1. Project Specification

This project aims to make a Java based program to enable user to make a reservation for a hotel room or spa services.

1. Solution Design

This program allows user to access some features to make hotel and spa reservations, and for admin to see all the transactional data. Functional features that should exist in this program can be seen in the Usecase Diagram below.

Diagram

Description automatically generated

1. Implementation
2. Program Scheme
3. Show main menu
4. If guest user pick main menu 1, run point c-l.
5. Get username and password input from user.
6. If username and password matched the data from admin database, run point d to-l.
7. Show menu for admin
8. If admin chooses menu 1, run point g-h.
9. Get name, username, and password input from admin.
10. Call update() function from class Admin. This will renew/update admin’s profile data.
11. If admin chooses menu 2, run point j.
12. Call function printData() from class Hotel. This will print all transaction data for hotel reservations.
13. If admin chooses menu 3, run point l.
14. Call function printData() from class Spa. This will print all transaction data for spa reservations.
15. If admin chooses menu 4, go back to point a.
16. If guess user picks menu 2, run point n-bb.
17. Ask guest user if they already have an account.
18. If guest user answers y (yes), then run point p-q.
19. Get username and password input from customer.
20. If username and password matched the data from customer database, run point r-z.
21. Show menu for customer.
22. If customer picked menu 1, run point t-u.
23. Get name, username, and password input from customer.
24. Call update() function from class Customer. This will renew/update customer’s profile data.
25. If customer picked menu 2, run point s.
26. Call reservationForm() function from Hotel class. This will ask user every necessary information form a hotel reservation and guides user to complete the transaction.
27. If customer picked menu 3, run point y.
28. Call reservationForm() function from Spa class. This will ask user every necessary information form a spa reservation and guides user to complete the transaction.
29. If guest user answers n (nay) then run point bb-
30. Ask user for name, username, and password, then store it in Customer database. Customer is successfully registered and can proceed to point s.
31. Data Structure
32. Primitive Data

This project uses many types of primitive data. From the main class (Reservation class), there are several primitive data variables. mainMenu variable for integer. inputName, inputUsername, and inputPassword for String.

1. Instance Variables and objects.

This project uses instance variables in different classes. In Customer and Admin class, there are 3 instance variables: name, username, and password. In Hotel and Spa class, there’s one instance variable: Scanner s.

This project also uses objects in the main class (Reservation class). admin is an object for Admin class, customer is an object for Customer class, hotel is an object for Hotel clas, and spa is an object for Spa class.

1. Use of imported classes.

This project uses some imported classes. Reservation class imports several classes: Hotel and Spa classes from the package Facility, and Admin and User classes from the package reservation.User.

1. Use of custom-built classes and methods.

This project has several custom-built classes: Facility, Hotel, Spa, User, and Customer classes. All these five classes are established with their own custom-built methods.

Class Facility contains two functions: transaksi(total), and transaksi(cardNumber, total). transaksi(total) method is used when a customer chooses to pay in cash, and transaksi(cardNumber, total) method is used when a customer chooses to pay with a credit card.

Class Hotel contains four functions: two inherited functions from facility, reservationForm() to process a reservation from customer, and printData() to call all the hotel reservation data from database and print them.

Class Spa contains four functions: two inherited functions from facility, reservationForm() to process a reservation from customer, and printData() to call all the spa reservation data from database and print them.

Class Admin contains 10 functions: setter and getter for name (getName(), setName()), setter and getter for username (getUsername(), setUsername()), and the setter and getter for password (getPassword(), setPassword()). The other functions are login() to handle login process, add() to add a new admin, update() to update admin data, and printMenu() to print menu for admin.

Class Admin contains 10 functions: setter and getter for name (getName(), setName()), setter and getter for username (getUsername(), setUsername()), and the setter and getter for password (getPassword(), setPassword()). The other functions are login() to handle login process, add() to add a new customer, update() to update customer data, and printMenu() to print menu for customer.

1. Use of Java Collection

This project uses Java Collection ‘ArrayList’ to store spa menu data taken from database in class Spa.

1. Use of Exception Handling

This project uses exception handling techniques in Hotel, Spa, Admin, and Customer classes to handle file opening, and to handle data writing to a file.

1. Use of Inheritance, Polymorphism, and Interfaces

This project implements inheritance, class Hotel and class Spa both inherit the Facility class. Polymorphism happened in class Facility as there are two transaksi() functions with different amount of parameters. There’s one Interface in this project, and it’s named User.java. This interface is used in Admin and Customer classes.

1. Evidence of Working Program

Graphical user interface, text

Description automatically generated

A picture containing graphical user interface

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated