

Restaurant Management System

advanced programming(CSE231S) project





Our team



Hazem Ayman Farouk

2100314(Classes)



Kirellos Bassem Samuel

2101522(Classes)

Adham Ahmed

2100725(GUI)

Mahmoud Yasser

2001200(Classes)

Amin Moustafa

2100483(GUI)





Table of contents

01 UML





  02 Encapsulation

  03 Abstraction

   04 Inheritance

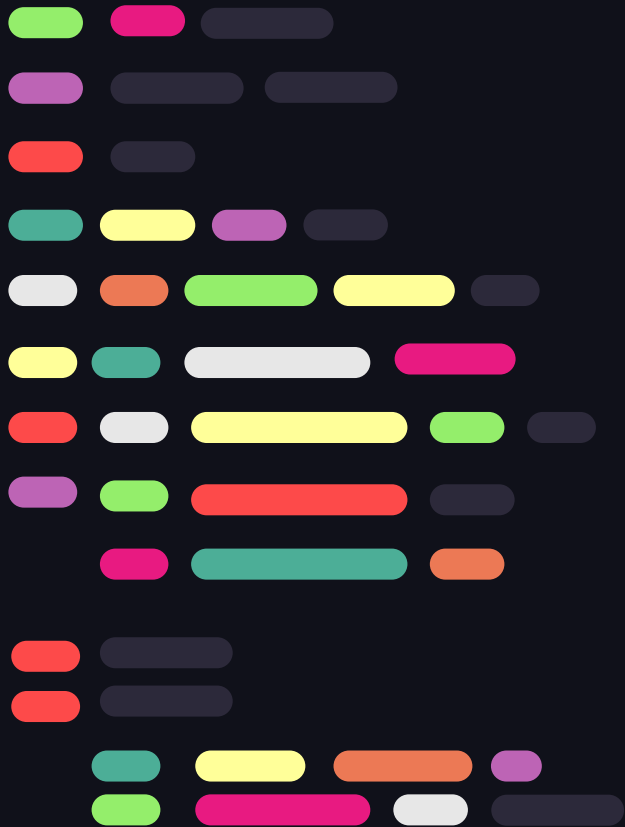
    05 Polymorphism

    06 Interface

    07 Exception handling

    08 GUI

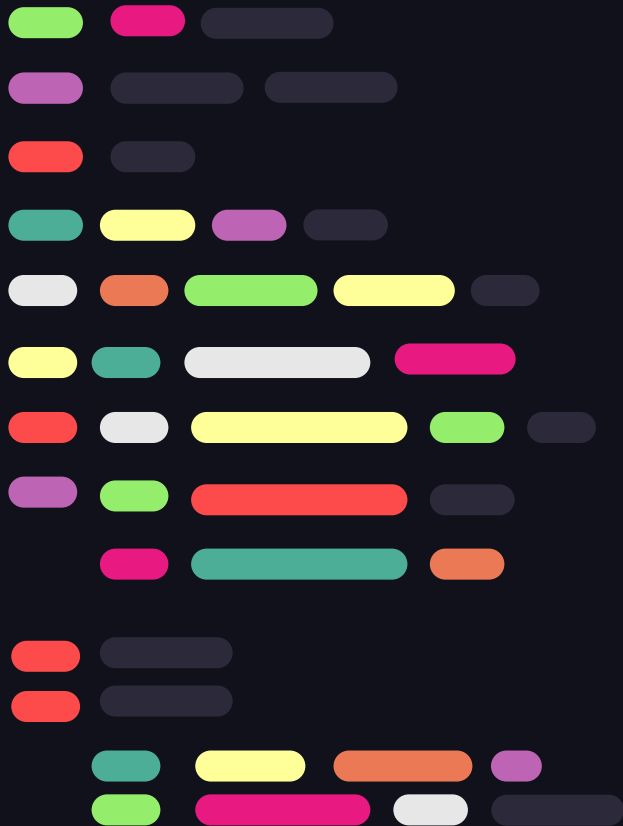


UML



<Unified Modeling Language>





Encapsulation }

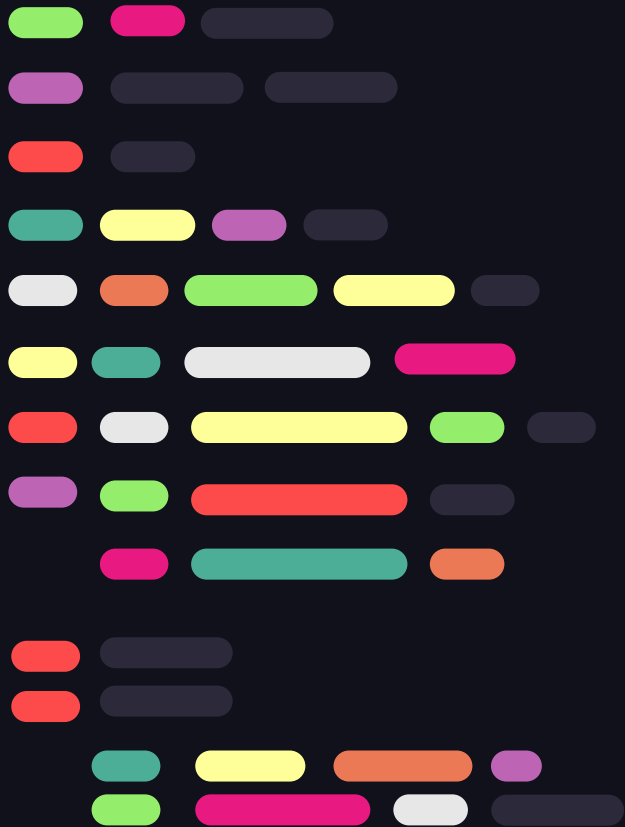


➤ Encapsulation



| Users |
|---|
