HOUSEBOAT MANAGEMENT SYSTEM

GROUP MEMBERS:

ABHIJITH SANTHOSH NARAYANAN(AM.EN.U4CSE20101)

ARAVIND RAJESH(AM.EN.U4CSE20112)

ROHITH K P(AM.EN.U4CSE20158)

COMPUTER SCIENCE (B-BATCH) DEPT.

AMRITA VISHWA VIDYAPEETHAM, KOLLAM

DECEMBER,2021

ABSTRACT

The System has been designed and developed to provide customers with an efficient and user-friendly portal for House Boat Reservation. The system mainly focuses on the customer and the boat owners. Boat owners can register and their boats to manage the boats and book the house boat with ease. A user can login or signup into the system with their details as an admin, Boat owner or customer where they will be provided with different access levels.

The house boat owner can register in the system providing all details including his licence number. They will be able to register any number of boats owned by them with the boat id. He can manage the boat fare. They will be able to add boats or delete the boat if they required.

The customer needs to create a new account or to sign in for booking house boats. The customer will be able to book the boat according to the fare and the availability. After the booking the customer will be provided with the payment option. The customer can also update or delete the booking.

The admin will have the ability to manage both the registration of the owner and the booking of the house boats by the customers. The admin will have the access to delete or add new boat registration. The admin can also be able to modify the booking by any customer.

We have successfully created a java project for houseboat management where the customers can login or create account which will be stored in the database and also can retrieve information from the database and the boat owners can successfully add their boat to the booking system. Using this the customers can book their trip with ease and the boat owners can manage their boat details smoothly.

INDEX

INTRODUCTION

The system has been designed and developed to provide customers with an efficient and user-friendly portal for Houseboat reservation. Here, the boat owners can register and manage their boats and the user can easily book these boats on any particular date. A common login page is provided for all the users and are further redirected to their respective pages.

PROBLEM STATEMENT:

An agency keeps track of house boats, its owners and the customers who rented it. An owner identified by owner number, name and city can have many house boats and a house boat is owned by only 1 person. A house boat is identified by an id, name and capacity (number of people it can hold). A customer can rent many house boats and a house boat can be rented by many customers on different dates.

Design a java application which allows customers and owners of the house boats efficiently book and rent the houseboats for the tourism promotion.

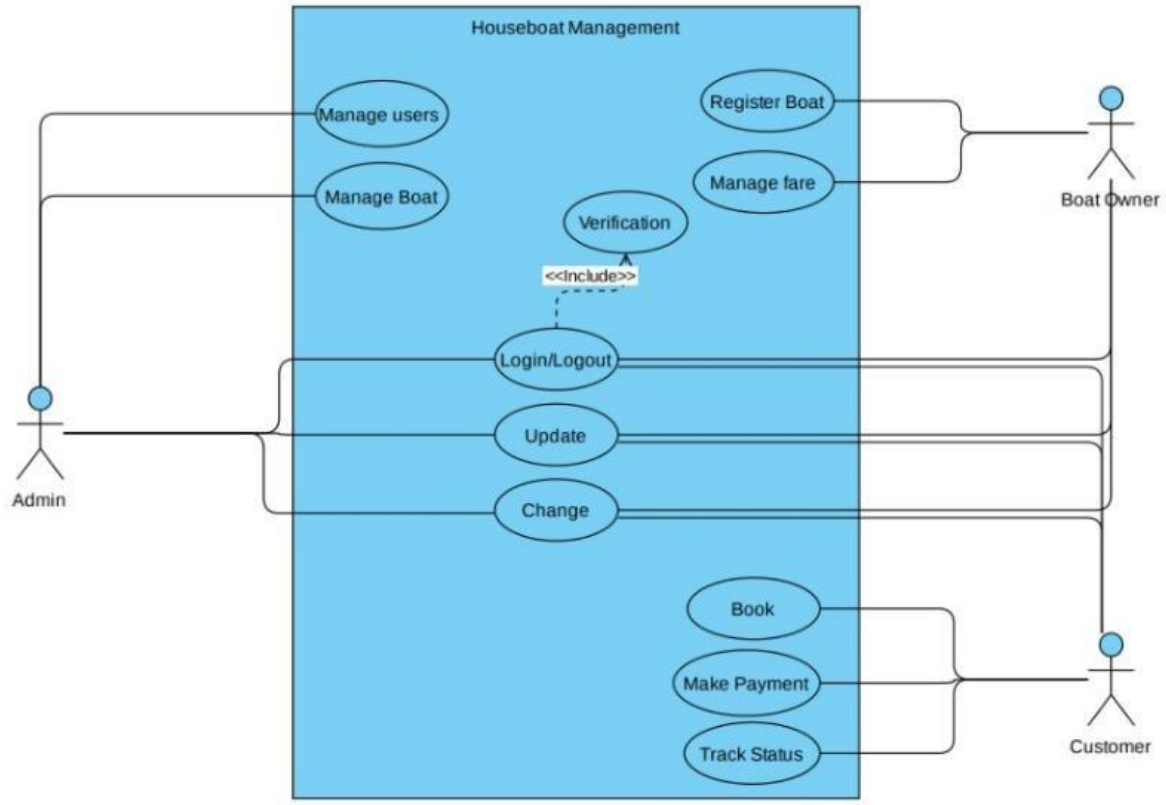
The customer can create a new account and sign-in using the existing account for both booking the houseboat and tracking the booking status. They will be redirected to the customer page from where they can book houseboats. After booking, they are redirected to the payment page. Then for tracking the status of their booking report, a separate page is provided.

The houseboat owner can also create account and sign-in in the same way as that of the customer. They are then redirected to the owner's page from where they can register their boats. They can register any number of boats and can also manage the boat's fare and other facilities. The registration details will be displayed on owner's page.

The admin can manage both the customer details and the boat details and can make necessary updates. After logging-in, they are redirected to the admin panel.

SYSTEM DESIGN

USECASE DIAGRAM:



USE CASE DESCRIPTION:

USE CASE	MANAGE USERS
ACTOR	ADMIN
INPUT	CUSTOMER DETAILS
OUTPUT	SYSTEM WILL VERIFY AND UPDATE THE DETAILS
DESCRIPTION	THE SYSTEM WILL VERIFY THE DETAILS AND UPDATE THE CORRESOPNDING CUSTOMER DETAILS AND THEIR BOOKING DETAILS

USE CASE	MANAGE BOATS
ACTOR	ADMIN
INPUT	BOAT DETAILS

OUTPUT	SYSTEM WILL VERIFY AND UPDATE THE DETAILS
DESCRIPTION	THE SYSTEM WILL VERIFY THE DETAILS AND UPDATE THE CORRESOPNDING BOAT DETAILS AND THEIR BOOKING DETAILS. THIS HELPS THE ADMIN TO NOT ONLY UPDATE BUT ALSO DELETE THE BOAT DETAILS IF REQUIRED

USE CASE	LOGIN
ACTOR	ADMIN , CUSTOMER , BOAT OWNER
INPUT	PASSWORD AND USER NAME
OUTPUT	DETAILED VERIFIED OR NOT
DESCRIPTION	THE SYSTEM WILL CHECK THE USER NAME AND PASSSWORD. THE SYSTEM WILL VALIDATE THE INPUTS AND GRAND ACCESS IF THE DETAILS ARE CORRECT AND DENY IF THE DETAILS ARE WRONG.

USE CASE	UPDATE
ACTOR	ADMIN , CUSTOMER , BOAT OWNER
INPUT	DETAILS OF ADMIN CUSTOMER OR THE BOAT OWNER
OUTPUT	DETAILES UPDATED
DESCRIPTION	AFTER THE LOGIN PROCESSS IF THE ACTOR WILL BE ABLE TO UPDATE AND CHANGE THEIR DETAILS IF REQUIRED.

USE CASE	UPDATE
ACTOR	ADMIN , CUSTOMER , BOAT OWNER
INPUT	DETAILS OF ADMIN CUSTOMER OR THE BOAT OWNER
OUTPUT	DETAILES UPDATED
DESCRIPTION	AFTER THE LOGIN PROCESSS IF THE ACTOR WILL BE ABLE TO UPDATE AND CHANGE THEIR DETAILS IF REQUIRED.

USE CASE	REGISTER BOAT
ACTOR	BOAT OWNER
INPUT	BOAT DETAILS SUCH AS BOAT LICENCE NUMBER AND CAPACITY

OUTPUT	BOAT REGISTERED SUCCESSFULLY
DESCRIPTION	THE BOAT OWNER CAN ADD THE BOAT DETAILS SUCH AS THE BOAT LICENCE NUMBER,BOAT CAPACITY FOR THE CUSTOMERS TO CHOOSE FROM

USE CASE	MANAGE FARE
ACTOR	BOAT OWNER
INPUT	FARES FOR THE SELECTED ROUTE
OUTPUT	DETAILES UPDATED
DESCRIPTION	THE BOAT OWNER CAN ADD,UODATE OR CHANGE THE FARES OF THE CORRESPONDING ROUTE THE CUSTOMER SELECTS.

