ABSTRACT:

Hospital Management System is an organized computerized system designed and programmed to deal with day-to-day operations and management of the hospital activities. The program can look after inpatients, outpatients, records, database treatments, status illness, billings in the pharmacy and labs. It also maintains hospital information such as ward id, doctors in charge and department administering. The major problem for the patient nowadays to get report after consultation , many hospital managing reports in their system but it's not available to the patient when he / she is outside. In this project we are going to provide the extra facility to store the report in the database and make available from anywhere in the world.

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.

Hospital Management System is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to hospitals.

Hospital Management System is designed for multispeciality hospitals, to cover a wide range of hospital administration and management processes. It is an integrated end-to-end Hospital Management System that provides relevant information across the hospital to support effective decision making for patient care, hospital administration and critical financial accounting, in a seamless flow.

Hospital Management System is a software product suite designed to improve the quality and management of hospital management in the areas of clinical process analysis and activity-based costing.

Hospital Management System enables you to develop your organization and improve its effectiveness and quality of work. Managing the key processes efficiently is critical to the success of the hospital helps you manage your processes.

1.Problem Description

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 in convenienceand wastage of time.

Lack of immediate information storage: -

The information generated by various transactions takes time and efforts to be stored at right place.

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.

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.

Preparation of accurate and prompt reports: -

This becomes a difficult task as information is difficult to collect from various register.

1.2 Objective:-

1) **Define hospital**
2) **Recording information about the Patients that come.**
3) **Generating bills.**
4) **Recording information related to diagnosis given to Patients.**
5) **Keeping record of the Immunization provided to children/patients.**
6) **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.

2 About the system

EXISTING SYSTEM:

Hospitals currently use a manual system for the management and maintainance of critical information. The current system requires numerous paper forms, with data stores spread through out the hospital management infrastructure. Often information is incomplete or does not follow management standards. Forms are

often lost in transit between departments requiring a comprehensive auditing process to ensure that no vital information is lost. Multiple copies of the same information exist in the hospital and may lead to inconsistencies in data in various data stores.

PROPOSED SYSTEM:

The Hospital Management System is designed for any hospital to replace their existing manual paper based system. The new system is to control the information of patients. Room availability, staff and operating room schedules and patient invoices. These services are to be provided in an efficient, cost effective manner, with the goal of reducing the time and resources currently required for such tasks .

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**
- ➢ **Technical Feasibility**
- ➢ **Operational Feasibility**

2.1 Goals

- **User friendly**
- **Simple fast**
- **Low cost and effective**
- **It deals with the collection of patient’s information**
- **Diagnosis**

2.2 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,**
4) **which contains Patient information. It is destroyed after some time period to decrease the paper load in the office.**
5) **Immunization records of children are maintained in pre-formatted sheets, which are kept in a file.**
6) **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

2.3 HARDWARE AND SOFTWARE REQUIREMENTS

HARDWARE REQUIREMENTS:

- **System: intel i5 processor**
- **Hard Disk: 500 GB**
- **Monitor:15'' LED**
- **Input Devices: Keyboard, mouse**
- **RAM: 8 GB**

SOFTWARE REQUIREMENTS

- **Operating System: windows 11**
- **Coding language: HTML, CSS, JSP**
- **Tool: Netbeans 8.2**
- **Database: MYSQL**

3. MODULES:

The entire project mainly consists of 7 modules, which are

- **Admin module**
- **User module (patient)**
- **Doctor module**
- **Nurse module**
- **Pharmacist module**
- **Laboratorist module**
- **Accountant module**

3.1Admin module:

- **manage department of hospitals, user, doctor, nurse, pharmacist, laboratorist accounts.**
- **watch appointment of doctors**
- **watch transaction reports of patient payment**

- **Bed ,ward, cabin status**
- **watch blood bank report**
- **watch medicine status of hospital stock**
- **watch operation report**
- **watch birth report**
- **watch diagnosis report**
- **watch death report**

3.2 User module(patient):

- **View appointment list and status with doctors**
- **View prescription details**
- **View medication from doctor**
- **View doctor list**
- **View blood bank status**
- **View operation history**
- **View admit history. like bed, ward icu etc**
- **Manage own profile**