| <ul style="list-style-type: none">-name: String-UserName: String-password: String-role: String-contactNumber: int |
| <ul style="list-style-type: none">+Users()+getContactNumber():int+setCotactNumber(contactNumber:int)+setRole(role:String)+getName():String+setName(name:String)+getUserName():String+setUserName(userName:String)+getPassword():String+setPassword(password:String)+getRole():String+toString():String |

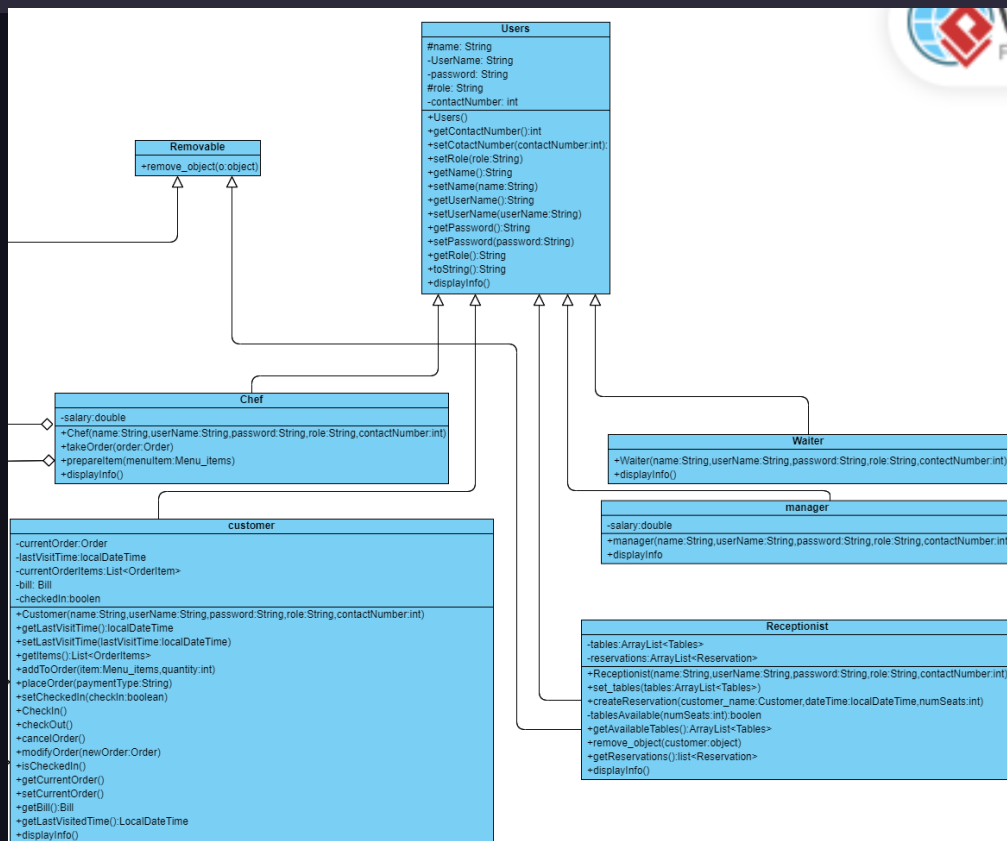
```
public abstract class Users {  
    protected String name;  
    private String userName;  
    private String password;  
    protected String role;  
    private int contactNumber ;  
}
```