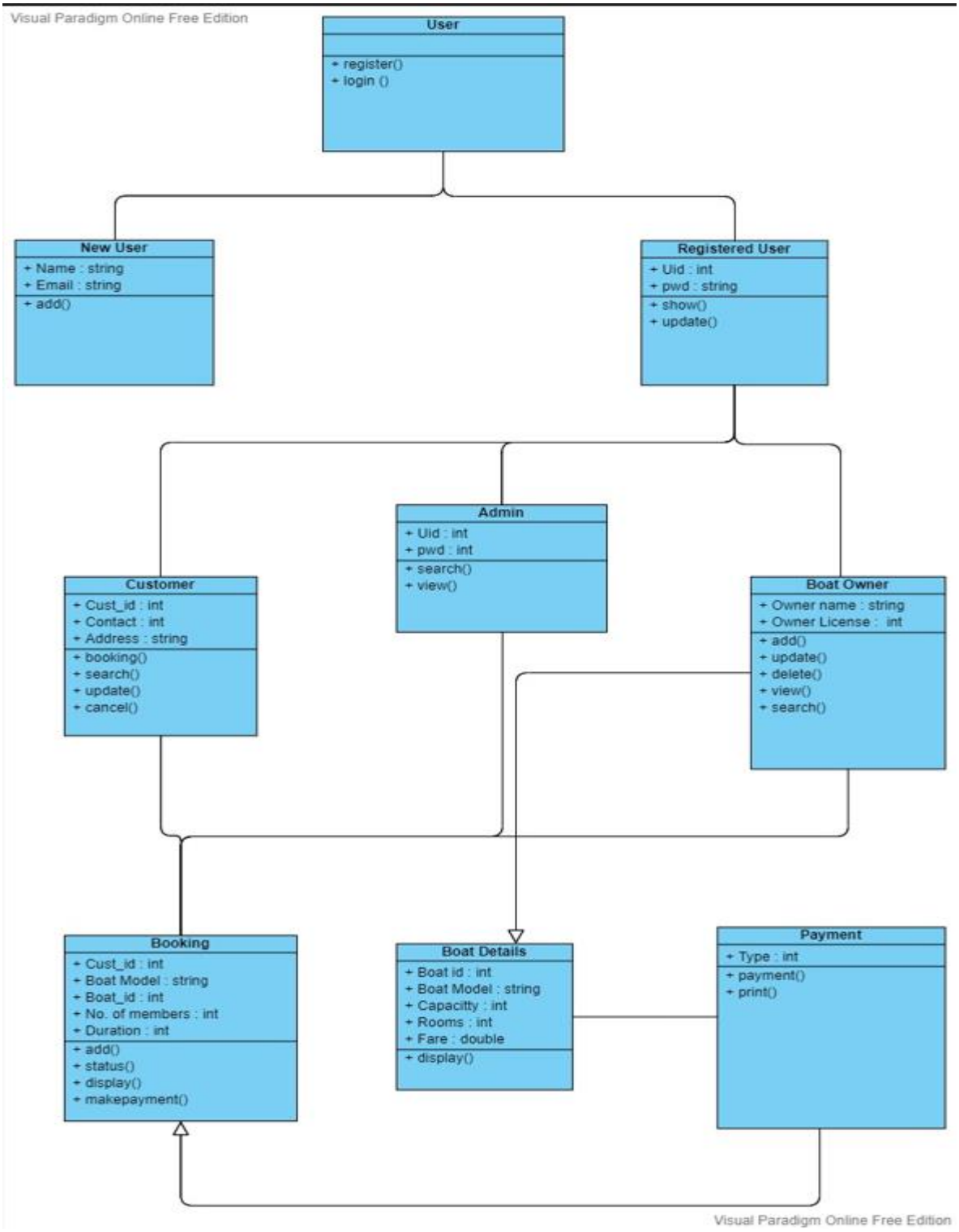
USE CASE	BOOK
ACTOR	CUSTOMER
INPUT	DETAILS OF CUSTOMER,DATE,BOAT SELECTION,ROUTE SELECTION
OUTPUT	BOOKING SUCCESSFULL
DESCRIPTION	THE CUSTOMERS WILL INPUT AND SELECT THE DETAILS FOR BOOKING SUCH AS CUSTOMER DETAILS ,DATE, ROUTE FOR THE SUCCESSFUL COMPLETION OF THE HOUSEBOAT BOOKING

USE CASE	PAYMENT
ACTOR	CUSTOMER
INPUT	BILLING METHOD, PAYMENT DETAILS
OUTPUT	PAYMENT SUCCESSFULL
DESCRIPTION	AFTER THE BOOKING PROCESS IS THE BILLING PROCESS WHERE THE CUSTOMER PAYS THE REQUIRED FARE ACCORDING TO THE ROUTE THE HOUSE BOAT THE CUSTOMER SELECTED

USE CASE	TRACK
ACTOR	CUSTOMER
INPUT	BOOKING DETAILS
OUTPUT	BOOKING STATUS

DESCRIPTION	AFTER THE COMPLETE BOOKING PROCESS THE CUSTOMER CAN CHECK THE BOOKING STATUS
	SYCH AS WHETHER THERES ANY DELAY OR CANCELLATION OF THEIR BOOKING

CLASS DIAGRAM:



User

+register()	To register new users
+login()	To sign-in by the registered users.

New User

+Name	To store the name of the user
+Email	To store the email-id of the user
+add()	To add the new user

Registered User

+Uid	To store the username of the registered user.
+pwd	To store the password of registered user
+show()	To display the user details of the registered user
+update()	To update user details

Admin

+Uid	To store the username of admin
+pwd	To store the password of admin
+search()	To search for a registered user
+view()	To display the user details

Customer

+Cust_id	To store the username of customer
+Contact	To store the password of customer
+Address	To store the address of the customer
+booking()	To book for a boat
+search()	To search for available boats
+update()	To update customer details
+cancel()	To cancel the booking

Boat Owner

+Owner name	To store the username of Boat owner
+Owner_License	To store the license.no of boat owner
+add()	To add a boat
+delete()	To delete an existing boat
+update()	To update boat owner details
+view()	To view the registered boat details
+search()	To search for the details of a specific boat

Booking

+Cust_id	To store the username of customer
+Boat_Model	To store the model name of the boat
+Boat_id	To store the id of the boat
+No_of_members	To store the no of passengers
+Duration	To store the duration of boating
+add()	To book for a boat
+status()	To view the status of the customer
+display()	To display the booking details
+makepayment()	To make the payment of booking

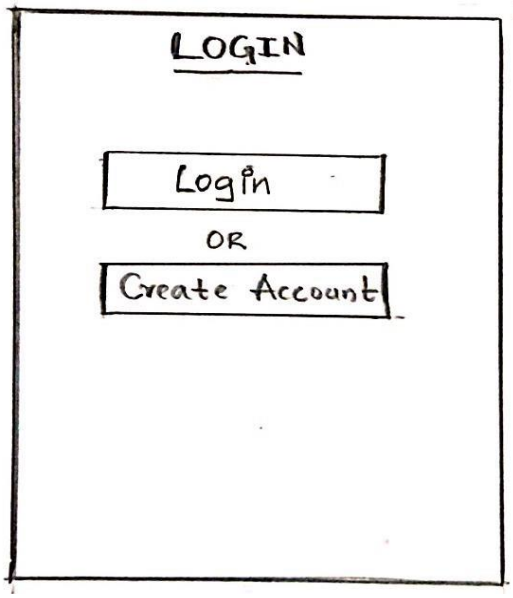
Boat details

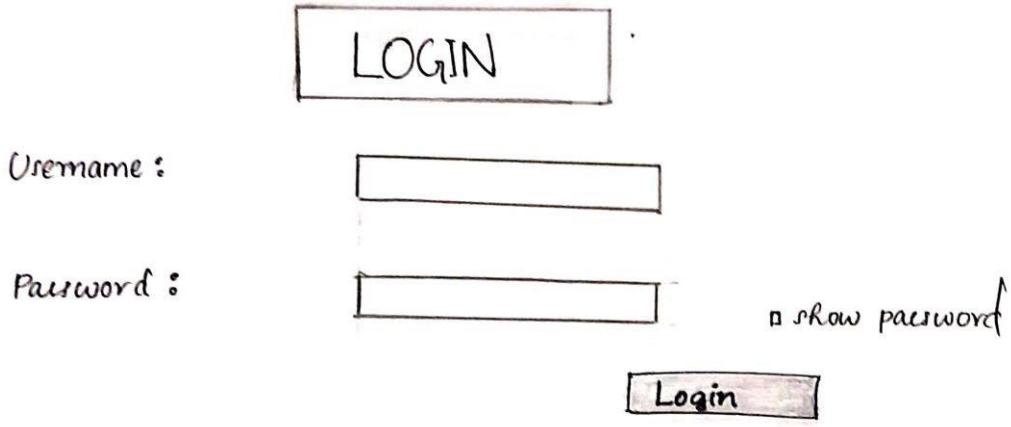
+Boat_id	To store the Boat ID
+Boat_Model	To store the model name of the boat
+Capacity	To store maximum no of persons that can be accommodated
+Rooms	To store the no of rooms available
+Fare	To store the fixed rate of the boat
+display()	To display the boat details

