LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

## Module Code & Module Title

## CS5004NI Emerging Programming Platforms and Technologies

## Assessment Weightage & Type

## 30% Group Coursework

## Year and Semester

## 2019-20 Autumn / 2020-21 Spring

**Title: Hospital Information System**

| Student Name | London met ID | College ID |
|---|---|---|
| Biraj Sapkota | 20048871 | NP01CP4S210229 |
| Niwahang Angbuhang | 20048942 | NP01CP4S210237 |
| Karmaraj Giri | 20048909 | NP01CP4S210233 |
| Sandesh Rai | 20049423 | NP01CP4S210222 |

**Assignment Due Date: 10th Jan 2022**

**Assignment Submission Date: 10th Jan 2022**

*I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.*

## Table of Contents

## List of Figures

## List of Tables

# 1. Proposal

**Title of the System:** Hospital Management System

## 1.1 Brief Introduction

In this group coursework, our group has decided to develop a program for storing hospital related information. The system is simply used for a patient who is trying to search for a hospital for check-up. The admin handles data relating to the hospital like name of the hospital, reputation, specialty of the hospital and many other information that is required for a patient to know before going to pick a hospital to go to. The data for the information relating to the hospital is stored in a JTable. The admin inputs and manipulates the data by using a JTextField. The data can be retrieved from an excel file or text file after it is stored in a JTable. The user can search for the hospital through specialty and reputation of the hospital.

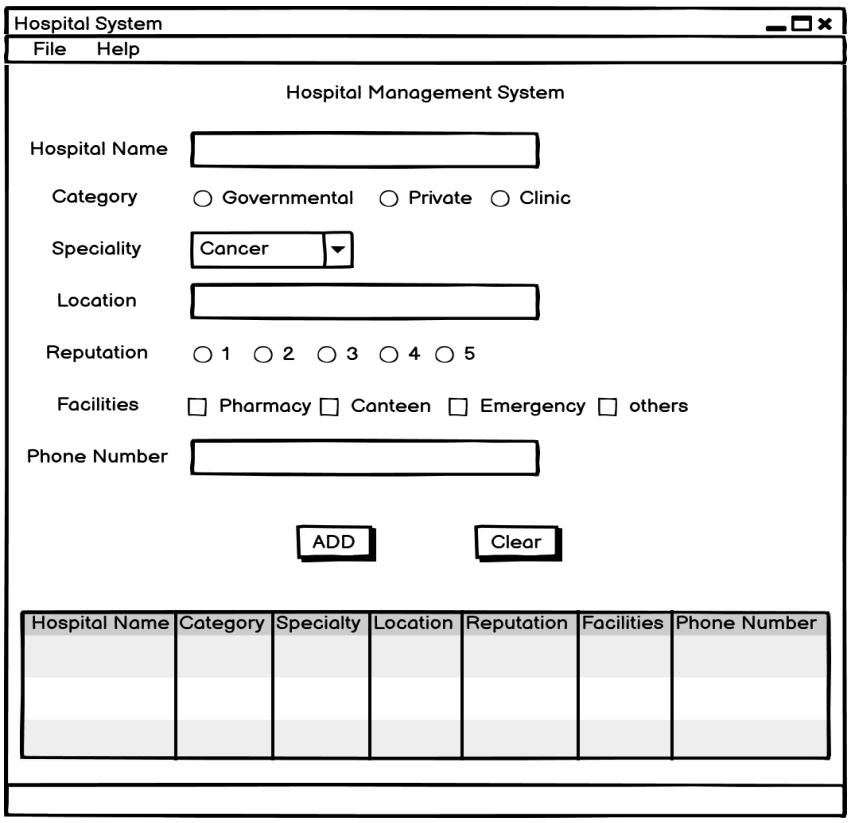



*Figure 1-1: Screenshot of Wireframe of the System GUI*

## 1.2 GUI Components

The following GUI components have been implemented in our system.

**I. JTextfield:**

The JTextfield is used to take certain data which might be added to the table later after a button is clicked. The JTextfield is used for the name of the hospital, location, and phone number. It takes and returns values in string.

**II. JLabel:**

The JLabel is used to provide a name for each component while displaying the GUI form. Several JLabels had been used for different components in the form including the title.

**III. JCheckBox:**

The JCheckBox is used to choose one or more facilities available in the hospital from the given options.

**IV. JRadioButton:**

The JRadioButton is used to choose a value for the reputation and category of the hospital.

**V. JComboBox:**

The JComboBox is used to choose an option which the hospital specializes in from a list of values.

**VI. JTable:**

The JTable is used for displaying the values accordingly from the GUI components and from an external source like excel or a text file.

**VII. JMenuBar:**

The JMenuBar contains two JMenus (File and Help) and inside these menus are JMenuItems.

## 1.3 List of Features

A. This system has the capability of adding, editing, and deleting details such as hospital name, location, and many more through the input of user.
B. The data inputted are shown in the JTable in the GUI.
C. A specific hospital can be searched through the reputation and specialty of the hospital.
D. There is a menu bar section and, in the menu, an open menu item where we can open any external file and display it in the JTable.
E. Similarly, a help menu item is used to show how to add information to the table.
F. The information in the JTable can be saved on a text file.

## 1.4 Justification of Tools

The tools used for the development of the system are Java-JDK as a programming language, Apache NetBeans as the IDE to code, Balsamiq for creating wireframes and MS Word for the documentation. Apache NetBeans is used as an IDE for the system as it is mainly focused on GUI and GUI making is quite easy in NetBeans as we can just drag and drop the GUI components needed for the system. MS Word is easy to use and widely popular for writing documents. Balsamiq consists of pre-built elements for creating wireframes which makes it easy for the user to create wireframes.

## 2. Individual Task

| Name | Task |
|---|---|
| Niwahang Angbuhang | ✓ Contributed to writing the method description.<br>✓ Helped on the proposal of the coursework.<br>✓ Helped on formatting the report.<br>✓ Developed the selection sort flowchart.<br>✓ Helped on adding functionalities to add and clear buttons.<br>✓ Helped on the proposal of the coursework.<br>✓ Developed the GUI of the system.<br>✓ Developed flowchart of the Selection sort. |
| Biraj Sapkota | ✓ Worked on creating and implementing clear data and clear all data buttons.<br>✓ Added functionalities to add, clear and menu bar buttons.<br>✓ Contributed on writing binary search.<br>✓ Worked on the testing part of the report.<br>✓ Worked on the Validation part.<br>✓ Developed flowchart of the binary search.<br>✓ Developed the GUI of the system.<br>✓ Developed the search hospital and search hospital by category method and implemented in the system.<br>✓ Helped on the proposal of the coursework. |

| | |
|---|---|
| Sandesh Rai | ✓ Worked on the introduction and conclusion part.<br>✓ Helped on the proposal of the coursework.<br>✓ Developed the add button, clear button, and menu bar button.<br>✓ Instructed and guided the whole members.<br>✓ Helped on formatting the report.<br>✓ Helped in programming section. |
| Karmaraj Giri | ✓ Helped on the proposal part of the report.<br>✓ Instructed and guided the whole members.<br>✓ Developed the add button, clear button, and menu bar button.<br>✓ Helped in programming section.<br>✓ Worked on the introduction and conclusion part.<br>✓ Helped on formatting the report. |

*Table 2-1: Table showing Individual Tasks*

## 3. Binary Search Algorithm

Binary search is an approach of finding a desired value from a list of sorted values by determining whether the value occurs in the first or second half and then repeating the search in one of the halves. The values must be in a sorted pattern to implement the binary search. The divide-by-two approach used in this algorithm made this algorithm named as binary search. Binary search goes through several recursions during the search. Recursion is a technique of calling function by itself to accomplish a given task.

In our project as well, the binary search had been implemented for searching the hospital according to their ratings as mentioned below:

- First, the ratings details of the hospitals in table were added in an array list and converted to an array of Integer type.
- Then, the array was sorted using Selection sort and the sorted array was passed in the method for searching the desired value in that array which involves several recursions of the search method for success completion of search.
- The index value of the desired item is returned when the value is found else an invalid message is returned by the method.
- With this the search completes on getting either an index value or an error message from the search method.

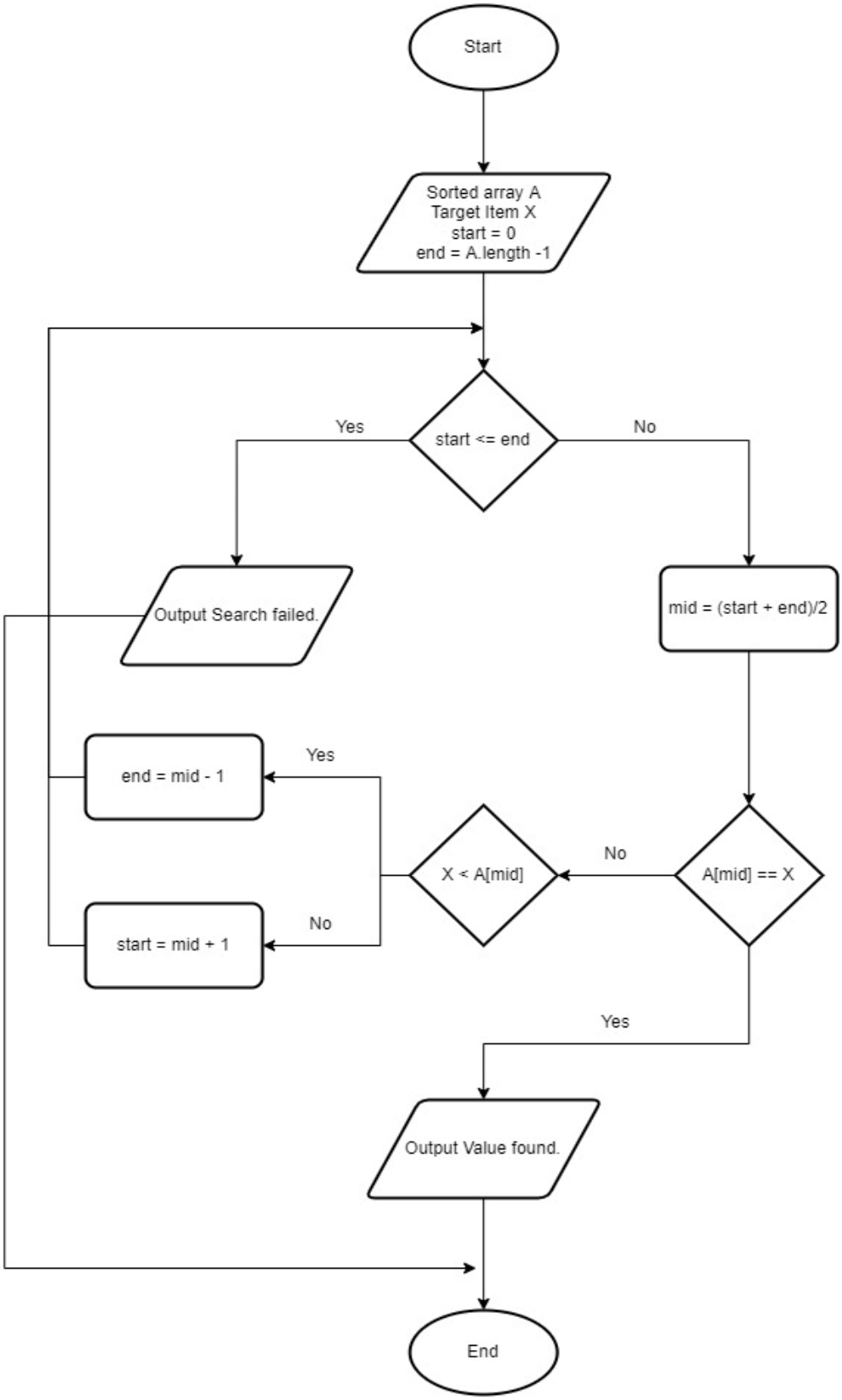



*Figure 3-1: Flowchart to represent binary search algorithm*

## 4. Selection Sort Algorithm

Since some people might have a heart disease or cancer, those people will require certain hospitals which specializes in those areas. Therefore, we have added a search button to sort between different specialty of hospitals. We have used selection sort algorithm to search for hospitals with a specific specialty. Selection Sort algorithm is used for arranging a list of elements in a particular order either in ascending or descending order. The first element in the list is chosen and is compared repeatedly with all the elements remaining in the list. If one element is smaller than the chosen element, then both are swapped so that the first position is taken by the smallest element in the sorted ascending order. Then the second element in the list is chosen and is compared again with the elements remaining in the list. If any element is smaller than the chosen element, then both are swapped. This process is repeated until the entire list is sorted in ascending or descending order.

Let's suppose an array A= [10,7,3,1] needs to be sorted in ascending order. The minimum element in the array 1 is searched for and then swapped with the element that is currently located at the first position, 10. Now the minimum element in the remaining unsorted array is searched for and put in the second position, and the process is repeated on loop. The process is shown in the flow chart below.

