

## **Specialized Hospital Booking System**

### **A MINI-PROJECT REPORT**

*Submitted by*

**GOWTHAAM B** **240701153**

**GOWTHAM S** **240701157**

*in partial fulfillment of the award of the degree*

*of*

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**



**RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI**

**An Autonomous Institute**

**CHENNAI**

**NOVEMBER 2025**

## BONAFIDE CERTIFICATE

Certified that this project “**Specialized Hospital Booking System**” is the bonafide work of “**GOWTHAAM B,GOWTHAM S**” who carried out the project work under my supervision.

### **SIGNATURE**

**Dr.ANITHA.M**

**ASSISTANT PROFESSOR**

Dept. of Computer Science and Engg,  
Rajalakshmi Engineering College  
Chennai

This mini project report is submitted for the viva voce examination to be held on

---

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

## ABSTRACT

Specialized Hospital Management is a Java Swing and MySQL based hospital management front-end that streamlines patient intake, emergency response, ambulance booking, hospital selection, doctor assignment, ward allocation, food service selection, billing, and review capture. The user-centered interface provides secure sign-up/sign-in (with placeholders for third-party OAuth), stepwise patient verification, fast emergency handling with ambulance booking, and clear transaction breakdowns. The system stores all records in a MySQL backend for retrieval and audit. Designed for a mini-project, it demonstrates full-stack integration, UI workflows, data validation, and simulated notification flows for real-world hospital operation.

## **ACKNOWLEDGEMENT**

We express our sincere thanks to our beloved and honorable chairman **MR. S. MEGANATHAN** and the chairperson **DR. M.THANGAM MEGANATHAN** for their timely support and encouragement.

We are greatly indebted to our respected and honorable principal **Dr. S.N. MURUGESAN** for his able support and guidance.

No words of gratitude will suffice for the unquestioning support extended to us by our Head Of The Department **Dr. E.M. MALATHY** and our Deputy Head Of The Department **Dr. J. MANORANJINI** for being ever supporting force during our project work

We also extend our sincere and hearty thanks to our internal guide **Dr.ANITHA.M**, for his valuable guidance and motivation during the completion of this project.

Our sincere thanks to our family members, friends and other staff members of computer science engineering.

**1. GOWTHAAM B**

**2. GOWTHAM S**

**TABLE OF CONTENTS**

<b>CHAPTER NO.</b>	<b>TITLE</b>	<b>PAGE NO</b>
	<b>ABSTRACT</b>	<b>iv</b>
<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
1.1	INTRODUCTION	8
1.2	SCOPE OF THE WORK	8
1.3	PROBLEM STATEMENT	8
1.4	AIM AND OBJECTIVES OF THE PROJECT	8
<b>2</b>	<b>SYSTEM SPECIFICATIONS</b>	<b>9</b>
2.1	HARDWARE SPECIFICATIONS	9
2.2	SOFTWARE SPECIFICATIONS	9
<b>3</b>	<b>MODULE DESCRIPTION</b>	<b>10</b>
<b>4</b>	<b>CODING</b>	<b>11</b>
<b>5</b>	<b>SCREENSHOTS</b>	<b>16</b>
<b>6</b>	<b>CONCLUSION AND FUTURE ENHANCEMENT</b>	<b>18</b>
	<b>REFERENCES</b>	<b>19</b>

## **LIST OF FIGURES**

<b>FIGURE NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
<b>5.1</b>	<b>INTRODUCTION PAGE</b>	<b>15</b>
<b>5.2</b>	<b>CUSTOMER DETAILS</b>	<b>15</b>
<b>5.3</b>	<b>BOOKING LOG</b>	<b>16</b>
<b>5.4</b>	<b>BOOKING CREATION</b>	<b>16</b>
<b>5.5</b>	<b>DELETION OF BOOKING</b>	<b>17</b>
<b>5.6</b>	<b>DATABASE CREATION</b>	<b>17</b>

# CHAPTER 1

## INTRODUCTION

### 1.1 INTRODUCTION

The Specialized Hospital Management System is a Java Swing and MySQL-based project that automates hospital operations like patient registration, ambulance booking, doctor assignment, room and food management, billing, and report generation through an interactive, user-friendly interface.

### 1.2 SCOPE OF THE WORK

The scope of this project is to digitalize hospital processes by managing patient records, doctor details, ambulance services, room and food preferences, and billing efficiently, ensuring smooth, secure, and user-friendly hospital management operations.