Inheritance



➤ Inheritance

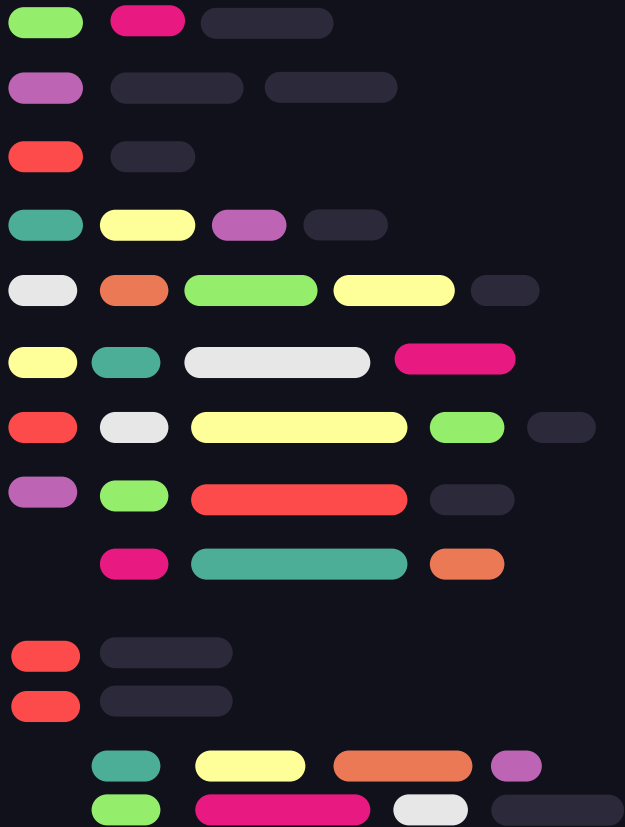


```
14 public class Customer extends Users {
15     private boolean checkedIn;
16     private Order currentOrder;
17     private LocalDateTime lastVisitTime;
18     private List<OrderItem> currentOrderItems;
19     private Bill bill;
20 }
```

```
0 public class Receptionist extends Users implements Removable{
1     private static ArrayList<Tables> tables;
2     private ArrayList<Reservation> reservations;
```

```
7 public final class Waiter extends Users {
8     private double salary;
```

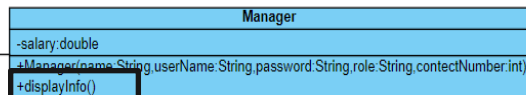
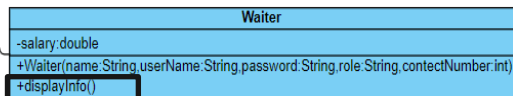
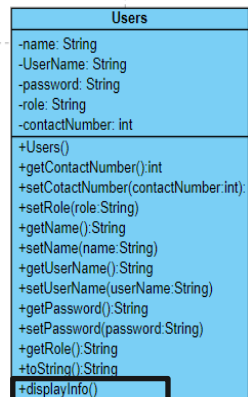
```
public class Receptionist extends Users implements Removable{
    private static ArrayList<Tables> tables;
    private ArrayList<Reservation> reservations;
```