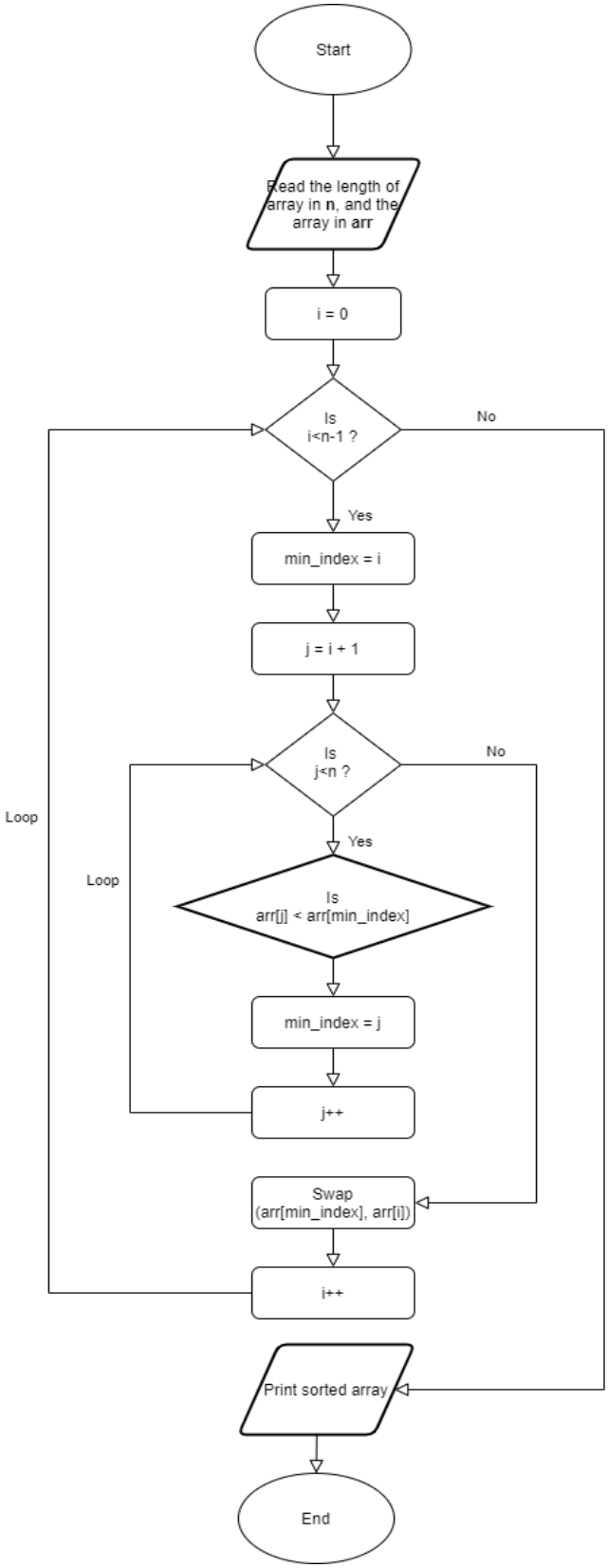



*Figure 4-1: Flowchart to represent Selection sort algorithm*

## 5. Method Description

| Method Name | Description |
|---|---|
| private void SearchSpecialty_JBActionPerformed() | This method is used to search for the hospitals which have specific specialty in some sectors. Selection sort is used in this method for searching. |
| private void Save_JMIActionPerformed() | This method is used to save the details of the hospital in the table in either .txt file or .csv file. |
| private void Exit_JMIActionPerformed() | This method is used to close the program. |
| private void Open_JMIActionPerformed() | This method is used to open a .txt file or .csv file which contains the details of the hospital. After opening the file, the details of the hospitals are automatically inserted in the table. |
| private void Help_JMIActionPerformed() | This method is used to open a user manual (a .pdf file) which shows the user how to insert data. |
| private void Add_JBActionPerformed() | This method is used to add details of the hospital to the table. If wrong inputs are given, an error message is displayed accordingly. |
| private void Clear_JBActionPerformed() | This method is used to clear the text fields, radio buttons, and check boxes in the information form. |
| private void ClearAll_JBActionPerformed() | This method is used to clear all the details of the hospitals from the table. |
| private void ClearData_JBActionPerformed() | This method is used to clear the details of the hospital of the selected row. A row needs to be selected in order to clear the data. |
| private void Search_BTNActionPerformed() | This method is used to search for the hospitals which have certain ratings. Binary search is used in this method for searching. |

| | |
|---|---|
| public int BinarySearch(Integer arr[], int start, int end, int search) | This method is used to search for the details (integer) in the table which is given by the user in the search text field. |
| public static void main(String args[]) | This is the main method. This method is used to run the program. |

*Table 5-1: Table for Method Description*

# 6. Testing

## 6.1 Test 1

| | |
|---|---|
| Test No: | 1 |
| Objective: | To run the program in NetBeans. |
| Action: | The program is compiled and run in NetBeans. |
| Expected Result: | The program would be compiled and run. |
| Actual Result: | The program got compiled and ran too. |
| Conclusion: | The test is successful. |

*Table 6-1: Test Table for Test 1*

## Screenshots:

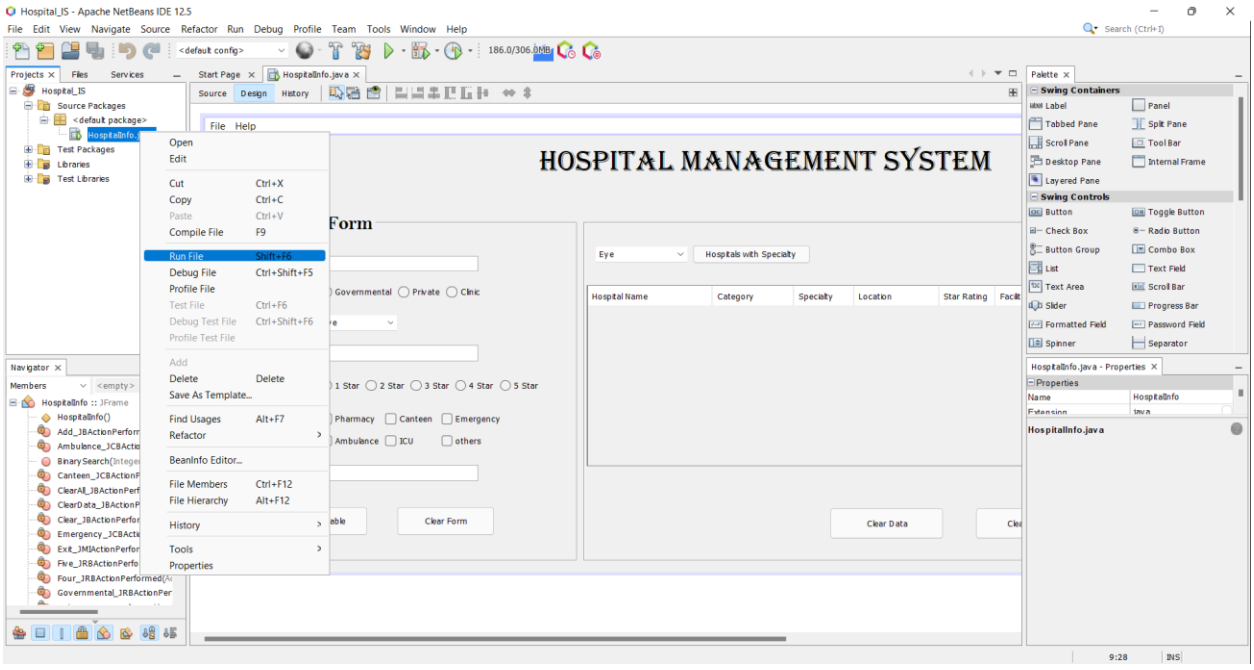



*Figure 6-1: Screenshot for running the system*

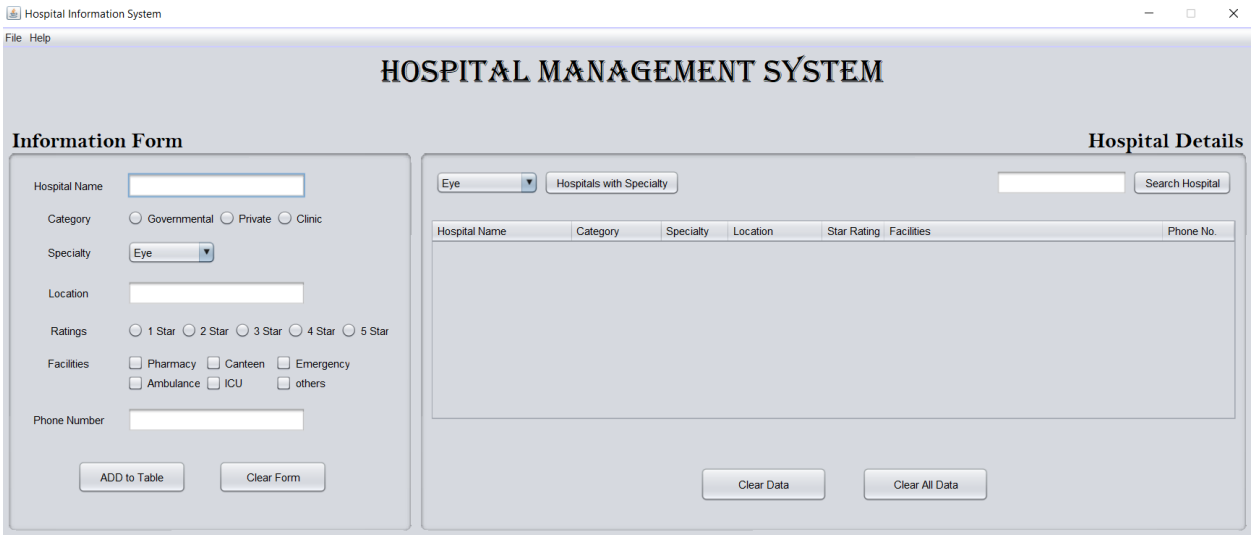



*Figure 6-2: Screenshot for output after running the system*

## 6.2 Test 2

| | |
|---|---|
| Test No: | 2 |
| Objective: | To add values in table by filling the form. |
| Action: | Necessary details were filled in the form and was added to table. |
| Expected Result: | The details would be added in the table. |
| Actual Result: | The details got added in the table. |
| Conclusion: | The test is successful. |

*Table 6-2: Test Table for Test 2*

## Screenshots:

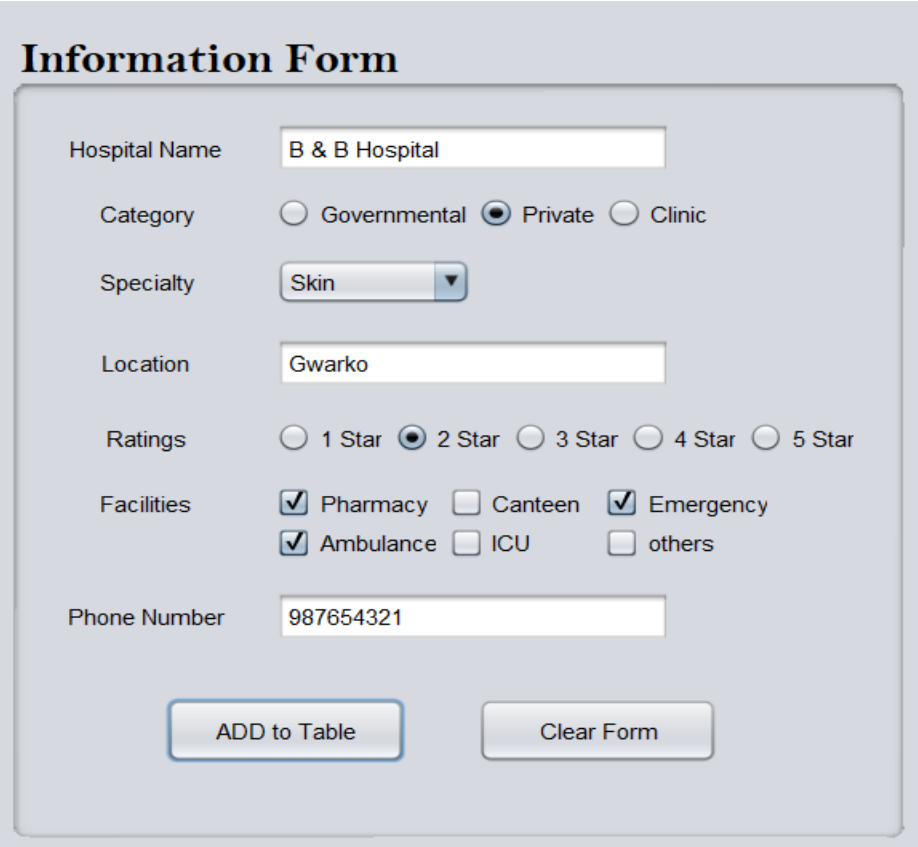



*Figure 6-3: Screenshot for adding details to table*

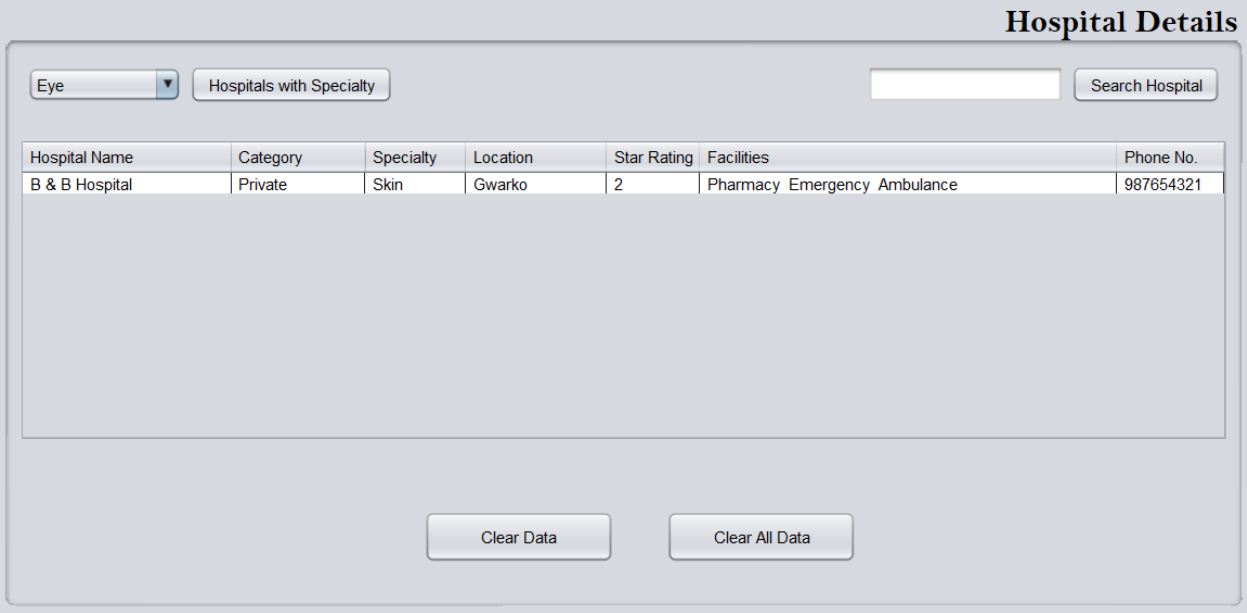



*Figure 6-4: Screenshot for output of adding details to table*

## 6.3 Test 3

| | |
|---|---|
| Test No: | 3 |
| Objective: | To search hospital according to their ratings. |
| Action: | A validate input was given in the text field and searched. |
| Expected Result: | The searched detail would pop-up. |
| Actual Result: | The searched detail got popped-up. |
| Conclusion: | The test is successful. |