### 1.3 PROBLEM STATEMENT

Manual hospital management often leads to data errors, delays, and poor coordination. This project aims to solve these issues by developing an automated system that efficiently handles patient details, hospital services, billing, and communication through a centralized digital platform.

### 1.4 AIM AND OBJECTIVES OF THE PROJECT

The aim of this project is to develop an automated hospital management system that simplifies and digitalizes hospital

**operations. Its objectives include managing patient records, ambulance services, doctor details, billing, and communication efficiently using a secure and user-friendly interface.**

## **CHAPTER 2**

### **SYSTEM SPECIFICATIONS**

#### **2.1 HARDWARE SPECIFICATIONS**

Processor	:	Intel i3
Memory Size	:	8GB (Minimum)
HDD	:	1 TB (Minimum)

#### **2.2 SOFTWARE SPECIFICATIONS**

Operating System	:	WINDOWS 10
Front – End	:	Java Swing
Back - End	:	MySql
Language	:	Java Swing,MySQL

## CHAPTER 3

### MODULE DESCRIPTION

This application consists of two modules. When the program runs, it will ask for a confirmation to the login window. The person who interacts can login as an Administrator or as a User. The description of the modules are as follows:

#### 1. Admin login

When the person who interacts tries to login as Admin then he needs to login with his username and password. The administrator only has the power to change and manipulate the data in the database.

#### 2. User login

When the person tries to login as a user then he/she will be prompted to enter the number of symptoms and the final result will be printed in the form of table.

CHAPTER 4

## SAMPLE CODING

```
import java.sql.*;  
  
import javax.swing.*;  
  
public class SampleLogin {  
  
    public static void main(String[] args) {  
  
        String user = JOptionPane.showInputDialog("Enter Username:");  
  
        String pass = JOptionPane.showInputDialog("Enter Password:");  
  
        try {  
  
            Class.forName("com.mysql.cj.jdbc.Driver");  
  
            Connection con = DriverManager.getConnection(  
                "jdbc:mysql://localhost:3306/hospital_management", "root",  
                "yourpassword");
```

```
PreparedStatement pst = con.prepareStatement(  
    "SELECT * FROM users WHERE username=? AND password=?");  
  
pst.setString(1, user);  
  
pst.setString(2, pass);  
  
ResultSet rs = pst.executeQuery();  
  
  
  
  
if (rs.next())  
  
    JOptionPane.showMessageDialog(null, "Login Successful!");  
  
else  
  
    JOptionPane.showMessageDialog(null, "Invalid Credentials!");  
  
  
  
  
con.close();  
  
} catch (Exception e) {  
  
    JOptionPane.showMessageDialog(null, "Error: " + e.getMessage());  
  
}  
  
}  
  
}
```

## Sample 1

Sample 1 depicts the display code, that gets the data from the database i.e. being stored there and represented on users demand with the layout and measurements i.e. being already specified.

```
import java.sql.*;

public class SampleSignUp {

    public static void main(String[] args) {

        String username = "john_doe";

        String email = "john@example.com";

        String password = "12345";

        try {

            Class.forName("com.mysql.cj.jdbc.Driver");

            Connection con = DriverManager.getConnection(
                "jdbc:mysql://localhost:3306/hospital_management", "root",
                "yourpassword");

            PreparedStatement pst = con.prepareStatement(
```

```
"INSERT INTO users (username, email, password) VALUES (?, ?, ?)");

pst.setString(1, username);

pst.setString(2, email);

pst.setString(3, password);

int rows = pst.executeUpdate();

if (rows > 0)

    System.out.println("User Registered Successfully!");

else

    System.out.println("Registration Failed!");

con.close();

} catch (Exception e) {

    System.out.println("Error: " + e.getMessage());

}

}

}

Sample 2
```

