19CSE204

OBJECT ORIENTED PROGRAMMING

HOUSEBOAT MANAGEMENT SYSTEM

BY

KEERTHANA A (AM.EN.U4CSE20241)

LAKSHMI V B (AM.EN.U4CSE20244)

N R NIRANJAN (AM.EN.U4CSE20249)

SIVACHANDRA K B (AM.EN.U4CSE20267)

CONTENTS

1. INTRODUCTION

This project focusses on applying OOPS concepts to build an application for a chain of houseboats. The Houseboat Management System provides a window for customers and owners to login and help them to access the information and implement certain functionalities.

The application is meant to be used by the customers and the owners of the houseboats. The owners can view the details of their houseboats that are booked and the staff working under them. The customers can register and login and book houseboats and make payment.

Java Swings have been used in this project to implement the UI for the application. The project also uses Java Database Connectivity to insert, access, update and delete data from the Houseboat Management System database.

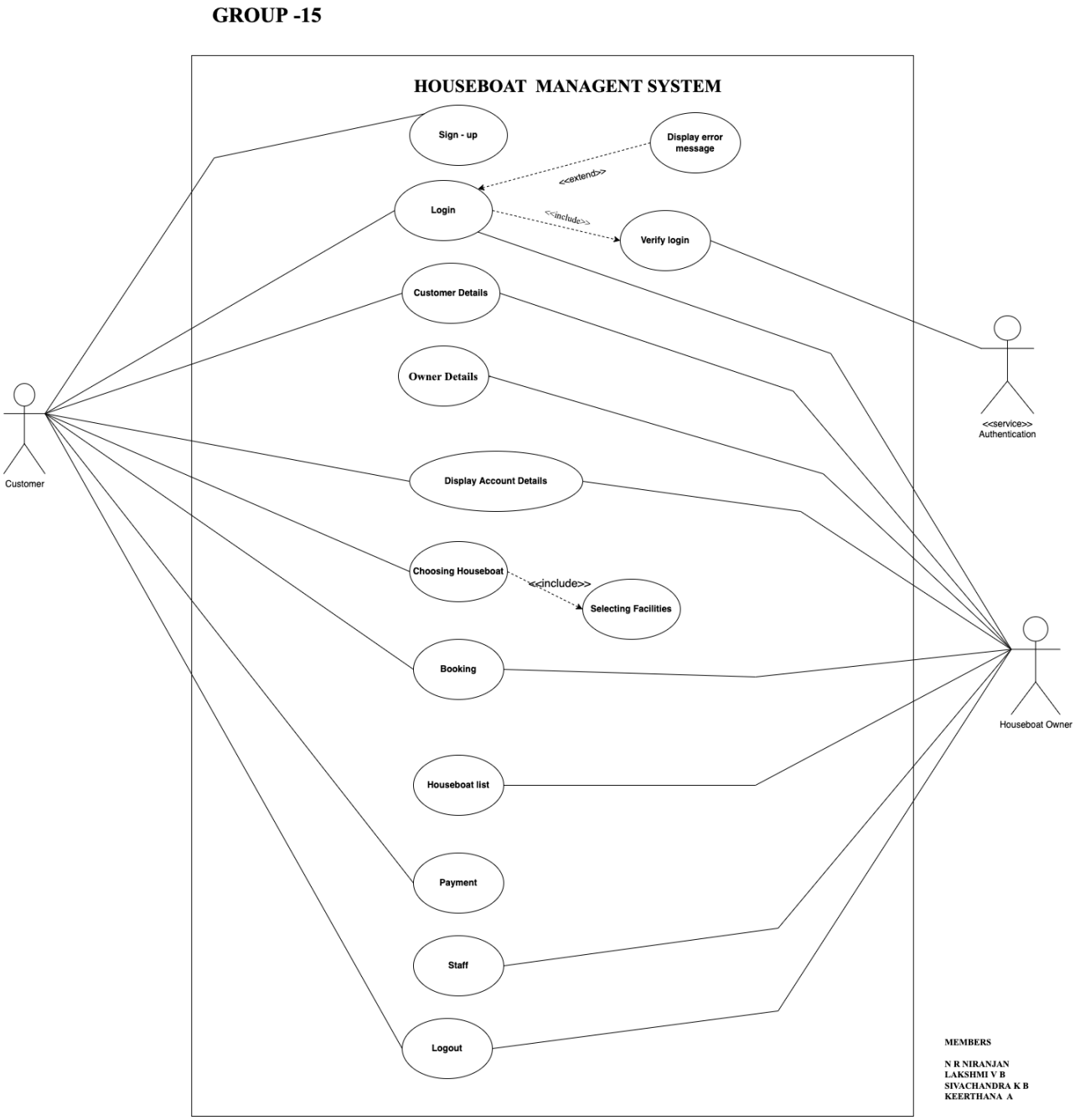
2. ABSTRACT

This project aims to develop a houseboat management system that can help an agency keep track of its customers, houseboat owners and their houseboats. When a customer books a houseboat through the agency, he/she is asked to sign-up and give his/her details. Similarly, if a houseboat owner wants to rent out his houseboat to the agency, he is asked to give his details and the houseboats details including the details of the facilities and staffs employed into the portal by signing up. The details that are provided are stored into the database. The database helps store this information for later use when the user has to access this through the portal after logging in. The customers can log-in to view their profile, carry out bookings after selecting the houseboat and going through their facilities and at last carry out payments through the portal.

After logging in, the customer is first brought into their profile where he/she is shown the details he/she gave when he/she registered. Then the customer can proceed to booking by selecting the houseboat that he/she might want depending on the number of passengers, then they can proceed to see the facilities the boat provides and select the required ones. The last part is where he is shown the total amount for renting the houseboat.

For the owner after logging in they are brought to their profile where they see their details. Then they can monitor their houseboats by seeing the boats that are rented and also their staff details.

3. USE CASE DIAGRAM



3.1 ACTORS

1. **CUSTOMER**
2. **HOUSEBOAT OWNER**

3.2 FUNCTIONALITIES

1. Sign-Up

Description: The customers can sign-up by entering the details.
Actors: Customer

2. Login

Description: The customers and houseboat owners can login using their user id and password.
Actors: Customer, Houseboat Owner

3. Customer Details

Description: The customers and houseboat owners can view the details of the customers who has login.
Actors: Customer, Houseboat Owner

4. Owner Details

Description: The owners can view the details of the customers who has login.
Actors: Houseboat Owner

5. Display Account Details

Description: Both houseboat owner and customer can viewtheir details after login.

Actors: Customer, Houseboat Owner

6. **Choosing Houseboats**
   Description: The customers choose a houseboat of their choice.
   Actor: Customer

7. **Selecting Facilities**
   Description: After selecting the houseboat, the customer can select the facilities required.
   Actors: Customer

8. **Booking**
   Description: The customer can book a houseboat of their choice and the houseboat owner can know if the booked houseboat belongs to him or not.
   Actors: Customer, Houseboat Owner

9. **Payment**
   Description: After the trip, the customer can view the amount of the trip and make payment.
   Actors: Customer