3.3 Doctor module:

- **Manage patient. account opening and updating**
- **Create, manage appointment with patient**
- **Create prescription for patient**
- **Provide medication for patients**
- **Issue for operation of patients and creates operation report**
- **Manage own profile**

3.4 Nurse module:

- **Manage patient. account opening and updating**
- **Allot bed, ward, cabin for patients**
- **Provide medication according to patient prescription**
- **Manage blood bank and update status**
- **Keep record of patient operation, baby born and death of patient**
- **Manage own profile**

3.5 Pharmacist module:

- **Maintain medicine**
- **Keep records of hospitals stock medicines and status**
- **Manage medicine categories**

- **Watch prescription of patient**
- **Provide medication to prescriptions**

3.6 Laboratorist module:

- **Watch prescription list**
- **Upload diagnostic report**
- **Preview of report files. like xray images, ct scan, mri reports**
- **Manage own profile**

3.7 Accountant module:

- **Create invoice for payment**
- **Order invoice to patient**
- **Take cash payment**
- **Watch payment history of patients**
- **Manage own profile**

4.CODING

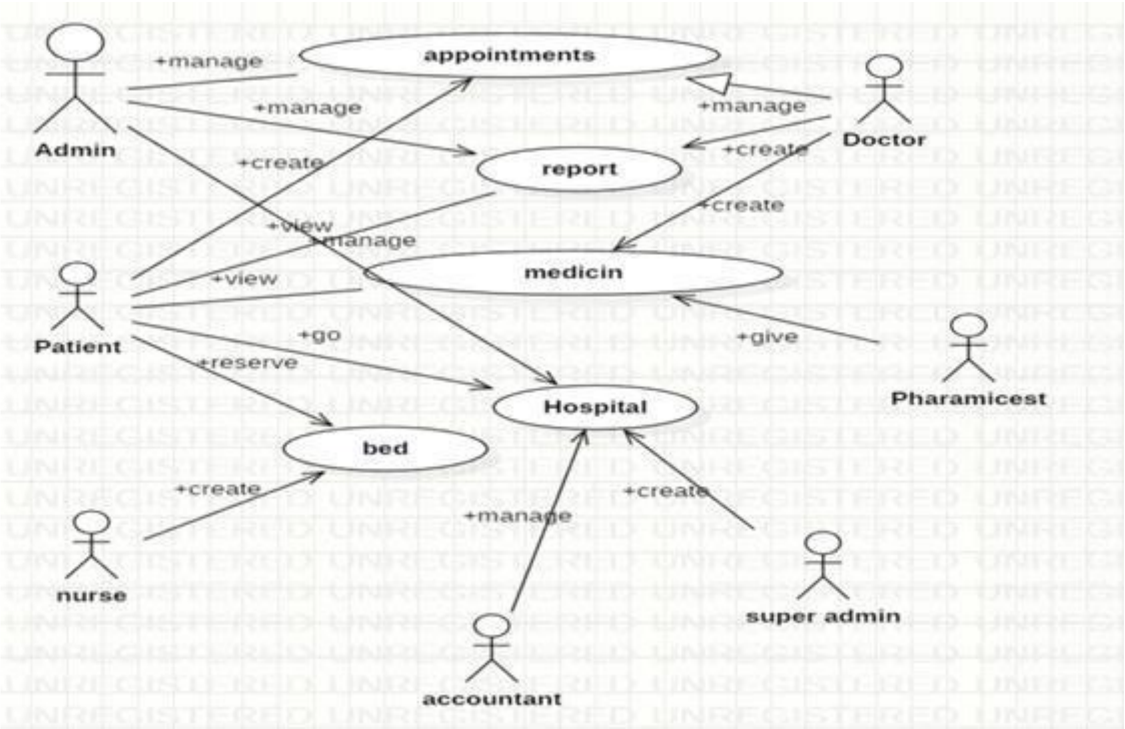
```
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.Point;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import javax.swing.JMenu;
import javax.swing.JMenuItem;
import javax.swing.table.DefaultTableModel;
import javax.swing.JMenuBar;
import javax.swing.*;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.*;

public class GUI extends JFrame implements ActionListener {
		JPanel northPanel= new JPanel();
		JPanel southPanel= new JPanel();
		JLabel welcomeText = new JLabel("Welcome to Lifeline! Pick up your choice from the
menu above, or double click on a patient's record.");
		static JTable patientsTable = new JTable() {
			private static final long serialVersionUID = 1L;
			public boolean isCellEditable(int row, int column) {
```

```
            return false;
        };
    };
