**HOSPITAL MANAGEMENT SYSTEM**

**Bonafide Certificate**

This work was done by Mr.G.Bharathchandra (*192111649*) from CSE department during slot A - Programming in Java (*CSA0915*) in the duration of four months from December 2023 to march 2024 under the guidance of Mr.T.VincentGnanaraj.

**Abstract**

A typical Hospital requires a management system to control its various operations such as maintaining an account of all people in its domain of services, attending to various needs of patients, and also achieving increased efficiency in the overall working of the Hospital itself. In the present time, there is a great rush in hospitals, as these have become necessities for the middle and upper class of society. Nowadays people are visiting the hospital because of this indecisive situation, even for a small issue, we need to consult a doctor. The booking is manually done using paperwork and direct human language communication by mouth to the hospital management. This delays the information in the hospital. Booking is done through phone calls or through visits to the hospital or online. The Hospital Management System aims to make simpler interactions between staff and patients. The system can be accessed by the admin and patients but the highest priority given to the admin is that they are allocated a login id and password. Hospital Management System provides various Appointments Booking Services, Managing the information of patients, Description regarding the Appointment booking.

**Keywords:** Hospital Management System, Patient Management, Online Booking, Appointment Description, Appointment Booking, Patient Records.

1. **Introduction**

A Hospital Management System (HMS) is a comprehensive software solution designed to streamline and optimize the administrative, operational, and clinical aspects of healthcare institutions. This robust system aims to enhance the efficiency of hospital management by automating various tasks such as patient registration, appointment scheduling, billing, and inventory management. Through the integration of advanced technologies, the HMS not only facilitates seamless communication among different departments but also ensures accurate and real-time data management, ultimately improving the overall quality of patient care.

Efficiently managing a multitude of tasks, the Hospital Management System contributes significantly to reducing manual errors and administrative burdens. This web-based solution offers healthcare professionals a centralized platform for accessing patient records, treatment history, and diagnostic reports, promoting quicker decision-making and enhancing the overall healthcare delivery process. With its user-friendly interface, the system empowers hospital staff to focus more on patient care while automating routine administrative tasks, thereby fostering a more effective and patient-centric healthcare environment.

In addition to optimizing internal hospital operations, the HMS plays a crucial role in improving the patient experience. Through features such as online appointment scheduling, electronic health records, and automated prescription management, patients benefit from a more convenient and accessible healthcare service. The system also enables hospitals to maintain compliance with regulatory requirements, enhance resource allocation, and adapt to the evolving landscape of healthcare technology, ensuring a sustainable and future-ready approach to hospital management.

1. **Existing System**

The current hospital management system in place relies heavily on manual processes and paper-based documentation. Patient information, appointments, and billing details are recorded manually, leading to a time-consuming and error-prone workflow. Additionally, communication between different departments is hindered as there is no centralized digital platform for data sharing. This manual system often results in inefficiencies, delayed access to critical information, and increased chances of errors in patient records and billing.

One of the primary challenges with the existing hospital management system is its lack of scalability and adaptability to modern technological advancements. The reliance on paper records not only consumes physical storage space but also poses a significant risk of data loss due to mishandling or environmental factors. Furthermore, the absence of real-time updates and integration hampers the hospital's ability to respond promptly to patient needs, impacting overall service delivery. The need for a more streamlined and automated approach becomes evident to address these limitations and enhance the overall effectiveness of hospital management.

1. **Proposed SystemTop of Form**

In response to identified challenges in the current hospital management system, our proposed solution involves the implementation of a centralized Electronic Health Record (EHR) system integrated with advanced analytics and machine learning algorithms. This comprehensive system aims to address issues related to information silos, inefficiencies in data retrieval, and manual errors in patient records. By consolidating patient data into a unified and secure digital platform, healthcare professionals can access real-time information, streamline communication, and improve decision-making. The incorporation of analytics and machine learning further enhances the system's capabilities, enabling predictive analysis for resource allocation, patient outcomes, and identifying potential healthcare trends, ultimately contributing to more effective and data-driven hospital management.

The novelty of our research lies in the integration of cutting-edge technologies to create a smart and adaptive hospital management system. By leveraging machine learning algorithms, our system goes beyond traditional data management, offering predictive insights that can aid in proactive healthcare decision-making. Additionally, the EHR system's user interface is designed to be intuitive and user-friendly, ensuring easy adoption by healthcare professionals. The research contributes to the evolving field of healthcare technology by not only addressing current challenges but also by providing a forward-looking and innovative approach to hospital management, fostering a more efficient and patient-centric healthcare ecosystem.

1. **REQUIREMENTS AND SPECIFICATION**

We need the following requirement for the developing, designing and testing the project.

