ANUC1110-Introduction to Software Systems

Assignment 1- Semester 2 – 2018

Due date: 7:00pm, 19 September 2018

This assignment will count for 10% of the final grade. Below you will find the

specifications.

Instructions:

- This assignment must be done in groups of two or three.

- Your report must include Assignment Cover Sheet providing the details of this assignment including student IDs. The report submission includes an electronic copy uploaded to Wattle and a hard copy handed in to the teacher during the lecture time in week 8 (19 / 09 / 2018).

- For the program, you are required to commit and push your works to your created GitLab repo for each task completion. Details of the tasks are in the Mark distributions sections.

- Late submission of assessment tasks without an extension are penalised at the rate of 10% of the possible marks available per working day or part thereof. Late submission of assessment tasks is not accepted after 10 working days after the due date, or on or after the date specified in the course outline for the return of the assessment item.

- Plagiarism will attract academic penalties in accordance with the ANU guidelines.

Good luck and enjoy the time you will spend on this assignment

Assignment general specifications:

An effective online hotel booking system for querying and recommendation is expected for this assignment. Three json files named day1.json, day2.json and day3.json contain the booking information. Each file corresponds to one day, documenting the booking records on that day. In a json file, each line is a booking record on that day, specifying the traveler id and the choice of hotel. You can open one of the three json files to see the structure of booking records.

The function of query module in your system is to retrieve booking record given traveler id. Before that, we need to specify which day user wants to refer to, that is, which json file user wants to retrieve. User can choose to retrieve the booking information from the json files by entering number 1 to 4, where the details are as following:

Entry from a user	Data retrieved from json files
1	Day 1 booking information
2	Day 2 booking information
3	Day 3 booking information
4	Day 1, Day 2 and Day 3 booking information

After the specifying which json file we need to refer to, we can read in the booking records (recognised by traveler id, hotel name) from json file and convert them into a collection of java objects. Starting code has been provided using Jackson to show how to get data from json file. You should complete the code to read attributes of each booking record and construct proper objects to save the information. (Hint: arraylist may be an efficient data structure for you)

After saving the given information, input a traveler id in console. Then output the hotel name which the user has reserved on that day. If 4 is specified, output his / her possible booking records for these three days.

An example is given here. If user enters 1, 2 or 3 to specify the day, then enters the traveler id (1 - 20), the console should print out the possible booking information as shown below. Your program does not need to print "[out]" and "[in]", they are used here to tell you whether the line is input that user types in or output that your program should print out.

```
[out]Please enter a number (1-4) to specify a day:
[in]1
[out]Please enter a traveler id:
[in]1
[out]Hotel name: Pacific; traveling time: Day 1
```

If no bookings have been found, the console should print out the following:

```
[out]Please enter a number (1-4) to specify a day:
[in]1
[out]Please enter a traveler id:
[in]20
[out]Hotel name: No record has been found
```

If user enters 4 to specify the day, then enter the traveler id (1 - 20), the console should print out all the possible booking information as shown below:

```
[out]Please enter a number (1-4) to specify a day:
[in]4
[out]Please enter a traveler id:
[in]1
[out]Hotel name: Pacific; traveling time: Day 1
[out]Hotel name: Pacific; Traveling time: Day 2
```

In the next stage, you are required to output an overview of the booking distribution over different hotels on that day or three days. For example, if user enters 1 at the first stage, your system should output the booking distribution on day1, that is, every hotel has 20 percent of travelers on that day.

Your console should look like this at this stage:

```
[out]Please enter a number (1-4) to specify a day:
[in]1
[out]Please enter a traveler id:
[in]1
[out]Hotel name: Pacific; traveling time: Day 1
[out]The overview of booking is:
[out]Hotel name: Pacific; Proportion 20%
[out]Hotel name: Novotel; Proportion 20%
[out]Hotel name: Crown; Proportion 20%
[out]Hotel name: ibis; Proportion 20%
[out]Hotel name: Meriton; Proportion 20%
```

Recommendation function is required in the next stage. Given the booking records, the system should provide useful information for potential consumers. Your program should output the hotel name with most empty rooms for recommendation. The capacity of each hotel (total number of rooms) is given in the json file named hotels.json, and the number of empty rooms is capacity minus the number of rooms which have been occupied. In this part, your program could only handle the situation when only one date is specified in the first stage.

