CSCI 222 – Group B4

Assignment 2 Code Report

Our coding is done in C++.

There are 7 files we have done for this assignment in .cpp and .h

* Admin.cpp/h
* BookedFaci.cpp/h
* ClubManager.cpp/h
* ClubMember.cpp/h
* Facilities.cpp/h
* System.cpp/h
* Text1.cpp/h

All of these files are compiled into Booking.exe.

Explanations of each file are located below.

Admin

Admin has only 2 variables of its own, username and password.

This data is stored in Admin.txt, so when an admin tries to login, data will be read from Admin.txt with readAdmin function and the account information must match one of the lines in the text. If a new admin account is made, edited or deleted, printAdmin will overwrite Admin.txt.

An admin can do these things:

* Add or delete club member and manager
* Edit club member and manager details (as club manager is a form of club member, editID, editName, etc are used for both club manager and member)
* View booking reports by a certain month or by recent booking.

Facilities

A facility consists of these variables:

* facID (facility ID)
* facName (facility name)
* facStatus (facility status)
* facNormalRate (facility normal rate)
* facPeakRate (facility peak rate)
* reqRank (rank required, in integer between 1 to 10)

A club manager can add, update or delete a club facility.

All these information are stored in Facilities.txt. Like with Admin.txt, Print will (over)write the file, while the read function takes in data for the program to read.

BookedFaci

A booked facility consists of these variables:

* refID (a random ID generated whenever you book a facility)
* facID (the ID of facility booked)
* memberID (the ID of member who booked a facility)
* date, time, duration (time-based attributes inputted by a member)
  + Date format in DD/MM/YYYY
  + Time in 24 hour format (eg 23:30)
  + Duration in x hour format (eg 4 hours)
* discountRate, totalCost (discounts given based on club member’s rankings, given by club manager)

Here is how a booking process works:

* Club member wants to book a facility
* A refID will be generated randomly
* Member chooses which facility to book (must exist and be available!)
* Member inputs his or someone else’s memberID (when booking for someone else)
* Member chooses the date, time and duration of booking
* Discount rate and total cost will be sorted by club manager based on member’s rankings)

All these information are stored in BookedFaci.txt. Like with Admin.txt, Print will (over)write the file, while the read function takes in data for the program to read.

Club Member

A club member has these following variables:

* memberID
* memberPass
* memberName
* memberEmail
* memberPhone
* memberRanking
* discountServ

Some information on these variables:

* Each member has a unique memberID and email
* memberPass and memberName must not be the same
* memberRanking is an integer, and must be between 1 and 10
* discountServ is the discount a member gets based on his/her ranking

A club member is able to:

* Search facilities, and with certain criterias the user wants
* Book facilities and edit his/her own booking if needed
* View a user’s own member information
* View a user’s own booking information
* View a user’s own discount

This is all of the criterias a club member can use to search for a facility:

* searchByID
  + search for a facility based on its ID
* searchByName
  + search for a facility based on its name
* searchByStatus
  + search for a facility based on its availability status
* searchByNorRate
  + search for a facility based on its normal rates, then the user can choose on whether to he wants a facility with lesser or higher normal rate than the amount he/she has searched with
* searchByPeakRate
  + search for a facility based on its peak rates, like with normal rate, can select whether user wants a facility with lesser or higher peak rate than the amount inputted
* searchByReqRank
  + search for a facility based on member ranking required to use it, the facilities displayed will be those will less rank required than the amount the inputted

If a search for a facility is successful, this information will be display:

* Facility ID
* Facility Name
* Facility Status
* Facility Normal Rate
* Facility Peak Rate
* Facility Rank Required

Club Manager

Club Manager is a form of a club member (as a subclass), and so he has all variables a club member would have. In addition, a club manager has a contract date. A variable to store a club manager’s contract end date is put into the class.

A club manager also has these functions for him to do his/her job:

* Add, delete or update facilities
* Edit a member’s information (including his own)
* Edit a member’s ranking
* Query (search) for a member’s information
* Get reports for booked facilities based on either:
  + A month the manager inputted
  + Recent bookings
* Process a club member’s bookings

Since a club manager is an extension of a club member, a club manager ID must NOT match a club member’s ID. Similarly, emails should also not match with another manager or member’s.

System

The main function to run the program, you will be presented with 4 choices upon running the program:

* Login as member
* Login as manager
* Login as Admin
* Quit the program