*Table 6-3: Test Table for Test 3*

## Screenshots:

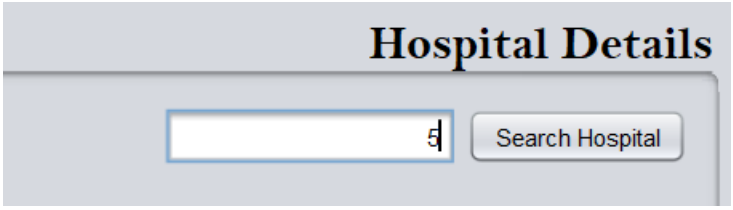



*Figure 6-5: Screenshot for giving input for test 3*

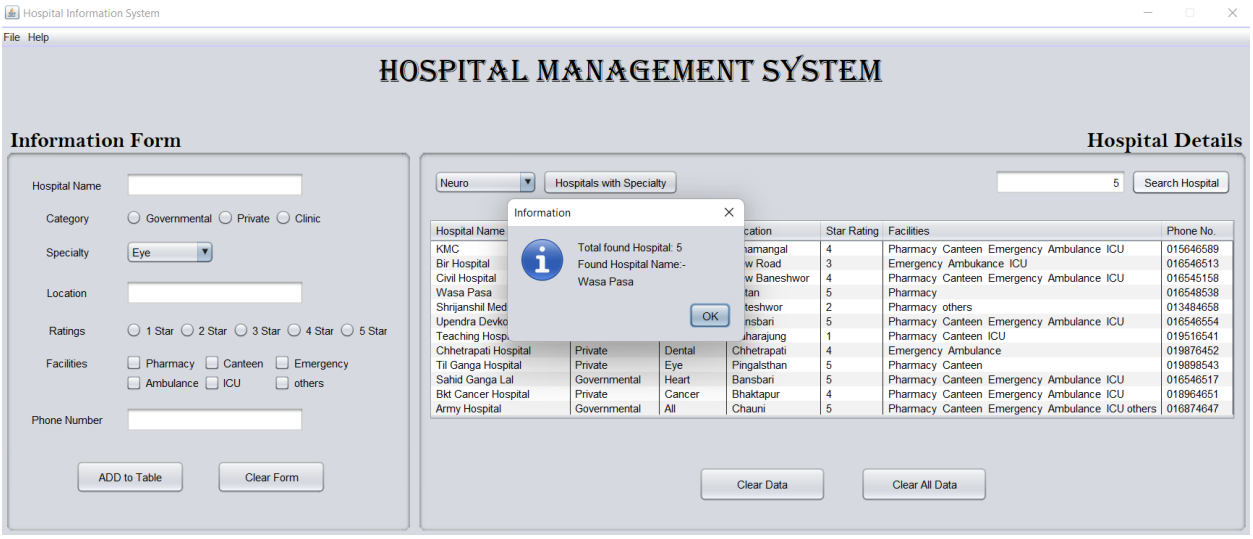



*Figure 6-6: Screenshot for output of test 3*

## 6.4 Test 4

| | |
|---|---|
| Test No: | 4 |
| Objective: | To search hospitals according to their specialty. |
| Action: | An option was selected from combo box and was searched. |
| Expected Result: | The searched detail would pop-up. |
| Actual Result: | The searched detail got popped-up. |
| Conclusion: | The test is successful. |

*Table 6-4: Test Table for Test 4*

## Screenshots:

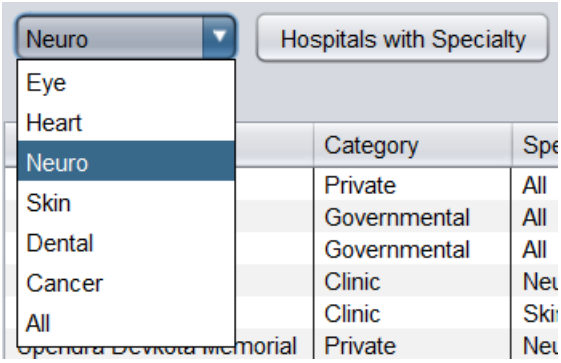



*Figure 6-7: Screenshot for giving input for test 4*

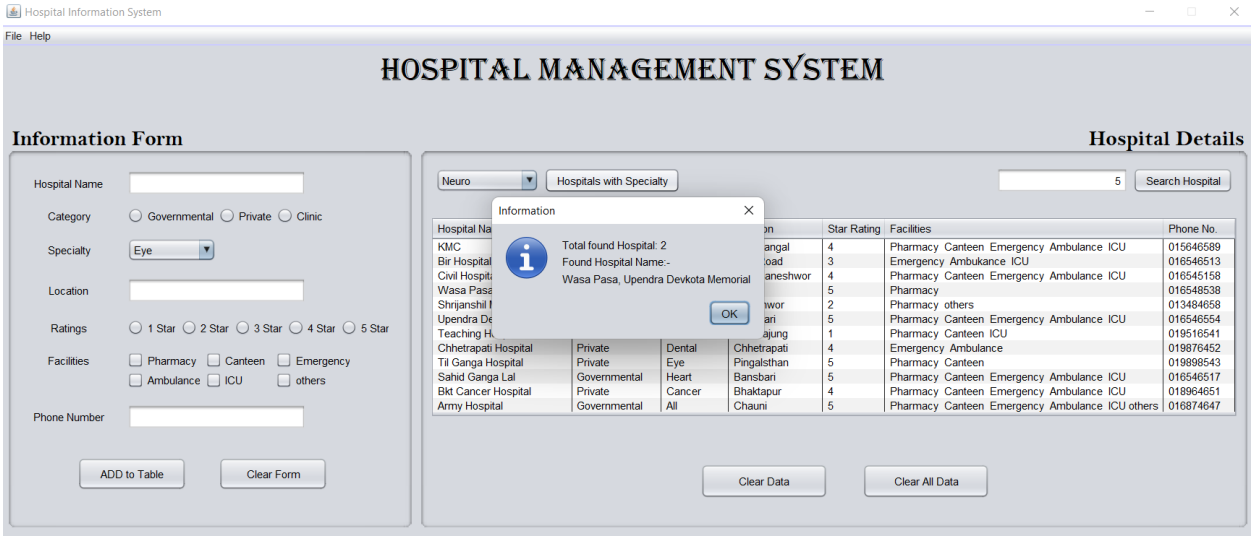



*Figure 6-8: Screenshot for output of test 4*

## 6.5 Test 5

| | |
|---|---|
| Test No: | 5 |
| Objective: | To open an external .txt file for importing data in table. |
| Action: | The required file was opened through the open menu item in file menu of the menu bar. |
| Expected Result: | The data would be imported to the table. |
| Actual Result: | The data got imported to the table. |
| Conclusion: | The test is successful. |

*Table 6-5: Test Table for Test 5*

## Screenshots:

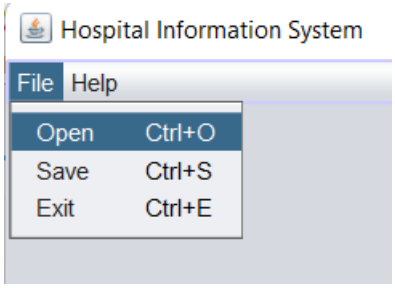



*Figure 6-9: Screenshot for selecting the open menu item*

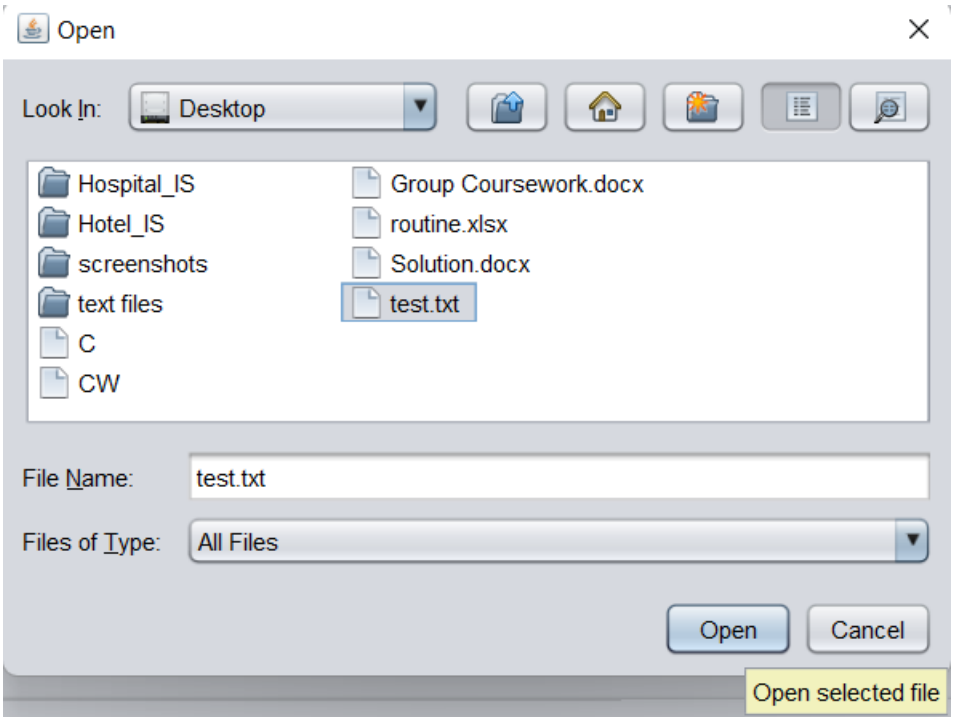
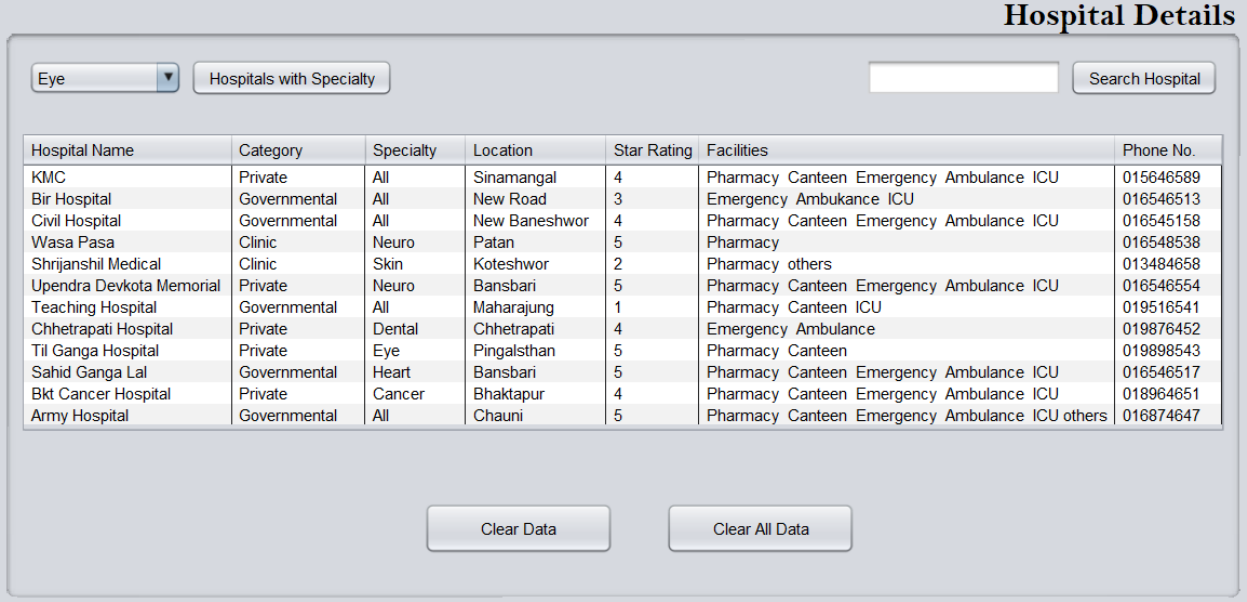



*Figure 6-10: Screenshot for opening the desired file*




| Hospital Name | Category | Specialty | Location | Star Rating | Facilities | Phone No. |
|---|---|---|---|---|---|---|
| KMC | Private | All | Sinamangal | 4 | Pharmacy Canteen Emergency Ambulance ICU | 015646589 |
| Bir Hospital | Governmental | All | New Road | 3 | Emergency Ambukance ICU | 016546513 |
| Civil Hospital | Governmental | All | New Baneshwor | 4 | Pharmacy Canteen Emergency Ambulance ICU | 016545158 |
| Wasa Pasa | Clinic | Neuro | Patan | 5 | Pharmacy | 016548538 |
| Shrijanshil Medical | Clinic | Skin | Koteshwor | 2 | Pharmacy others | 013484658 |
| Upendra Devkota Memorial | Private | Neuro | Bansbari | 5 | Pharmacy Canteen Emergency Ambulance ICU | 016546554 |
| Teaching Hospital | Governmental | All | Maharajung | 1 | Pharmacy Canteen ICU | 019516541 |
| Chhetrapati Hospital | Private | Dental | Chhetrapati | 4 | Emergency Ambulance | 019876452 |
| Til Ganga Hospital | Private | Eye | Pingalsthan | 5 | Pharmacy Canteen | 019898543 |
| Sahid Ganga Lal | Governmental | Heart | Bansbari | 5 | Pharmacy Canteen Emergency Ambulance ICU | 016546517 |
| Bkt Cancer Hospital | Private | Cancer | Bhaktapur | 4 | Pharmacy Canteen Emergency Ambulance ICU | 018964651 |
| Army Hospital | Governmental | All | Chauni | 5 | Pharmacy Canteen Emergency Ambulance ICU others | 016874647 |

