Helper Document to explain the code

Control Class:-

It controls the overflow of the program. Contains Storage elements for storage of

- UserData(ArrayList<Integer> UserInfo)
- Properties(ArrayList<Property>AllProperties)
- Bookings(ArrayList<booking> AllBookings)
- UserCredentials(HashMap<String,String>UserCredentials) .

(We will call them by just "containers" in this doc). These are made static so that easily accessible in other classes also. Data is loaded in these elements whenever Constructor of control is being called (in the main method). It contains methods to parse strings of data in files.

It also has an object of User class. This User object initially helps in adding data to files. In Login()/Signup() methods this object gets re Instantiated with constructor of manager Class or Customer class.

We have three comma delimited input text files for the same:

1.)UserData:Stores objects of User Class, fills UserData and UserCredentials

2.)Bookings:Stores objects of Booking Class, fills AllBookings

3.)AllProperties:Stores Object of Property Class, fills AllProperties

Important Methods

- parseData,parseData1,parseData2 methods are used to parse Strings of these files and load them in containers.
- addToFile method is used to store data again in these files just before termination.
- SignUp() takes input from console and creates a new User account
- SignUp(User u) just adds this new User to file as well as elements.
- Login() take input from console and display the main menu
- VerifyLogin() to verify Login Credentials using HashMap. HashMap was created specifically for this purpose(for faster operations)
- Logout() to add all data from containers in respective file and terminate the program.

Main menu displays(6/7) options based on User is Manager or Customer.

- Search
- Cancel Past Booking
- See Previous Booking
- Check Balance
- Add Balance
- Logout

- Add new Property(for manager)

User Class:-

Fields(protected to ensure encapsulation)

- Decider-will take value 0/1 depending upon manager or User on respective constructor call
- Name
- UserName
- Pasword
- Balance

Methods

- Update Balance-different overloaded methods to Update balance of current user
- Show Balance-to view UserBalance
- Search(Starting Date, Number Of Nights)-to search for properties (from property container)
  Display available properties and Take input property choice from user.
- Book(Property name,Starting Date, Number of nights) to check user account for sufficient balance, and then confirm booking by deducting money from user account and passing to property manager's account.
- CancelBooking(Property name,Starting Date, Number of nights) – to cancel previous bookings of user and immediate transactions again. Complete amount is not refunded from manager's account and Refund fees deducted.

SubClasses

Customer(decider=0) and manager(decider=1)

Manager has an extra method used to add property.

Property Class :

Fields

- UserName-to Store UseName of manager who added this property
- Name(assumed every property has distinct names)
- Address
- Price Per Night
- Available Dates(instance of calender class)

No important methods as such

Booking Class :

Fields

- UserName(of person who booked the property)

- Name
- Address
- Starting Date
- Number Of Nights

Calender Class:

Fields

- ArrayList <Integer> Available Dates – to store Available dates in for each property

Methods

- Book(Starting Date,Number of nights) – to book property calender needs to be updated. While booking, booked dates are removed from calender.
- Cancel(Starting Date,Number of nights) – While cancelling a booking available dates are added back in the calender.

Link of Video Demonstration:

https://drive.google.com/file/d/1eCSioyoAUda068ICAQ7gd6aT_Ey8f_wE/view?usp=sharing