**Hardware requirements:**

**CPU** : 11th Gen Intel(R) Core(TM) i5-11400F @ 2.60GHz

**RAM** : 8.0 GB or 16.0 GB

**Storage Space**: 100-150GB

**Monitor**  : LCD/LED

**Internet** : 4g/5g Internet or Wi-Fi

**Mouse**  : Wired or Wireless

**Band width** : 5mhps – 50mhps

**Software Requirements:**

**Development SDK :** Java JDK 21

**Programming IDE :** Eclipse & JDK 21

**Runtime Environment:** JRE-8

**Operating System :** Windows 11

**Packages :** MySQL, Java server pages

**Human Requirements:**

**Team size :** 1

**Supervisor :** 1

**Total :** 2

**Financial Requirements:**

**Budget:** 1000 to 15000 depending on the project implementation and quality metrics.

1. **Methodology**

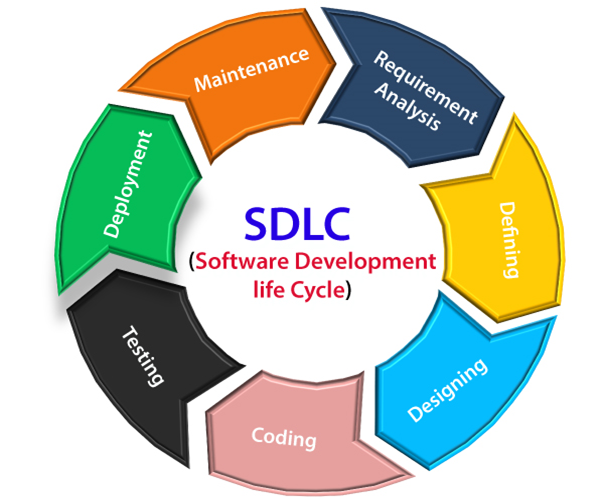
****

Figure -SOFTWARE DEVELOPMENT LIFE CYCLE

**Software Development Life Cycle** **MODEL:**

The Software Development Life Cycle (SDLC) is a systematic process that guides the development and maintenance of software. It typically involves stages such as planning, analysis, design, implementation, testing, deployment, and maintenance. This structured approach ensures a methodical progression from idea conception to the final product, fostering efficiency, quality, and effective collaboration among development teams. Each phase in the SDLC has specific objectives and deliverables, providing a framework that helps manage resources, mitigate risks, and produce a reliable and high-quality software solution.

Top of Form

**Software Development Life Cycle** **FOR HOSPITAL MANAGEMENT SYSTEM:**

* Define the scope, objectives, and requirements of the Hospital Management System (HMS) project. Identify key stakeholders, allocate resources, and establish a timeline for development.
* Conduct a thorough analysis of hospital processes, workflows, and user needs. Gather requirements for patient registration, appointment scheduling, billing, and other essential functionalities to ensure the HMS meets the specific needs of the healthcare environment.
* Create detailed system architecture, outlining the database structure, user interfaces, and integration points. Design user-friendly interfaces to enhance the user experience for hospital staff, ensuring efficient navigation and data entry.
* Develop the HMS according to the design specifications, utilizing appropriate programming languages and frameworks. Integrate necessary features such as electronic health records, pharmacy management, and reporting functionalities.
* Perform comprehensive testing, including unit testing, integration testing, and system testing, to identify and rectify any defects. Conduct user acceptance testing to ensure the HMS meets the expectations and requirements of hospital staff.

1. **ANALYSIS**

* Thoroughly analyze the current hospital management system to identify existing challenges and inefficiencies, such as manual data entry, communication gaps, or delays in patient care.
* Conduct stakeholder interviews and gather detailed requirements from hospital staff, administrators, and IT personnel to ensure a comprehensive understanding of user needs and expectations.
* Evaluate the technical, operational, and economic feasibility of the proposed Hospital Management System project, considering factors such as technology compatibility, budget constraints, and potential benefits.

**1. DATA ANALYSIS**

* Gather relevant data from various sources within the hospital, including patient records, financial transactions, and operational data, ensuring a comprehensive dataset for analysis.
* Scrub the collected data to eliminate inconsistencies, errors, or missing information, ensuring the accuracy and reliability of the dataset for subsequent analysis.
* Integrate data from different hospital departments and systems to create a unified dataset, enabling a holistic view of hospital operations for comprehensive analysis.
* Perform descriptive statistical analysis to understand the basic characteristics of the data, providing insights into trends, patterns, and key metrics within the hospital management system.

**2. SOFTWARE ANALYSIS**

* Identify and gather detailed requirements for the hospital management system, including user needs, workflow specifications, and regulatory compliance.
* Conduct a feasibility study to assess the practicality, economic viability, and technical feasibility of implementing the hospital management system within the existing hospital infrastructure.
* Identify potential risks associated with the development and implementation of the software, such as data security, system integration challenges, or resistance from users, and develop mitigation strategies.
* Clearly define both functional requirements (features and capabilities) and non-functional requirements (performance, scalability, and security) for the hospital management system.
* Develop a comprehensive system design, outlining the architecture, database schema, and user interfaces to ensure alignment with the identified requirements and specifications.