*Figure 6-11: Screenshot for output of test 5*

## 6.6 Test 6

| | |
|---|---|
| Test No: | 6 |
| Objective: | To add blank data in the table. |
| Action: | The add button was fired without any details. |
| Expected Result: | An error message would appear. |
| Actual Result: | An error message got appeared. |
| Conclusion: | The test is successful. |

*Table 6-6: Test Table for Test 6*

## Screenshots:

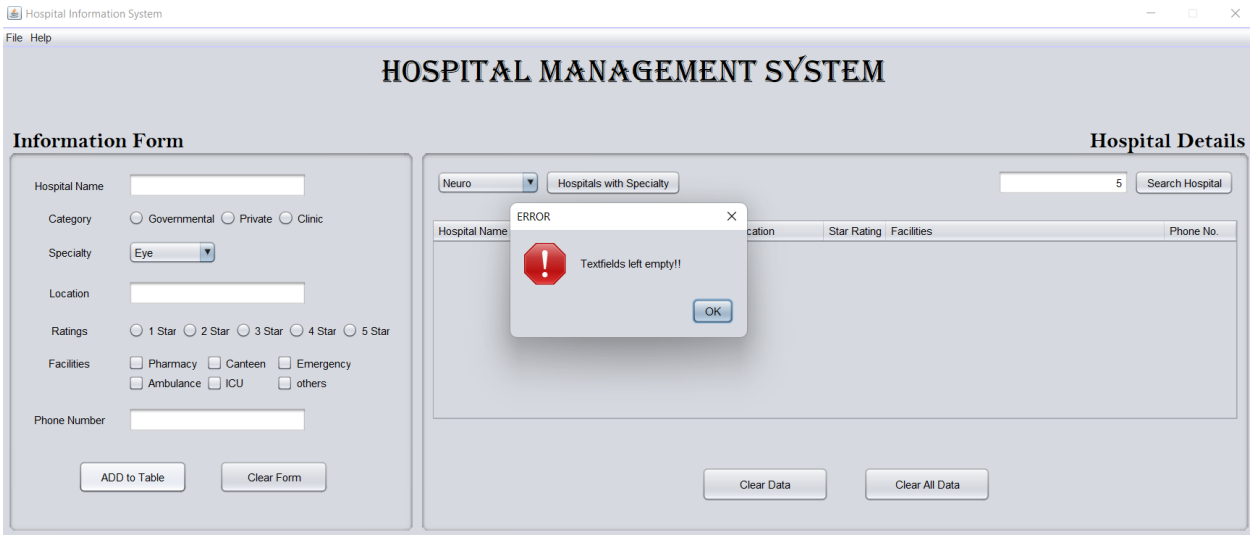



*Figure 6-12: Screenshot for output of test 6*

## 6.7 Test 7

| | |
|---|---|
| Test No: | 7 |
| Objective: | To add a duplicate data in the table. |
| Action: | Same details were given and add button was pressed. |
| Expected Result: | An error message would come up. |
| Actual Result: | An error message came up. |
| Conclusion: | The test is successful. |

*Table 6-7: Test Table for Test 7*

## Screenshots:

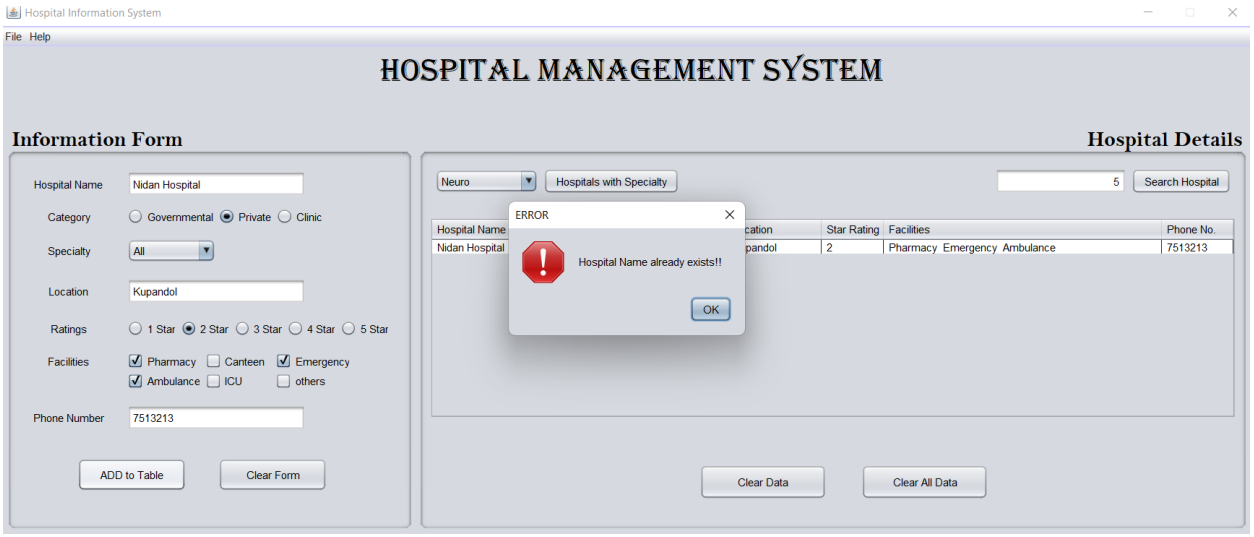



*Figure 6-13: Screenshot for output of test 7*

## 6.8 Test 8

| | |
|---|---|
| Test No: | 8 |
| Objective: | To search for hospitals according to category that are not present in the table. |
| Action: | An incompatible option was selected from combo box and searched. |
| Expected Result: | A suitable message would pop-up. |
| Actual Result: | A suitable message popped-up. |
| Conclusion: | The test is successful. |

*Table 6-8: Test Table for Test 8*

## Screenshots:

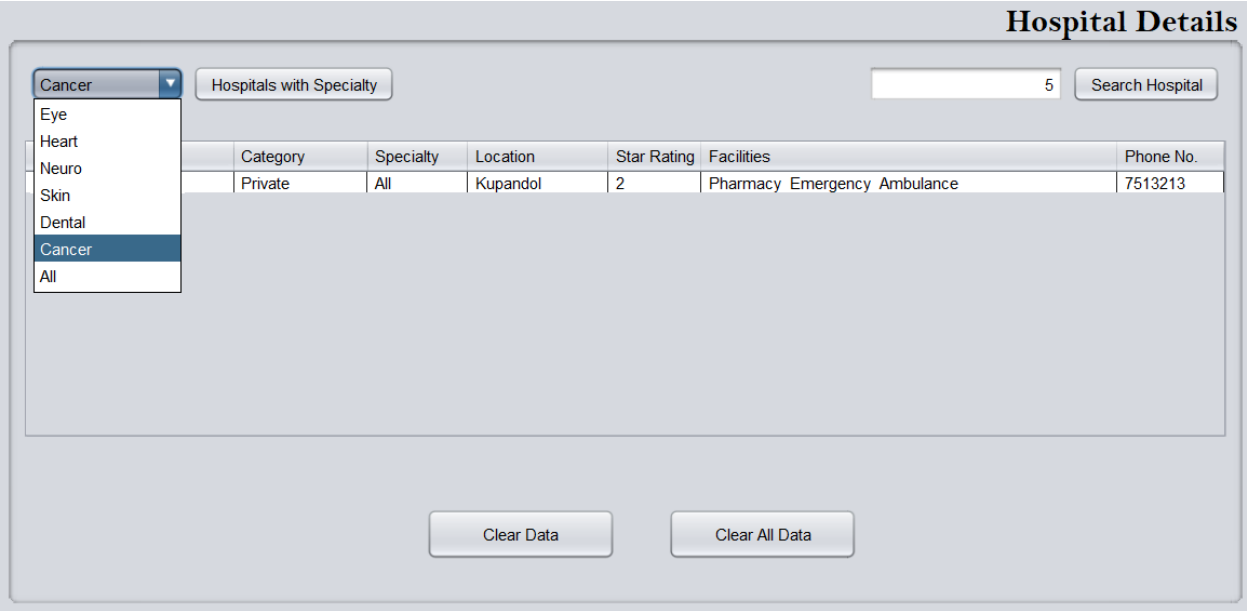



*Figure 6-14: Screenshot for giving input for test 8*

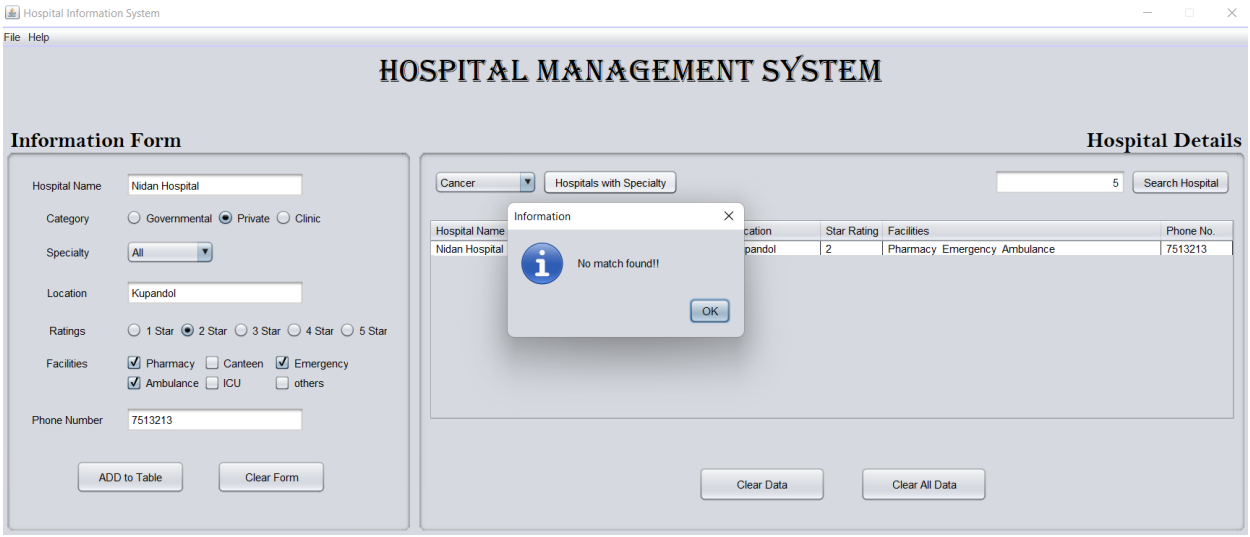



*Figure 6-15: Screenshot for output of test 8*

## 6.9 Test 9

| | |
|---|---|
| Test No: | 9 |
| Objective: | To check the correctness for unwanted value while searching by ratings. |
| Action: | Incompatible value was provided while searching by ratings. |
| Expected Result: | An error message would be displayed. |
| Actual Result: | An error message got displayed. |
| Conclusion: | The test is successful. |

*Table 6-9: Test Table for Test 9*

## Screenshots:

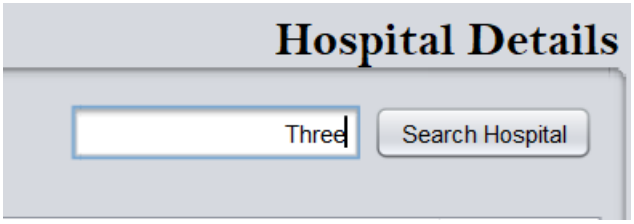



*Figure 6-16: Screenshot for giving input for test 9*

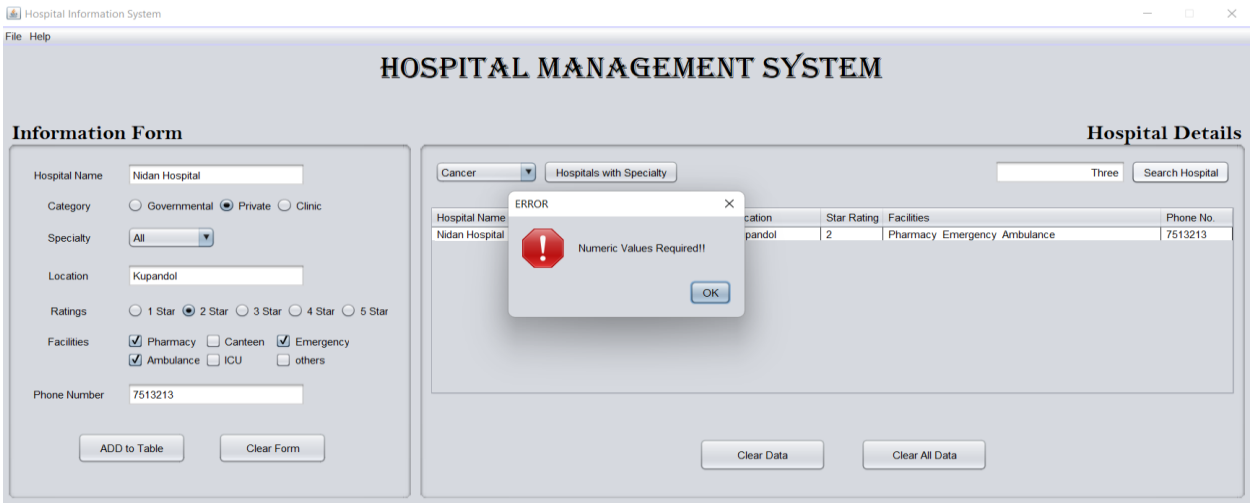



*Figure 6-17: Screenshot for output of test 9*

## 6.10 Test 10

| | |
|---|---|
| Test No: | 10 |
| Objective: | To check the correctness for not present value while searching by ratings. |
| Action: | Different value than in the table was provided while searching by ratings. |
| Expected Result: | A suitable message would be displayed. |
| Actual Result: | A suitable message got displayed. |
| Conclusion: | The test is successful. |