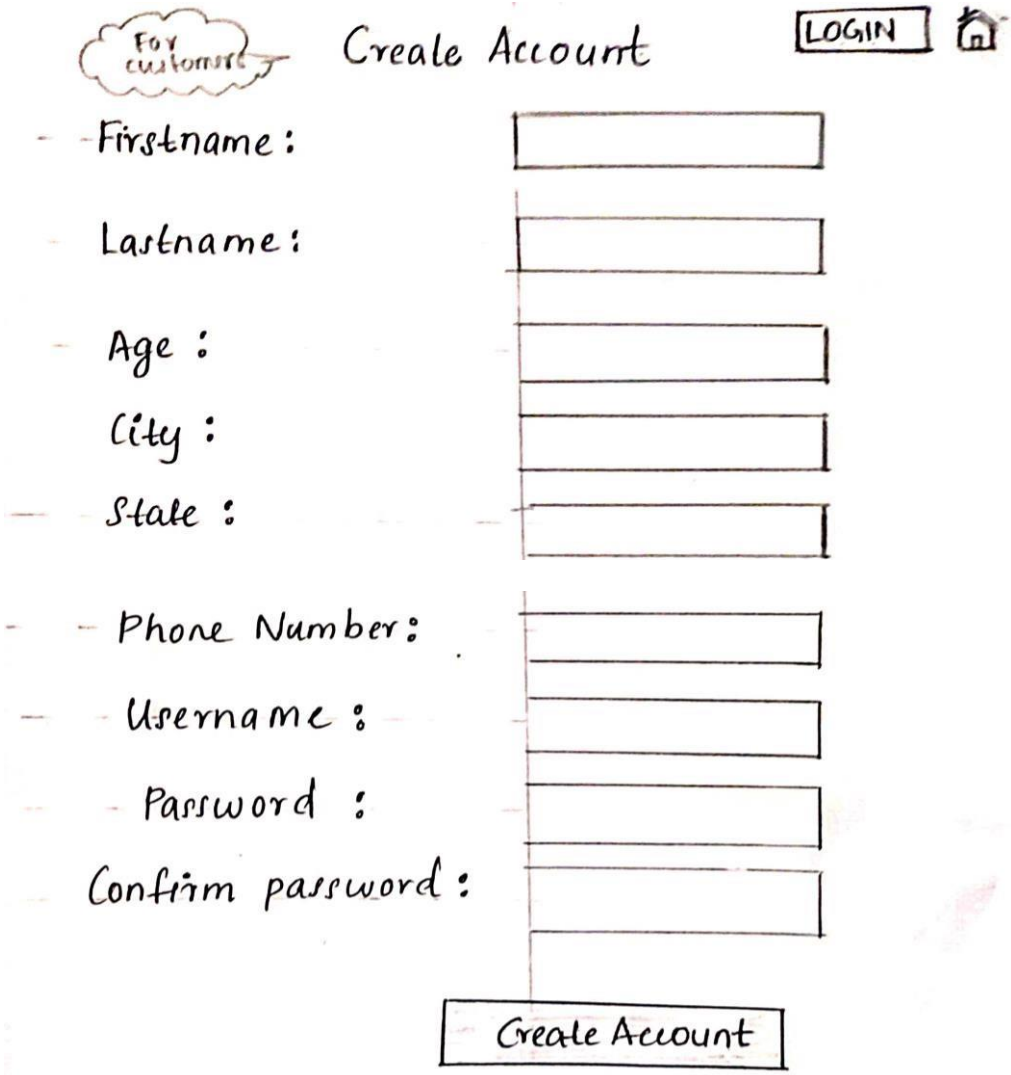
Payment

+Type	To store the mode of payment
+payment()	To confirm the payment
+print()	To view the payment slip

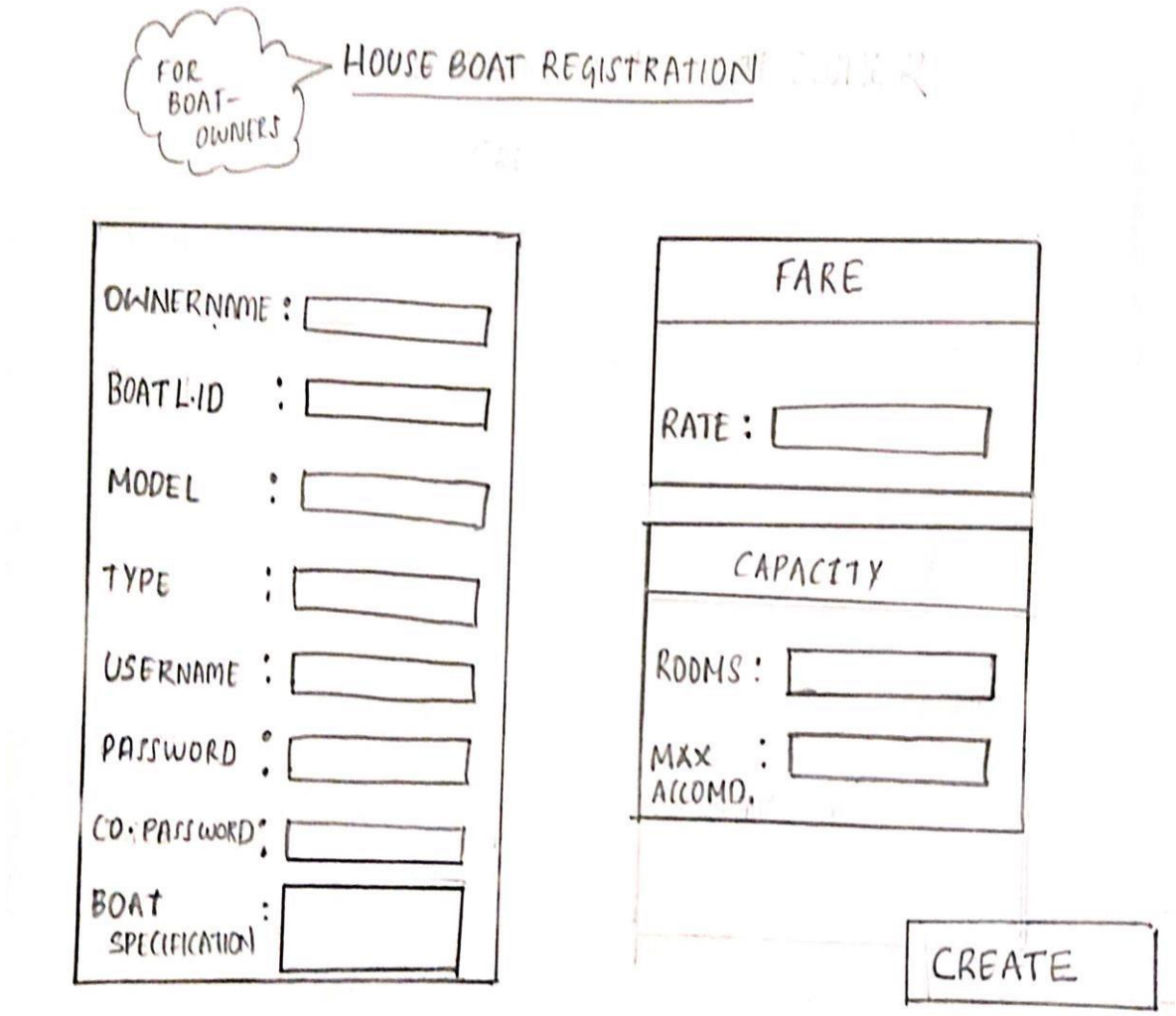
VISUAL LAYOUT

1. 

2. 