**3. HARDWARE ANALYSIS**

* Hospital Management Systems require robust hardware to support their operations. Hardware analysis involves evaluating server performance, storage capacity, and network infrastructure to ensure seamless data flow and system responsiveness.
* Scalability of the hardware is crucial to accommodate the growing volume of data and users. Analyzing the load handling capacity and implementing redundancy measures helps prevent system failures and downtime.
* Security measures such as firewalls, encryption protocols, and regular hardware maintenance are integral to safeguard against potential cyber threats and ensure the continuous availability of the Hospital Management System.

**4. HUMAN RESOURCES**

We have gained enough experience in Java Programming to complete this Project. The Supervisor is capable of training this project.

**5. EXTERNAL RESOURCES**

Onsite training, Internships, Workshops, and Funded Projects could be very helpful resources for this project.

1. **MAINTANANCE**

The software is under good maintenance

1. **DESIGN**
2. **ARCHITECTURAL DIAGRAM FOR HOSPITAL MANAGEMENT**

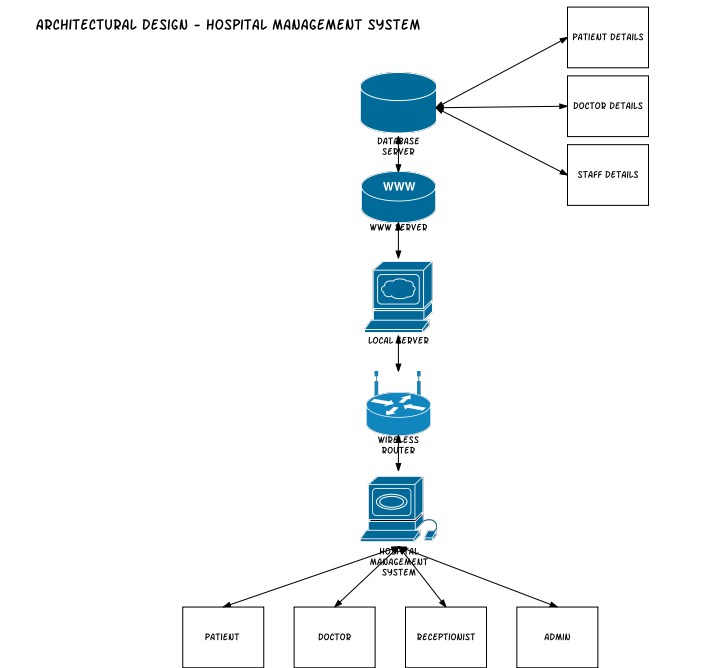
****

Figure - ARCHITECTURL DIAGRAM FOR HOSPITAL MANAGEMENT SYSTEM

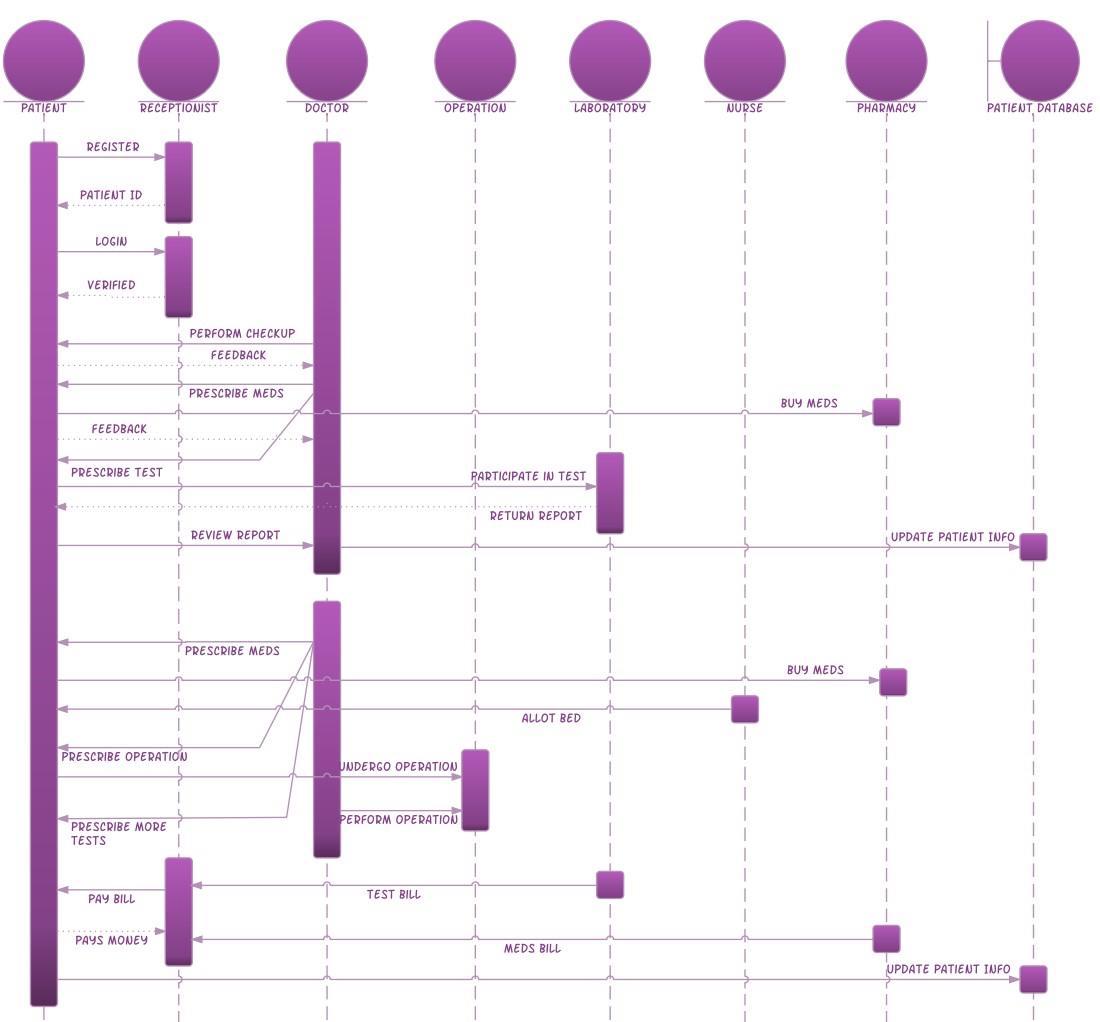
1. SEQUENCE DIAGRAM FOR HOSPITALMANAGEMENT