Polymorphism }



➤ Polymorphism



```
@Override
public void displayInfo() {

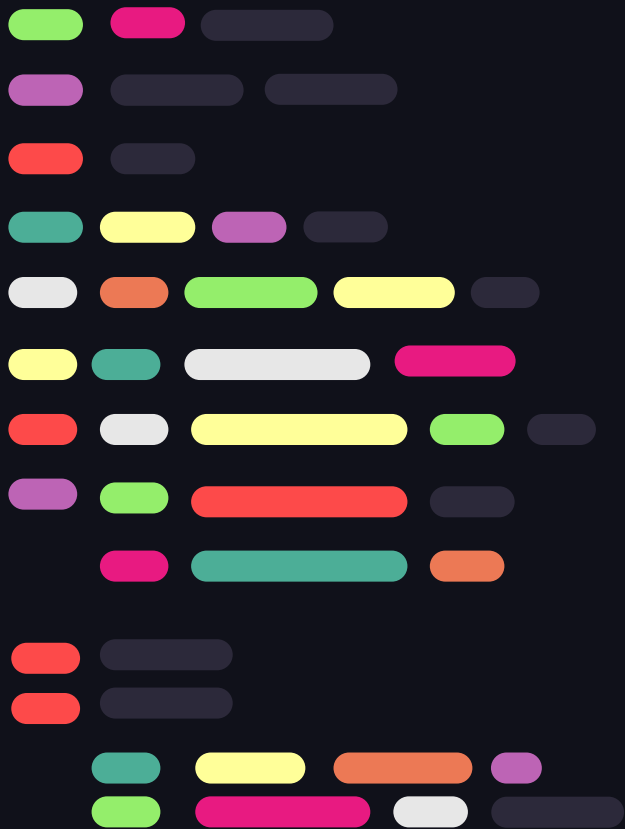
    super.displayInfo();
    System.out.println("Salary: $" + salary);

}
```

```
//add manager
Users managerInstance = new manager("George", "george", "password", "manager", 1205495697,20000);
managerInstance.displayInfo();

//add chef
Users Main_Chef = new Chef("Hassan", "hassan", "chef_password", "Chef", 1205495697,10000);
Main_Chef.displayInfo();

//add Waiter
Users Waiter1 = new Waiter("Ibrahim", "ibrahim", "waiter_password", "Waiter", 1205495697,5000);
Main_Chef.displayInfo();
```