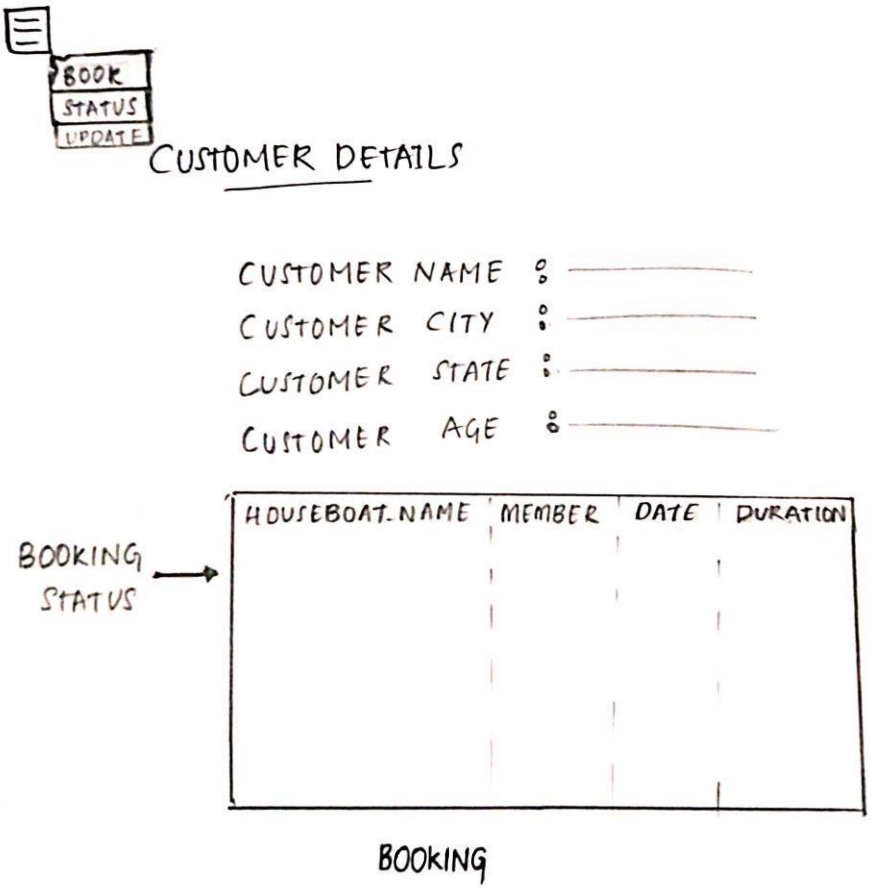

For customers
Create Account
LOGIN
Firstname :
Lastname :
Age :
City :
State :
Phone Number :
Username :
Password :
Confirm password :
Create Account

3.

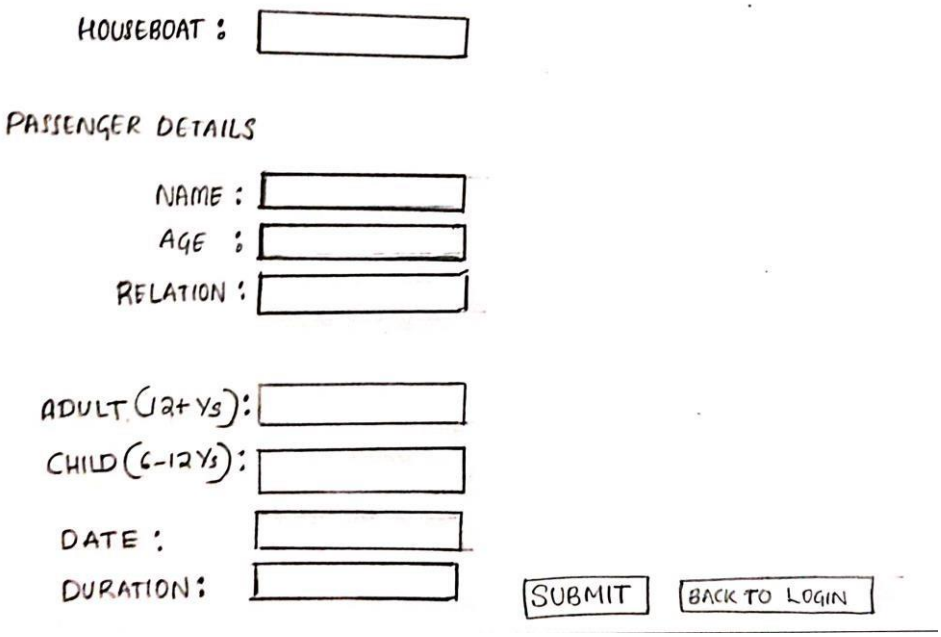

FOR BOAT-OWNERS
HOUSE BOAT REGISTRATION
OWNER NAME :
BOAT I.ID :
MODEL :
TYPE :
USERNAME :
PASSWORD :
CO-PASSWORD :
BOAT SPECIFICATION :
FARE
RATE :
CAPACITY
ROOMS :
MAX ACCOMD. :
CREATE

4.

5.


BOOK
STATUS
UPDATE
LOGIN
CUSTOMER DETAILS
CUSTOMER NAME :
CUSTOMER CITY :
CUSTOMER STATE :
CUSTOMER AGE :
BOOKING STATUS
HOUSEBOAT_NAME
MEMBER
DATE
DURATION
BOOKING

6.


HOUSEBOAT :
PASSENGER DETAILS
NAME :
AGE :
RELATION :
ADULT (12+ Yrs) :
CHILD (6-12 Yrs) :
DATE :
DURATION :
SUBMIT
BACK TO LOGIN

BOAT OWNER PANEL

ADD BOAT | DELETE | UPDATE

BOAT ID	OWNER NAME	CAPACITY	FARE	NO·OF EMPLOYEES

7.

ADMIN PANEL

BOAT DETAILS : UPDATE

HOUSE BOAT ID	OWNER NAME	CAPACITY	FARE
[TABLE SHOWING BOAT DETAILS]			

CUSTOMER DETAILS : UPDATE

HOUSE BOAT -NAME	MEMBER	DATE	DURATION
[TABLE SHOWING CUSTOMER DETAILS]			

8.

The front-end and back-end tools used:

- We used Java Eclipse as our IDE for doing the project.
- We implemented Jframe to design the layout of the pages.
- The javax.swing package provides classes for java swing API such as JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser etc.
- For carrying out the background connectivity with the database, we used the JDBC API.

RESULT

♣ LOGIN PAGE

- LOGIN BUTTON

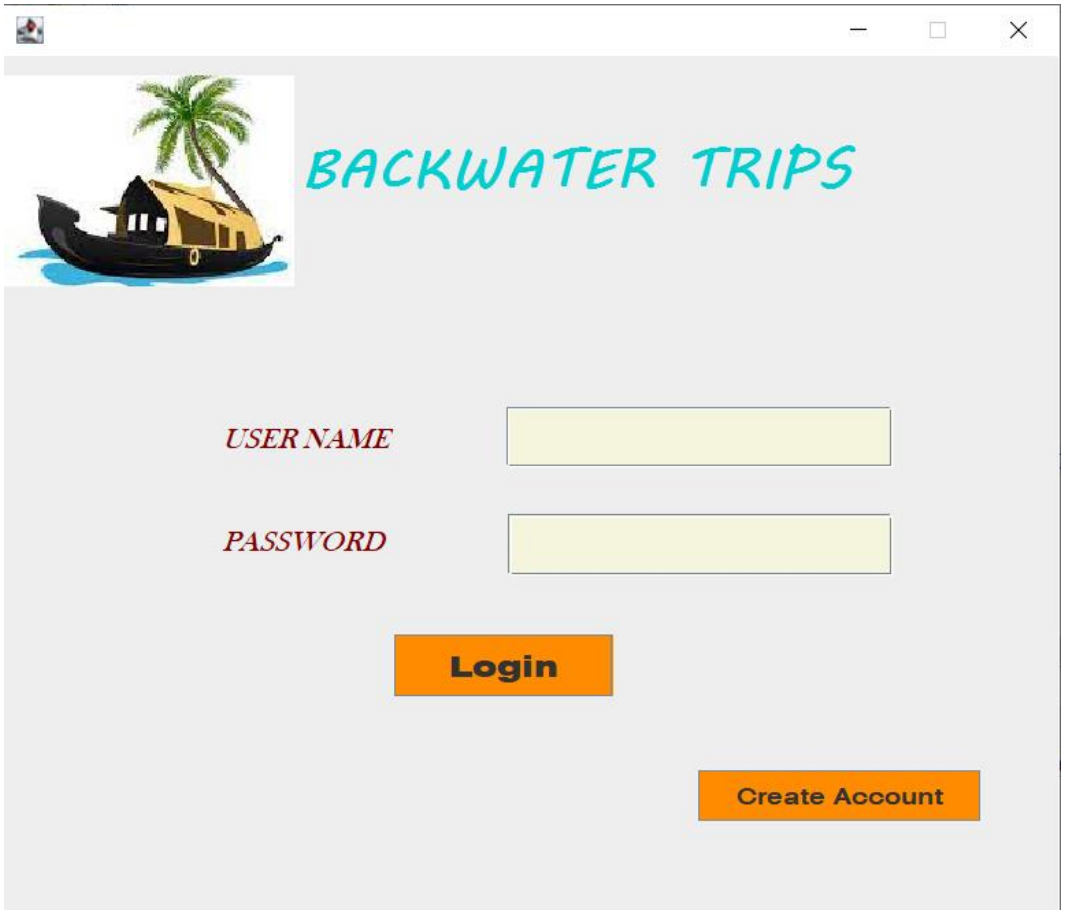
```
JButton btnNewButton = new JButton("Login\r\n");
btnNewButton.setBackground(new Color(255, 140, 0));
btnNewButton.addActionListener(new ActionListener() {
    @SuppressWarnings("deprecation")
    public void actionPerformed(ActionEvent e) {

    try {

            Class.forName("org.postgresql.Driver");

            Connection connection = DriverManager.getConnection("jdbc:postgresql://localhost:5432/houseboat","postgres","andumadom");
            String sql1 = "SELECT * FROM login where  username = ? and password = ? ";
            String sql2 = "SELECT authority from login where username = ?";



        PreparedStatement pst=connection.prepareStatement(sql1);
        PreparedStatement pst1 = connection.prepareStatement(sql2);


        pst.setString(1, user_name.getText());
        pst.setString(2, passwordField.getText());
        pst1.setString(1, user_name.getText());

        ResultSet rs = pst.executeQuery();
        ResultSet rs1 = pst1.executeQuery();
```

```
             int count = 0;
             while(rs.next()) {

                 count = count + 1;
             };
                 if(count == 1)
                 {

                     if(rs1.next()) {

                         System.out.print("Name of the Employee: "+rs1.getString("authority")+", ");
                         String str1 = rs1.getString("authority");
                     JOptionPane.showMessageDialog(null, "Password and username is correct");
                     if(str1 == "customer") {
                     customerdetails customerpage = new customerdetails();
                     customerpage.setVisible(true);
                     System.out.println("customer");
                     dispose();
                     }
                      if(str1 == "owner") {
                          //btnNewButton.addActionListener(new ActionListener() {};
                         ownerpg ownerspage = new ownerpg();
                         ownerspage.setVisible(true);
                         dispose();
                     }
                      if(str1=="admin") {
                         Admin adminpage = new Admin();
                         adminpage.setVisible(true);

                         }
                     }
```

```
        int count = 0;
        while(rs.next()) {

            count = count + 1;
        };
            if(count == 1)
            {

                if(rs1.next()) {

                    System.out.print("Name of the Employee: "+rs1.getString("authority")+", ");
                    String str1 = rs1.getString("authority");
                JOptionPane.showMessageDialog(null, "Password and username is correct");
                if(str1 == "customer") {
                customerdetails customerpage = new customerdetails();
                customerpage.setVisible(true);
                System.out.println("customer");
                dispose();
                }
                 if(str1 == "owner") {
                     //btnNewButton.addActionListener(new ActionListener() {};
                    ownerpg ownerspage = new ownerpg();
                    ownerspage.setVisible(true);
                    dispose();
                }
                 if(str1=="admin") {
                    Admin adminpage = new Admin();
                    adminpage.setVisible(true);

                    }
                }
```