Figure -SEQUENCE DIAGRAM FOR HOSPITAL MANAGEMENT SYSTEM

* For designing this project we have developed architecture diagram and sequence diagram.

**FROUNTEND DESIGN:** Java server pages

**BACKEND DESIGN:** MySQL

1. **CODING**

**Syntax:**

* Public class My Class { } – This is a class declaration. My Class is the name of the class.
* Public static void main (String [] args) { } – This is the main method that gets executed when you run your Java program.
* Int myVar = 5; – This is a variable declaration. My Var is a variable of type int and is assigned the value 5.
* Final double PI = 3.14; – This is a constant declaration. PI is a constant of type double and is assigned the value 3.14.
* If (condition) { } – This is an if statement. The code inside the braces {} is executed if the condition is true.
* If (condition) {} else {} – This is an if-else statement. If the condition is true, the code in the first block is executed; otherwise, the code in the else block is executed.
* Switch (variable) {case value: break; default: break ;} – This is a switch statement. It allows a variable to be tested for equality against a list of values.
* For (initialization; condition; increment) {} – This is a for loop. It is used to repeatedly execute a block of code until a certain condition is met.
* While (condition) {} – This is a while loop. It repeatedly executes a block of code as long as a certain condition is true.
* Do {} while (condition); – This is a do-while loop. It is similar to a while loop, but the condition is tested after the execution of the block of code.
* Int [] my Array = new int [10]; – This is an array declaration. My Array is an array of int type with a size of 10.
* Public return Type function Name (parameters) {} – This is a function declaration. Function Name is the name of the function, return Type is the data type of the value the function returns, and parameters are input to the function.
* Try {} catch (Exception Type e) {} – This is a try-catch block. It is used to handle exceptions and errors that occur in a block of code.
* My Class obj = new My Class (); – This is how to create an object. Obj is an object of class My Class.
* Obj.memberName; – This is how to access an object’s members. Member Name is the name of a member (variable or method) of the object obj.

**Source code:**

1. **To create a login module of Hospital management system here is the source code:**

import java.awt.HeadlessException;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import javax.swing.JOptionPane;

import java.sql.\*;

public class login extends javax.swing.JFrame {

public login() {

initComponents();

}

@SuppressWarnings("unchecked")

private void initComponents() {

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

txtUsername = new javax.swing.JTextField();

jButton1 = new javax.swing.JButton();

jPassword = new javax.swing.JPasswordField();

jLabel1.setFont(new java.awt.Font("Arial", 0, 18));

jLabel1.setText("Username");

jLabel2.setFont(new java.awt.Font("Arial", 0, 18));

jLabel2.setText("Password");

jButton1.setText("OK");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(27, 27, 27)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel1)

.addComponent(jLabel2))

.addGap(35, 35, 35)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(txtUsername)

.addComponent(jPassword, javax.swing.GroupLayout.DEFAULT\_SIZE, 122, Short.MAX\_VALUE))

.addContainerGap(61, Short.MAX\_VALUE))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton1, javax.swing.GroupLayout.PREFERRED\_SIZE, 95, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(24, 24, 24))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(62, 62, 62)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel1)

.addComponent(txtUsername, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(29, 29, 29)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel2)

.addComponent(jPassword, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(44, 44, 44)

.addComponent(jButton1)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

pack();

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

Connection con=null;

ResultSet rs=null;

PreparedStatement pst=null;

if (txtUsername.getText().equals("")) {

JOptionPane.showMessageDialog( this, "Please enter user name");

return;

}

if (jPassword.getText().equals("")) {

JOptionPane.showMessageDialog( this, "Please enter password");

return;

}

con=Connect.ConnectDB();

