/2015/07/java-swing-tutorial/