- CREATE ACCOUNT

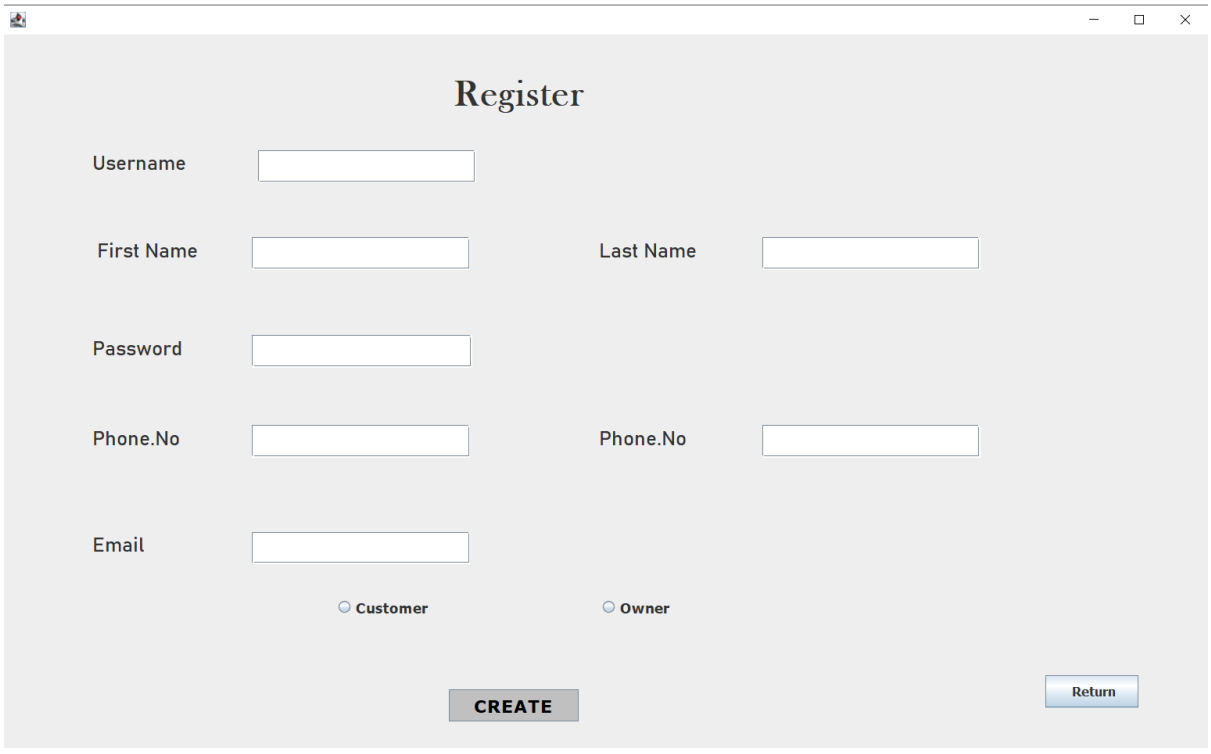
```
btnNewButton.setFont(new Font("Swis721 BlkEx BT", Font.PLAIN, 17));
btnNewButton.setBounds(228, 338, 128, 36);
contentPane.add(btnNewButton);

JButton btnNewButton_1 = new JButton("Create Account");
btnNewButton_1.setBackground(new Color(255, 140, 0));
btnNewButton_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        createaccount accpage = new createaccount();
        accpage.setVisible(true);
        dispose();

    }
});
```



♣ CREATE ACCOUNT

CREATE ACCOUNT BUTTON

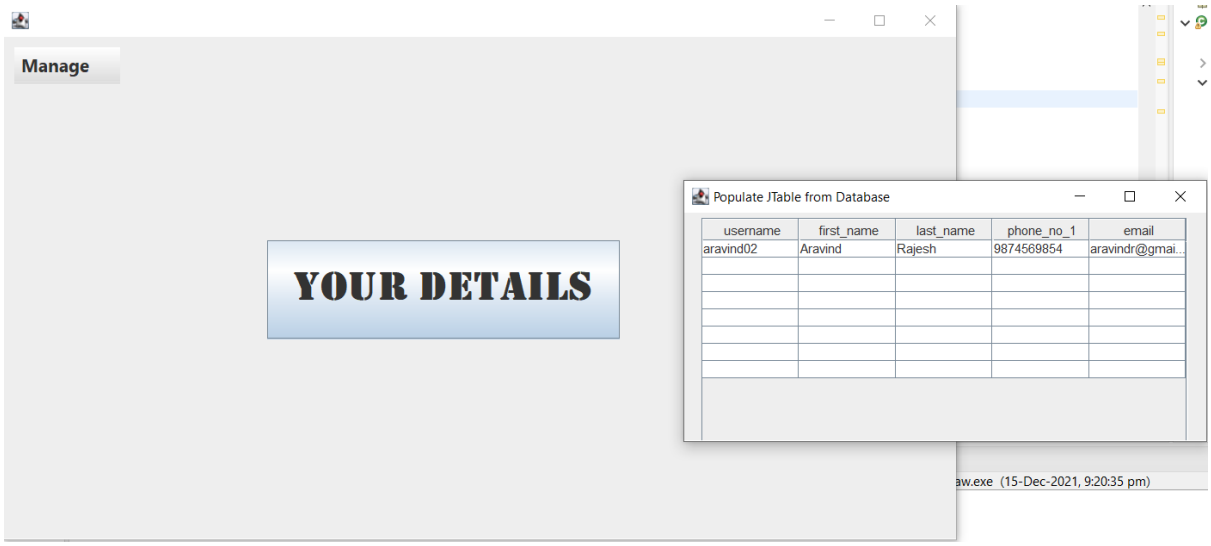
```
JButton createbtn = new JButton("CREATE");
createbtn.setForeground(Color.BLACK);
createbtn.setBackground(Color.LIGHT_GRAY);
createbtn.setFont(new Font("Verdana", Font.BOLD, 18));
createbtn.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e) {

    try {

        Class.forName("org.postgresql.Driver");

        Connection connection = DriverManager.getConnection("jdbc:postgresql://localhost:5432/houseboat","postgres","andumadom");
        String sql1 = "insert into login( username,first_name,last_name,password,phone_no_1,phone_no_2,email) values(?,?,?,?,?,?,?)";
    PreparedStatement pst=connection.prepareStatement(sql1);

    pst.setString(1, us_name.getText());
    pst.setString(2, f_name.getText());
    pst.setString(3, l_name.getText());
    pst.setString(4, passwordField_1.getText());
    pst.setString(5, ph_1.getText());
    pst.setString(6, ph_2.getText());
    pst.setString(7, email.getText());


    ResultSet rs1 = pst.executeQuery();

    rs1.close();
    pst.close();
    connection.close();

    }

    catch(Exception e1)
    {
        JOptionPane.showMessageDialog(null, e1);
        System.out.println("unknown error");
    }
}
});
```



