Calm Wind Spa

RESORT & SPA

SPA management system

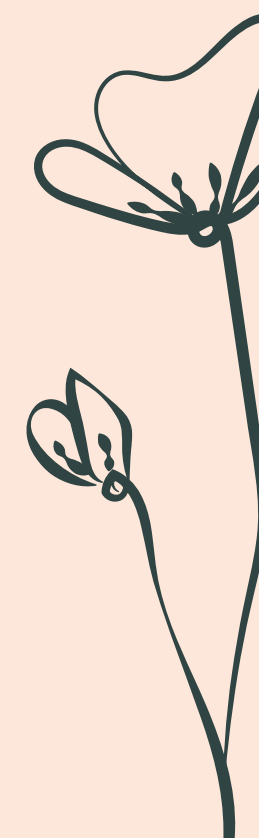# Team's members:

Esraa Ahmed Alamoudi   ID: 441015532

Jenan Ibrahim Mustafa   ID: 442015713

Ruba Hassan Balubaid   ID: 442017269

Teif Turki Aldadi   ID: 442002404

Shahad Maher Alharbi   ID: 442001123

Description
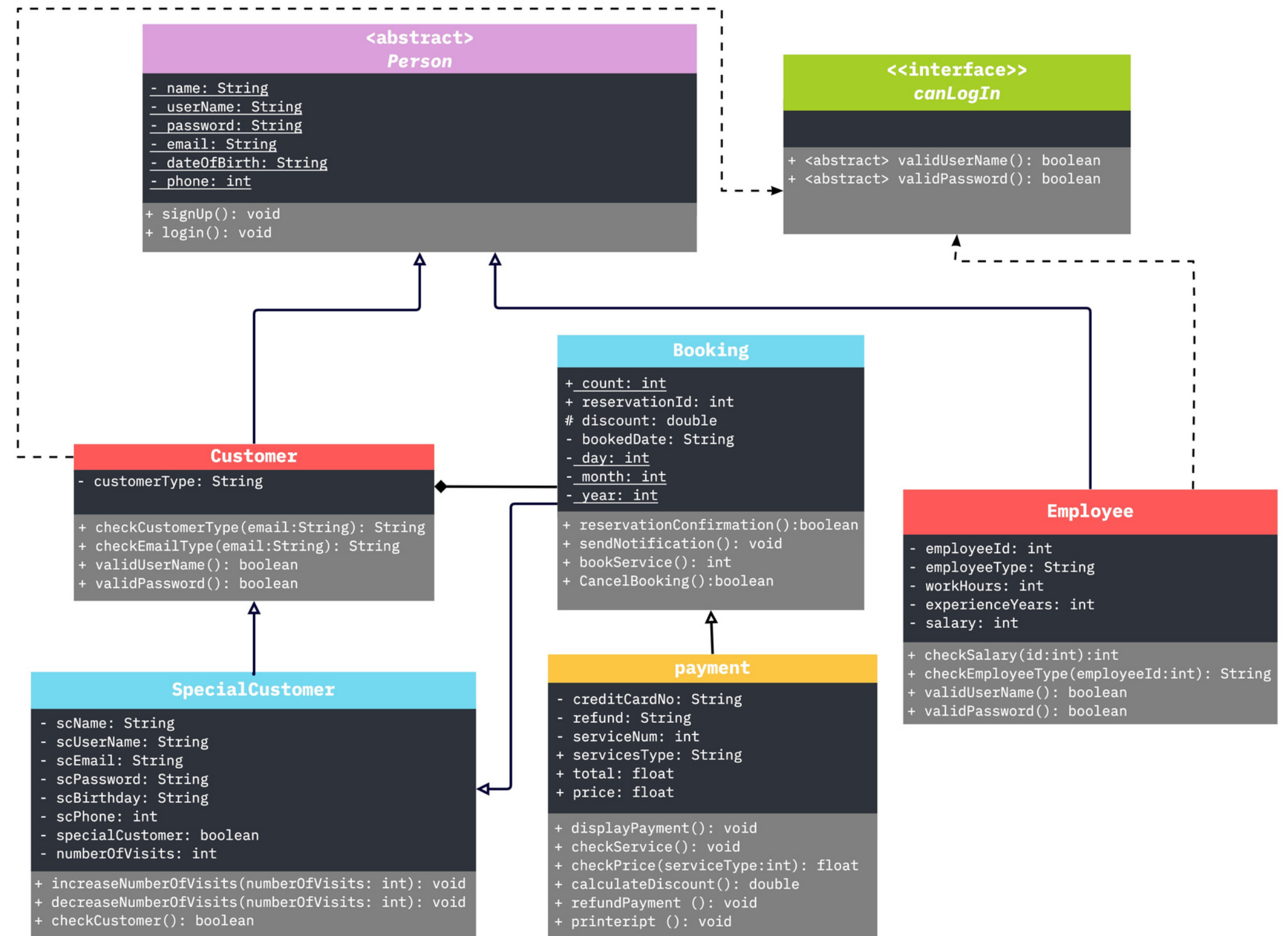
What's the SPA management system doing?