Sample 2 depicts the booking part of the code, where it displays booking details and enter user data and store it in database

```
import java.sql.*;
import javax.swing.*;

public class SampleBooking {

    public static void main(String[] args) {

        String patientName = JOptionPane.showInputDialog("Enter Patient Name:");
        String disease = JOptionPane.showInputDialog("Enter Disease:");
        String ambulance = JOptionPane.showInputDialog("Need Ambulance?  
(Yes/No):");

        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con = DriverManager.getConnection(
                "jdbc:mysql://localhost:3306/hospital_management", "root",
                "yourpassword");

            PreparedStatement pst = con.prepareStatement(
                "INSERT INTO bookings (patient_name, disease, ambulance_required)
                VALUES (?, ?, ?)");
                pst.setString(1, patientName);
                pst.setString(2, disease);
                pst.setString(3, ambulance);
                pst.executeUpdate();
        }
    }
}
```

```
    pst.setString(2, disease);

    pst.setString(3, ambulance);

    int rows = pst.executeUpdate();

    if (rows > 0)

        JOptionPane.showMessageDialog(null, "Booking Successful!\nPatient: "
+ patientName +

        "\nDisease: " + disease + "\nAmbulance: " + ambulance);

    else

        JOptionPane.showMessageDialog(null, "Booking Failed!");

    con.close();

} catch (Exception e) {

    JOptionPane.showMessageDialog(null, "Error: " + e.getMessage());

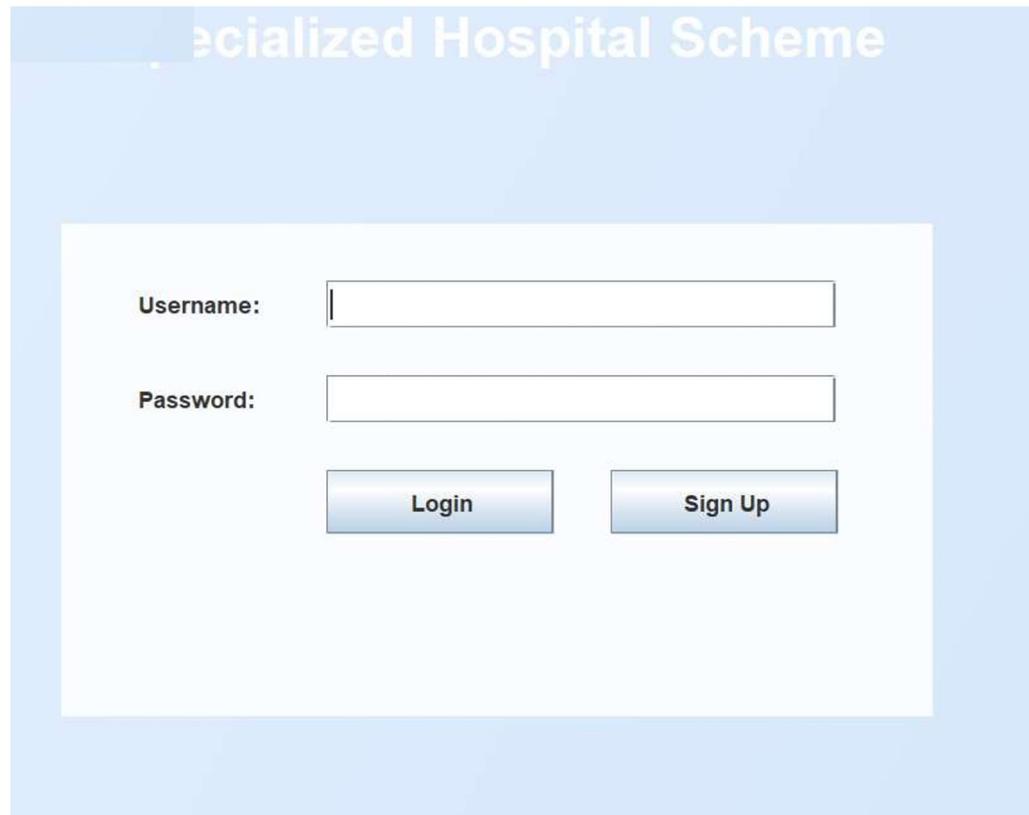
}

}

}
```

