ANUC1110-Introduction to Software Systems

Assignment 1- Semester 1 – 2018

Due date: 7:00pm, 03 April 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.
- 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 7 (03 / 04 / 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:

Shared riding deals with traffic congestion as well as reduces emissions. We expect you to develop travelling plans for passengers according to their start points, destinations and travelling time, i.e. if the passenger’s route is covered by a driver’s (for both x axis and y axis) and passenger’s travelling time is in the period of the driver’s, then he/she could send requirement to that driver. And maybe the passenger will meet other sharing-friend whose route and time is also covered by the drivers.

We provide two json files, saving passengers’ information and drivers’ information respectively. Passengers are recognised by user id, name, start point, destination, travelling time, while drivers are recognized by driver id, name, start point, destination, start time and end time. We have provided you with start code to show how to get data from json file. You should complete the code to read attributes of each people 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 user id in console. The output should contain: user’s name, start point, destination and travelling time. After searching for available driver (you can output anyone of them), decide whether there are other sharing friends with the passenger (assume the car can take all of them), then print result.

If there is no available driver, your console should look like:

```
Please enter user id:
12
Passenger's name: Jack
Start point: ANU
Destination: UCLodge
Travelling time: 2018/1/16 9:00

Sorry, there is no available driver right now.
```

If there is available driver, your console should look like:

```
Please input user id:
12
Passenger's name: Jack
Start at: ANU
Destination: UCLodge
Travelling time: 2018/1/16 9:00

Available driver: 2
Sharing friend: 34 67
```

“2” is the driver id of the available driver. “34” and “67” are user id of sharing friends whose route and travelling time are also covered by the same driver.

You are required to:

- Log in GitLab, fork the assignment repository from the course GitLab repository in the following link: (1 mark)
  https://gitlab.cecs.anu.edu.au/anuc1110/Assignment1
- Change the membership of your forked project by adding the teacher and tutor, Ray Lau (u4561496) and Yujia (u6075459), as Reporter. (1 mark)

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

- Fix the program as following:
  1. Read json files and save the information in proper objects. You need user class and driver class to save information from json. (2 marks)
  2. Search passenger information according to user id, input from console. And output the passenger's name, start point, destination and travelling time. (2 marks)
  3. Determine whether there is a driver can take this passenger. To simplify, assume all cars run from left to right, from bottom to top. So, "cover" means the driver's start point is on the bottom left of the passenger's and destination is on the top right of the passenger's. Besides, passenger's travelling time should in the period of the driver's start time and end time. (Hint: there is java api to save and compare time) After search, print the result like above. (3 marks)
  4. Determine whether there are other sharing friends with the passenger, whose trip is also covered by the same driver. Print the result like above. (2 marks)

- For each solved problem, test to make sure it runs properly as required (2 mark)

- Commit and push the changes to your forked GitLab repository with relevant comments. (2 marks)

- Contribute to your team works (if work in pairs). (1 mark)
  1. Clearly demonstration of work load distribution and overall contribution in percentage

- 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, 3 April 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 have to cite the source in both GUI and your report.