♣ CUSTOMER DETAILS

YOUR DETAILS BUTTON

```
JButton btnNewButton_1 = new JButton("YOUR DETAILS");
btnNewButton_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            Connection connection = DriverManager.getConnection("jdbc:postgresql://localhost:5432/houseboat","postgres","andumadom");
            String query = "SELECT * FROM login where authority = 'customer'";

              java.sql.Statement stm = connection.createStatement();
              ResultSet res = stm.executeQuery(query);

              String columns[] = { "username", "first_name", "last_name","phone_no_1","email" };
              String data[][] = new String[8][5];

              int i = 0;
              while (res.next()) {
                String id = res.getString("username");
                String f_name = res.getString("first_name");
                String l_name = res.getString("last_name");
                String phone_no_1 = res.getString("phone_no_1");
                String email = res.getString("email");
                data[i][0] = id + "";
                data[i][1] = f_name;
                data[i][2] = l_name;
                data[i][3] = phone_no_1;
                data[i][4] = email;

                i++;
              }


               DefaultTableModel model = new DefaultTableModel(data, columns);
               JTable table = new JTable(model);
               table.setShowGrid(true);
               table.setShowVerticalLines(true);
               JScrollPane pane = new JScrollPane(table);
               JFrame f = new JFrame("Populate JTable from Database");
               JPanel panel = new JPanel();
               panel.add(pane);
               f.getContentPane().add(panel);
               f.setSize(500, 250);
               f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
               f.setVisible(true);
        } catch (HeadlessException e1) {
            // TODO Auto-generated catch block
            e1.printStackTrace();
        } catch (SQLException e1) {
            // TODO Auto-generated catch block
            e1.printStackTrace();
        }

    }
});
```



♣ ADMIN PANEL

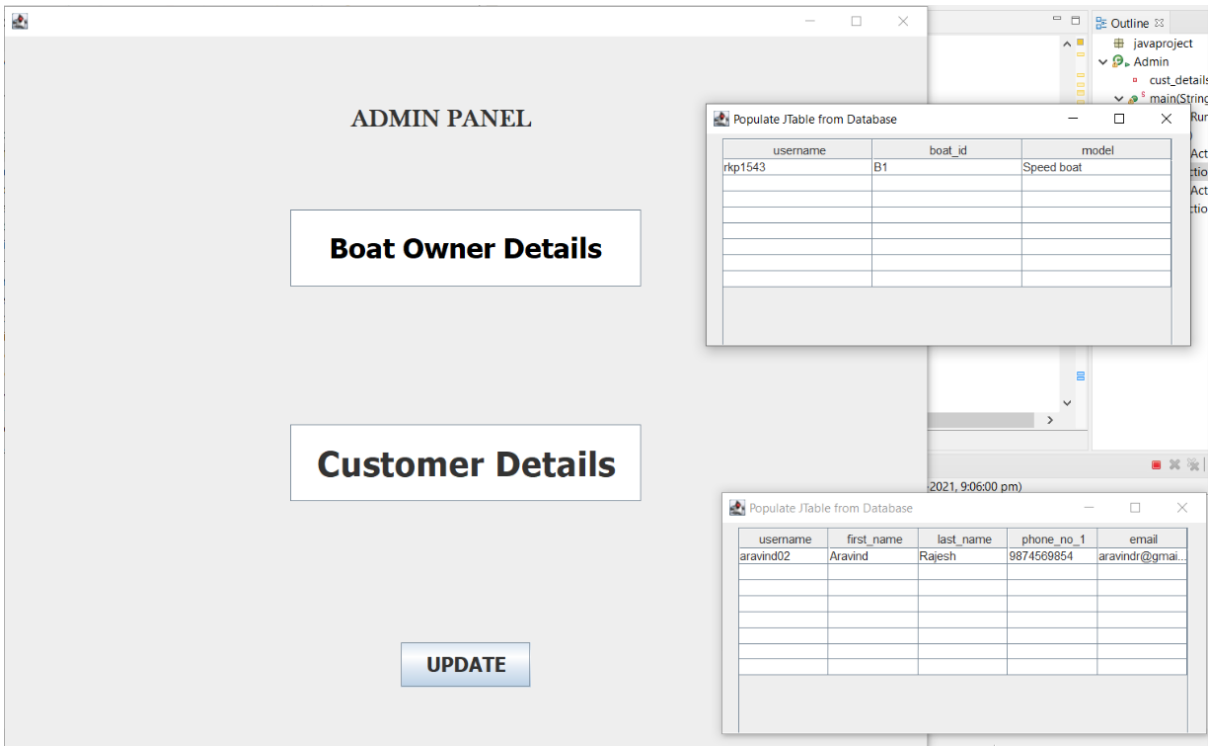
- BOAT OWNER DETAILS BUTTON

```
JButton btnNewButton_1 = new JButton("Boat Owner Details");
btnNewButton_1.setForeground(Color.BLACK);
btnNewButton_1.setFont(new Font("Tahoma", Font.BOLD, 28));
btnNewButton_1.setBackground(new Color(245, 222, 179));
btnNewButton_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        try {
            Connection connection = DriverManager.getConnection("jdbc:postgresql://localhost:5432/houseboat","postgres","andumadom");
            String query = "SELECT * FROM boat_registration";

              java.sql.Statement stm = connection.createStatement();
              ResultSet res = stm.executeQuery(query);

              String columns[] = { "username", "boat_id", "model" };
              String data[][] = new String[8][3];

              int i = 0;
              while (res.next()) {
                String id = res.getString("username");
                String nom = res.getString("boat_id");
                String model = res.getString("model");
                data[i][0] = id + "";
                data[i][1] = nom;
                data[i][2] = model;
                i++;
              }


              DefaultTableModel model = new DefaultTableModel(data, columns);
              JTable table = new JTable(model);
              table.setShowGrid(true);
              table.setShowVerticalLines(true);
              JScrollPane pane = new JScrollPane(table);
              JFrame f = new JFrame("Populate JTable from Database");
              JPanel panel = new JPanel();
              panel.add(pane);
              f.getContentPane().add(panel);
              f.setSize(500, 250);
              f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
              f.setVisible(true);
        } catch (HeadlessException e1) {
            // TODO Auto-generated catch block
            e1.printStackTrace();
        } catch (SQLException e1) {
            // TODO Auto-generated catch block
            e1.printStackTrace();
        }


    }
});
```

- CUSTOMER DETAILS BUTTON

```
144     JButton boat_det = new JButton("Customer Details");
145     boat_det.setForeground(new Color(0, 0, 0));
146     boat_det.setFont(new Font("Tahoma", Font.BOLD, 28));
147     boat_det.setBackground(new Color(245, 222, 179));
148     boat_det.addActionListener(new ActionListener() {
149         public void actionPerformed(ActionEvent e) {
150
151             try {
152                 Connection connection = DriverManager.getConnection("jdbc:postgresql://localhost:5432/houseboat","postgres","andumadom");
153                 String query = "SELECT * FROM login where authority = 'customer'";
154
155                   java.sql.Statement stm = connection.createStatement();
156                   ResultSet res = stm.executeQuery(query);
157
158                   String columns[] = { "username", "first_name", "last_name","phone_no_1","email" };
159                   String data[][] = new String[8][5];
160
161                   int i = 0;
162                   while (res.next()) {
163                     String id = res.getString("username");
164                     String f_name = res.getString("first_name");
165                     String l_name = res.getString("last_name");
166                     String phone_no_1 = res.getString("phone_no_1");
167                     String email = res.getString("email");
168                     data[i][0] = id + "";
169                     data[i][1] = f_name;
170                     data[i][2] = l_name;
171                     data[i][3] = phone_no_1;
172                     data[i][4] = email;
173
174                     i++;
```

```
                DefaultTableModel model = new DefaultTableModel(data, columns);
                JTable table = new JTable(model);
                table.setShowGrid(true);
                table.setShowVerticalLines(true);
                JScrollPane pane = new JScrollPane(table);
                JFrame f = new JFrame("Populate JTable from Database");
                JPanel panel = new JPanel();
                panel.add(pane);
                f.getContentPane().add(panel);
                f.setSize(500, 250);
                f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                f.setVisible(true);
            } catch (HeadlessException e1) {
                // TODO Auto-generated catch block
                e1.printStackTrace();
            } catch (SQLException e1) {
                // TODO Auto-generated catch block
                e1.printStackTrace();
            }

        }
    });
```