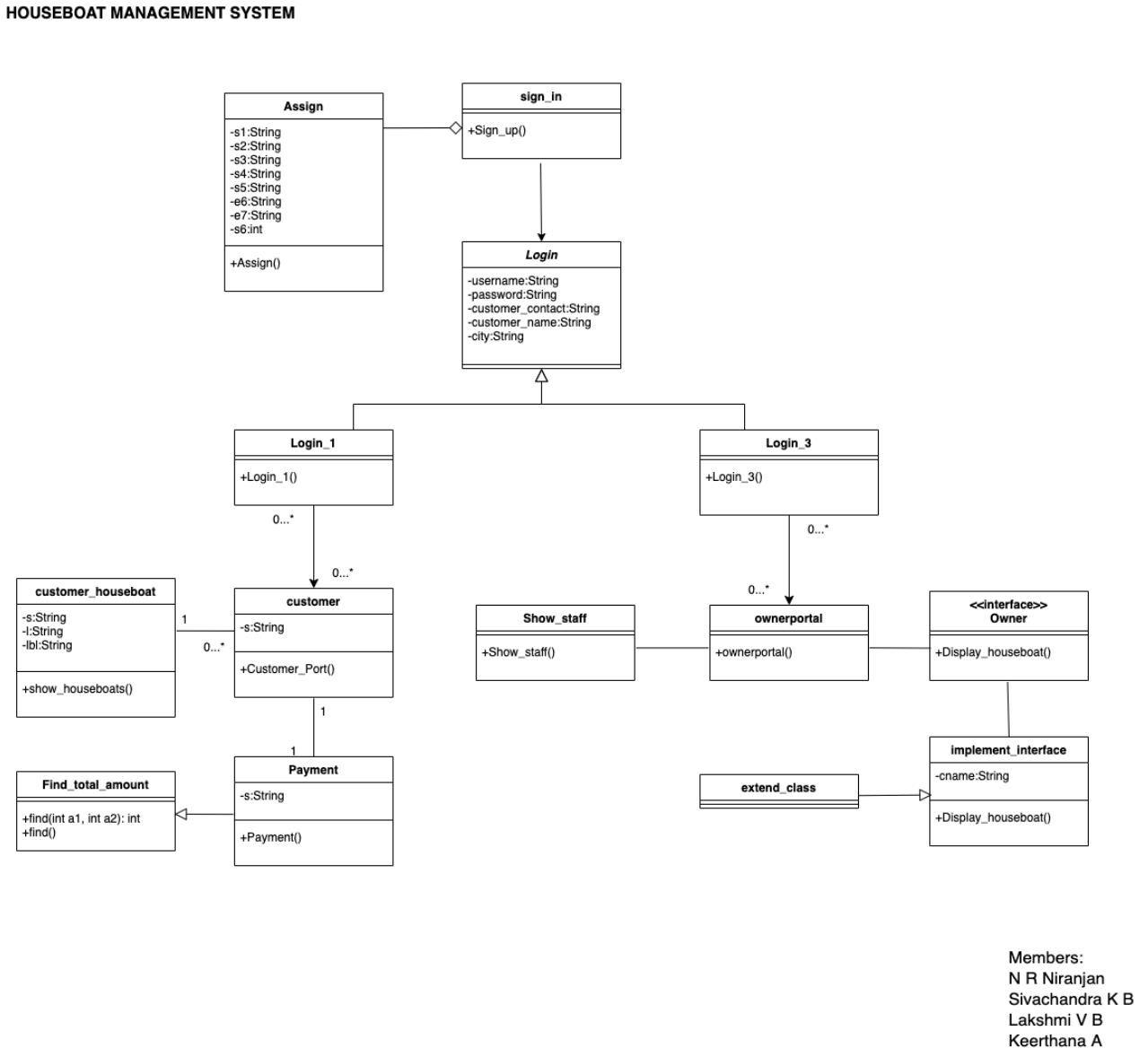
10. **Houseboat List**
    Description: The houseboat owner can view the list of booked houseboats.
    Actors: Houseboat Owner

11. **Staff List**
    Description: The houseboat owner can view the staff working under them.
    Actors: Houseboat Owner

12. **Logout**
    Description: Both the houseboat owner and customer can logout from the portal.

Actors: Customer, Houseboat Owner

4. CLASS DIAGRAM



Members:
N R Niranjan
Sivachandra K B
Lakshmi V B
Keerthana A

5. OOP FEATURES IMPLEMENTED

5.1 AGGREGATION

Aggregation is a special type of association that models a whole- part relationship between aggregate and its parts.

It is implemented using the sign_in and assign class where assign class is a part of the sign_in class. We used the assign class to input the values of the customer's sign_in details to the database.

5.2 GENERALIZATION

A generalization helps to connect a subclass to its superclass. A sub-class is inherited from its superclass. Generalization relationship can't be used to model interface implementation. Class diagram allows inheriting from multiple super classes.

It is implemented in the login portal, where both the classes Login_1 and Login_3 inherit the data members from the Login class.

5.3 INHERITANCE

Java inheritance refers to the ability in Java for one class to inherit from another class.

It is used in the customer portal, where the payment class inherits the member functions from the Find_total _amount class.

It is also implemented in the owner portal, where extend_class inherits the member functions from implement_interface class.

5.4 POLYMORPHISM

The compiler resolves the polymorphism during the compilation of the program. Also, called static binding. In Java, it can be achieved with the help of Method Overloading.

It is implemented in the payment portal, where the payment class calls the member function find() from the class Find_total_amount. Depending on the parameters passed on to the function, the respective find() is called. Thereby implemented method overloading.

5.5 ABSTRACTION

ABSTRACTION is the concept of object-oriented programming that "shows" only essential attributes and "hides" unnecessary information.

It is implemented using an interface Owner which is implemented by a class called implement_interface. This is used to display the houseboat list to the owner.
It is also implemented in the login portal where the login class is defined as an abstract class.

5.6 ENCAPSULATION

The process of binding data and corresponding methods (behaviour) together into a single unit is called encapsulation in Java.

Since we're using classes in our code, Encapsulation is satisfied by default.

6. DATABASE DETAILS

There are 8 tables in the Houseboat Management System. The details of each of them are given below:

6.1 OWNER TABLE

COLUMN_NAME	DATATYPE
owner_id	varchar(25)
license_no	varchar(25)
city	varchar(25)
owner_name	varchar(25)
owner_contact	varchar(10)
no_of_boats	int
owner_password	varchar(25)

6.2 CUSTOMER TABLE

COLUMN_NAME	DATATYPE
customer_id	varchar(25)
customer_name	varchar(25)
customer_city	varchar(25)
customer_country	varchar(25)
customer_contact	varchar(10)
no_of_passengers	int
customer_password	varchar(25)

6.3 AVAILABLE_HOUSEBOAT TABLE

COLUMN_NAME	DATATYPE
houseboat_id	varchar(25)
owner_id	varchar(25)
houseboat_name	varchar(25)
houseboat_capacity	int
no_of_rooms	int
typeof_boat	varchar(25)
price_per_head	int

6.4 BOOKED_HOUSEBOAT TABLE

COLUMN_NAME	DATATYPE
houseboat_id	varchar(25)
owner_id	varchar(25)
houseboat_name	varchar(25)
houseboat_capacity	int
no_of_rooms	int
typeof_boat	varchar(25)
price_per_head	int

6.5 FACILITIES TABLE

COLUMN_NAME	DATATYPE
customer_id	varchar(25)
houseboat_id	varchar(25)
wifi	varchar(10)
breakfast	varchar(25)
lunch	varchar(25)
snacks	varchar(25)
dinner	varchar(25)
entertainment	varchar(25)

6.6 STAFF TABLE

COLUMN_NAME	DATATYPE
staff_id	varchar(25)
owner_id	varchar(25)
staff_name	varchar(25)
gender	varchar(10)
job	varchar(25)
salary	decimal(7,2)

6.7 CURRENT TABLE

COLUMN_NAME	DATATYPE
cust_id	varchar(25)
name	varchar(25)
city	varchar(25)
contact_no	varchar(10)

6.8 HOUSEBOAT_CURRENT TABLE

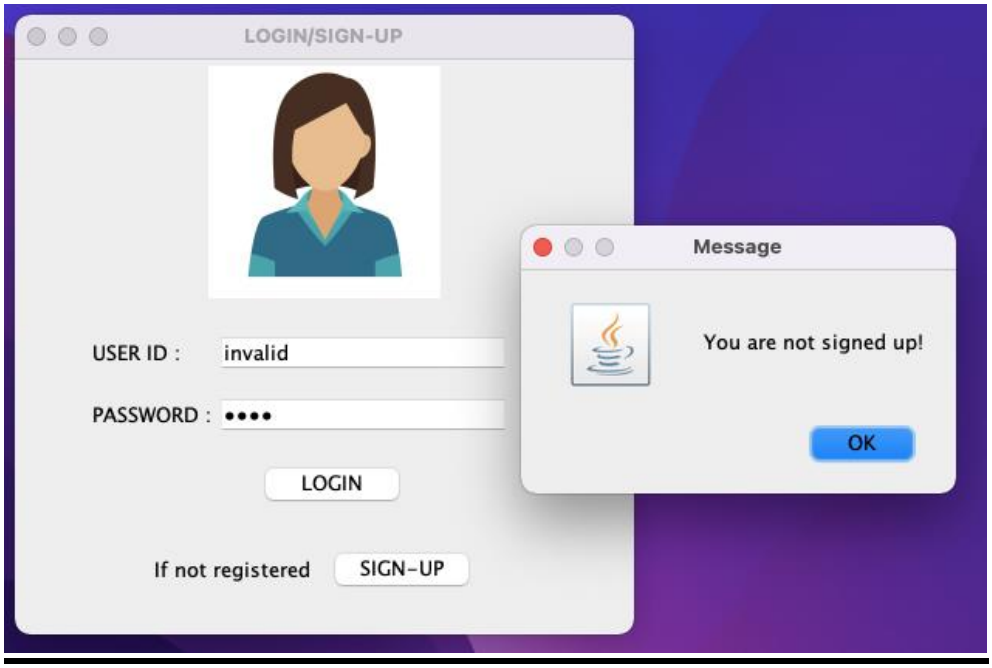
COLUMN_NAME	DATATYPE
houseboat_id	varchar(25)