Your console should look like this at this stage:

```
[out]Please enter a number (1-4) to specify a day:
[in]1
[out]Please enter a traveler id:
[in]1
[out]Hotel name: Pacific; traveling time: Day 1
[out]The overview of booking is:
[out]Hotel name: Pacific; Proportion 20%
[out]Hotel name: Novotel; Proportion 20%
[out]Hotel name: Crown; Proportion 20%
[out]Hotel name: ibis; Proportion 20%
[out]Hotel name: Meriton; Proportion 20%
[out]The recommended hotel is : Pacific
```

You are required to:

Assignment Project Exam Help

1) Log in GitLab, fork the assignment repository from the course GitLab repository in the following link: (1 mark)

https://powcoder.com

https://gitlab.cecs.anu.edu.au/anuc1110/Assignment_1.git

Add WeChat powcoder

2) Change the membership of your forked project by adding the teacher and tutor, Ray Lau (u4561496) and Yujia (u6075459), as Reporter. (1 mark)

3) Open eclipse, import the project from your forked project. (1 mark)

4) Fix the program as following:

a) Print appropriate instructions for user's input, that is, tell the user to type in appropriate number to specify which day he/she wants to refer to. Then according to the input from user, read specified json file(s) (day1 to day3) and save the information in proper objects. You need *record* class to save booking information from json. You may also need other classes to solve the problem. (2 marks)

b) Print appropriate instructions for user's input to specify which traveler he/she wants to retrieve. Search traveler information according to traveler id, input from console. And output the hotel name which he/her has reserved on that day or three days with the format given in previous figures. If there is no record for that traveler on that day, your system should print prompt message as shown in previous figure. You will get part of marks if your system can only handle the situation when only one day is specified in the first stage but cannot deal with three days' records. (3 marks)

c) Print booking overview for specified date(s). Every hotel with booking record should be printed like above. You will get part of marks for this part if your program can print the overview of each single day but does not work for three days' records. But hard code is not accepted, such as specifying hotel names in your program instead of reading them from json file directly. (3 marks)

d) Print the recommended hotel based on the number of empty rooms in each hotel. You need to construct appropriate container for hotel names and read them from json file directly, hard code is not permitted. You can use the intermediate result from c) for the number of occupied rooms of each hotel. Together with the capacity of each hotel read from json file, calculate the number of empty rooms in each hotel. Use reasonable method to find the hotel with most empty rooms and print it out. (2 marks)

5) For part c) in question 4, design a JUnit test to check whether the distribution for Day 1 is correct. You can use the given distribution information of day 1 (each hotel occupies 20% of the market on that day). (1 mark)

6) For each solved problem, make sure it runs properly as required, then commit and push the changes to your forked GitLab repository with relevant comments. (2 marks)

7) Contribute to your team works. (1 mark)

8) Report quality (3 marks)

Assignment submissions:

Your submission needs to include the program via **GitLab** and a report that includes:

1) Assignment Cover Sheet (available on the courses Wattle site).
2) The report that details your project including:
   a) Structure of your project and its classes.
   b) Report of how each part of your project has been done. For example, how do you search passenger's information, how to determine whether there is a driver can take this passenger etc.
   c) Screen shots of the running program for each major function fix.

The report is required to be submitted on **Wattle** by the **due date** (7:00pm, 19 September 2018) and a **hard copy** of it needs to be handed in to the teacher on the due date in the classroom. One report per group.
Note: Failed to submit the report (either the electronic or hard copy) will apply 5 mark deduction out of your possible assignment result.

3) If you have used external resources for your assignment, you must cite the source in both GUI and your report.