String sq1= "select \* from users where user\_name= '" + txtUsername.getText() + "' and password ='" + jPassword.getText() + "'";

try{

pst=con.prepareStatement(sq1);

rs= pst.executeQuery();

if (rs.next()){

this.hide();

Main frm=new Main();

frm.setVisible(true);

}

else{

JOptionPane.showMessageDialog(null, "Login Failed..Try again !");

}

}catch(SQLException | HeadlessException e){

JOptionPane.showMessageDialog(null, e);

// TODO add your handling code here:

}

}

public static void main(String args[]) {

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Metal".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new login().setVisible(true);

}

});

}

private javax.swing.JButton jButton1;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JPasswordField jPassword;

private javax.swing.JTextField txtUsername;

}

1. **To create a Patient module of Hospital management system here is the source code:**

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import javax.swing.JOptionPane;

import net.proteanit.sql.DbUtils;

public class PatientRec extends javax.swing.JFrame {

Connection con=null;

ResultSet rs=null;

PreparedStatement pst=null;

public PatientRec() {

initComponents();

con= Connect.ConnectDB();

Get\_Data();

}

private void Get\_Data(){

String sql="select PatientID as 'Patient ID', PatientName as 'Patient Name',FatherName as 'Father Name',Address,ContactNo as 'Contact No',Email as 'Email ID',Age,Gen as 'Gender',BG as 'Blood Group',Remarks from Patientregistration";

try{

pst=con.prepareStatement(sql);

rs= pst.executeQuery();

jTable1.setModel(DbUtils.resultSetToTableModel(rs));

}catch(Exception e){

JOptionPane.showMessageDialog(null, e);

}

}

@SuppressWarnings("unchecked")

private void initComponents() {

jScrollPane1 = new javax.swing.JScrollPane();

jTable1 = new javax.swing.JTable();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE);

addWindowListener(new java.awt.event.WindowAdapter() {

public void windowClosing(java.awt.event.WindowEvent evt) {

formWindowClosing(evt);

}

});

jTable1.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

jTable1MouseClicked(evt);

}

});

jScrollPane1.setViewportView(jTable1);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jScrollPane1, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT\_SIZE, 612, Short.MAX\_VALUE)

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT\_SIZE, 388, Short.MAX\_VALUE)

);

pack();

}

private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {

//GEN-FIRST:event\_jTable1MouseClicked

try{

con=Connect.ConnectDB();

int row= jTable1.getSelectedRow();

String table\_click= jTable1.getModel().getValueAt(row, 0).toString();

String sql= "select \* from PatientRegistration where PatientID = '" + table\_click + "'";

pst=con.prepareStatement(sql);

rs= pst.executeQuery();

if(rs.next()){

this.hide();

Registration frm = new Registration();

frm.setVisible(true);

String add1=rs.getString("PatientID");

frm.txtId.setText(add1);

String add2=rs.getString("Patientname");

frm.txtName.setText(add2);

String add3=rs.getString("Fathername");

frm.txtFname.setText(add3);

String add5=rs.getString("Email");

frm.txtEmail.setText(add5);

int add6 = rs.getInt("Age");

String add= Integer.toString(add6);

frm.txtAge.setText(add);

String add7=rs.getString("Remarks");

frm.txtInfo.setText(add7);

String add9=rs.getString("BG");

frm.cmbBG.setSelectedItem(add9);

String add11=rs.getString("Gen");

frm.cmbGender.setSelectedItem(add11);

String add15=rs.getString("Address");

frm.txtAdd.setText(add15);

String add16=rs.getString("ContactNo");

frm.txtContact.setText(add16);

frm.btnUpdate.setEnabled(true);

frm.btnDelete.setEnabled(true);

frm.btnSave.setEnabled(false);

}

}catch(Exception ex){

JOptionPane.showMessageDialog(this,ex);

}

}//GEN-LAST:event\_jTable1MouseClicked

private void formWindowClosing(java.awt.event.WindowEvent evt) {//GEN-FIRST:event\_formWindowClosing

}

public static void main(String args[]) {

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(PatientRec.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(PatientRec.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(PatientRec.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(PatientRec.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new PatientRec().setVisible(true);

}

});

}

}

1. **To create a Doctor module of Hospital management system here is the source code:**

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import javax.swing.JOptionPane;

import net.proteanit.sql.DbUtils;

public class DocRec extends javax.swing.JFrame {

Connection con=null;

ResultSet rs=null;

PreparedStatement pst=null;

public DocRec() {

initComponents();

con= Connect.ConnectDB();

Get\_Data();

}

private void Get\_Data(){

String sql="select DoctorID as 'Doctor ID', DoctorName as 'Doctor Name',FatherName as 'Father Name',Address,ContacNo as 'Contact No',Email as 'Email ID',Qualifications,Gender,BloodGroup as 'Blood Group',DateOfJoining as 'Joining Date' from Doctor order by DoctorName";

try{

pst=con.prepareStatement(sql);

rs= pst.executeQuery();

jTable1.setModel(DbUtils.resultSetToTableModel(rs));