static DefaultTableModel model = new DefaultTableModel(){};
JMenuBar menuBar = new JMenuBar();
JMenu fileMenu = new JMenu("File");
JMenu patientsMenu = new JMenu("Patients");
JMenu lifeLineMenu = new JMenu("Lifelines");
JMenu aboutMenu = new JMenu("About Us");
JMenu contactMenu = new JMenu("Contact Us");
JMenuItem fileExit = new JMenuItem("Exit");
JMenuItem addPatientMenu = new JMenuItem("Add patient");
JMenuItem removePatientMenu = new JMenuItem("Remove Patient");
JMenuItem updatePatientMenu = new JMenuItem("Update Patient");
JMenuItem addLifelineMenu = new JMenuItem("Add a Lifeline Record");
JMenuItem showLifelineMenu = new JMenuItem("Show the Lifeline of a user");
JMenuItem showAboutUs = new JMenuItem("Click Here");
JMenuItem showContactUs = new JMenuItem("Click Here");

public static void main(String[] args) {
            GUI graphicUserInterface = new GUI();
            patientsTable.setModel(model);
            loadData();
            }


//Graphical user interface constructor
public GUI(){
            //Adding Menus Mnemonics
            fileMenu.setMnemonic('F');
            patientsMenu.setMnemonic('P');
            lifeLineMenu.setMnemonic('L');
            aboutMenu.setMnemonic('A');
            contactMenu.setMnemonic('C');
            setLayout(new BorderLayout());
            setSize(800,600);
            setTitle("Lifeline");
            setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            setLocationRelativeTo(null);
            setVisible(true);
            setResizable(false);
            add("North", northPanel);
            add("South", southPanel);
            northPanel.add(menuBar);
            southPanel.add(welcomeText);

            //Adding the Table
            add(new JScrollPane(patientsTable));
            //End of adding the Table

            //Adding menus to menubar
            menuBar.add(fileMenu);
            menuBar.add( Box.createHorizontalStrut( 30 ) );
            menuBar.add(patientsMenu);
            menuBar.add( Box.createHorizontalStrut( 30 ) );
            menuBar.add(lifeLineMenu);
```

