TCP1201 Objected-Oriented Programming and Data Structures

Assignment 1 (20%)

Trimester 2, Session 2020/2021
Faculty of Computing and Informatics
Multimedia University

DUE DATE: 10 January 2021, 11:59pm

1. FORWARD

- This is a group assignment with maximum of **TWO (2)** students per group.
- The deadline for the submission is **10 January 2021 (Sunday) at 11:59pm**. The time given is more than enough if you start early. Hence, no late work will be accepted and no deadline will be extended.
- Do not share your code with any other group. If detected all parties involved may get zero marks.

2. TASKS

You are developing a simplified contact tracing system for COVID-19. The three (3) main roles in the system are listed below:

1) Customer – a customer has name, phone, and status. A customer can:
    a) Register an account in the system.
    b) Sign in the system
    c) Check-in a shop
    d) View the history of the shops he/she visited
    e) View his/her status. A customer status can have 3 possible values: Normal, Case, Close.
        i. Normal – The customer is normal.
        ii. Case – The customer is a case of COVID-19 positive.
        iii. Close – The customer is a close contact of a case. For simplicity, a close contact is anyone who checks in the same shop within one-hour range (inclusive) with a case. The seconds in check-in time are ignored.
        For example, if a case checks in a shop today at 12pm, then:
            - A customer who checks in yesterday is not a close contact.
            - A customer who checks in today at 10.59am is not a close contact.
            - A customer who checks in today at 11am is a close contact.
            - A customer who checks in today at 1pm is a close contact.
            - A customer who checks in today at 1.01pm is not a close contact.
2) Shop – A shop has name, phone, status, and manager. A shop status can have 2 possible values: Normal, Case.
3) Admin – a government agency who can:
    a) View the details of all customers, the details of all shops, the master visit history.
    b) Flag (change status) a customer. When a customer is flagged as a Case, the system shall automatically flag:
        i. The shop as Case.
        ii. All close contacts as Close.

The histories and records shall be displayed in the following format during execution:
Master visit history:

```
No   Date       Time      Customer  Shop
1    2020-12-01 15:10:15  Ali       Tesco
```

History viewed by a customer:

```
No   Date       Time      Shop
1    2020-12-01 15:10:15  Tesco
```

List of customers viewed by admin:

```
No   Name       Phone       Status
1    Ali        0121112222  Normal
```

List of shops viewed by admin:

```
No   Name       Phone       Manager   Status
1    Tesco      0388889999  Siti      Normal
```

Design your classes, data fields, and methods wisely. You may add classes or data fields to support inheritance, association, aggregation, composition, or significant features. Do not over design (adding classes, data fields, or methods that do not help solving the assignment, e.g. full address).

To make testing easier and save time during interview, your program shall have the following features:

i. Never clear screen.
ii. Pre-load with 5 customers (when the program starts).
iii. Pre-load with 4 shops.
iv. Have a feature to add 30 random visits to the master visit history (random shops, random visitors, time ranging from yesterday until current system time). Sort the master visit history by date and time.

3. RECOMMENDED TASK DISTRIBUTION

- A group member programs the customer and shop roles, while another member programs the admin role.
- To enable maximum separation of work, each member writes his/her program separately and share the data in files (you have 2 programs in this way). Design your data files wisely so that it can be easily used by the 2 programs.

4. SUBMISSION

i. Java source code. Make sure the code can be compiled and run.
ii. Java documentation (Javadoc) for a class. Choose a class that have methods with parameter and return type.
iii. A file named Members.txt that lists down the group members' id, name, and contribution.

```
Member 1 Id Name
Member 2 Id Name

TASK DISTRIBUTION
1. Customer and shop – contributor & percentage
2. Admin – contributor & percentage
```

3. Javadoc – contributor & percentage

Zip the THREE (3) files above and label it in the following format:

TTX_Student1ID_Student1Name_TTX_Student2ID_Student2Name.zip

For example:

TT1L_1191112222_Ali_Yusof_TT3V_1192221111_Pravin_Sriram.zip

Mark Sheet (20%)

Criteria	Items (Mark for an item is awarded if it works and student can explain)
1. Program Execution (13 marks)	**1.1. Correct program features and output (11m)** The values of all useful relevant data must be shown when displaying info. i. Register and sign in as Customer. (1m] ii. A customer can check-in more than one shop sequentially. (1m) iii. More than one customer can check-in a shop sequentially. (1m) iv. Customers can view their respective status and visit history. (1m) v. Shops can view their respective status. (1m) vi. Admin can view list of customers, list of shops, and master visit history. (2m) vii. Can add a history of 30 random visits with sorting by date & time. (1m) viii. Admin can flag customers as cases (change status to Case). (1m) ix. The system automatically identifies and flags both the shop (change status to Case) and the close contacts (change status to Close). (1m) x. Customer's and shop's status are correctly updated. (1m)
	1.1. User friendliness (2m) 2.0m – Input and output are clear without ambiguity. 1.5m – One input or output are unclear with ambiguity. 1.0m – Many input or output are unclear with ambiguity. 0.5m – The program is unusable.
2. OOP Design and Java Documentation (5 marks)	**2.1. Association, Aggregation, and/or Composition (1.5m)** 1.5m – Excellent design (sensible modeling with good identifier name) 1m – Fine design (sensible modeling with minor issue) 0.5m – Poor design (insensible modeling) 0m – Do not use
	2.2. Inheritance (1.5m) 1.5m – Excellent design (sensible modeling with good identifier name) 1m – Fine design (sensible modeling with minor issue) 0.5m – Poor design (insensible modeling, poor identifier name) 0m – Do not use
	2.3. Java Documentation for one class (2m) 0.5m – Correct class description 0.5m – Correct method description 0.5m – Correct parameter description 0.5m – Correct return description
3. Bonus (2 marks)	1m for each of the following items: i. Use files to save and share data. ii. JavaFX/Android for customer registration and sign-in. iii. JavaFX/Android for customer's visit history admin's master visit history, admin's list of customers. or admin's list of shops.
4. Presentation & Interview (2 marks)	2m – Very smooth and well prepared 1.5m – Overall fine with minor issues 1m – Average 0.5m – Poorly prepared or has major issues

5. Late submission, not attending interview, or plagiarism	0 mark for this assignment