7. PROJECT FEATURES

7.1

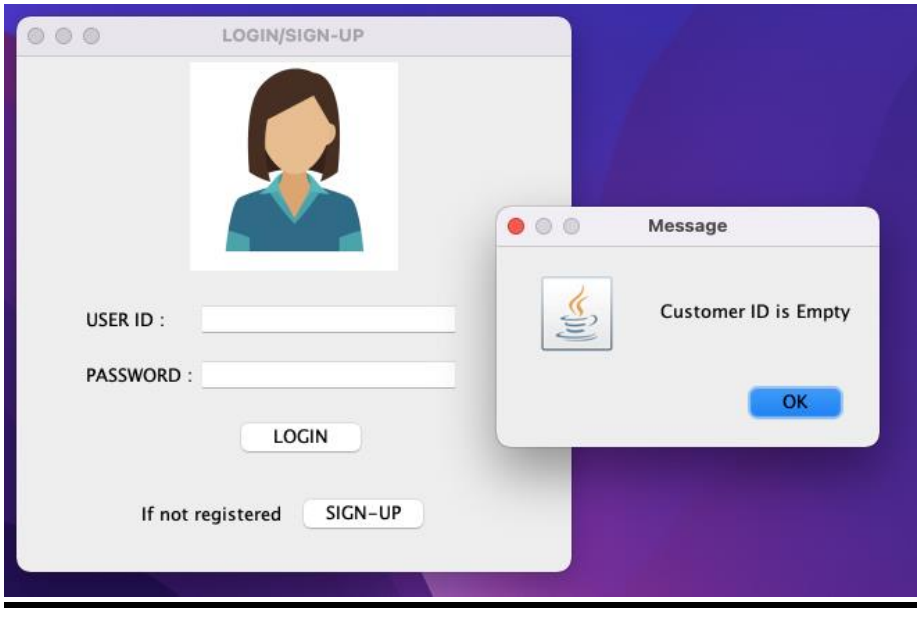
We start our code from the page above where the user can either select the Customer or the Houseboat Owner. Here we go ahead with Customer.

7.2



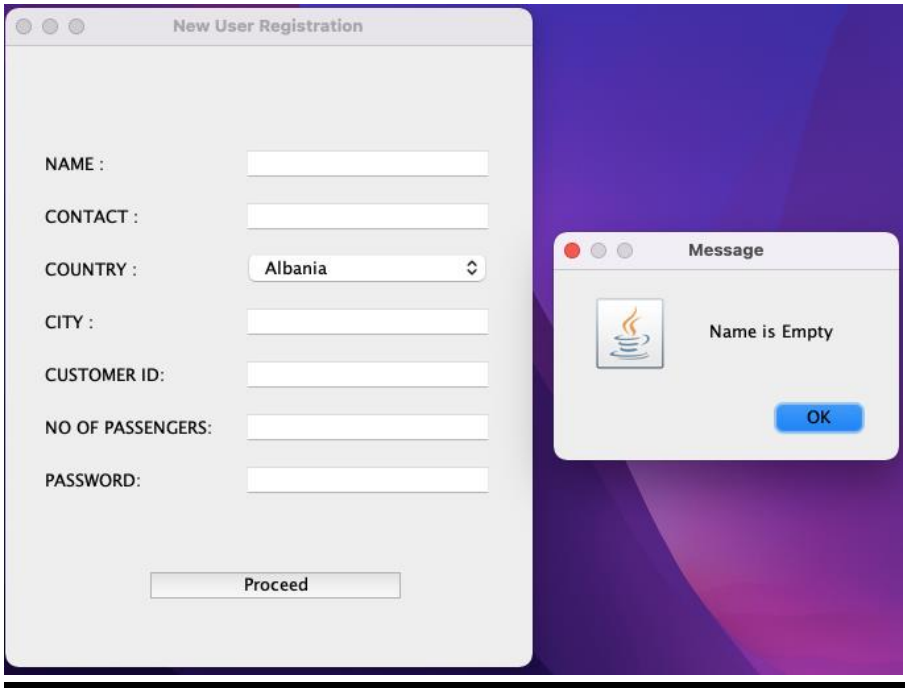
If we give the invalid credentials the above pop-up will appear, prompting us that the credentials are invalid.

7.3



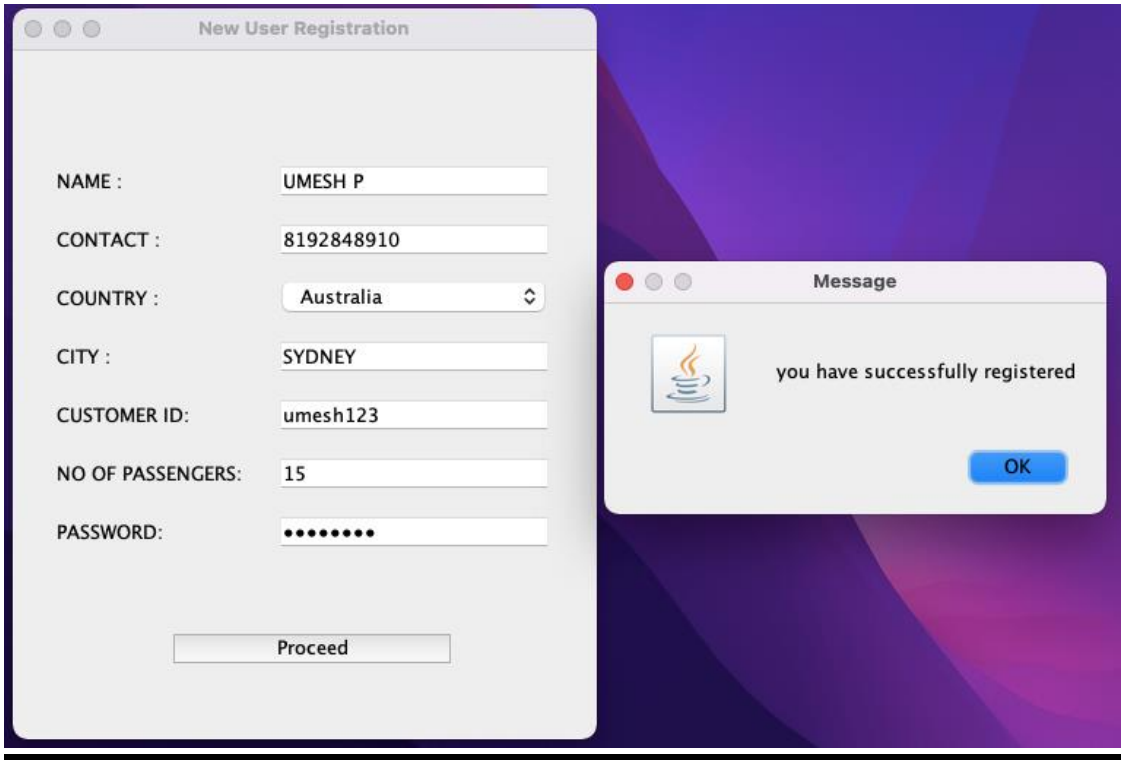
If we give any field empty the code gives us a prompt saying that.
If we need a new account as a customer we can click the button sign-up.

7.4



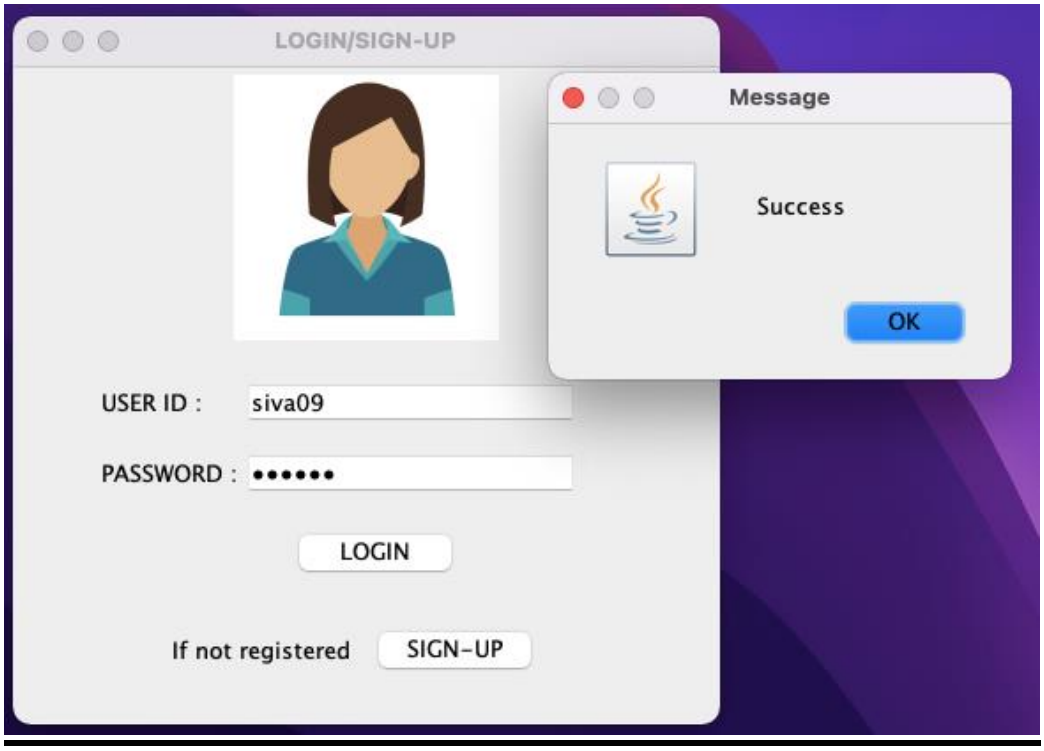
Inside the sign up page we can enter the necessary details and click proceed if any value is not given the following pop-up is shown.