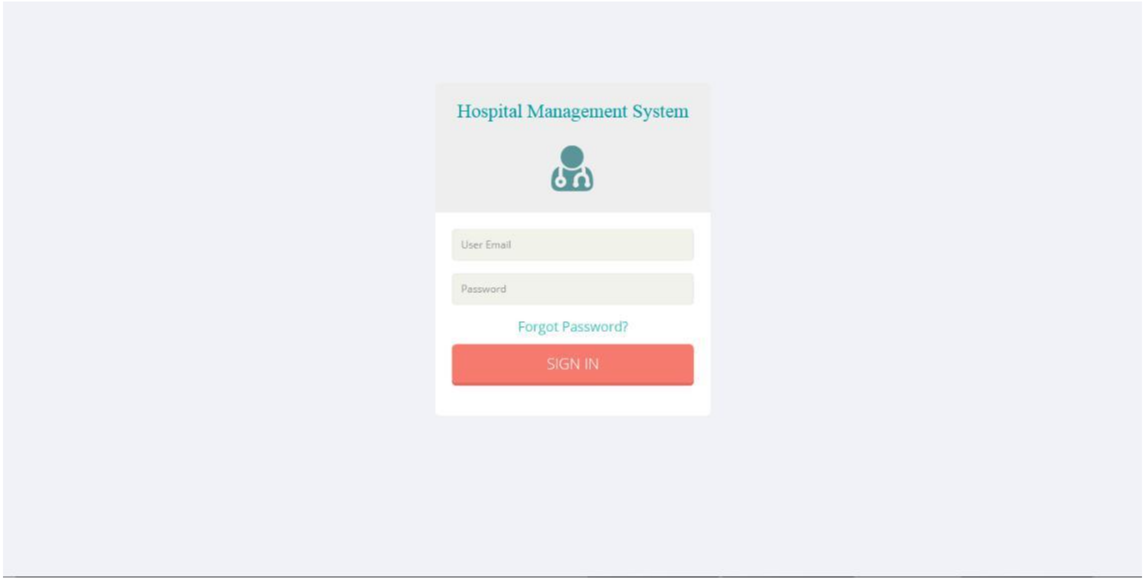
```
                menuBar.add( Box.createHorizontalStrut( 30 ) );
                menuBar.add(aboutMenu);
                menuBar.add( Box.createHorizontalStrut( 30 ) );
                menuBar.add(contactMenu);

                //Adding submenus to menus
                fileMenu.add(fileExit);
                patientsMenu.add(addPatientMenu);
                patientsMenu.add(updatePatientMenu);
                patientsMenu.add(removePatientMenu);
                lifeLineMenu.add(addLifelineMenu);
                lifeLineMenu.add(showLifelineMenu);
                aboutMenu.add(showAboutUs);
                contactMenu.add(showContactUs);

                //Adding the ActionListeners
                fileExit.addActionListener(this);
                addPatientMenu.addActionListener(this);
                updatePatientMenu.addActionListener(this);
                removePatientMenu.addActionListener(this);
                addLifelineMenu.addActionListener(this);
                showLifelineMenu.addActionListener(this);
                showAboutUs.addActionListener(this);
                showContactUs.addActionListener(this);
                aboutMenu.addActionListener(this);

                //Mouse listener for the JTable
                patientsTable.addMouseListener(new MouseAdapter() {\

                          public void mousePressed(MouseEvent me) {
                           int row= patientsTable.getSelectedRow();
                            if (me.getClickCount() == 2) {
                               Object[] possibleValues = {"Remove the Patient", "Update the
Patient", "Show Lifeline", "Add a Lifeline"};
                               Object SelectedValue= JOptionPane.showInputDialog(null,
"Choose your option", "Select an Action", JOptionPane.INFORMATION_MESSAGE, null,
possibleValues, possibleValues[0]);
                               String id = (String) patientsTable.getModel().getValueAt(row, 0);
                               if(SelectedValue=="Remove the Patient"){
                               RemovePatientForm removePatient= new
RemovePatientForm();
                                       removePatient.setIdText(id);
                               }
                               if(SelectedValue=="Update the Patient"){
                              UpdatePatientForm updatePatient = new UpdatePatientForm();
                              String name = (String) patientsTable.getModel().getValueAt(row,
1);
                               String surname = (String)
patientsTable.getModel().getValueAt(row, 2);
                               String birth = (String) patientsTable.getModel().getValueAt(row,
3);
                               String blood = (String)
patientsTable.getModel().getValueAt(row, 4);
                               String phone = (String)
patientsTable.getModel().getValueAt(row, 5);
                               String email = (String)
patientsTable.getModel().getValueAt(row, 6);
                               String country = (String)
patientsTable.getModel().getValueAt(row, 7);
                              updatePatient.setAllText(id, name, surname, birth, blood, phone,
email, country);
                               }
                               if(SelectedValue=="Show Lifeline"){
                                       ShowLifelineRecords record = new
ShowLifelineRecords();
                                       record.showLifeline(id);
                               }
                               if(SelectedValue=="Add a Lifeline"){