SPA management system

The system lets the user create an account and select if he's a customer or an employee, then if he's a customer the system will be checking if a student or a special customer and display the list of services the user can be chosen and booking a day with other procedures to completing the booking.

UML

The UML diagram of SPA management

****
Person
- name: String
- userName: String
- password: String
- email: String
- dateOfBirth: String
- phone: int
+ signUp(): void
+ login(): void

<<interface>>
canLogIn
+ validUserName(): boolean
+ validPassword(): boolean

Customer
- customerType: String
+ checkCustomerType(email:String): String
+ checkEmailType(email:String): String
+ validUserName(): boolean
+ validPassword(): boolean

Booking
+ count: int
+ reservationId: int
# discount: double
- bookedDate: String
- day: int
- month: int
- year: int
+ reservationConfirmation():boolean
+ sendNotification(): void
+ bookService(): int
+ CancelBooking():boolean

Employee
- employeeId: int
- employeeType: String
- workHours: int
- experienceYears: int
- salary: int
+ checkSalary(id:int):int
+ checkEmployeeType(employeeId:int): String
+ validUserName(): boolean
+ validPassword(): boolean

SpecialCustomer
- scName: String
- scUserName: String
- scEmail: String
- scPassword: String
- scBirthday: String
- scPhone: int
- specialCustomer: boolean
- numberOfVisits: int
+ increaseNumberOfVisits(numberOfVisits: int): void
+ decreaseNumberOfVisits(numberOfVisits: int): void
+ checkCustomer(): boolean

payment
- creditCardNo: String
- refund: String
- serviceNum: int
+ servicesType: String
+ total: float
+ price: float
+ displayPayment(): void
+ checkService(): void
+ checkPrice(serviceType:int): float
+ calculateDiscount(): double
+ refundPayment (): void
+ printeript (): void

List of all important methods in main()

Method	Description
increaseNumberOfVisits	This method increases the number of customer visits by 1
checkCustomer	This method to check if a customer is special or not by checking the number of visits
checkEmailType	This method checks if the email is a student email or not.

Method	Description
checkService	This method checks the chosen services
calculateDiscount	This method calculates the total discount for special customers and students
reservationConfirmation	This method is for confirming the customer's reservation.

Method	Description
displayPayment	This method allows the customer to pay by credit card or cash
cancelBooking	This method cancels the booking
sendNotification	This method sends a notification to a customer after booking

Method	Description
printRipt	This method prints the final receipt to a customer with all the needed details
refundPayment	This method displays the refund of the payment when the customer cancels the booking
validUserName	This method is implemented in the Employee class and Customer class

Method	Description
validPassword	This method is implemented in the Employee class and Customer class
checkEmployeeType	This method checks the type of the employee depending on the employee's department ID
checkSalary	This method checks the amount of salary for each employee depending on the employee's work hours

Calm Wind Spa
RESORT & SPA

All scenarios of main()

All possible outputs

```
run:
**Welcome to Calm Wind SPA**
Enter Account Type:
Employee
Customer
```

GUI

Graphic user interface

we created the Frame from the JFrame class, and then we designed the interface according to our needs using the tools in the same user interface class. We created a basic interface class called JF1 that contains two JButton tools, one for the customer class and the other for the employee class. Next, we used the properties of each tool to change colors and add texts and fonts.

```java
public class JF1 extends javax.swing.JFrame {

    private void button2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        new FEmployee ().setVisible(true);

    }


    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {

        new JF1 ().setVisible(true);
        /* Set the Nimbus look and feel */
        // Look and feel setting code (optional)

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new JF1().setVisible(true);
            }
        });

    }


    // Variables declaration - do not modify
    private java.awt.Button button1;
    private java.awt.Button button2;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    // End of variables declaration
}
```

Calm Wind Spa
RESORT & SPA

Thanks for watching