7.5



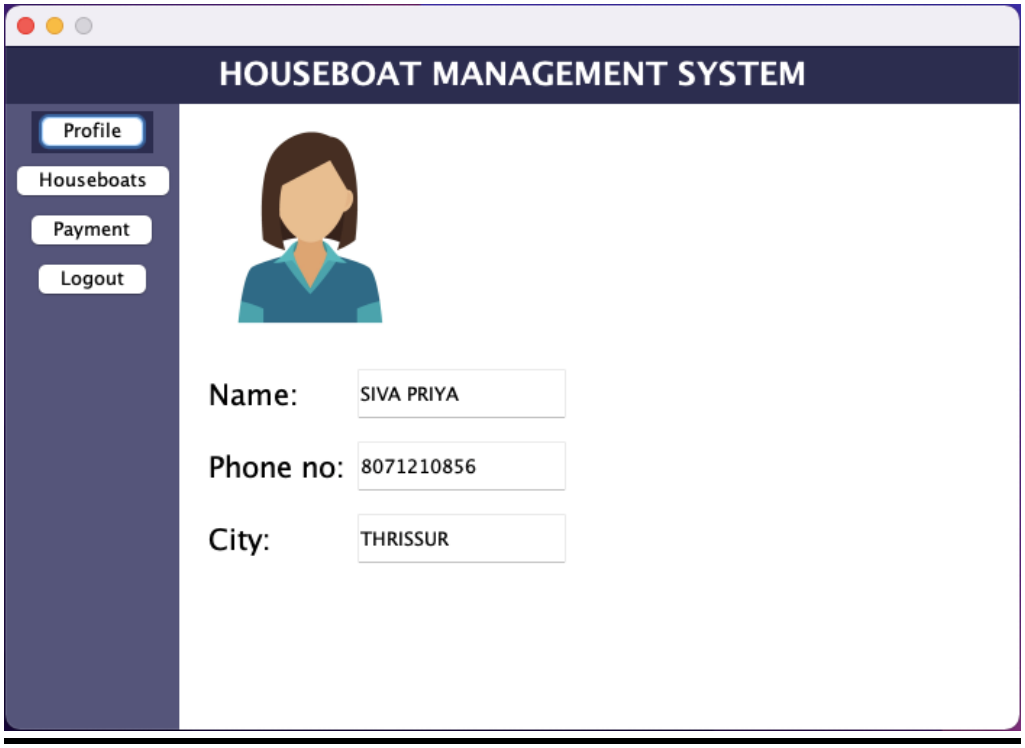
Once we give the proper details and proceed the following pop-up comes and our details are entered into the database thus making an account.

7.6



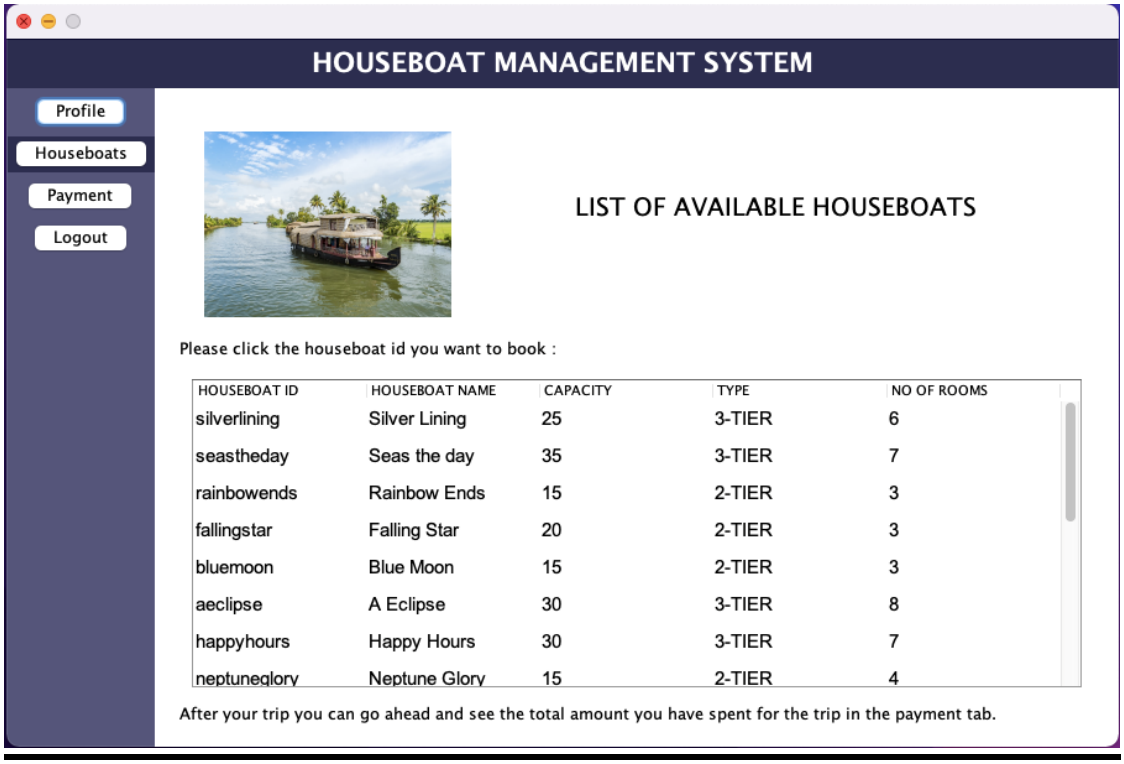
Next we can login to the customer portal with the proper credentials.

7.7



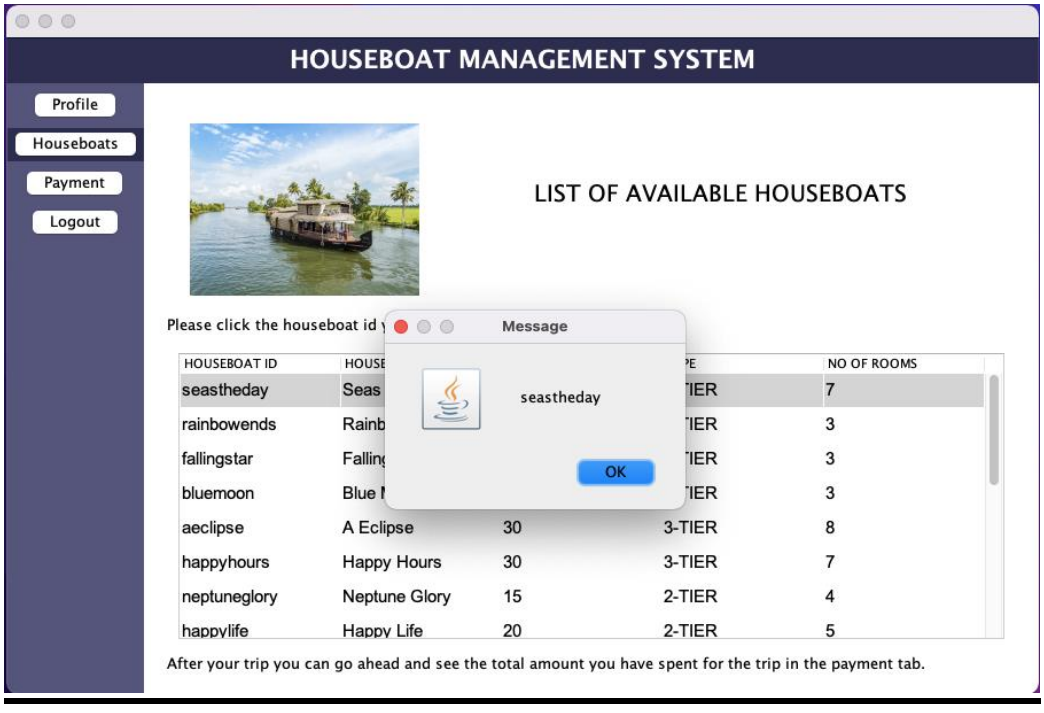
Inside the customer portal we have 4 options to view which are profile, houseboats, payment and logout. In the profile we are given the basic details of the customer as shown above.