*Table 6-10: Test Table for Test 10*

## Screenshots:

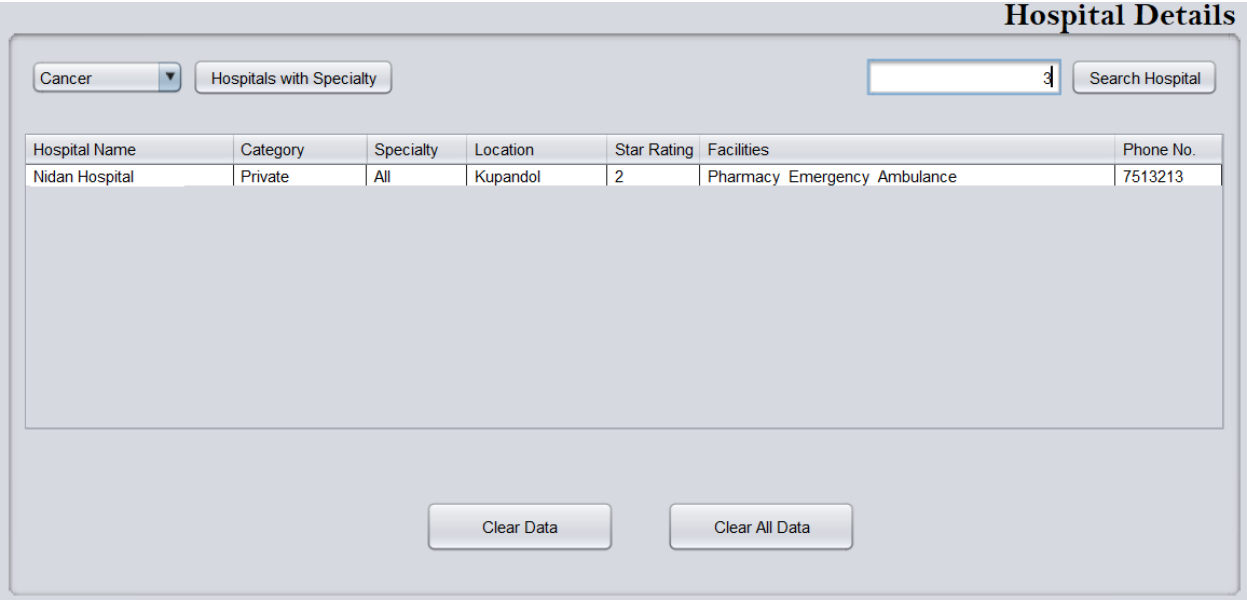



*Figure 6-18: Screenshot for giving input for test 10*

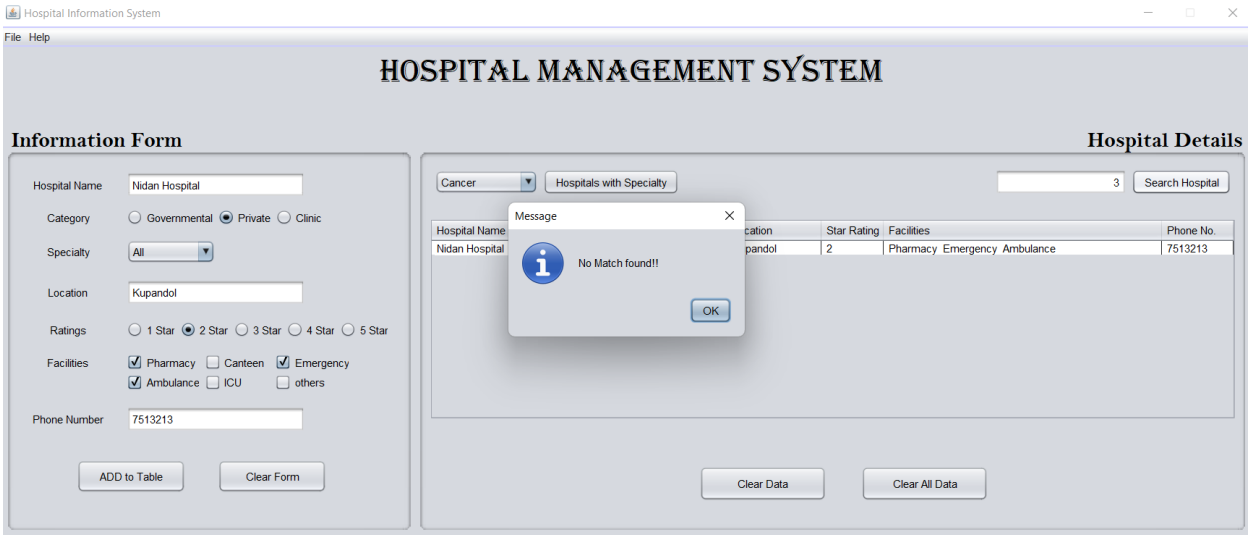



*Figure 6-19: Screenshot for output of test 10*

## 6.11 Test 11

| | |
|---|---|
| Test No: | 11 |
| Objective: | To see the correctness for clear data button. |
| Action: | No rows were selected, and clear data button was pressed. |
| Expected Result: | A suitable message dialog would appear. |
| Actual Result: | A suitable message dialog appeared. |
| Conclusion: | The test is successful. |

*Table 6-11: Test Table for Test 11.*

## Screenshots:

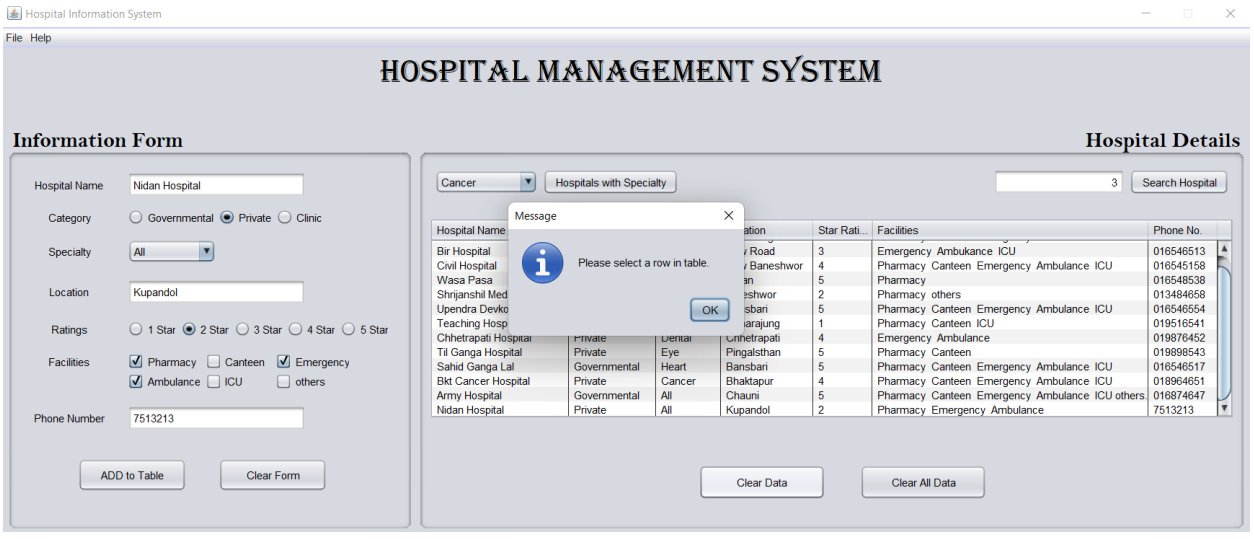



*Figure 6-20: Screenshot for output of test 11*

## 6.12 Test 12

| | |
|---|---|
| Test No: | 12 |
| Objective: | To clear selected row from the table. |
| Action: | A specific row was selected, and clear data button was fired. |
| Expected Result: | The selected row would be removed. |
| Actual Result: | The selected row got removed. |
| Conclusion: | The test is successful. |

*Table 6-12: Test Table for Test 12*

## Screenshots:

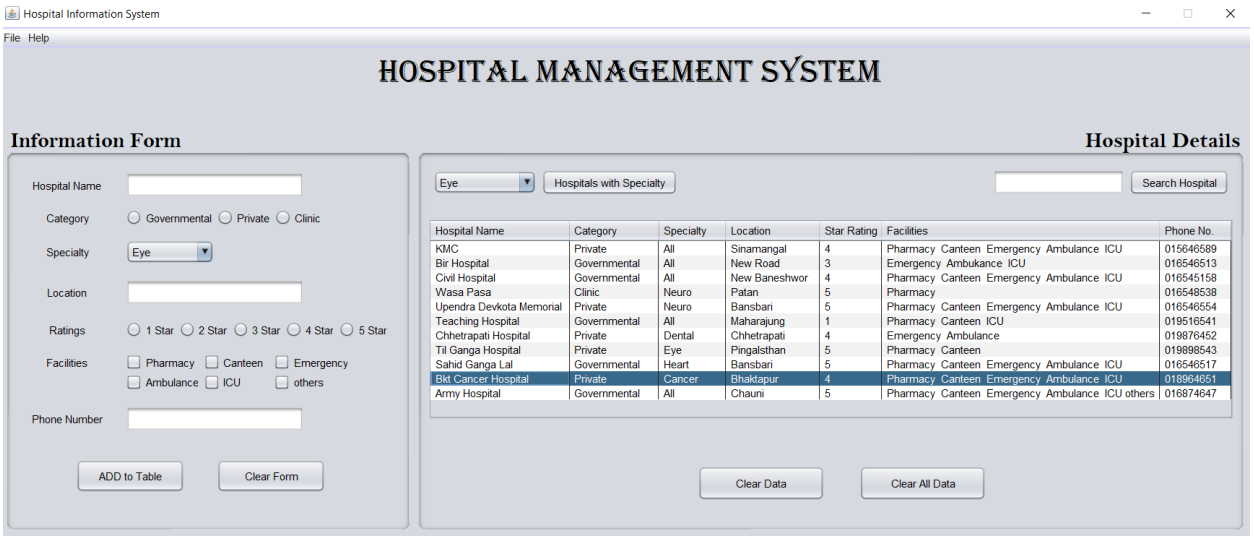



*Figure 6-21: Screenshot for giving input for test 12*

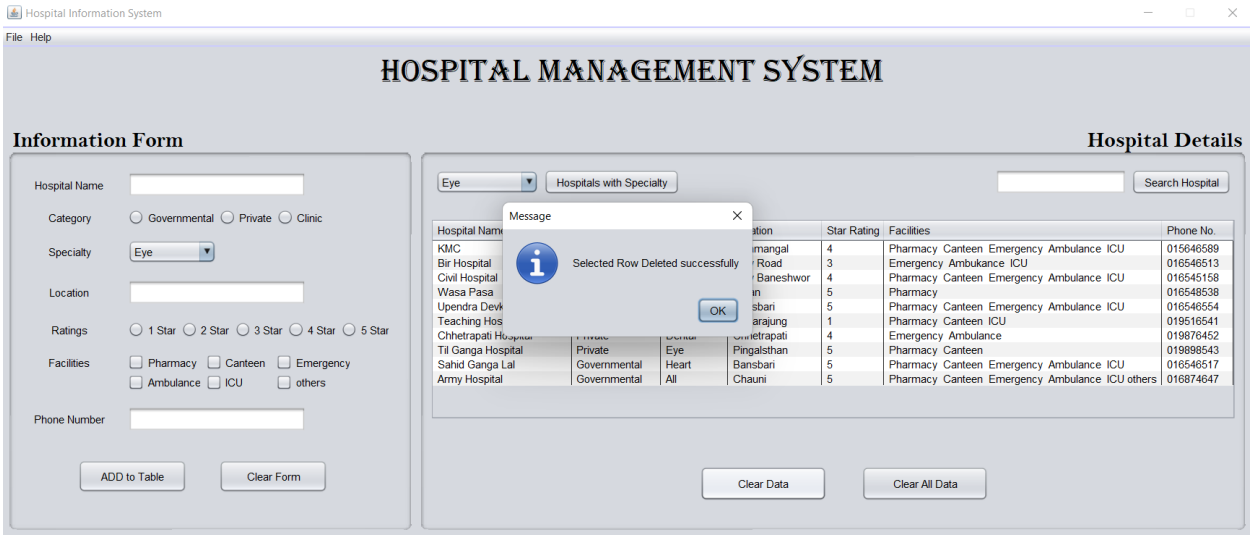



*Figure 6-22: Screenshot for output of test 12*

## 6.13 Test 13

| | |
|---|---|
| Test No: | 13 |
| Objective: | To remove all data from the table. |
| Action: | The Clear All Data button was fired. |
| Expected Result: | All the table data would be removed. |
| Actual Result: | All the table data got removed. |
| Conclusion: | The test is successful. |