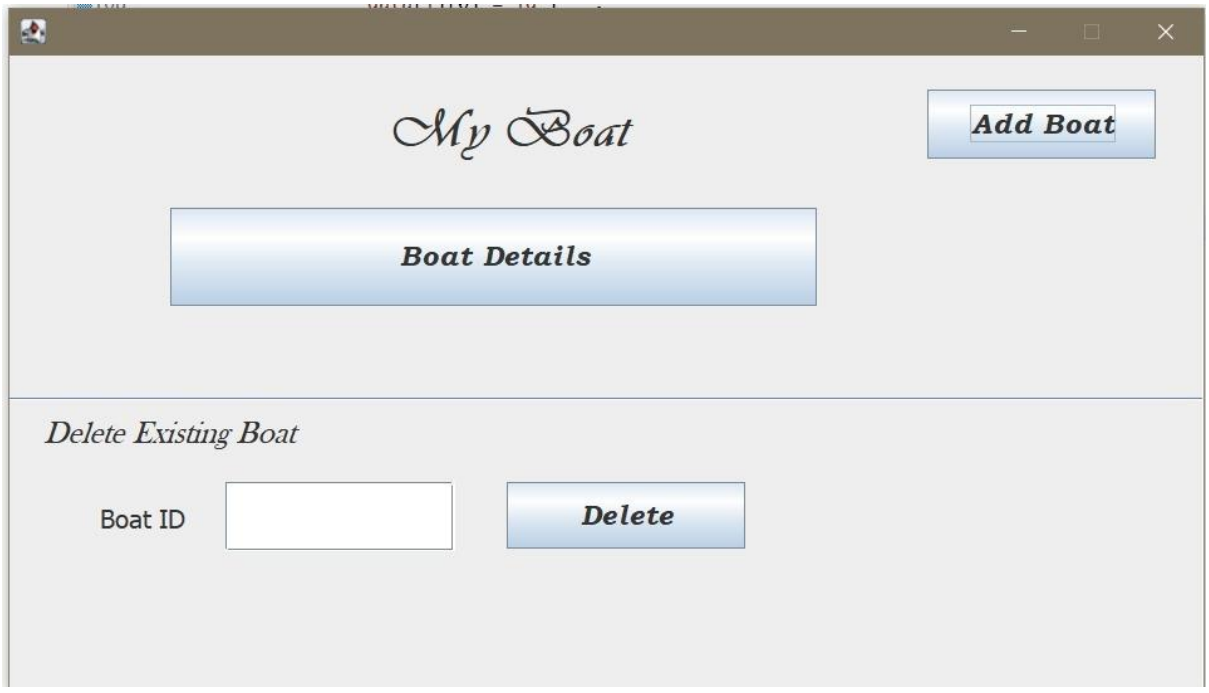
♣ OWNER PAGE

- ADD BOAT BUTTON

```
92 JButton addboat = new JButton("Add Boat");
93 addboat.setFont(new Font("Bookman Old Style", Font.BOLD | Font.ITALIC, 18));
94 addboat.addActionListener(new ActionListener() {
95 public void actionPerformed(ActionEvent e) {
96 boatregistration br = new boatregistration();
97 br.setVisible(true);
98 br.setResizable(false);
99 dispose();
.00 }
.01 });
.02 addboat.setBounds(585, 22, 146, 44);
.03 contentPane.add(addboat);
```

- DELETE BUTTON

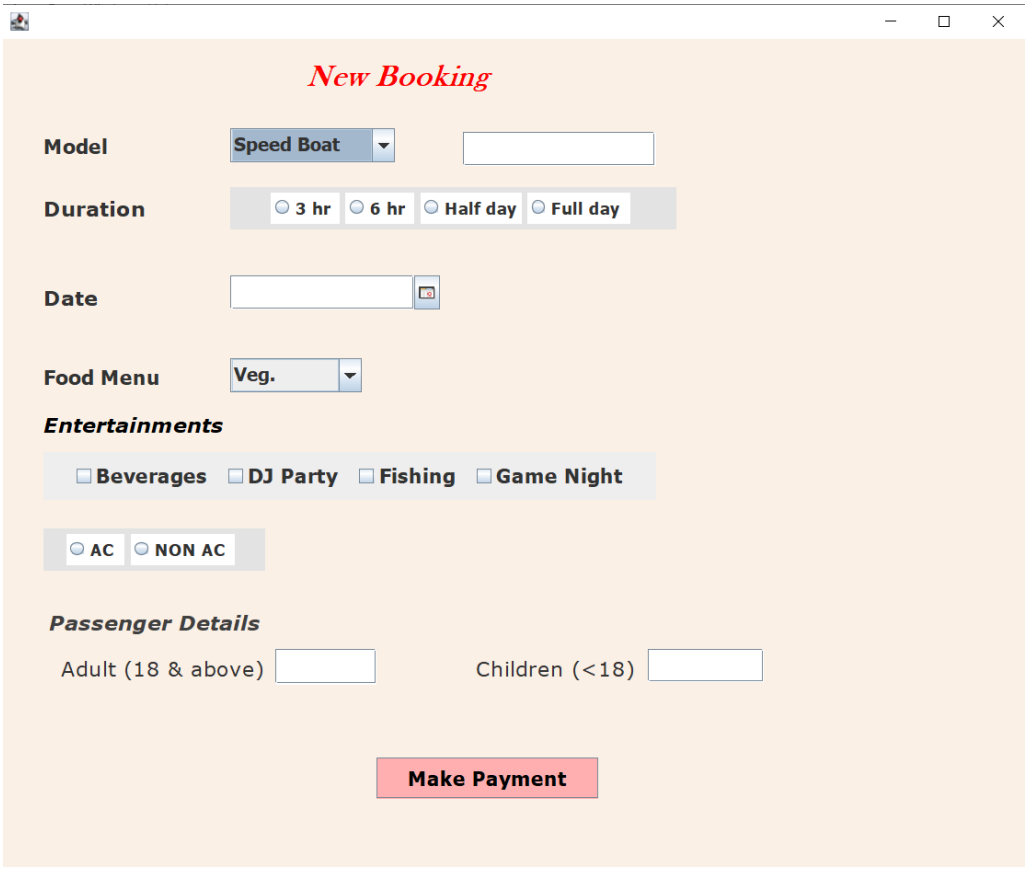
```
105 JButton btnDelete = new JButton("Delete");
106 btnDelete.addActionListener(new ActionListener() {
107     public void actionPerformed(ActionEvent e) {
108         try {
109             JOptionPane.showMessageDialog(null,"You wanna delete the boat ?");
110
111             Class.forName("org.postgresql.Driver");
112
113             Connection connection = DriverManager.getConnection("jdbc:postgresql://localhost:5432/houseboat","postgres","andumadom");
114             String sql1 = "delete FROM boat_registration where boat_id= ? ";
115
116         PreparedStatement pst=connection.prepareStatement(sql1);
117
118
119
120         pst.setString(1, textField.getText());
121
122
123         ResultSet rs = pst.executeQuery();
124
125
126
127         rs.close();
128
129         pst.close();
130
131         connection.close();
132         }
133         catch(Exception e1)
134         {
135             JOptionPane.showMessageDialog(null, e1);
136             System.out.println("unknown error");
137
```



♣ BOOKING PAGE

- MAKE PAYMENT BUTTON

```
JButton btnNewButton = new JButton("Make Payment");
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

            try {

            String value1 =(String)comboBox1.getSelectedItem();//model combobox
            String value2 =(String)comboBox2.getSelectedItem();//food menu combox
             System.out.println(value1+value2);

             Date date=dateChooser.getDate();//date

                System.out.println(date);


                //duration radibutton
        String duration = null;
                    if(t1.isSelected()){
                         duration =t1.getText();
                        }

                    else if(t2.isSelected()){
                         duration =t2.getText();
                        }

                    else if(t3.isSelected()){
                         duration =t3.getText();
                    }
                    else if(t4.isSelected()){
                         duration =t4.getText();
                    }
                    System.out.println(duration);

        //

    //      ac/non ac
                 String cooler = null;
                                if(f1.isSelected()){
                                     cooler =f1.getText();
                                    }

                                else if(f2.isSelected()){
                                     cooler =f2.getText();
                                    }

                  System.out.println(cooler);


// entertainments checkbox
                String partyitem1= null,partyitem2=null,partyitem3=null,partyitem4=null;
                if( check1.isSelected()){
                     partyitem1=check1.getText();
                    }

                if(check2.isSelected()){
                     partyitem2 =check2.getText();
                    }

                if(check3.isSelected()){
                    partyitem3 =check3.getText();
                }
                 if(check4.isSelected()){
                     partyitem4 =check4.getText();
                }
                System.out.println(partyitem1+" "+partyitem2+" "+partyitem3+" "+ partyitem4);
```