7.8



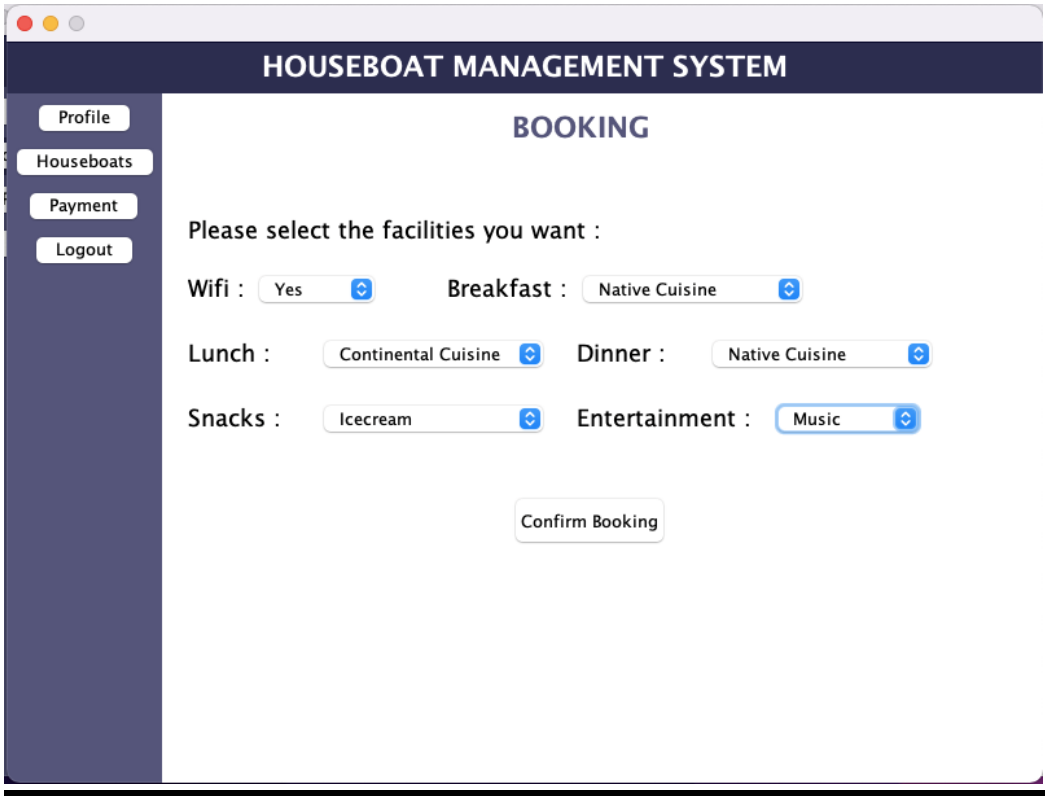
The next option is to view the houseboat. In this tab we can go ahead and click the row out of the desired houseboat to proceed to booking.

7.9



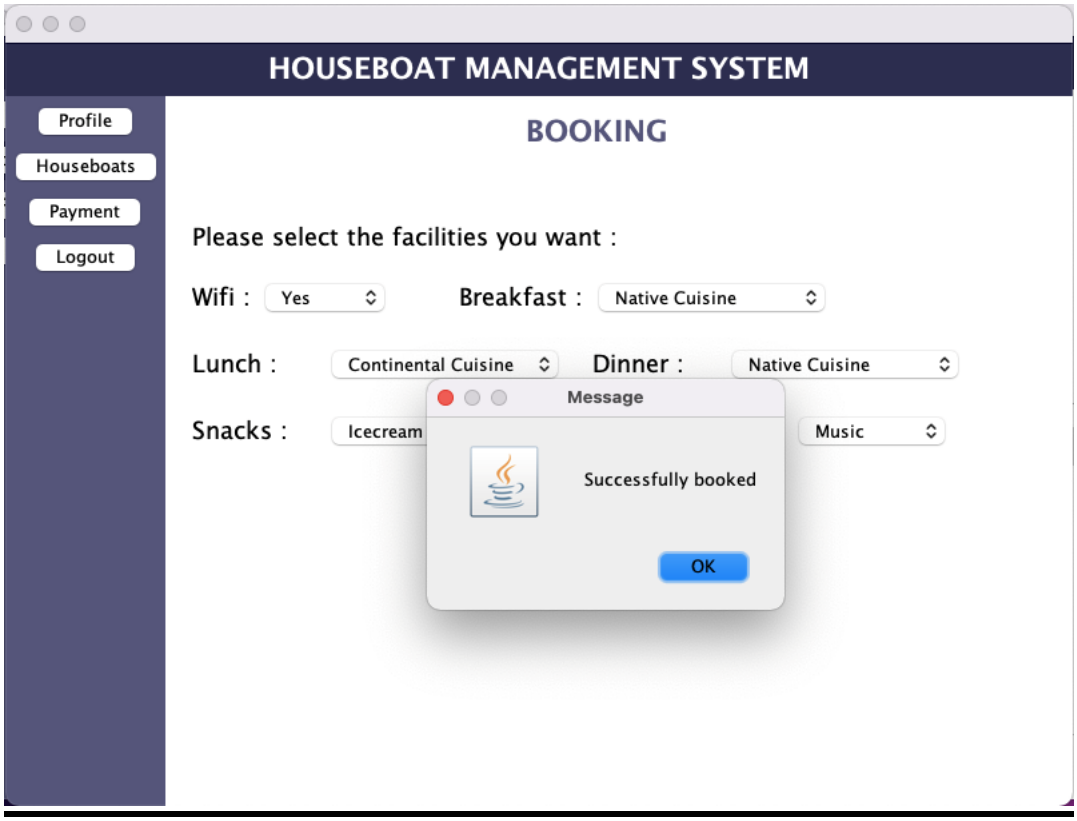
The following tab comes up confirming the houseboats id and proceeds to the next page.

7.10



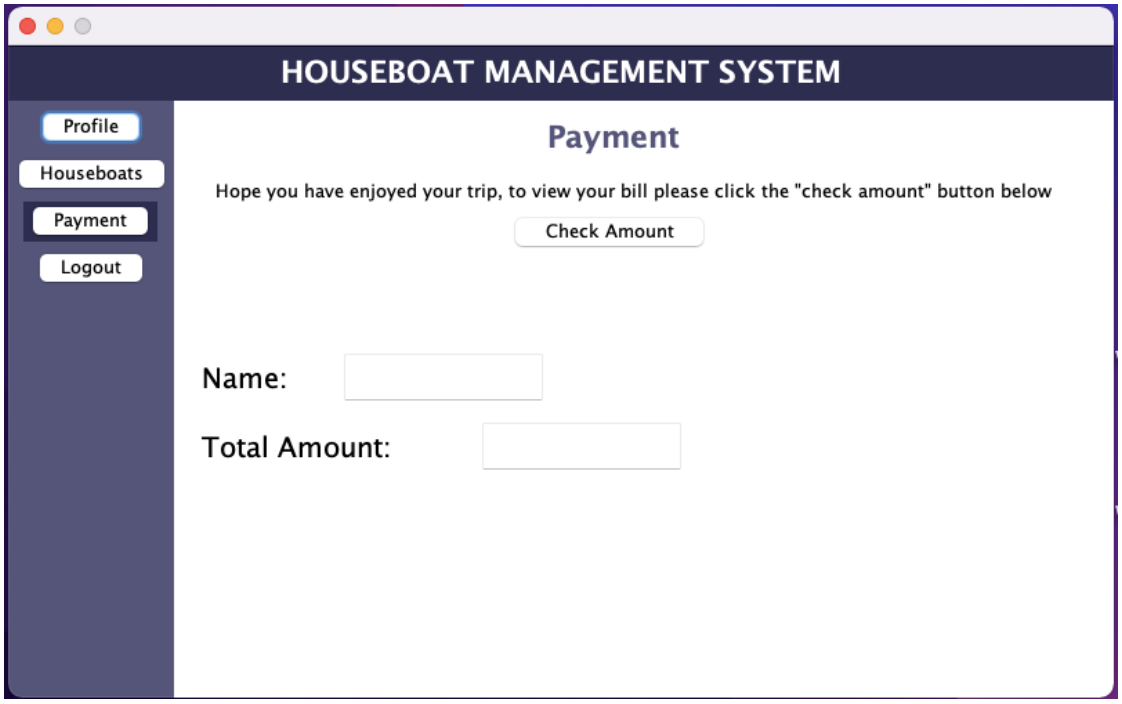
Here we can select the desired facilities we want in our houseboat and click confirm booking.