*Table 6-13: Test Table for Test 13*

## Screenshots:

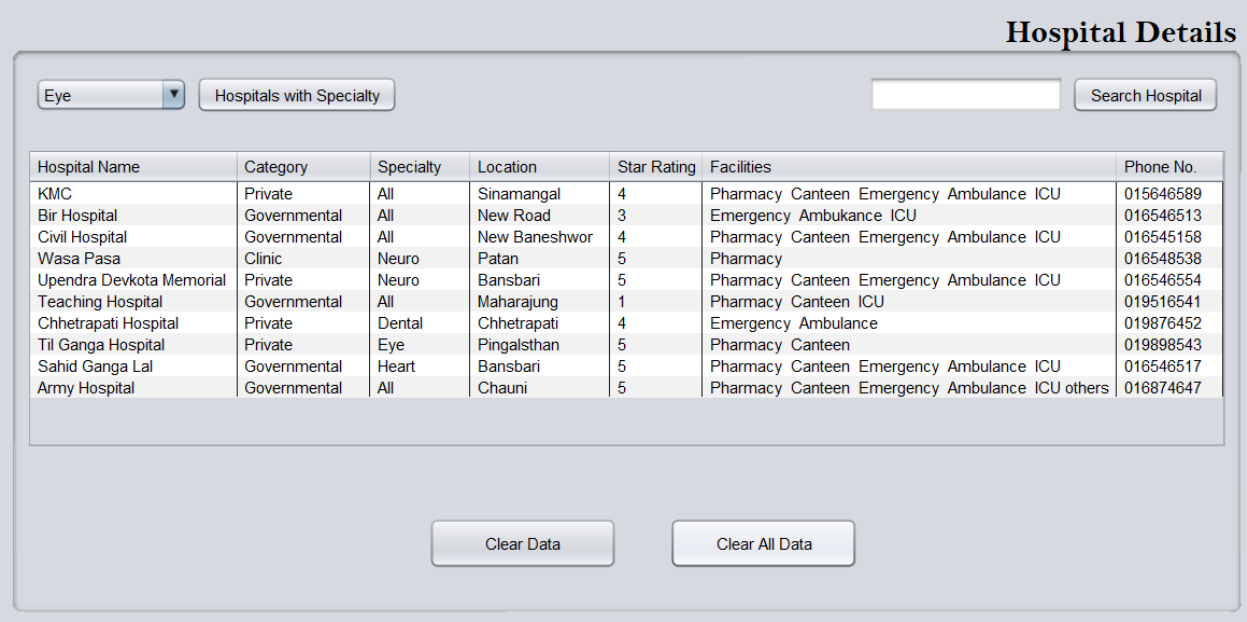



*Figure 6-23: Screenshot of the table before deleting data*

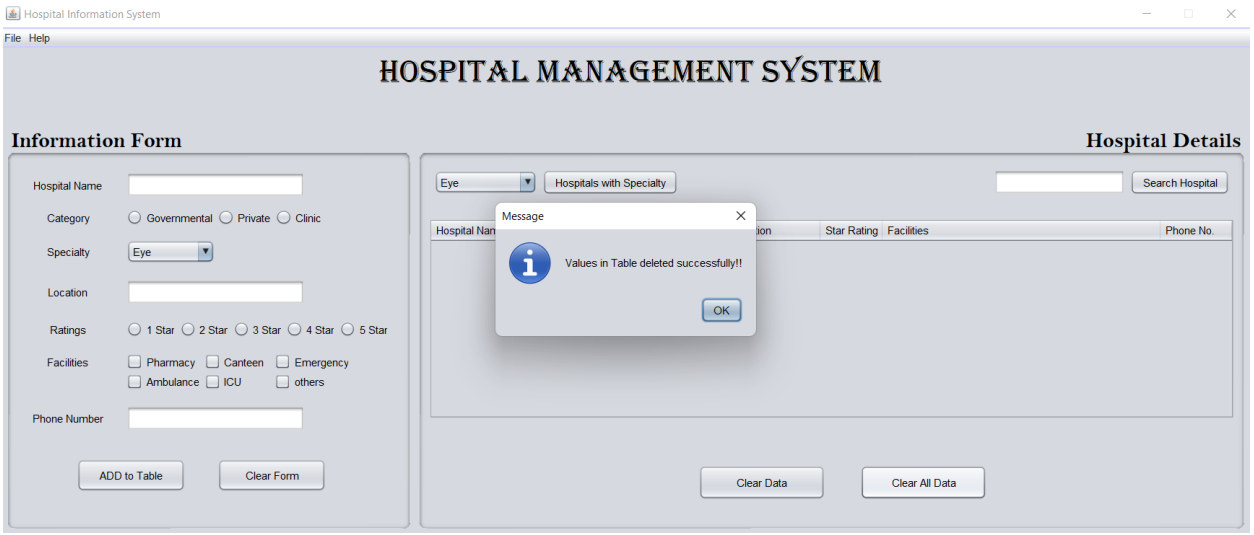



*Figure 6-24: Screenshot of the table after deleting data*

## 6.14 Test 14

| | |
|---|---|
| Test No: | 14 |
| Objective: | To check the correctness of the Clear All Data button. |
| Action: | The button was fired when no data are present in the table. |
| Expected Result: | An error message would come up. |
| Actual Result: | An error message came. |
| Conclusion: | The test is successful. |

*Table 6-14: Test Table for Test 14*

## Screenshots:

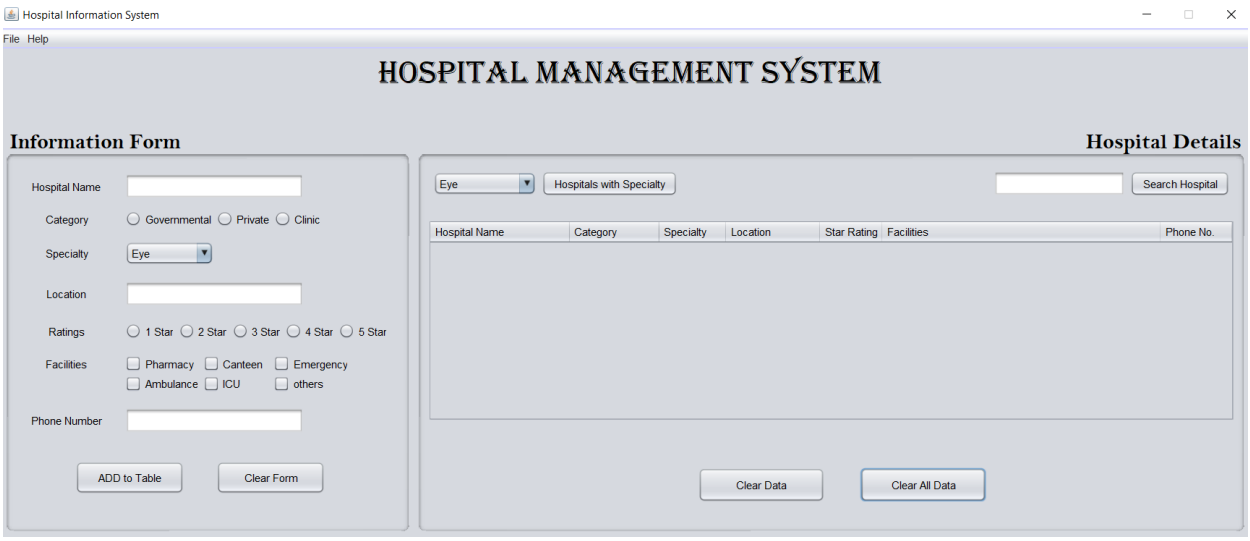



*Figure 6-25: Screenshot for action made for test 14*

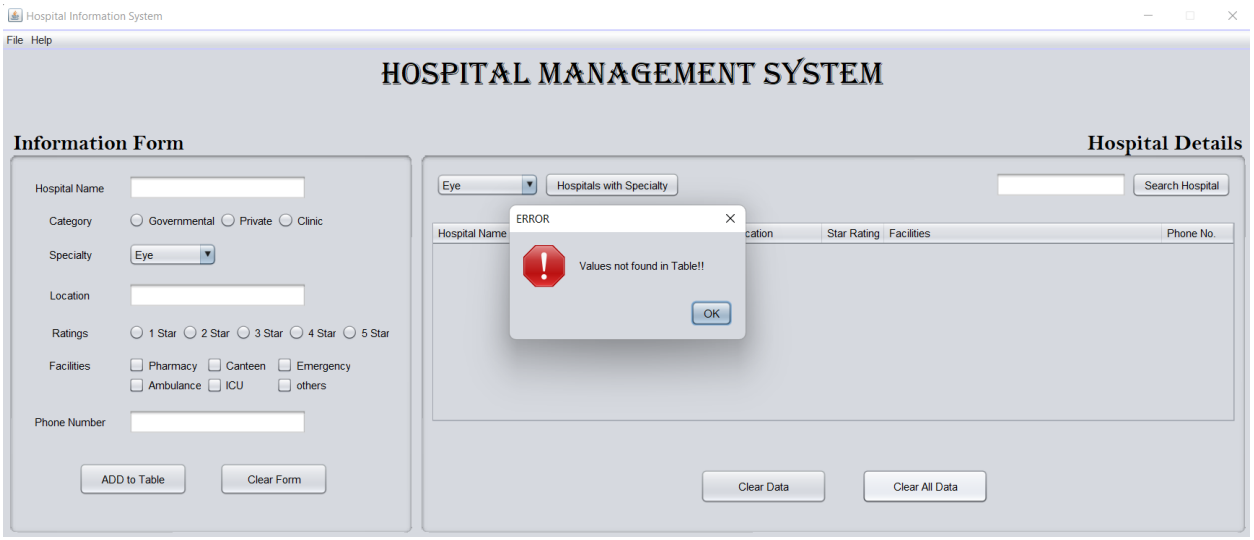



*Figure 6-26: Screenshot for output of test 14*

## 6.15 Test 15

| | |
|---|---|
| Test No: | 15 |
| Objective: | To see what happens when a different value is provided. |
| Action: | A different numeric value was provided and was searched for. |
| Expected Result: | An error message dialog would come. |
| Actual Result: | An error message dialog came. |
| Conclusion: | The test is successful. |

*Table 6-15: Test Table for Test 15*

## Screenshots:

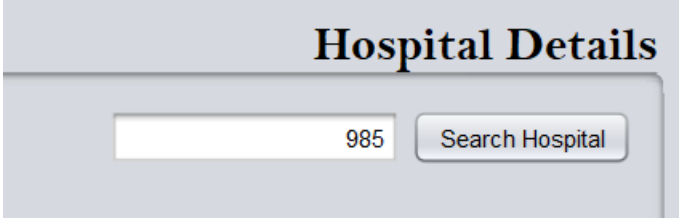



*Figure 6-27: Screenshot of the input given in test 15*

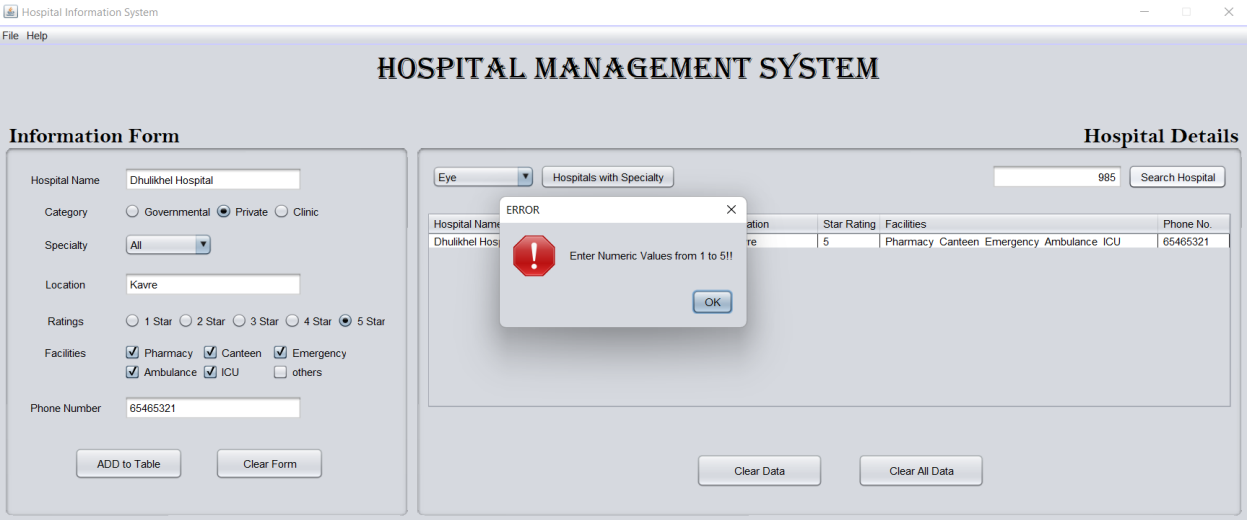



*Figure 6-28: Screenshot for output of test 15*

## 7. Conclusion

In this project, we had followed all the instructions and guidelines provided to make it a success. We had put all my effort and hard work on making this project a good one. Here, we had used several java features to meet the requirements of the project. Several software was used for making this project like NetBeans: for coding, Java JDK for providing a suitable java platform, etc. Similarly, bunch of codes were written to meet the requirement of the project. Several tests and debugging were carried out to ensure the accuracy of the program.

Doing this project on one hand was a hard job but on the other hand it made me to learn several java codes and features. Firstly, we knew about is the concept of GUI (Graphical User Interface) in java. Making a GUI was very simple by using NetBeans. Similarly, there is no need of importing several packages that would be used in the project manually. Not only this, but it also comprises of large number of facilities for handling different sorts of events.

To sum up, although there were many problems while creating the project, we defended those problems by taking reference to our teachers, articles, tutorials, friends, and many sources to bring the project to a success. We learned many things from several sources and used those things to enhance the accuracy of our project. We visited through our lecture slides and got to know about many things which helped us a lot in making this project accurate for acquiring its requirements.