}catch(Exception e){

JOptionPane.showMessageDialog(null, e);

}

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents

private void initComponents() {

jScrollPane1 = new javax.swing.JScrollPane();

jTable1 = new javax.swing.JTable();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE);

addWindowListener(new java.awt.event.WindowAdapter() {

public void windowClosing(java.awt.event.WindowEvent evt) {

formWindowClosing(evt);

}

});

jTable1.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

jTable1MouseClicked(evt);

}

});

jScrollPane1.setViewportView(jTable1);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT\_SIZE, 612, Short.MAX\_VALUE)

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT\_SIZE, 345, Short.MAX\_VALUE)

);

pack();

}

private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {

//GEN-FIRST:event\_jTable1MouseClicked

try{

con=Connect.ConnectDB();

int row= jTable1.getSelectedRow();

String table\_click= jTable1.getModel().getValueAt(row, 0).toString();

String sql= "select \* from Doctor where DoctorID = '" + table\_click + "'";

pst=con.prepareStatement(sql);

rs= pst.executeQuery();

if(rs.next()){

this.hide();

Entry frm = new Entry();

frm.setVisible(true);

String add1=rs.getString("DoctorID");

frm.txtId.setText(add1);

String add2=rs.getString("Doctorname");

frm.txtName.setText(add2);

String add3=rs.getString("Fathername");

frm.txtFname.setText(add3);

String add5=rs.getString("Email");

frm.txtE.setText(add5);

String add6=rs.getString("Qualifications");

frm.txtQ.setText(add6);

String add9=rs.getString("BloodGroup");

frm.cmbB.setSelectedItem(add9);

String add11=rs.getString("Gender");

frm.cmbG.setSelectedItem(add11);

String add14=rs.getString("DateOfJoining");

frm.txtD.setText(add14);

String add15=rs.getString("Address");

frm.txtAd.setText(add15);

String add16=rs.getString("ContacNo");

frm.txtC.setText(add16);

frm.btnUpdate.setEnabled(true);

frm.btnDelete.setEnabled(true);

frm.btnSave.setEnabled(false);

}

}catch(Exception ex){

JOptionPane.showMessageDialog(this,ex);

}

}//GEN-LAST:event\_jTable1MouseClicked

private void formWindowClosing(java.awt.event.WindowEvent evt) {//GEN-FIRST:event\_formWindowClosing

}

public static void main(String args[]) {

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(DocRec.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(DocRec.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(DocRec.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(DocRec.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

java.awt.EventQueue.invokeLater(new Runnable() {

@Override

public void run() {

new DocRec().setVisible(true);

}

});

}

}

1. **TESTING**
2. **Unit Testing**

* Unit tests focus on individual components or functions to ensure they work as expected.
* In this code, you might create unit tests for methods like update in the Doctor Module and patient module.
* Since the code doesn't have many standalone methods or functions, unit testing is somewhat limited.

1. **Integration Testing**

* Integration tests verify that different components of the system work together correctly.
* For this code, you might want to simulate the entire flow of adding a patient, updating medical bills, and displaying patients.

1. **BLACKBOX TESTING**

* Evaluates the overall functionality of the hospital management system without considering its internal code structure.
* Focuses on assessing the system's inputs, outputs, and user interfaces to ensure they meet specified requirements.
* Identifies potential issues related to data validation, user permissions, and system integration by treating the software as a closed box.

1. **WHITE BOX TESTING**

* Examines the internal logic, code paths, and data structures of the hospital management system.
* Aims to uncover flaws in the system's design and implementation by scrutinizing the source code.
* Identifies vulnerabilities such as coding errors, security loopholes, and inefficiencies within the software's architecture.

1. **Module Testing**

* Ensure that each module within the hospital management system, such as patient registration, appointment scheduling, billing, and others, is tested individually. Verify that each module functions correctly and meets the specified requirements.

1. **IMPLEMENTATION**

The implementation of the hospital management system involved the deployment of a comprehensive and integrated software solution tailored to streamline healthcare processes. It encompassed the development and integration of modules for patient registration, appointment scheduling, billing, and other critical functionalities. The implementation process focused on ensuring user-friendly interfaces and adherence to regulatory standards for data security and privacy. The successful deployment required collaboration with healthcare professionals to tailor the system to their specific needs, ensuring seamless adoption and functionality within the hospital's existing infrastructure. Regular testing, training sessions, and feedback loops were essential components of the implementation strategy, resulting in a well-optimized and efficient hospital management system that enhances overall healthcare service delivery.Top of Form