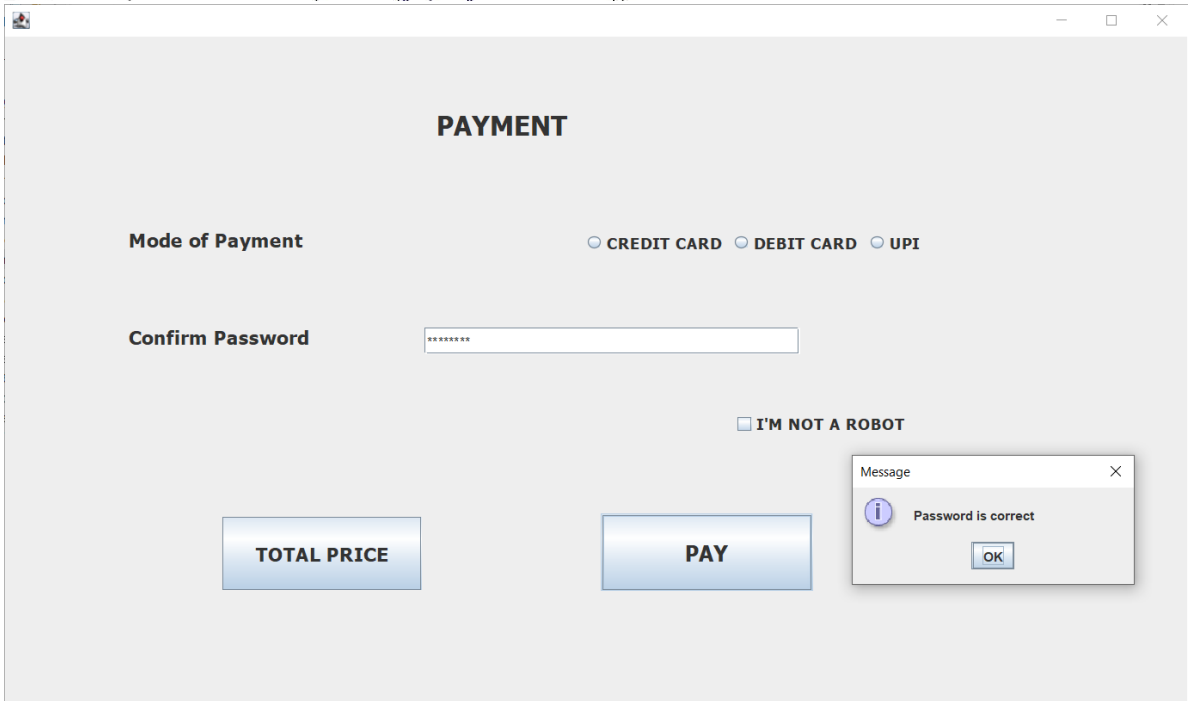

New Booking
Model
Speed Boat
Duration
3 hr
6 hr
Half day
Full day
Date
Food Menu
Veg.
Entertainments
Beverages
DJ Party
Fishing
Game Night
AC
NON AC
Passenger Details
Adult (18 & above)
Children (<18)
Make Payment

♣ PAYMENT

- PAY BUTTON

```
JButton btnNewButton = new JButton("PAY");
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            Class.forName("org.postgresql.Driver");

            Connection connection = DriverManager.getConnection("jdbc:postgresql://localhost:5432/houseboat","postgres","andumadom");

            String sql = "SELECT password from login where password=?";



                PreparedStatement pst=connection.prepareStatement(sql);

                pst.setString(1,passwordField.getText());


                ResultSet rs = pst.executeQuery();

                int count = 0;
                while(rs.next()) {

                    count = count + 1;
                };
                    if(count == 1)
                    {

                        JOptionPane.showMessageDialog(null, "Password is correct");
                    }
```

```
157
158                                 else if(count>1)
159                                 {
160                                     JOptionPane.showMessageDialog(null, "redundancy detected");
161                                 }
162                                 else
163                                 {
164                                     JOptionPane.showMessageDialog(null,"Incorrect credentials");
165                                 }
166
167                             rs.close();
168
169                 } catch (ClassNotFoundException e1) {
170                     // TODO Auto-generated catch block
171                     e1.printStackTrace();
172                 } catch (SQLException e1) {
173                     // TODO Auto-generated catch block
174                     e1.printStackTrace();
175                 }
176
177
178                 customerdetails accpage = new customerdetails();
179                 accpage.setVisible(true);
180             }
181         });
```



♣ BOAT REGISTRATION

- REGISTER BOAT BUTTON

```
146     JButton reg_boat = new JButton("Register Boat");
147     reg_boat.addActionListener(new ActionListener() {
148         public void actionPerformed(ActionEvent e) {
149
150             try {
151
152                 Class.forName("org.postgresql.Driver");
153
154                 Connection connection = DriverManager.getConnection("jdbc:postgresql://localhost:5432/houseboat","postgres","andumadom");
155                 String sql1 = "insert into boat_registration(max_capacity,rooms,fare,model,boat_id,username) values(?,?,?,?,?,?)";
156             PreparedStatement pst=connection.prepareStatement(sql1);
157
158             pst.setString(1, max_capacity.getText());
159             pst.setString(2, rooms.getText());
160             pst.setString(3, fare.getText());
161             pst.setString(4, model.getText());
162             pst.setString(5, boat_id.getText());
163             pst.setString(6, u_name.getText());
164
165
166
167             int rs1 = pst.executeUpdate();
168             JOptionPane.showMessageDialog(null,"values updated");
169             ownerpg ownerspage = new ownerpg();
170             ownerspage.setVisible(true);
171             dispose();
172             //return rs1;
173             //rs1.close();
174             pst.close();
175             connection.close();
```

```
178
179                 catch(Exception e1)
180                 {
181                     JOptionPane.showMessageDialog(null, e1);
182                     System.out.println("unknown error");
183                 }
184             }
185         });
```

- RETURN BUTTON

```
213     JButton btnNewButton = new JButton("Return");
214     btnNewButton.addActionListener(new ActionListener() {
215         public void actionPerformed(ActionEvent e) {
216
217                 ownerpg ownerpage = new ownerpg();
218                ownerpage.setVisible(true);
219                dispose();
220             }
221     });
222     btnNewButton.setFont(new Font("Verdana", Font.BOLD, 16));
223     btnNewButton.setBounds(545, 561, 161, 40);
224     contentPane.add(btnNewButton);
225
226   }
227 }
```


Model
Boat Registration
Username
Boat ID
Model
Fare
Rooms
Max Capacity
Register Boat
Return

CONCLUSION

In this project we were successfully able create a java project which implements various API's such as java swings and JDBC API to retrieve and export boat owner and customer details into a database.

With this system a customer login, create account and make a booking with ease, wherein the boat owner can add any number or delete any number of boats. The admin act as a supervisor who has the authority to regulate boat owners and check information of the customer and the boat owners.

Our future aim is to release new updates which will make the system faster and more efficient. We will also make the app connected to the internet. Our ultimate aim is to make a project which will meet the industry standards and have a chance to compete professional app developers.

REFERENCES

- ✓ https://www.javatpoint.com/java-swing
- ✓ https://www.w3schools.com/java/default.asp
- ✓ https://www.oracle.com/in/java/
- ✓ https://www.java.com/en/
- ✓ https://www.geeksforgeeks.org/java/
- ✓ https://www.tutorialspoint.com/java/
- ✓ https://www.edureka.co/blog/java-swing
- ✓ https://www.meritschool.com/aprender-ingles/frases-vida-ingles/?cid=175&shop=java+spring+tutorialspoint&xi=1&xc=29&pr=72.99&you=0
- ✓ https://www.javatpoint.com/java-jdbc
- ✓ https://www.javatpoint.com/java-swing
- ✓ https://www.javatpoint.com/java-jradiobutton
- ✓ https://www.guru99.com/java-tutorial.html
- ✓ https://www.youtube.com/watch?v=niJ1tANWwFk

APPENDIX

- **GITHUB LINK**

1. **ABHIJITH SANTHOSH NARAYANAN**
   https://github.com/abhijith8201/Houseboatmanagementsystem

2. **ARAVIND RAJESH**
   https://github.com/arav02/Houseboat-Management-System?classId=ffd2e566-2f59-4142-9193-6ea52f9687ce&assignmentId=15e57631-e285-4e2b-9223-ed91a908c452&submissionId=3ef2666d-fe16-4dbb-448a-a7be2d1ad7e1&classId=ffd2e566-2f59-4142-9193-6ea52f9687ce&assignmentId=15e57631-e285-4e2b-9223-ed91a908c452&submissionId=3ef2666d-fe16-4dbb-448a-a7be2d1ad7e1

3. ROHITH K P
   https://github.com/rkp-1801/HOUSEBOAT-MANAGEMENT