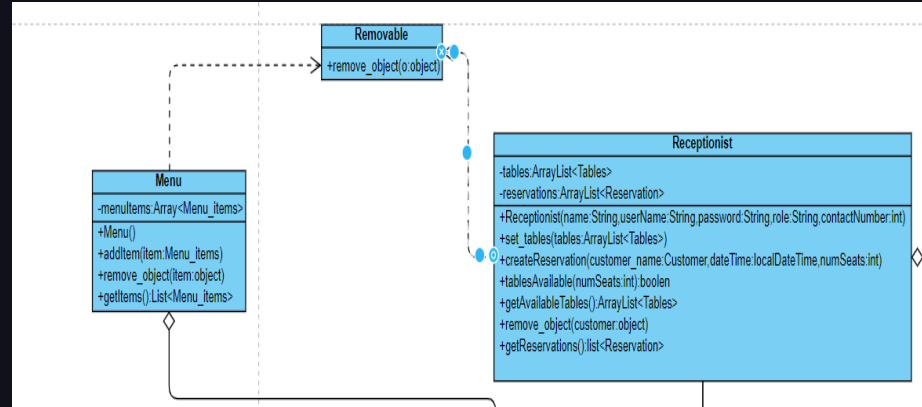
Interface



➤ Interface

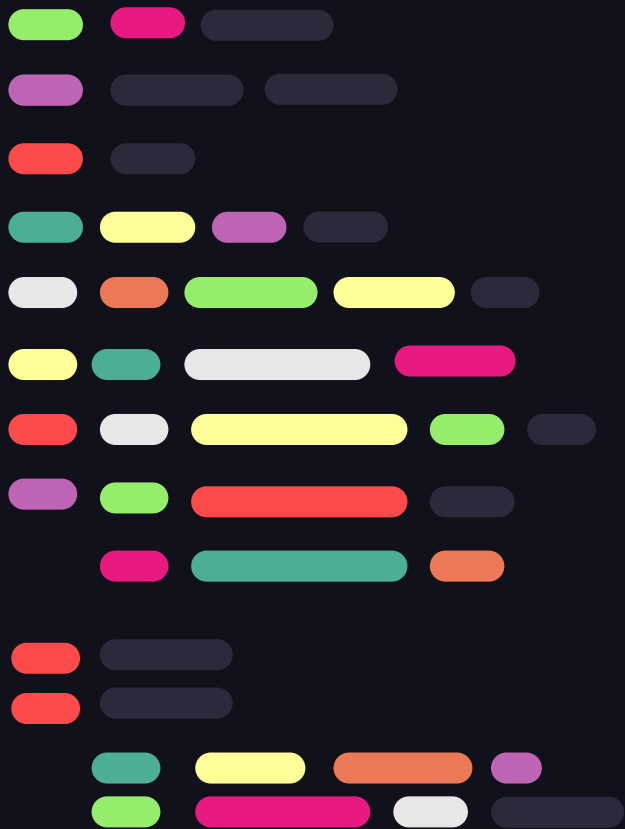


```
public interface Removable {  
    public void remove_object (Object o);  
}
```



```
Receptionist receptionist = new Receptionist("Receptionist", "receptionist", "password", "receptionist",  
987654321);
```

```
Removable removable = receptionist; // Polymorphic reference  
removable.remove_object(customer1);
```



Sorting



➤ Sorting



```
public class Menu_items implements Comparable<Menu_items> {
```

```
    public int compareTo(Menu_items item)
    {
        if (this.price > item.price)
            return 1;
        else if (this.price < item.price)
            return -1;
        else
            return 0;
    }
    @Override
    public boolean equals(Object obj){
        return this.dishType.equals(((Menu_items)obj).getdishType())
            && this.dishName.equals(((Menu_items)obj).getDishName())
            && this.price == ((Menu_items)obj).getPrice();
    }
}
```

Dish type: Salads

Dish name: Salad

Dish price: 6

Dish type: Main Course

Dish name: Spaghetti

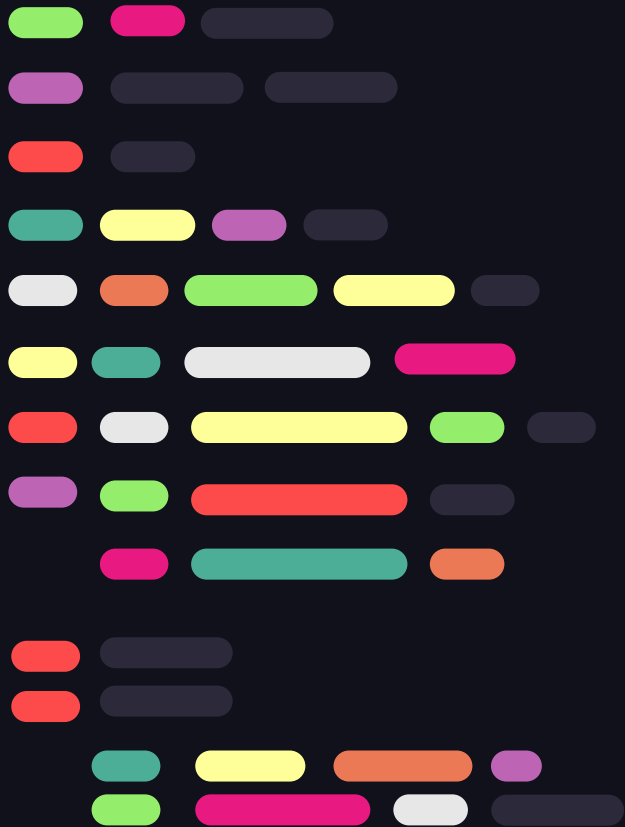
Dish price: 10

Dish type: Main dishes

Dish name: Pizza

Dish price: 12

```
Collections.sort(menu.getItems());
```



Exception handling





Exception handling

```
public void start(Stage primaryStage) throws Exception {  
    try { ...  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