                                       AddLifelineRecord add = new AddLifelineRecord();
                                       add.setId(id);
                               }
                            }
                          }
                       });

                this.addWindowListener(new WindowAdapter()
                        public void windowActivated(WindowEvent e) {
                                loadData();
                           }
                        });
}
        public void actionPerformed(ActionEvent arg0) {
        if(arg0.getSource()==fileExit){
                       System.exit(DO_NOTHING_ON_CLOSE);
               }
               if(arg0.getSource()==addPatientMenu){
                       addPatientForm f1= new addPatientForm();
               }
               if(arg0.getSource()== updatePatientMenu){
                       UpdatePatientForm f2 = new UpdatePatientForm();
               }
               if(arg0.getSource()==removePatientMenu){
                       RemovePatientForm f3 = new RemovePatientForm();
               }
               if(arg0.getSource()==addLifelineMenu){
                       AddLifelineRecord f4 = new AddLifelineRecord();
               }
               if(arg0.getSource()==showLifelineMenu){
                       ShowLifelineRecords f5 = new ShowLifelineRecords();
               }
               if(arg0.getSource()== showAboutUs){
                               AboutUs f6 = new AboutUs();
               }
               if(arg0.getSource()==showContactUs){
                       ContactUs f7 = new ContactUs();
               }
        }
        public static void loadData(){
                //Connecting to the database
               final String DATABASE_URL="jdbc:mysql://localhost/javaproject";
               Connection connection = null;
               Statement statement = null;
               ResultSet resultSet = null;
               Object[] columnsName = new Object[8];
        columnsName[0] = "ID";
        columnsName[1] = "Name";
        columnsName[2] = "Surname";
        columnsName[3] = "Date of Birth (yyyy-mm-dd)";
        columnsName[4]="Blood Type";
        columnsName[5]="Phone Number";
        columnsName[6]="Email";
        columnsName[7]="Country";
        model.setColumnIdentifiers(columnsName);
        Object[] rowData = new Object[8];
    try{
            connection=DriverManager.getConnection(DATABASE_URL,"root","");
                       statement = connection.createStatement();
                       resultSet= statement.executeQuery("select * from lifeliners");
                       model.setRowCount(0);
                       while(resultSet.next()){
                                   rowData[0] = resultSet.getString("p_id");
                                   rowData[1] = resultSet.getString("p_name");
                                   rowData[2] = resultSet.getString("p_surname");
                                   rowData[3] = resultSet.getString("p_date_of_birth");
                                   rowData[4] = resultSet.getString("p_blood_type");
                                   rowData[5] = resultSet.getString("p_phone");
                                   rowData[6] = resultSet.getString("p_email");
                                   rowData[7] = resultSet.getString("p_country");
                                   model.addRow(rowData);
                       }
               }
               catch(Exception exc){
                       exc.printStackTrace();
               }
            } }
```