7.11



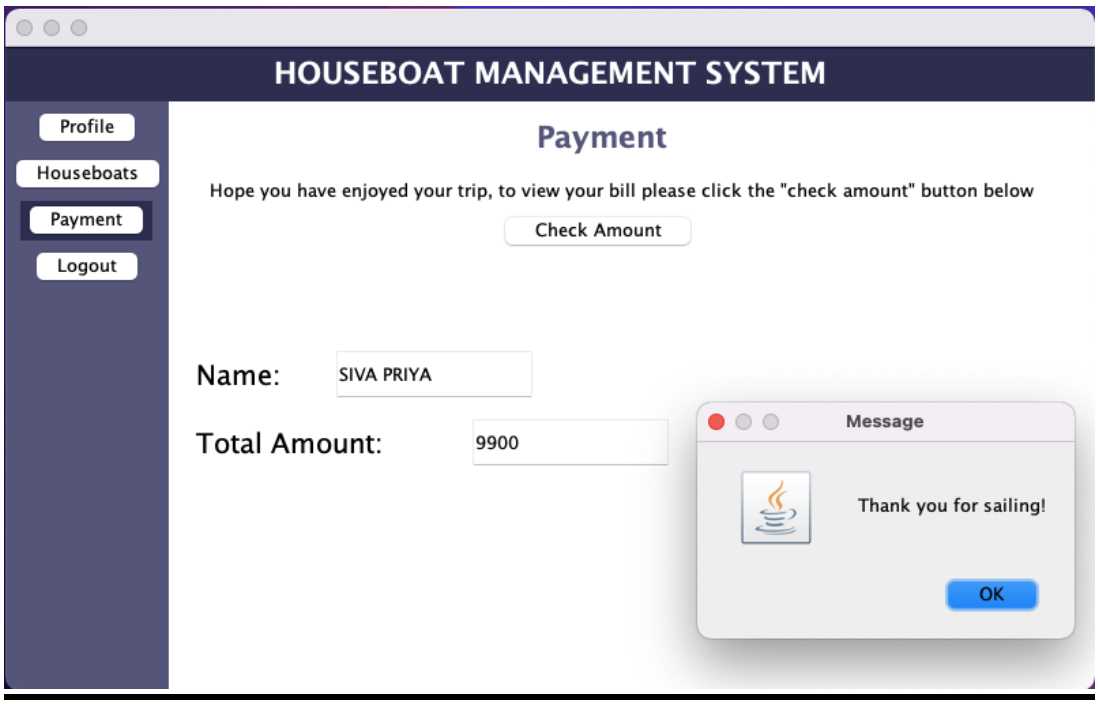
After that a pop-up confirms that we have successfully booked the houseboat.
The booked houseboat is taken away from the table in the houseboats tab as it is already booked by another customer and is not returned until after the trip.

7.12



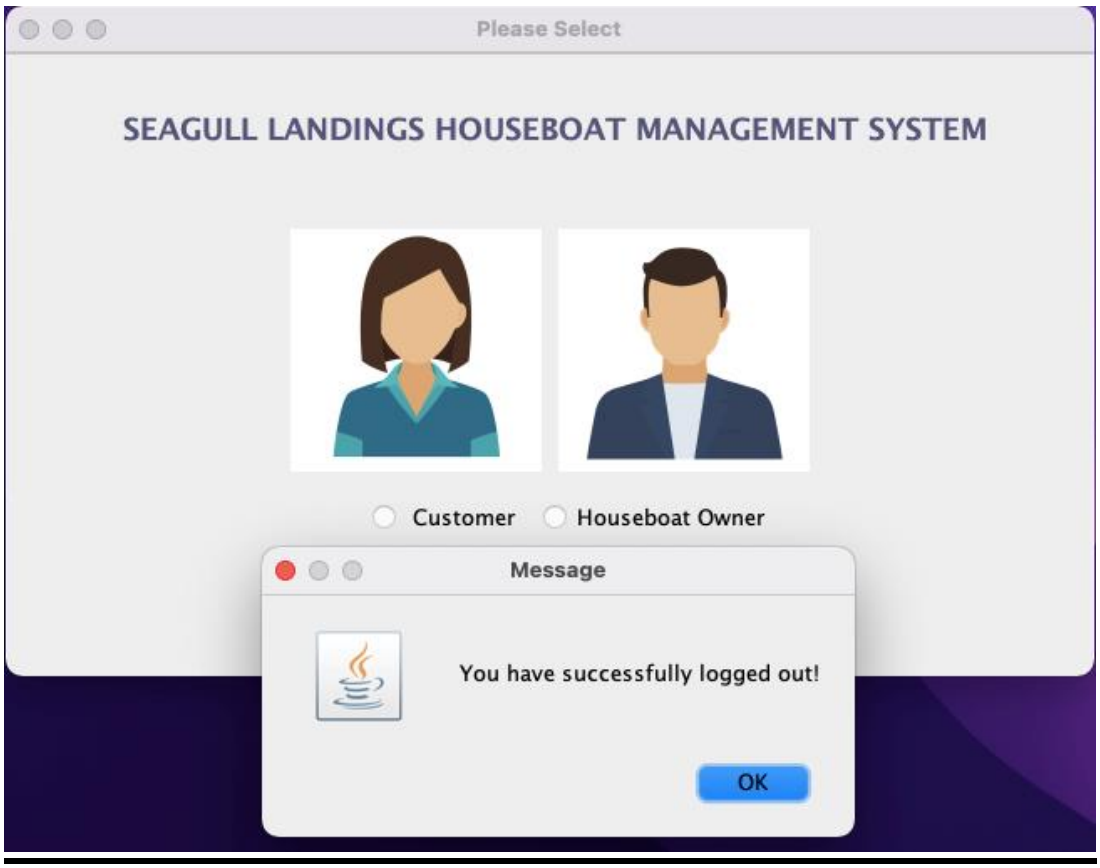
The customer is asked only to pay the amount after he/she is back from the trip and after that they can see their amount by the check amount button.

7.13



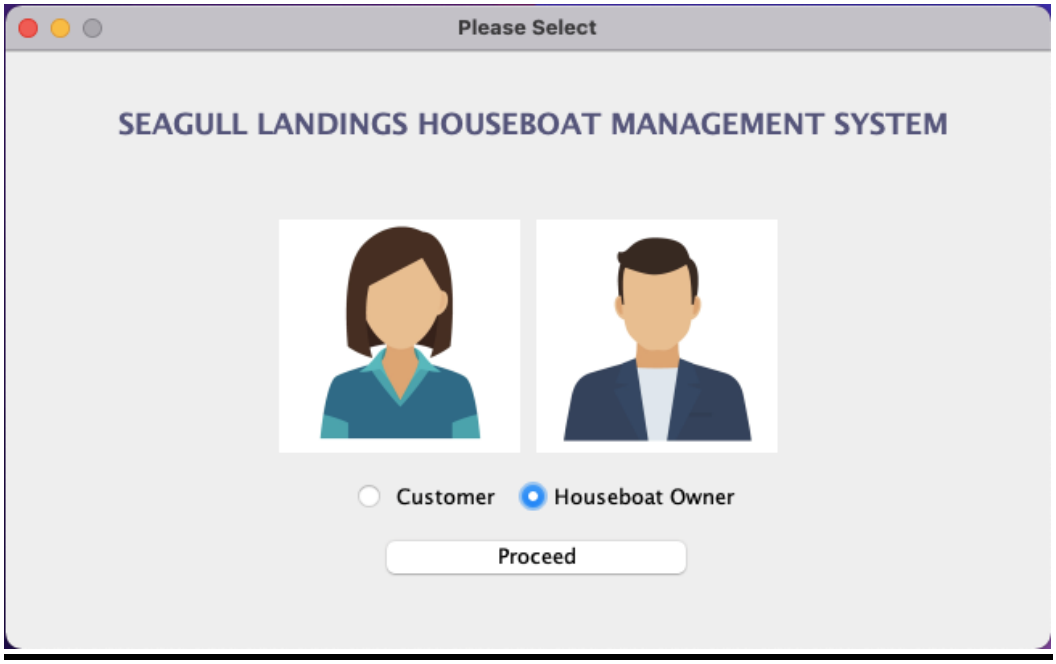
It shows us the total amount for the trip and a pop-up that says thank you. The 4th option is to logout, if you click this it'll take you to the starting page.