```
try {  
    quantity = Integer.parseInt(quantityInput);  
    if (quantity <= 0) {  
        // Display error for negative or zero quantity  
        showError("Please enter a positive quantity.");  
        quantityField.setText("");  
        return;  
    }  
} catch (NumberFormatException e) {  
    // Display error for non-integer quantity  
    quantityField.setText("");  
    showError("Please enter an integer quantity.");  
    return;  
}
```

The image shows a UI mockup of a shopping cart interface. At the top, there is a text input field containing the value "-5". Below the input field is a purple button labeled "Add item". Underneath the button are two radio button options: "Cash" and "Visa". Below these is another purple button labeled "Order". At the bottom of the interface is a purple button labeled "Cancel Order". A red error message, "Please enter a positive quantity.", is displayed at the very bottom of the screen.

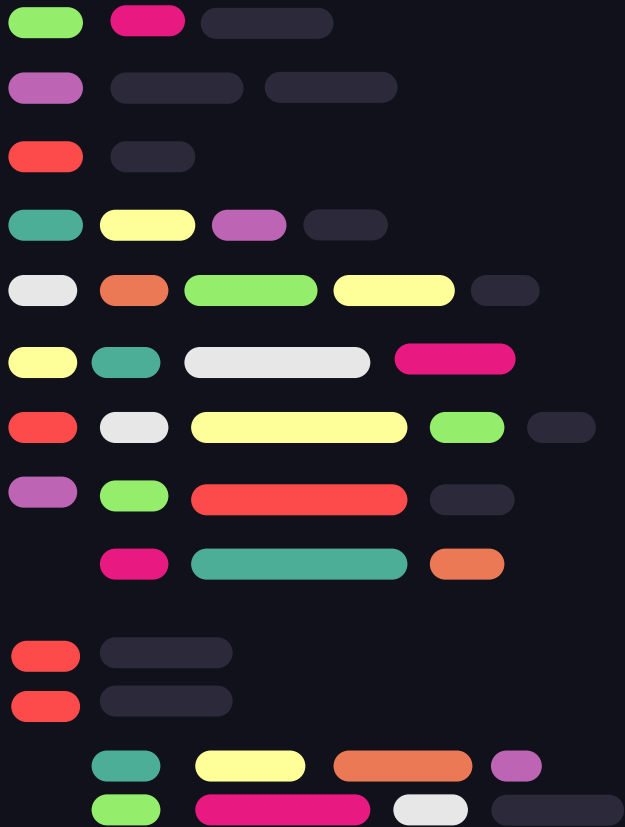
Exception handling