## Appendix

```
import java.awt.Desktop;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */
/**
 *
 * @author
 */
public class HospitalInfo extends javax.swing.JFrame {

    /**
     * Creates new form HospitalInfo
```

```
     */
    public HospitalInfo() {
        initComponents();
        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("Hospital Information System");
        setResizable(false);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        Category_JBG = new javax.swing.ButtonGroup();
        Ratings_JBG = new javax.swing.ButtonGroup();
        jPanel1 = new javax.swing.JPanel();
        HospitalName_JL = new javax.swing.JLabel();
        HospitalName_JTF = new javax.swing.JTextField();
        Location_JTF = new javax.swing.JTextField();
        Category_JL = new javax.swing.JLabel();
        Speciality_JL = new javax.swing.JLabel();
        Location_JL = new javax.swing.JLabel();
        Ratings_JL = new javax.swing.JLabel();
        Facilities_JL = new javax.swing.JLabel();
        PhNo_JL = new javax.swing.JLabel();
        PhNo_JTF = new javax.swing.JTextField();
        Governmental_JRB = new javax.swing.JRadioButton();
```

```
Private_JRB = new javax.swing.JRadioButton();
Clinic_JRB = new javax.swing.JRadioButton();
Speciality_JCombo = new javax.swing.JComboBox<>();
One_JRB = new javax.swing.JRadioButton();
Three_JRB = new javax.swing.JRadioButton();
Two_JRB = new javax.swing.JRadioButton();
Four_JRB = new javax.swing.JRadioButton();
Five_JRB = new javax.swing.JRadioButton();
Pharmacy_JCB = new javax.swing.JCheckBox();
Canteen_JCB = new javax.swing.JCheckBox();
Emergency_JCB = new javax.swing.JCheckBox();
Ambulance_JCB = new javax.swing.JCheckBox();
ICU_JCB = new javax.swing.JCheckBox();
Others_JCB = new javax.swing.JCheckBox();
Add_JB = new javax.swing.JButton();
Clear_JB = new javax.swing.JButton();
Title_JL = new javax.swing.JLabel();
jPanel3 = new javax.swing.JPanel();
jScrollPane1 = new javax.swing.JScrollPane();
Hospital_JTable = new javax.swing.JTable();
Spl_JCB = new javax.swing.JComboBox<>();
SearchSpecialty_JB = new javax.swing.JButton();
Search_TF = new javax.swing.JTextField();
ClearAll_JB = new javax.swing.JButton();
ClearData_JB = new javax.swing.JButton();
Search_BTN = new javax.swing.JButton();
jMenuBar = new javax.swing.JMenuBar();
File_JM = new javax.swing.JMenu();
Open_JMI = new javax.swing.JMenuItem();
Save_JMI = new javax.swing.JMenuItem();
Exit_JMI = new javax.swing.JMenuItem();
```

```
        Help_JM = new javax.swing.JMenu();
        Help_JMI = new javax.swing.JMenuItem();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "Information Form", javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION, javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Bell MT", 1, 24))); // NOI18N

        HospitalName_JL.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        HospitalName_JL.setText("Hospital Name");

        Category_JL.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        Category_JL.setText("Category");

        Speciality_JL.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        Speciality_JL.setText("Specialty");

        Location_JL.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        Location_JL.setText("Location");

        Ratings_JL.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        Ratings_JL.setText("Ratings");

        Facilities_JL.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        Facilities_JL.setText("Facilities");

        PhNo_JL.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        PhNo_JL.setText("Phone Number");
```

```
        Category_JBG.add(Governmental_JRB);
        Governmental_JRB.setText("Governmental");

        Category_JBG.add(Private_JRB);
        Private_JRB.setText("Private");

        Category_JBG.add(Clinic_JRB);
        Clinic_JRB.setText("Clinic");

        Speciality_JCombo.setModel(new javax.swing.DefaultComboBoxModel<>(new
String[] { "Eye", "Heart", "Neuro", "Skin", "Dental", "Cancer", "All" }));

        Ratings_JBG.add(One_JRB);
        One_JRB.setText("1 Star");

        Ratings_JBG.add(Three_JRB);
        Three_JRB.setText("3 Star");

        Ratings_JBG.add(Two_JRB);
        Two_JRB.setText("2 Star");

        Ratings_JBG.add(Four_JRB);
        Four_JRB.setText("4 Star");

        Ratings_JBG.add(Five_JRB);
        Five_JRB.setText("5 Star");

        Pharmacy_JCB.setText("Pharmacy");

        Canteen_JCB.setText("Canteen");
```

```
        Emergency_JCB.setText("Emergency");

        Ambulance_JCB.setText("Ambulance");

        ICU_JCB.setText("ICU");

        Others_JCB.setText("others");

        Add_JB.setText("ADD to Table");
        Add_JB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                Add_JBActionPerformed(evt);
            }
        });

        Clear_JB.setText("Clear Form");
        Clear_JB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                Clear_JBActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
```

```
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                        .addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TR
AILING, false)
                                .addComponent(Facilities_JL,
javax.swing.GroupLayout.DEFAULT_SIZE,   javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                                .addComponent(Ratings_JL,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE,   javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                                .addComponent(Location_JL,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE,   javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                                .addComponent(Speciality_JL,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE,   javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                                .addComponent(Category_JL,
javax.swing.GroupLayout.DEFAULT_SIZE,   javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                                .addComponent(PhNo_JL,
javax.swing.GroupLayout.DEFAULT_SIZE, 100, Short.MAX_VALUE))
                            .addGap(15, 15, 15)
```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                    .addComponent(PhNo_JTF,
javax.swing.GroupLayout.PREFERRED_SIZE,                    197,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                        .addComponent(Pharmacy_JCB)
                        .addComponent(Ambulance_JCB))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                        .addComponent(ICU_JCB)
                        .addComponent(Canteen_JCB))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                        .addComponent(Others_JCB)
                        .addComponent(Emergency_JCB)))

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                    .addGroup(jPanel1Layout.createSequentialGroup()
```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                                        .addGroup(jPanel1Layout.createSequentialGroup()
                                            .addComponent(Governmental_JRB)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                            .addComponent(Private_JRB))
                                        .addComponent(Speciality_JCombo,
javax.swing.GroupLayout.PREFERRED_SIZE,                                 98,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                    .addComponent(Clinic_JRB,
javax.swing.GroupLayout.PREFERRED_SIZE,                                 93,
javax.swing.GroupLayout.PREFERRED_SIZE))
                                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
                                    .addComponent(One_JRB)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                    .addComponent(Two_JRB)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                    .addComponent(Three_JRB)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                    .addComponent(Four_JRB)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                    .addComponent(Five_JRB))
```

```
                        .addComponent(Location_JTF,
javax.swing.GroupLayout.PREFERRED_SIZE,                                    197,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(HospitalName_JTF,
javax.swing.GroupLayout.PREFERRED_SIZE,                                    197,
javax.swing.GroupLayout.PREFERRED_SIZE))))
                    .addComponent(HospitalName_JL,
javax.swing.GroupLayout.PREFERRED_SIZE,                                    100,
javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addGap(66, 66, 66)
                    .addComponent(Add_JB, javax.swing.GroupLayout.PREFERRED_SIZE,
120, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(36, 36, 36)
                    .addComponent(Clear_JB,
javax.swing.GroupLayout.PREFERRED_SIZE,                                    120,
javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addContainerGap(33, Short.MAX_VALUE))
        );
        jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BA
SELINE)
                .addComponent(HospitalName_JL)
                .addComponent(HospitalName_JTF,
javax.swing.GroupLayout.PREFERRED_SIZE,
```

```
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(13, 13, 13)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BA
SELINE)
                    .addComponent(Category_JL)
                    .addComponent(Governmental_JRB)
                    .addComponent(Private_JRB)
                    .addComponent(Clinic_JRB))
                .addGap(16, 16, 16)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BA
SELINE)
                    .addComponent(Speciality_JL)
                    .addComponent(Speciality_JCombo,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BA
SELINE)
                    .addComponent(Location_JL)
                    .addComponent(Location_JTF,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)
```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BA
SELINE)
                .addComponent(Ratings_JL)
                .addComponent(One_JRB)
                .addComponent(Two_JRB)
                .addComponent(Three_JRB)
                .addComponent(Four_JRB)
                .addComponent(Five_JRB))
            .addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BA
SELINE)
                .addComponent(Facilities_JL)
                .addComponent(Pharmacy_JCB)
                .addComponent(Canteen_JCB)
                .addComponent(Emergency_JCB))
            .addGap(4, 4, 4)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BA
SELINE)
                .addComponent(Ambulance_JCB)
                .addComponent(ICU_JCB)
                .addComponent(Others_JCB))
            .addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BA
SELINE)
                .addComponent(PhNo_JL)
                .addComponent(PhNo_JTF,
javax.swing.GroupLayout.PREFERRED_SIZE,
```

```
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(31, 31, 31)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BA
SELINE)
                    .addComponent(Add_JB,     javax.swing.GroupLayout.PREFERRED_SIZE,
36, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(Clear_JB,   javax.swing.GroupLayout.PREFERRED_SIZE,
36, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(27, 27, 27))
        );

        Title_JL.setFont(new java.awt.Font("Algerian", 0, 36)); // NOI18N
        Title_JL.setText("Hospital Management System");

        jPanel3.setBorder(javax.swing.BorderFactory.createTitledBorder(null,      "Hospital
Details",                                    javax.swing.border.TitledBorder.RIGHT,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Bell MT", 1,
24))); // NOI18N

        Hospital_JTable.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {

            },
            new String [] {
                "Hospital   Name",   "Category",   "Specialty",   "Location",   "Star   Rating",
"Facilities", "Phone No."
            }
        ) {
            boolean[] canEdit = new boolean [] {
```

```
            false, false, false, false, false, false, false
        };

        public boolean isCellEditable(int rowIndex, int columnIndex) {
            return canEdit [columnIndex];
        }
    });

Hospital_JTable.setAutoResizeMode(javax.swing.JTable.AUTO_RESIZE_ALL_COLUMNS);
    Hospital_JTable.setShowGrid(true);
    Hospital_JTable.setShowHorizontalLines(false);
    Hospital_JTable.getTableHeader().setReorderingAllowed(false);
    jScrollPane1.setViewportView(Hospital_JTable);
    if (Hospital_JTable.getColumnModel().getColumnCount() > 0) {
        Hospital_JTable.getColumnModel().getColumn(0).setResizable(false);
        Hospital_JTable.getColumnModel().getColumn(0).setPreferredWidth(100);
        Hospital_JTable.getColumnModel().getColumn(1).setResizable(false);
        Hospital_JTable.getColumnModel().getColumn(1).setPreferredWidth(45);
        Hospital_JTable.getColumnModel().getColumn(2).setResizable(false);
        Hospital_JTable.getColumnModel().getColumn(2).setPreferredWidth(20);
        Hospital_JTable.getColumnModel().getColumn(3).setResizable(false);
        Hospital_JTable.getColumnModel().getColumn(3).setPreferredWidth(50);
        Hospital_JTable.getColumnModel().getColumn(4).setResizable(false);
        Hospital_JTable.getColumnModel().getColumn(4).setPreferredWidth(15);
        Hospital_JTable.getColumnModel().getColumn(5).setResizable(false);
        Hospital_JTable.getColumnModel().getColumn(5).setPreferredWidth(255);
        Hospital_JTable.getColumnModel().getColumn(6).setResizable(false);
        Hospital_JTable.getColumnModel().getColumn(6).setPreferredWidth(25);
    }
```

```
        Spl_JCB.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Eye", "Heart", "Neuro", "Skin", "Dental", "Cancer", "All" }));

        SearchSpecialty_JB.setText("Hospitals with Specialty");
        SearchSpecialty_JB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                SearchSpecialty_JBActionPerformed(evt);
            }
        });

        Search_TF.setHorizontalAlignment(javax.swing.JTextField.RIGHT);
        Search_TF.setName(""); // NOI18N

        ClearAll_JB.setText("Clear All Data");
        ClearAll_JB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                ClearAll_JBActionPerformed(evt);
            }
        });

        ClearData_JB.setText("Clear Data");
        ClearData_JB.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                ClearData_JBActionPerformed(evt);
            }
        });

        Search_BTN.setText("Search Hospital");
        Search_BTN.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                Search_BTNActionPerformed(evt);
```

```
        }
    });

    javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);
    jPanel3.setLayout(jPanel3Layout);
    jPanel3Layout.setHorizontalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel3Layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(Spl_JCB,       javax.swing.GroupLayout.PREFERRED_SIZE,
114, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(SearchSpecialty_JB)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
350, Short.MAX_VALUE)
            .addComponent(Search_TF,    javax.swing.GroupLayout.PREFERRED_SIZE,
145, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(Search_BTN)
            .addContainerGap())
        .addComponent(jScrollPane1)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel3Layout.createSequentialGroup()
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
            .addComponent(ClearData_JB,
javax.swing.GroupLayout.PREFERRED_SIZE,                    140,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(39, 39, 39)
```

```
            .addComponent(ClearAll_JB,   javax.swing.GroupLayout.PREFERRED_SIZE,
140, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(275, 275, 275))
      );
      jPanel3Layout.setVerticalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
         .addGroup(jPanel3Layout.createSequentialGroup()
            .addContainerGap()

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BA
SELINE)
               .addComponent(Spl_JCB,   javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
               .addComponent(SearchSpecialty_JB)
               .addComponent(Search_TF,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
               .addComponent(Search_BTN))
            .addGap(26, 26, 26)
            .addComponent(jScrollPane1,  javax.swing.GroupLayout.PREFERRED_SIZE,
223, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
51, Short.MAX_VALUE)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BA
SELINE)
```

```
                .addComponent(ClearData_JB,
javax.swing.GroupLayout.PREFERRED_SIZE,                                    38,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(ClearAll_JB,
javax.swing.GroupLayout.PREFERRED_SIZE,                                    38,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(24, 24, 24))
        );

        jMenuBar.setBorder(new  javax.swing.border.LineBorder(new  java.awt.Color(204,
204, 255), 2, true));

        File_JM.setText("File");


Open_JMI.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEven
t.VK_O, java.awt.event.InputEvent.CTRL_DOWN_MASK));
        Open_JMI.setText("Open");
        Open_JMI.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                Open_JMIActionPerformed(evt);
            }
        });
        File_JM.add(Open_JMI);


Save_JMI.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEven
t.VK_S, java.awt.event.InputEvent.CTRL_DOWN_MASK));
        Save_JMI.setText("Save");
        Save_JMI.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
            Save_JMIActionPerformed(evt);
        }
    });
    File_JM.add(Save_JMI);


Exit_JMI.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.
VK_E, java.awt.event.InputEvent.CTRL_DOWN_MASK));
    Exit_JMI.setText("Exit");
    Exit_JMI.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            Exit_JMIActionPerformed(evt);
        }
    });
    File_JM.add(Exit_JMI);

    jMenuBar.add(File_JM);

    Help_JM.setText("Help");