7.14



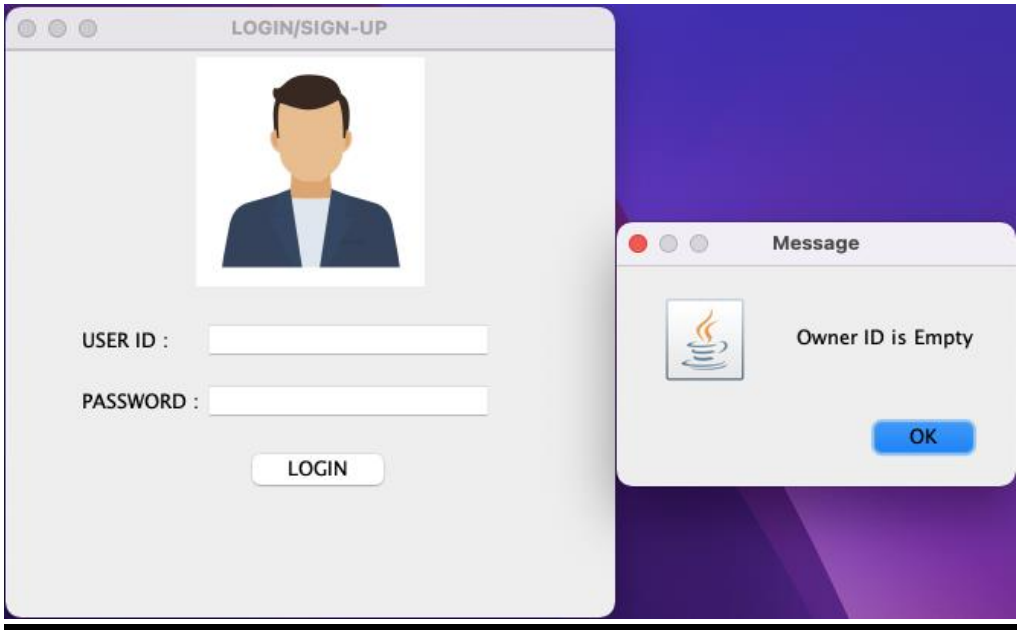
This the page after you click logout.
Any number of customers can login and logout to book their houseboats just like this.

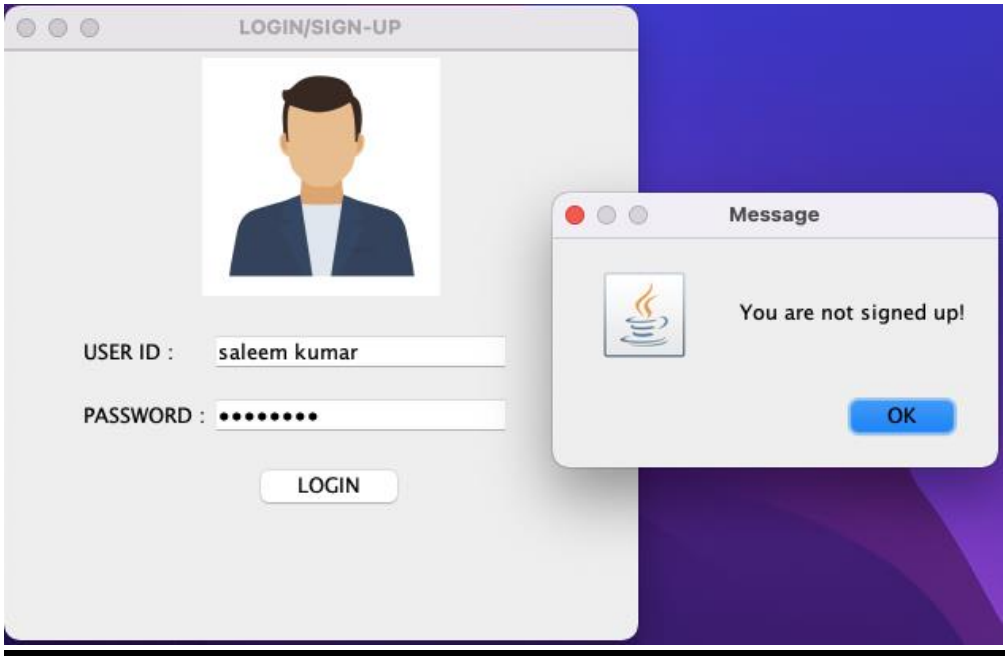
7.15



Next we'll see the owner part of the code.

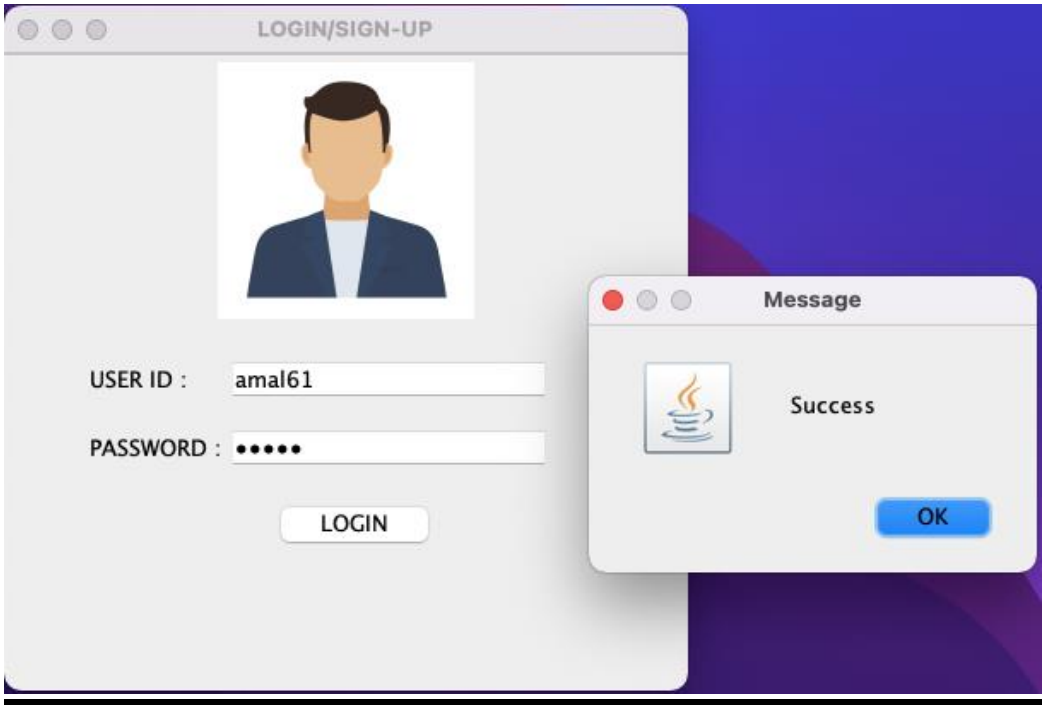
7.16





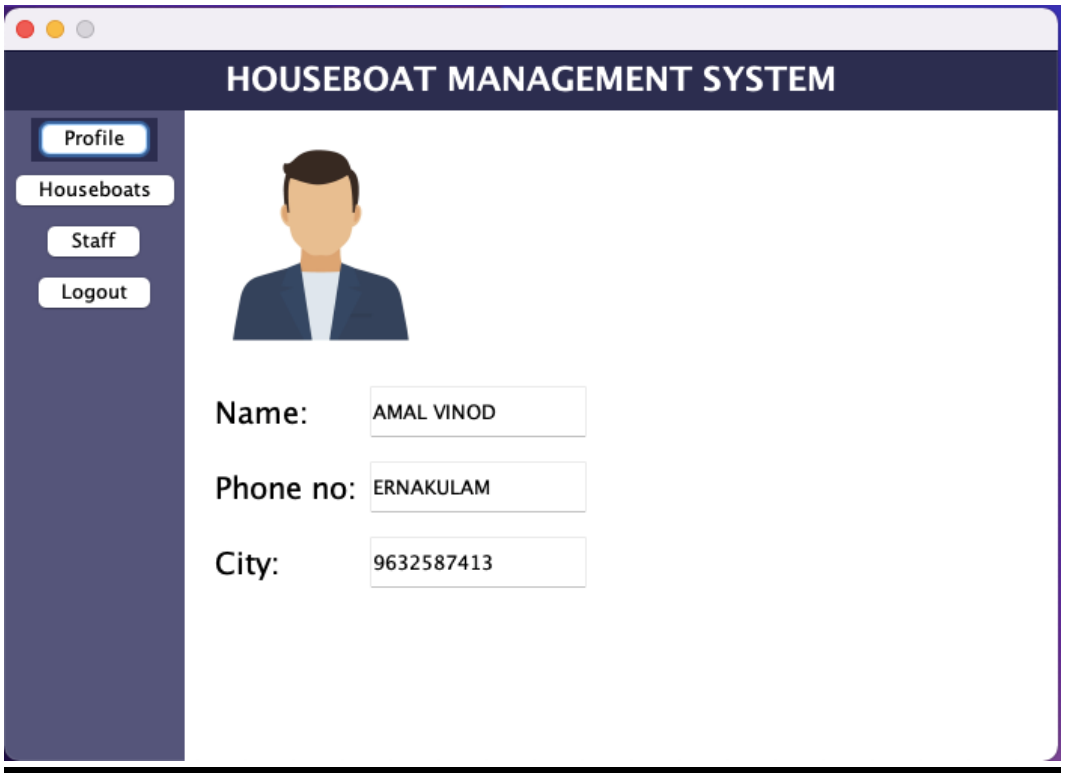
Just as in customer it gives you corresponding pop-ups for invalid or empty textfield.