6. SAMPLE OUTPUT SCREENSHOTS

Use case diagram of our project:



LOGIN PAGE:


Hospital Management System

DOCTOR MODULE:

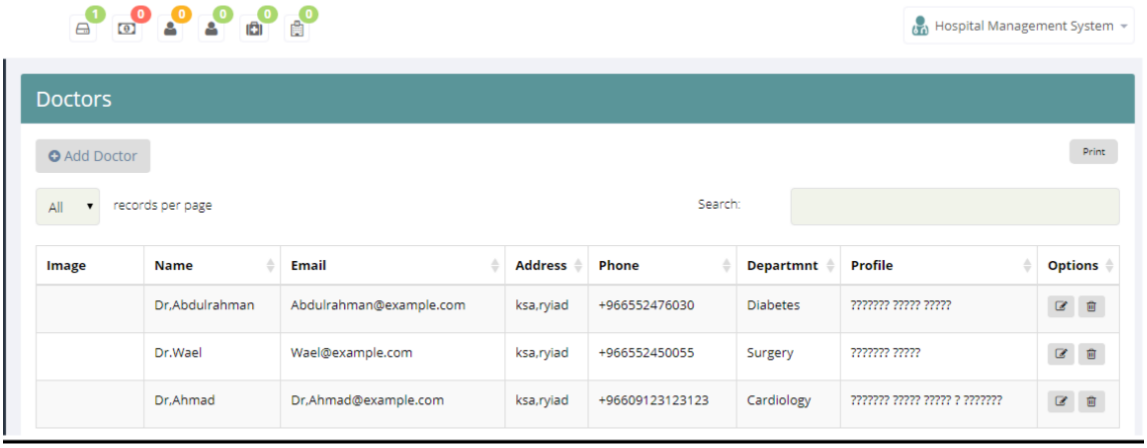
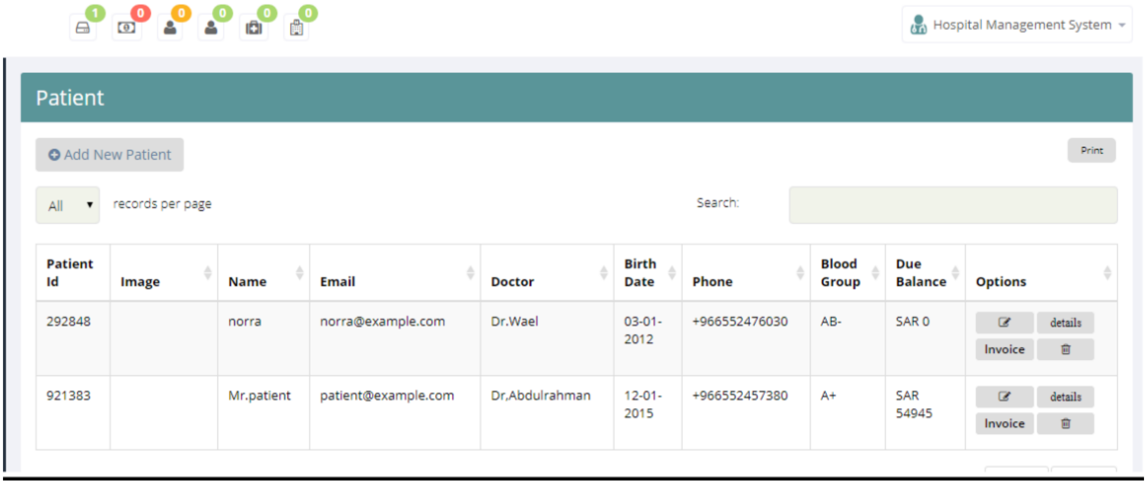


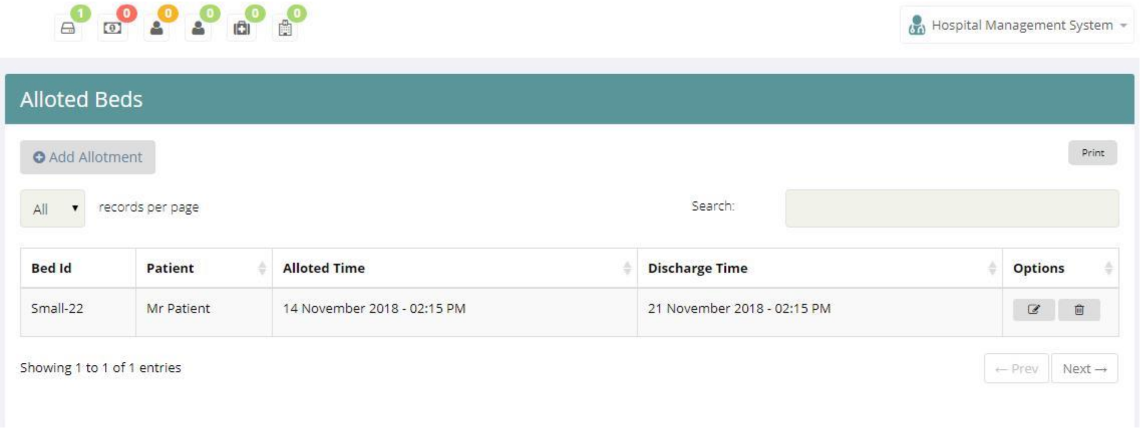
Image	Name	Email	Address	Phone	Departmnt	Profile	Options
	Dr,Abdulrahman	Abdulrahman@example.com	ksa,ryiad	+966552476030	Diabetes	??????? ????? ?????	
	Dr.Wael	Wael@example.com	ksa,ryiad	+966552450055	Surgery	??????? ?????	
	Dr,Ahmad	Dr,Ahmad@example.com	ksa,ryiad	+96609123123123	Cardiology	??????? ????? ????? ? ???????	