## CHAPTER 5

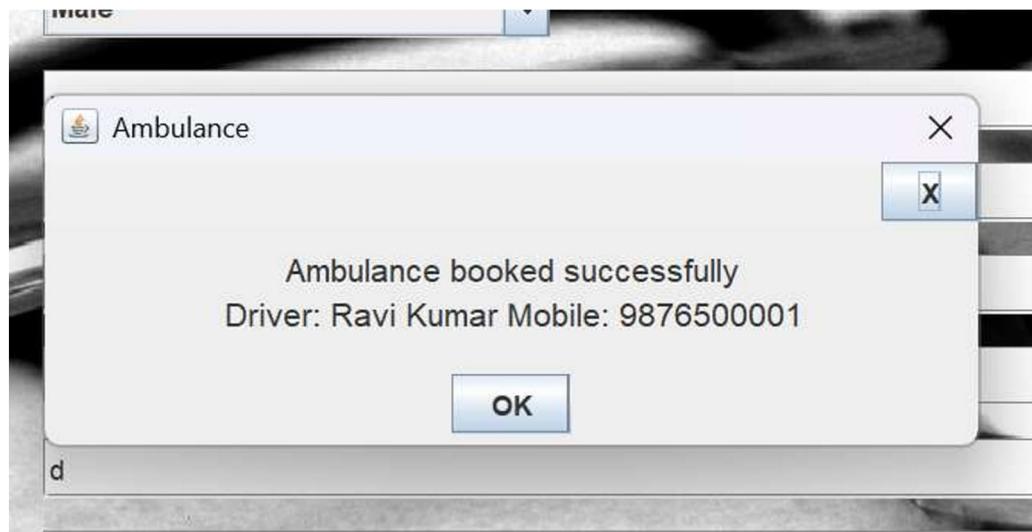
### Screenshots



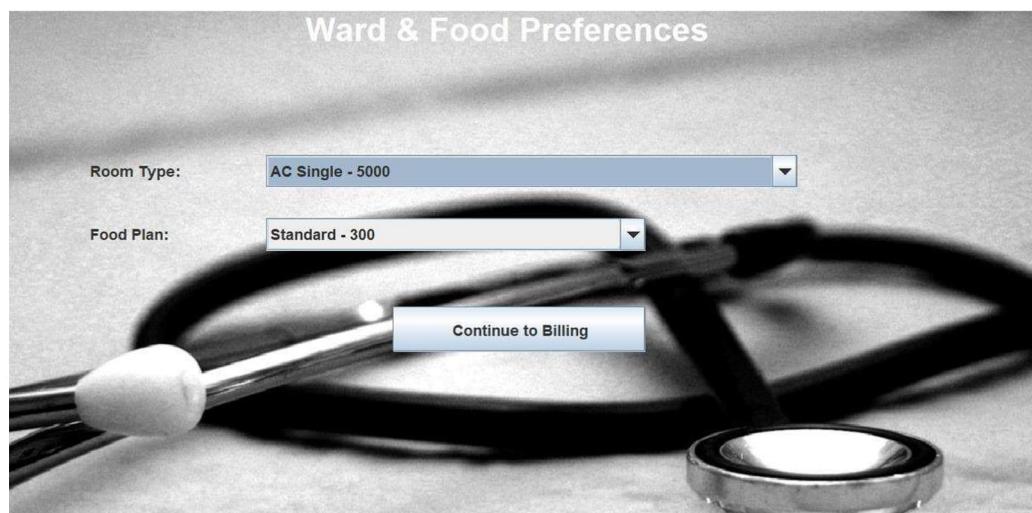
**Fig 5.1 Introduction page**

The image shows a 'Patient Details' form page. The background features a black and white photograph of a stethoscope coiled on a surface. The form itself has a light gray background with various input fields. The fields include: 'Name:' with a text input box; 'Age:' with a text input box; 'Gender:' with a dropdown menu set to 'Male'; 'Address:' with a text input box; 'Mobile:' with a text input box; 'Guardian Name:' with a text input box; 'Guardian Relation:' with a text input box; 'Guardian Mobile:' with a text input box; and 'Disease / Symptoms:' with a text input box. At the bottom of the form, there are two checkboxes: 'Emergency' and 'Require Ambulance'. Below the checkboxes are two blue rectangular buttons: 'Logout' on the left and 'Verify & Continue' on the right.