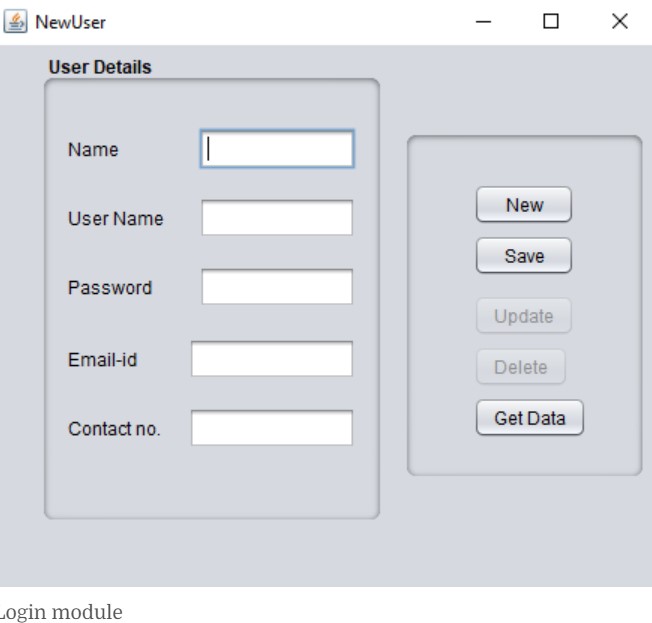


Figure 4-Login module

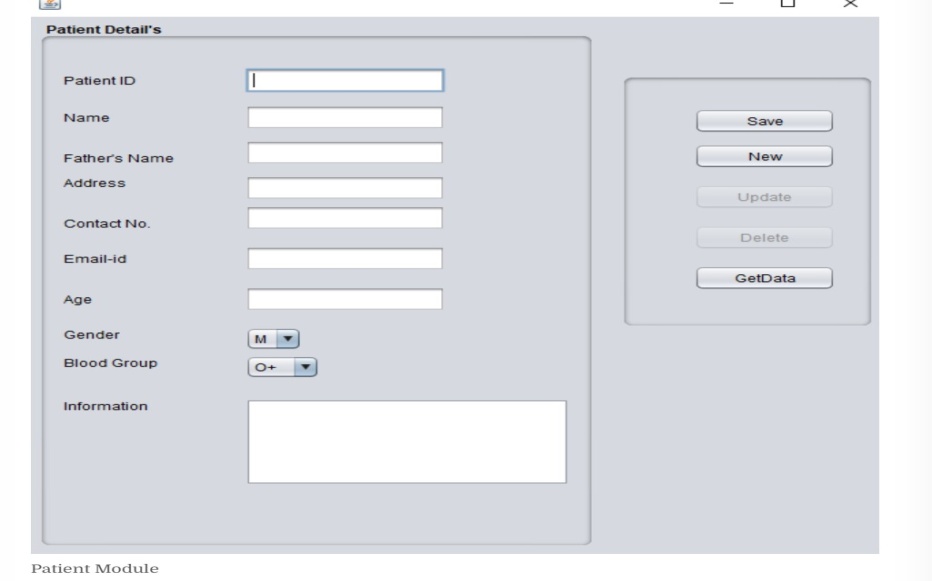
****

Figure 5-Patient Module

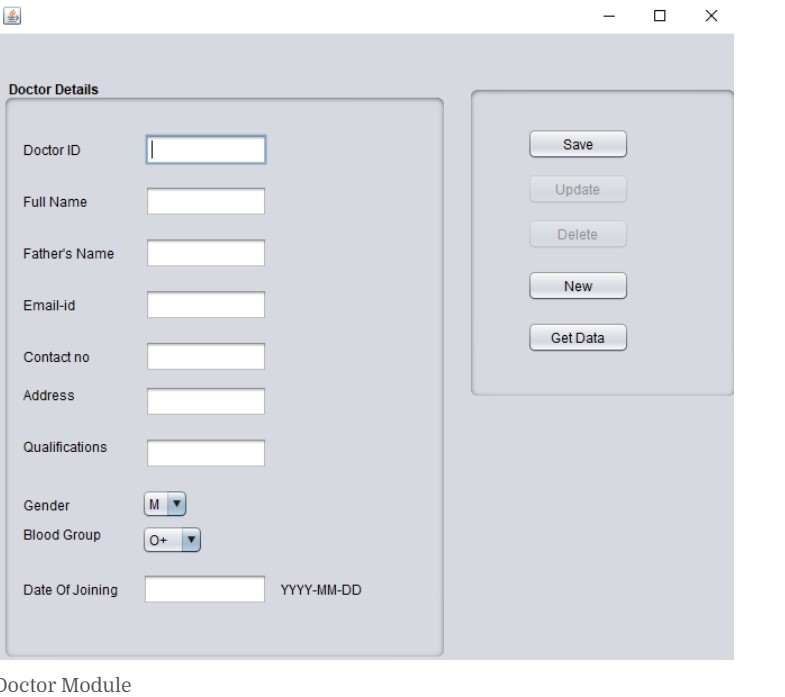


Figure 6-Doctors Module

1. **MAINTENANCE**

This project requires maintenance according to software update and hardware changes.

1. **FUTURE ENHANCEMENT**

* In Future work, This application to develop a cross platforms like IOS, etc.
* In adding the more features of hospital management system to develop access with user’s flexibility.
* To authenticate the users based on the system users list which is maintained by the operating system
* To restrict the usage of all files by the users based on their privileges on the system

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. The limited time and resources have restricted us to incorporate, in this project, only main activities that are performed in a Hospital Management System, but utmost care has been taken to make the system efficient and user friendly.