OPERATION DETAILS:



Patient Id	Image	Name	Email	Doctor	Birth Date	Phone	Blood Group	Due Balance	Options
292848		norra	norra@example.com	Dr.Wael	03-01-2012	+966552476030	AB-	SAR 0	details Invoice
921383		Mr.patient	patient@example.com	Dr,Abdulrahman	12-01-2015	+966552457380	A+	SAR 54945	details Invoice

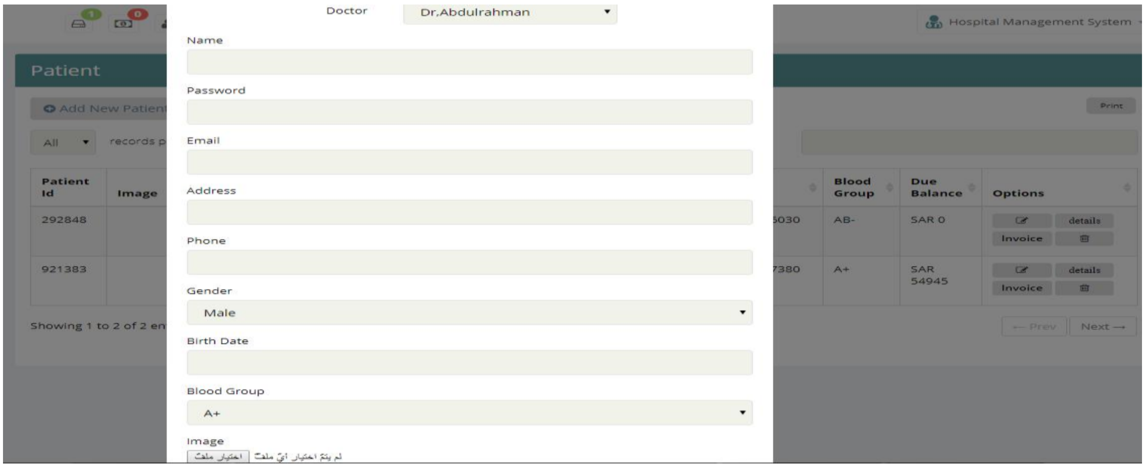
BED ALLOMENT:



Bed Id	Patient	Alloted Time	Discharge Time	Options
Small-22	Mr Patient	14 November 2018 - 02:15 PM	21 November 2018 - 02:15 PM	

Showing 1 to 1 of 1 entries

Create operation details:



Doctor: Dr.Abdulrahman

Name

Password

Email

Address

Phone

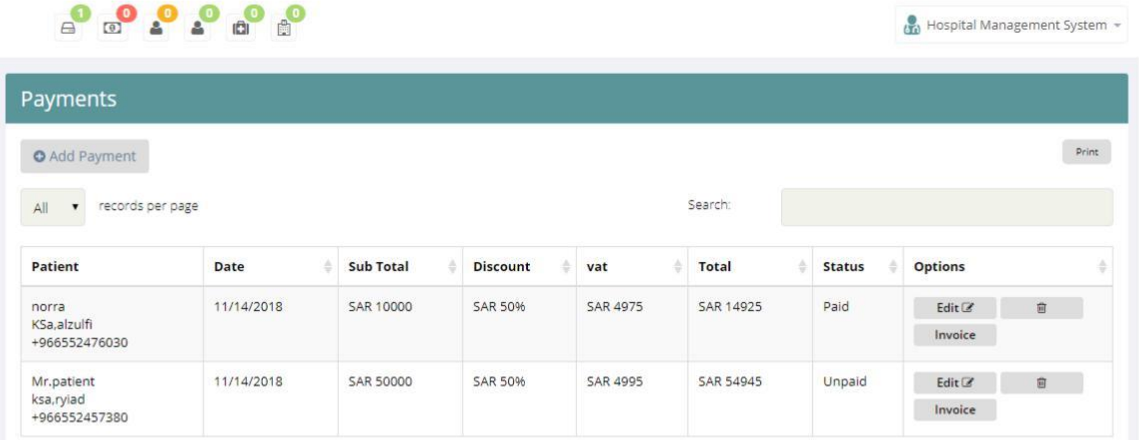
Gender: Male

Birth Date

Blood Group: A+

Image: اختيار ملف لم يتم اختيار أيّ ملف

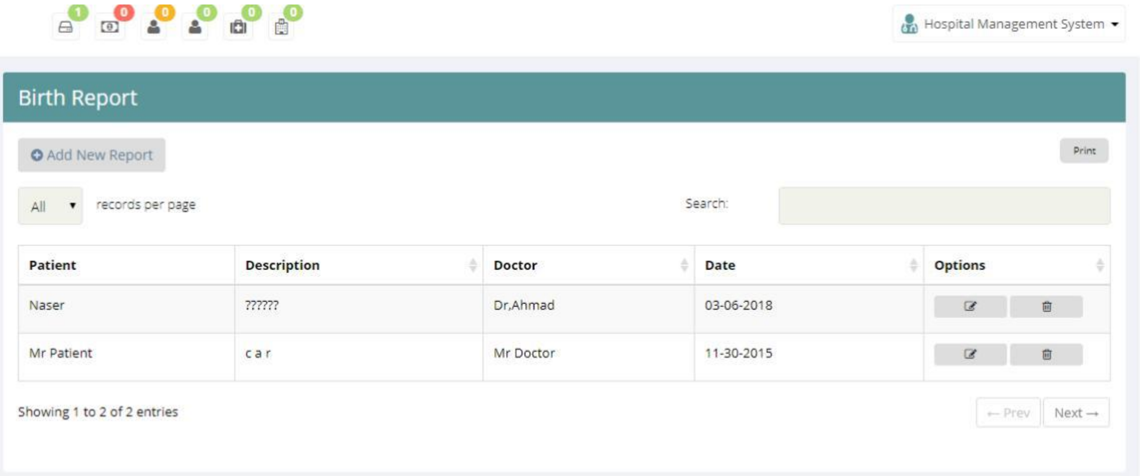
PAYMENYT DETAILS:



All records per page

Patient	Date	Sub Total	Discount	vat	Total	Status	Options
norra KSa,alzulfi +966552476030	11/14/2018	SAR 10000	SAR 50%	SAR 4975	SAR 14925	Paid	Edit Invoice
Mr.patient ksa,ryiad +966552457380	11/14/2018	SAR 50000	SAR 50%	SAR 4995	SAR 54945	Unpaid	Edit Invoice

BIRTH REPORT:



All records per page

Patient	Description	Doctor	Date	Options
Naser	??????	Dr.Ahmad	03-06-2018	
Mr Patient	c a r	Mr Doctor	11-30-2015	

Showing 1 to 2 of 2 entries

7.CONCLUSION

In this project, we developed a GUI-based project, a HOSPITAL MANAGEMENT SYSTEM Project in *Java and MySQL*. The users are able to perform the operations such as Admin Login, Patient details, Doctor details, Operation details, and Services.

A hospital management system allows for efficient storage and organization of patient information, and other relevant data. This ensures that information is readily accessible to authorized users when needed, enabling quicker and more effective decision-making. By centralizing and maintaining accurate and up-to-date client records, a hospital database management system facilitates better patient & doctors information. A well-designed database management system automates and simplifies these processes, reducing paperwork and freeing up valuable time for counselors to focus on patient details. A robust database management system can implement security measures to protect sensitive patient & doctor information from unauthorized access. For that purpose, login credentials are created for each individual access of system. In conclusion, Hospital database management system offers numerous benefits, including improved data organization and accessibility, enhanced data security and privacy. These advantages can lead to more efficient and effective counselling services and ultimately improve client outcomes.

8.REFERENCES

- ❖ https://copyassignment.com/hospital-management-system-project-in-java/
- ❖ https://github.com/yasserkabbout/Hospital-Management-System/blob/master/LifeLine%20(DEMO)/src/GUI.java
- ❖ https://www.coursehero.com/tutors-problems/Java-Programming/28032885-make-a-simple-Hospital-management-system-in-java-netbeans-preffered/
- ❖ https://www.slideshare.net/HimaniChopra/hospital-management-system-project