```
55         int contactNumber,  
56         try {  
57             contactNumber = Integer.parseInt(contactNumberField.getText());  
58         } catch (Throwable e) {  
59             errorMsg.setText("Invalid Number");  
60             return;  
61         }  
62         Customer customer = new Customer(name, userName, password, role, contactNumber);  
63         registeredCustomers.add(customer);  
64         System.out.println("New customer registered: " + customer.getName());  
65  
66         // Optionally, you can perform further actions with the customer object  
67     }  
68 }  
69
```

Invalid number type
exception handling

A screenshot of a web form. At the top is a text input field containing "0qwe". Below the field are two buttons: "Register" and "Home". Below the buttons, the text "Invalid Number" is displayed in red, indicating an error message.

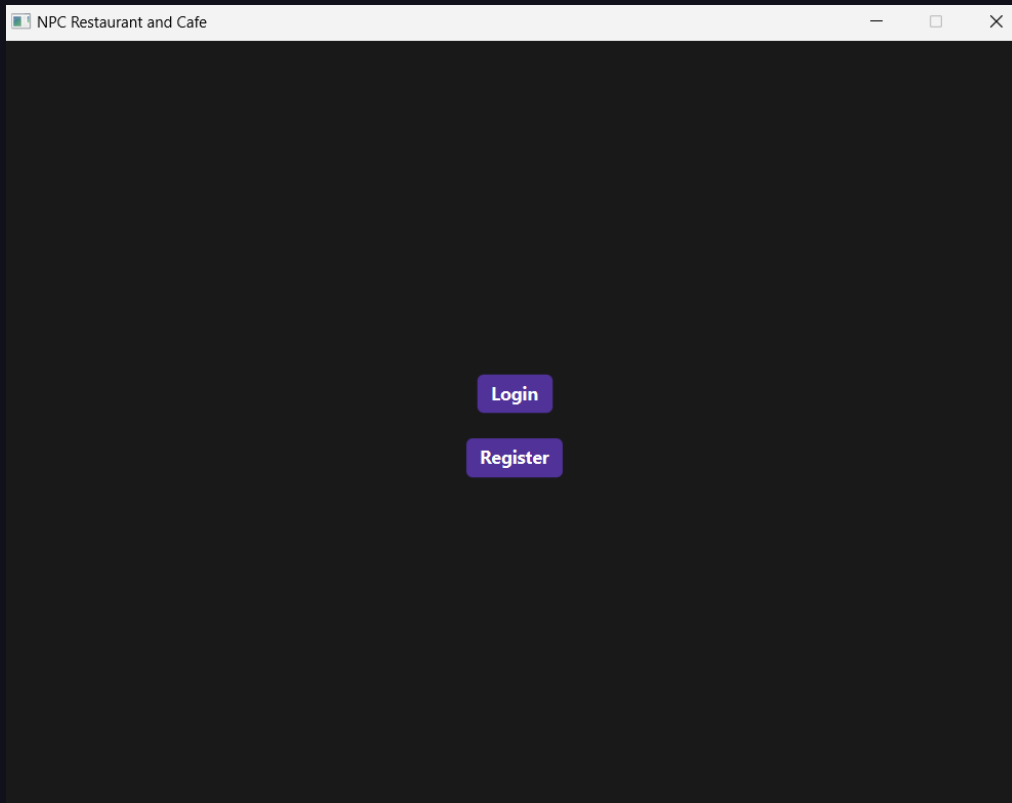


GUI

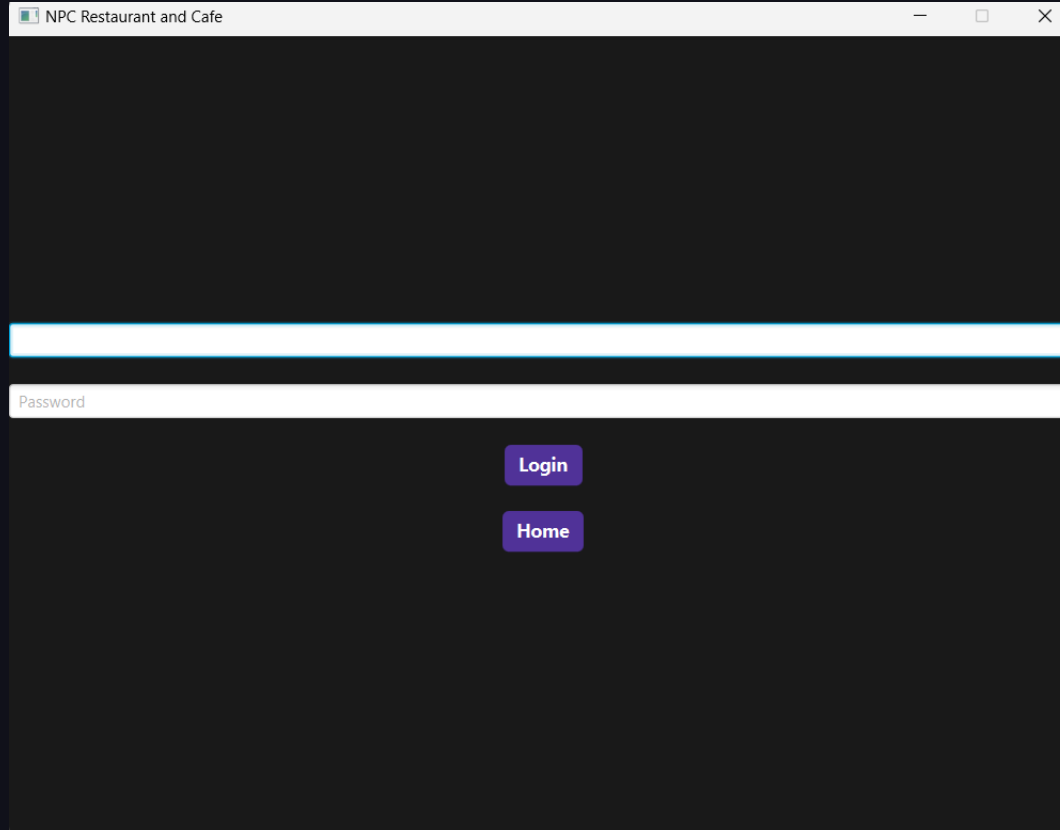




GUI

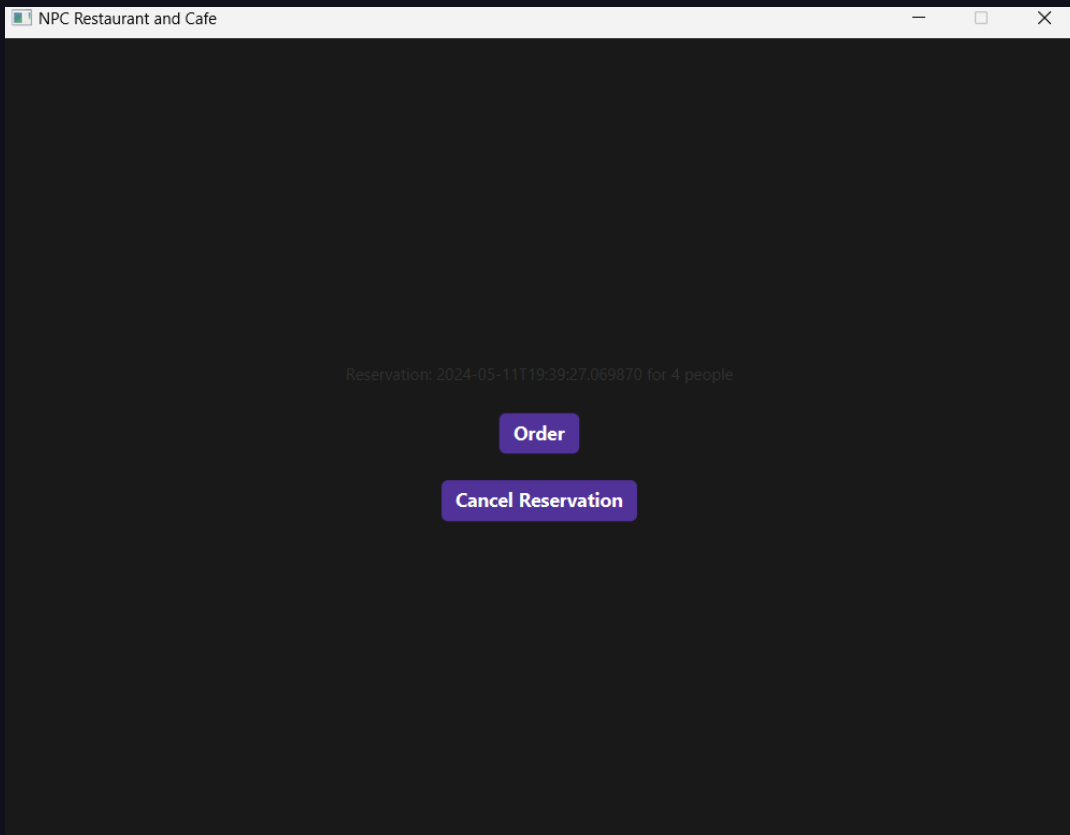








GUI





GUI



NPC Restaurant and Cafe

Dish type: Salads
Dish name: Salad
Dish price: 6

Dish type: Salads
Dish name: Salad
Dish price: 6

Dish type: Main Course
Dish name: Spaghetti
Dish price: 10

Dish type: Main dishes
Dish name: Pizza
Dish price: 12



GUI



NPC Restaurant and Cafe

Dish type: Main Course
Dish name: Spaghetti
Dish price: 10

2

Add item

☒ Cash
☐ Visa

Order

Total Price: \$38.00

Cancel Order

| | | | | |
|-----------|-----------|-------------|--------|-----------------|
| Salad | Price: 6 | Quantity: 3 | Remove | Change quantity |
| Spaghetti | Price: 10 | Quantity: 2 | Remove | Change quantity |



GUI

[illegible]



THANKS