Help_JMI.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent
.VK_H, java.awt.event.InputEvent.CTRL_DOWN_MASK));
    Help_JMI.setText("Help");
    Help_JMI.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            Help_JMIActionPerformed(evt);
        }
    });
    Help_JM.add(Help_JMI);
```

```
        jMenuBar.add(Help_JM);

        setJMenuBar(jMenuBar);

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(3, 3, 3)
                .addComponent(jPanel3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                .addComponent(Title_JL)
                .addGap(411, 411, 411))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
```

```
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
            .addComponent(Title_JL)
            .addGap(35, 35, 35)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
                .addComponent(jPanel1,          javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jPanel3,     javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addContainerGap())
        );

        pack();
    }// </editor-fold>

    private void SearchSpecialty_JBActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        try {
            int i = 0;
            ArrayList<String> list = new ArrayList<String>();
            String spl = Spl_JCB.getSelectedItem().toString();
            while (i < Hospital_JTable.getRowCount()) {
                String s = (String) Hospital_JTable.getValueAt(i, 2);
                if (s != null && s.length() != 0) {
                    list.add(s);
                }
                i++;
            }
```

```
        String search = "";
        int cnt = 0;
        int j = 0;

        for (String c : list) {
            if (spl.equals(c)) {
                if (cnt == 0) {
                    search = search + Hospital_JTable.getValueAt(j, 0);
                } else {
                    search = search + ", " + Hospital_JTable.getValueAt(j, 0);
                }
                cnt++;
            }
            j++;
        }

        if (cnt == 0) {
            JOptionPane.showMessageDialog(this, "No match found!!", "Information",
JOptionPane.INFORMATION_MESSAGE);
        } else {
            JOptionPane.showMessageDialog(this, "Total found Hospital: " + cnt +
"\nFound Hospital Name:- " + "\n" + search, "Information",
JOptionPane.INFORMATION_MESSAGE);
        }
      } catch (Exception expt) {
        JOptionPane.showMessageDialog(this, "Table found Empty!!", "Information",
JOptionPane.INFORMATION_MESSAGE);
      }
   }
```

```
    private void Save_JMIActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        JFileChooser chooser = new JFileChooser();
        chooser.setCurrentDirectory(new File("."));
        int reply = chooser.showSaveDialog(null);
        if (reply == JFileChooser.APPROVE_OPTION) {
            File savefile = chooser.getSelectedFile();
            try {
                FileWriter fw = new FileWriter(savefile);
                BufferedWriter bw = new BufferedWriter(fw);
                for (int i = 0; i < Hospital_JTable.getRowCount(); i++) {
                    for (int j = 0; j < Hospital_JTable.getColumnCount(); j++) {
                        bw.write((String) Hospital_JTable.getValueAt(i, j) + ",");
                    }
                    bw.newLine();
                }
                bw.close();
                fw.close();
            } catch (IOException ex) {
                JOptionPane.showMessageDialog(this, "ERROR", "ERROR MESSAGE",
JOptionPane.ERROR_MESSAGE);
            }
        }
    }

    private void Exit_JMIActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        System.exit(0);
    }

    private void Open_JMIActionPerformed(java.awt.event.ActionEvent evt) {
```

```
        // TODO add your handling code here:
        JFileChooser chooser = new JFileChooser();
        chooser.setCurrentDirectory(new File("."));
        int reply = chooser.showOpenDialog(null);
        DefaultTableModel model = (DefaultTableModel) Hospital_JTable.getModel();
        if (reply == JFileChooser.APPROVE_OPTION) {
            File file = new File(chooser.getSelectedFile().getAbsolutePath());
            try {
                BufferedReader br = new BufferedReader(new FileReader(file));
                Object[] row = br.lines().toArray();
                for (int i = 0; i < row.length; i++) {
                    String line = row[i].toString().trim();
                    String[] data = line.split(",");
                    model.addRow(data);
                }
            } catch (FileNotFoundException ex) {
                Logger.getLogger(HospitalInfo.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
    }

    private void Add_JBActionPerformed(java.awt.event.ActionEvent evt) {
        String hospitalName = HospitalName_JTF.getText();
        String location = Location_JTF.getText();
        String number = PhNo_JTF.getText();

        if (hospitalName.isBlank() || location.isBlank() || number.isBlank()) {
            JOptionPane.showMessageDialog(this, "Textfields left empty!!", "ERROR",
JOptionPane.ERROR_MESSAGE);
        } else {
            try {
```

```
int x = 0;
boolean present = false;
ArrayList<String> al = new ArrayList<String>();
while (x < Hospital_JTable.getRowCount()) {
    String s = (String) Hospital_JTable.getValueAt(x, 0);
    if (s != null && s.length() != 0) {
        al.add(s);
    }
    x++;
}

Governmental_JRB.setActionCommand("Governmental");
Private_JRB.setActionCommand("Private");
Clinic_JRB.setActionCommand("Clinic");
String category = Category_JBG.getSelection().getActionCommand();
String specialty = (String) Speciality_JCombo.getSelectedItem();
One_JRB.setActionCommand("1");
Two_JRB.setActionCommand("2");
Three_JRB.setActionCommand("3");
Four_JRB.setActionCommand("4");
Five_JRB.setActionCommand("5");
String facilities = "";

try {
    for (String name : al) {
        if (hospitalName.equals(name)) {
            present = true;
        }
    }
    if (present == true) {
```

```
                JOptionPane.showMessageDialog(this, "Hospital Name already exists!!",
"ERROR", JOptionPane.ERROR_MESSAGE);
            } else {
                int rating =
Integer.parseInt(Ratings_JBG.getSelection().getActionCommand());
                int phone = Integer.parseInt(number);
                DefaultTableModel def_model = (DefaultTableModel)
Hospital_JTable.getModel();
                def_model.addRow(new Object[]{null, null, null, null, null, null, null});

                if (Pharmacy_JCB.isSelected()) {
                    facilities += Pharmacy_JCB.getText() + "  ";
                }
                if (Canteen_JCB.isSelected()) {
                    facilities += Canteen_JCB.getText() + "  ";
                }
                if (Emergency_JCB.isSelected()) {
                    facilities += Emergency_JCB.getText() + "  ";
                }
                if (Ambulance_JCB.isSelected()) {
                    facilities += Ambulance_JCB.getText() + "  ";
                }
                if (ICU_JCB.isSelected()) {
                    facilities += ICU_JCB.getText() + "  ";
                }
                if (Others_JCB.isSelected()) {
                    facilities += Others_JCB.getText();
                }

                String[] value = {hospitalName, category, specialty, location,
String.valueOf(rating).toString(), facilities, number};
```

```
                int rows = Hospital_JTable.getRowCount();
                int columns = Hospital_JTable.getColumnCount();
                int index = 0;
                do {
                    String s = (String) Hospital_JTable.getValueAt(index, 0);
                    if (s != null && s.length() != 0) {
                        index++;
                    } else {
                        for (int i = 0; i < columns; i++) {
                            Hospital_JTable.setValueAt(value[i], index, i);
                        }
                        break;
                    }
                } while (index < rows);
            }
        } catch (NumberFormatException expt) {
            JOptionPane.showMessageDialog(this, "Phone Number must be an integer value!!", "ERROR", JOptionPane.ERROR_MESSAGE);
        }
    } catch (NullPointerException e) {
        JOptionPane.showMessageDialog(this, "Enter all the values first!!", "ERROR", JOptionPane.ERROR_MESSAGE);
    }
  }
}

  private void Clear_JBActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    HospitalName_JTF.setText(null);
    Category_JBG.clearSelection();
    Speciality_JCombo.setSelectedIndex(0);
```

```
        Location_JTF.setText(null);
        Ratings_JBG.clearSelection();
        Pharmacy_JCB.setSelected(false);
        Canteen_JCB.setSelected(false);
        Emergency_JCB.setSelected(false);
        Ambulance_JCB.setSelected(false);
        ICU_JCB.setSelected(false);
        Others_JCB.setSelected(false);
        PhNo_JTF.setText(null);
        JOptionPane.showMessageDialog(this, "Form Cleared!!");
    }

    private void ClearAll_JBActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        DefaultTableModel def_model = (DefaultTableModel) Hospital_JTable.getModel();

        if (def_model.getRowCount() == 0) {
            JOptionPane.showMessageDialog(this, "Values not found in Table!!", "ERROR",
JOptionPane.ERROR_MESSAGE);
        } else {
            def_model.setRowCount(0);
            JOptionPane.showMessageDialog(this, "Values in Table deleted
successfully!!");
        }
    }

    private void ClearData_JBActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        DefaultTableModel def_model = (DefaultTableModel) Hospital_JTable.getModel();

        if (Hospital_JTable.getSelectedRow() != -1) {
```

```
            // remove the selected row from the table model
            def_model.removeRow(Hospital_JTable.getSelectedRow());
            JOptionPane.showMessageDialog(this, "Selected Row Deleted successfully");
        } else {
            JOptionPane.showMessageDialog(this, "Please select a row in table.");
        }
    }

    private void Search_BTNActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        try {
            int search = Integer.parseInt(Search_TF.getText());
            if (0 < search && search < 6) {
                String name = "";
                int note = 0;
                ArrayList<Integer> rate = new ArrayList<Integer>();

                int x = 0;
                while (x < Hospital_JTable.getRowCount()) {
                    int s = Integer.parseInt((String) Hospital_JTable.getValueAt(x, 4));
                    rate.add(s);
                    x++;
                }

                Integer arr[] = rate.toArray(new Integer[0]);

                for (int i = 0; i < arr.length - 1; i++) {
                    int min = i;
                    for (int j = i + 1; j < arr.length; j++) {
                        if (arr[j] < arr[min]) {
                            min = j;
```

```
                }
            }
            if (i != min) {
                int temp = arr[i];
                arr[i] = arr[min];
                arr[min] = temp;
            }
        }

        int start = 0;
        int end = arr.length - 1;
        int result = BinarySearch(arr, start, end, search);
        if (result == -1) {
            JOptionPane.showMessageDialog(this, "No Match found!!");
        } else {
            for (Integer each : rate) {
                if (arr[result] == each) {
                    note++;
                }
            }

            for (int i = 0; i < Hospital_JTable.getRowCount(); i++) {
                if  (arr[result]  ==  Integer.parseInt((String)  Hospital_JTable.getValueAt(i,
4))) {
                    name = name + Hospital_JTable.getValueAt(i, 0);
                    break;
                }
            }
```

```
                JOptionPane.showMessageDialog(this, "Total found Hospital: " + note + "\nFound Hospital Name:- " + "\n" + name, "Information", JOptionPane.INFORMATION_MESSAGE);

            }
        } else {
            JOptionPane.showMessageDialog(this, "Enter Numeric Values from 1 to 5!!", "ERROR", JOptionPane.ERROR_MESSAGE);
        }
    } catch (NumberFormatException e) {
        JOptionPane.showMessageDialog(this, "Numeric Values Required!!", "ERROR", JOptionPane.ERROR_MESSAGE);
    }
  }

  private void Help_JMIActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        if (Desktop.isDesktopSupported()) {
            File help = new File("User Manual.pdf");
            Desktop.getDesktop().open(help);
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(this, "File not found!!!", "ERROR", JOptionPane.ERROR_MESSAGE);
    }
  }

  public int BinarySearch(Integer arr[], int start, int end, int search) {
    if (start <= end) {
        int mid = (start + end) / 2;
        if (arr[mid] == search) {
```

```
            return mid;
        } else if (search < arr[mid]) {
            return BinarySearch(arr, start, mid - 1, search);
        } else {
            return BinarySearch(arr, mid + 1, end, search);
        }
    }
    return -1;
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    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(HospitalInfo.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(HospitalInfo.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(HospitalInfo.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(HospitalInfo.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new HospitalInfo().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton Add_JB;
    private javax.swing.JCheckBox Ambulance_JCB;
    private javax.swing.JCheckBox Canteen_JCB;
    private javax.swing.ButtonGroup Category_JBG;
```

```
private javax.swing.JLabel Category_JL;
private javax.swing.JButton ClearAll_JB;
private javax.swing.JButton ClearData_JB;
private javax.swing.JButton Clear_JB;
private javax.swing.JRadioButton Clinic_JRB;
private javax.swing.JCheckBox Emergency_JCB;
private javax.swing.JMenuItem Exit_JMI;
private javax.swing.JLabel Facilities_JL;
private javax.swing.JMenu File_JM;
private javax.swing.JRadioButton Five_JRB;
private javax.swing.JRadioButton Four_JRB;
private javax.swing.JRadioButton Governmental_JRB;
private javax.swing.JMenu Help_JM;
private javax.swing.JMenuItem Help_JMI;
private javax.swing.JLabel HospitalName_JL;
private javax.swing.JTextField HospitalName_JTF;
private javax.swing.JTable Hospital_JTable;
private javax.swing.JCheckBox ICU_JCB;
private javax.swing.JLabel Location_JL;
private javax.swing.JTextField Location_JTF;
private javax.swing.JRadioButton One_JRB;
private javax.swing.JMenuItem Open_JMI;
private javax.swing.JCheckBox Others_JCB;
private javax.swing.JLabel PhNo_JL;
private javax.swing.JTextField PhNo_JTF;
private javax.swing.JCheckBox Pharmacy_JCB;
private javax.swing.JRadioButton Private_JRB;
private javax.swing.ButtonGroup Ratings_JBG;
private javax.swing.JLabel Ratings_JL;
private javax.swing.JMenuItem Save_JMI;
private javax.swing.JButton SearchSpecialty_JB;
```

```
    private javax.swing.JButton Search_BTN;
    private javax.swing.JTextField Search_TF;
    private javax.swing.JComboBox<String> Speciality_JCombo;
    private javax.swing.JLabel Speciality_JL;
    private javax.swing.JComboBox<String> Spl_JCB;
    private javax.swing.JRadioButton Three_JRB;
    private javax.swing.JLabel Title_JL;
    private javax.swing.JRadioButton Two_JRB;
    private javax.swing.JMenuBar jMenuBar;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel3;
    private javax.swing.JScrollPane jScrollPane1;
    // End of variables declaration
}
```