Most of the analysis and interpretations, made for this report, are based on secondary data obtained. This data could have some inherent mistakes and errors. Finally, although due care has been taken those can be typing and compilation errors in the report itself. The tasks specified were not well defined because nothing was mentioned regarding validations in the project. Though we gave maximum effort to check the software. But it in no way alters the ultimate aim of the project and because it's highly USER FRIENDLY, it would be the choice of all kinds of personnel.

1. **CONCLUSION**

The hospital management system project exhibits technical excellence by seamlessly integrating modules such as patient registration, appointment scheduling, and billing. Challenges faced during development included ensuring the security of sensitive patient data and optimizing the system's performance under varying workloads. Additionally, the implementation of necessary regulatory compliance measures presented complexities. Despite these challenges, the project successfully navigated through deviations by incorporating rigorous testing methodologies, addressing unforeseen issues, and maintaining a focus on user acceptance. The resulting hospital management system stands as a testament to resilience in overcoming challenges and delivering a robust solution for efficient healthcare management

1. **ACKNOWLEDGEMENT**

We thank Oracle for providing java software as free of cost for developing the project. We thank course faculty who guided our project in a successful manner to complete. We thank mentor for moral support, supervisor for handling technical support. We also thank the management for providing required accommodations.

1. **REFERENCES**

# MS K.Ramani, “HOSPITAL MANAGEMENT: TEXT AND CASES”, 1 January 2013, “<https://www.amazon.in/HOSPITAL-MANAGEMENT-TEXT-CASES-Ramani/dp/8131794016>”

# Ms Manisha saxena, “ Hospital Management, Volume 1 (4th reprint)”, 1 January 2019, “<https://www.amazon.in/Hospital-Management-Vol-M-Saxena/dp/8123923015/ref=pd_lpo_sccl_1/258-7607998-9930052?pd_rd_w=FaV0c&content-id=amzn1.sym.646ecf30-bd56-461f-a4dc-7f0c041d20b3&pf_rd_p=646ecf30-bd56-461f-a4dc-7f0c041d20b3&pf_rd_r=BH4EGJX55C64BNSVN1YY&pd_rd_wg=f7xqI&pd_rd_r=73367ad7-1102-4f5c-b489-3823ff848373&pd_rd_i=8123923015&psc=1>”

# Mr.Joydeep Das Gupta, “Hospital Administration and Management: A Comprehensive Guide”, 11 January 2023, “<https://www.amazon.in/Hospital-Administration-Management-Comprehensive-Guide/dp/9356960801/ref=pd_lpo_sccl_2/258-7607998-9930052?pd_rd_w=FaV0c&content-id=amzn1.sym.646ecf30-bd56-461f-a4dc-7f0c041d20b3&pf_rd_p=646ecf30-bd56-461f-a4dc-7f0c041d20b3&pf_rd_r=BH4EGJX55C64BNSVN1YY&pd_rd_wg=f7xqI&pd_rd_r=73367ad7-1102-4f5c-b489-3823ff848373&pd_rd_i=9356960801&psc=1>”

# Mr.Michael S. Leeman, Mr.Linda Collins, “Journal of Hospital Management and Health Policy”, 30 December 2023, “<https://jhmhp.amegroups.org/article/view/8441/html>”

# K.Nishanthan, S.Mathyvathana, R.Priyanthi, “The Hospital Management System”, October 2022,“<https://www.researchgate.net/publication/367460409_The_Hospital_Management_System>”

# Najeh LAKHOUA, “REVIEW ON SMART HOSPITAL MANAGEMENT SYSTEM TECHNOLOGIES”, “<file:///C:/Users/User/Downloads/17.%20Najeh%20LAKHOUA%20-REVIEW%20ON%20SMART%20HOSPITAL%20MANAGEMENT%20SYSTEM%20TECHNOLOGIES.pdf>”

# Digvijay H. Gadhari, Yadnyesh P. Kadam, Prof. Parineeta Suman, “HOSPITAL MANAGEMENT SYSTEM”, 11FEB 2016, “<https://ijream.org/papers/INJRV01I11006.pdf>”

# Esha Bisht, Monika Chaudhary, Akansha Rathi, “Hospital Management System Software Engineering Project Report”, “<https://mscw.ac.in/NAAC/Criteria1/Samples-of-ProjectWork_Fieldwork/Computer_Science/software_Engineering/Software%20Engineering/Hospital%20Managment%20Project%20SE-converted.pdf>”

1. [ANUJ KUMAR](https://phpgurukul.com/author/anujk305/), “Hospital Management System In PHP”, 08 Nov 2022 ,“<https://phpgurukul.com/hospital-management-system-in-php/>”

# **Apoorv chowdhry, “Hospital Management System SQL”, Dec 17 2023, “**<https://medium.com/@apoorvchowdhry55/hospital-management-system-f21b978a1b8c>**”**