7.17



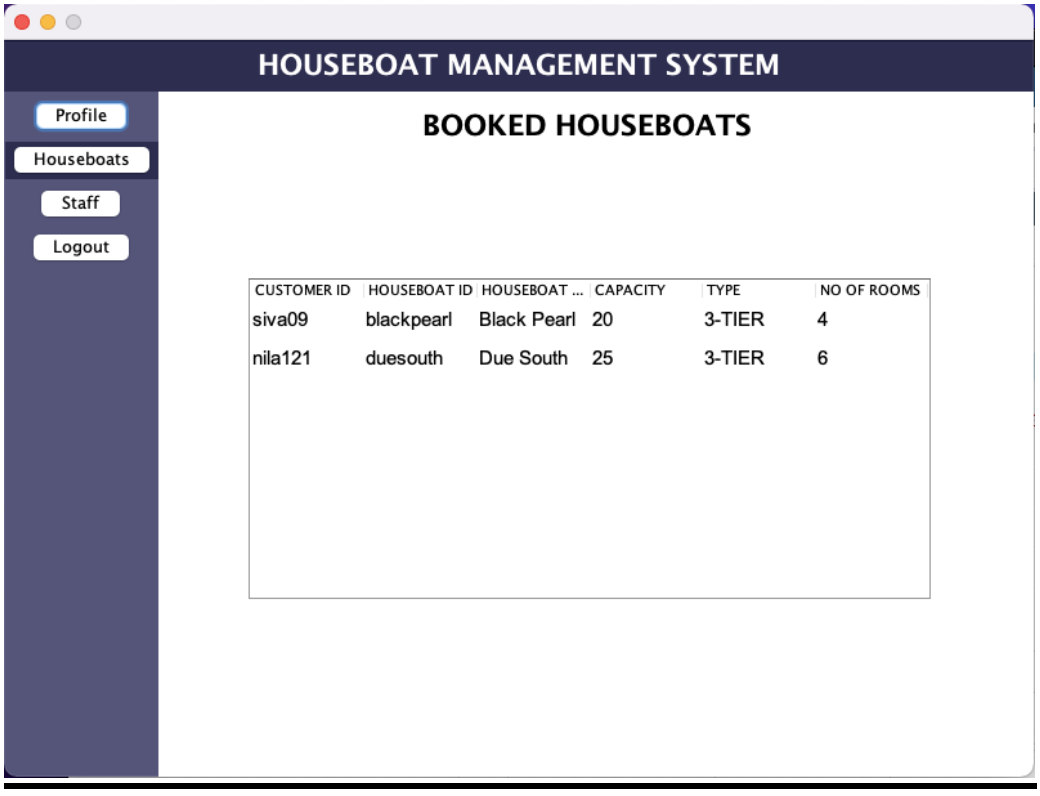
After logging in we are taken to the portal.

7.18



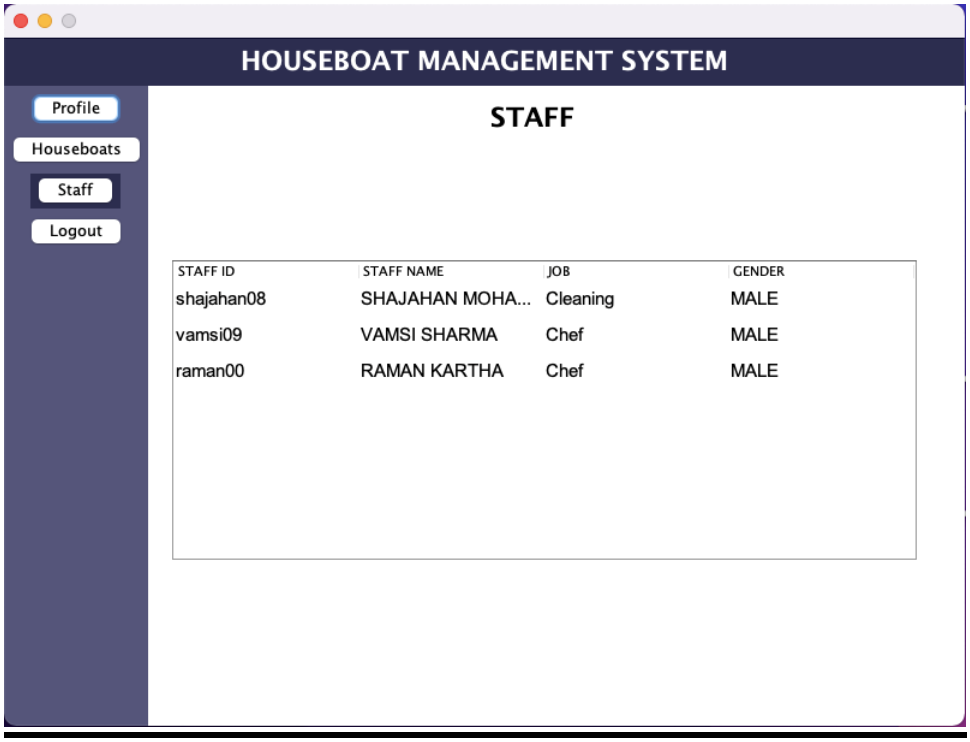
It shows us the details just as in the portal seen before.

7.19



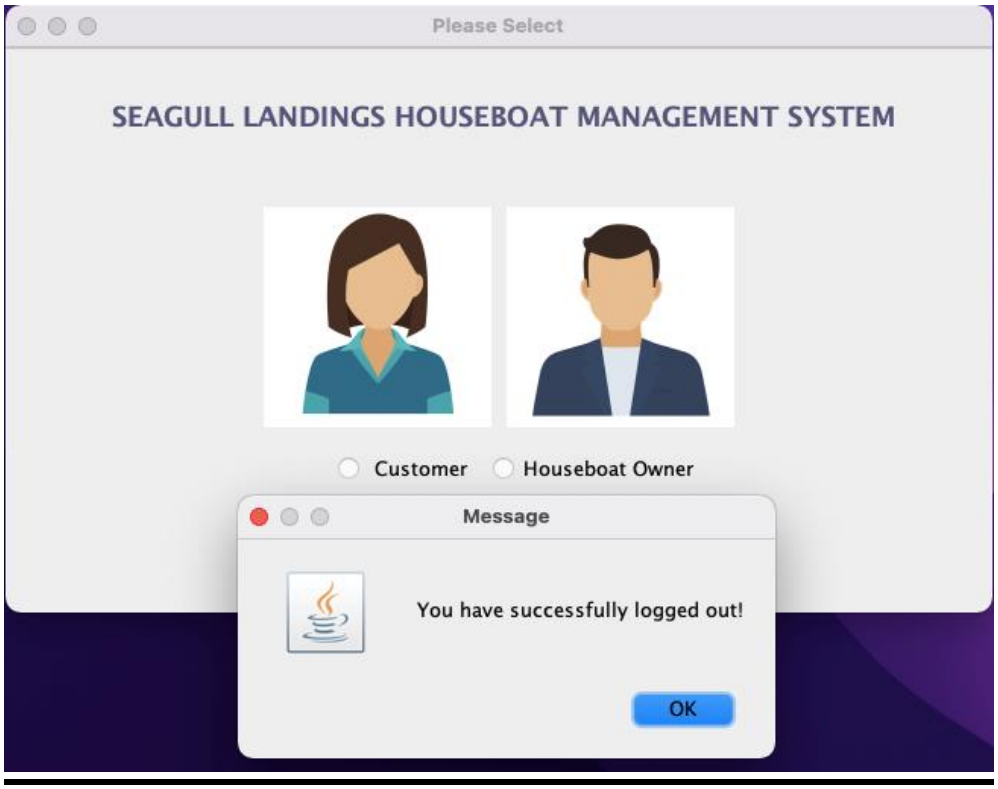
Once we click the houseboat option we can see which all houseboats of the owner are booked by which customers using the customers id, just as shown above.

7.20



In the staff page we can see the staff and their details, working under the houseboat owner.

7.21



After this you can click the logout option to leave the portal.Any number of owners can login and logout to manage their houseboats just like this.