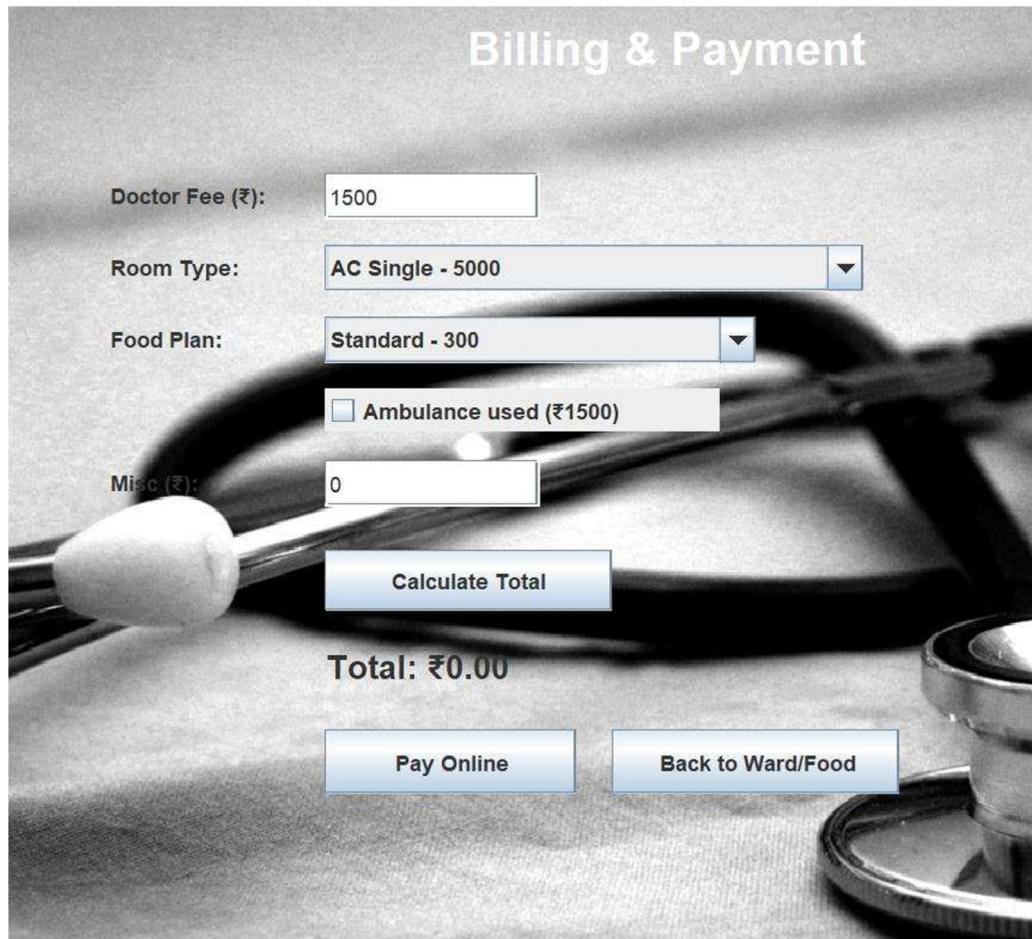
**Fig 5.2 Patients details**



**Fig 5.3 Booking Log**



**Fig 5.4 Booking creation**



## **Fig 5.5 Billing and Payments**

The screenshot shows the MySQL Workbench interface. On the left, the database tree displays 'shm\_db' with tables like ambulance, bookings, doctors, food, hospitals, patients, payments, reviews, users, and wards. Under 'Stored Procedures' and 'Functions', there are no entries. At the bottom, tabs for Administration, Schemas, and Information are visible, along with a search bar for 'patients 1 x'.

1 ▼ SELECT \* FROM shm\_db.patients;

shm\_db

Tables

- ambulance
- bookings
- doctors
- food
- hospitals
- patients
- payments
- reviews
- users
- wards

Views

Stored Procedures

Functions

Result Grid | Filter Rows: Edit:

	id	user_id	name	age	gender	address	mobile
▶	1	NULL	Rithesh	19	NULL	Chennai	9843464
	2	NULL	pranav	19	NULL	chennai	754364
	3	NULL	thendral	19	NULL	10/mogapapir	9000385
	4	NULL	jfc	14	NULL	ua	a
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Result Grid Form Editor Field Types

**Fig 5.6 Database creation****CHAPTER 6****CONCLUSION AND FUTURE ENHANCEMENT**

The Specialized Hospital Management System efficiently digitalizes hospital operations, improving accuracy and reducing manual workload through automated data handling. In the future, this system can be enhanced with features like cloud-based storage, AI-assisted diagnosis, mobile app integration, real-time ambulance tracking, and advanced data security for better scalability and user experience.

## **REFERENCES**

1. [HTTPS://WWW.W3SCHOOLS.COM/SQL/](https://www.w3schools.com/sql/)
2. [HTTPS://WWW.TUTORIALSPPOINT.COM/SQLITE/  
INDEX.JAVA](https://www.tutorialspoint.com/sqlite/index.java)
3. [HTTPS://WWW.WIKIPEDIA.ORG/](https://www.wikipedia.org/)
4. [HTTPS://WWW.LEARNJAVA.ORG/](https://www.learnjava.org/)
5. [HTTPS://WWW.CODECADEMY.COM/LEARN/LEARN-JAVA](https://www.codecademy.com/learn/learn-javascript)