NATIONAL UNIVERSITY OF SINGAPORE

CS2103/T – SOFTWARE ENGINEERING

(Semester 2: AY2018/19)

Part 2

Time Allowed: 1 Hour

INSTRUCTIONS TO STUDENTS

1. Please write your Student Number only. Do not write your name.
2. This assessment paper contains **THREE** questions and comprises **TEN** printed pages.
3. You are required to answer **ALL** questions.
4. Write your answers in the space provided.
5. This is an **OPEN BOOK** assessment.
6. You may **use pencils** to write answers; but write legibly.

STUDENT NO:	**A**								

This portion is for examiner's use only

Question	Marks	Remarks
Q1	/16	
Q2	/14	
Q3	/10	
Total	/40	

Description used for all questions in the paper (Q1-Q3):

You are developing a **server-based application for food orders and deliveries**, named *Foodgle* (similar to Deliveroo system). The users interact with the application through mobile applications. The application suite has a few parts to manage the following:

- Restaurants
- Customers - order food from different restaurants
- Couriers - bring the food from the restaurant to the customer
- Orders - posted by customers to various restaurants
- Payments - received and made among customers, restaurants and couriers.

You are assigned to work on the **mobile application used by food couriers**. However, you are involved in designing the architecture of the whole system as well. The payments are managed through an external system that is not developed by you.

The API among components includes (but not limited to) the following:

(i) The courier gets informed by the central server about areas with restaurants in high demand for deliveries. The central server assigns orders to couriers for delivery.
(ii) The courier accepts orders, delivers the food, and marks an order as completed.
(iii) The courier retrieves from the central server the details and status of an order with a restaurant; however the restaurant is unaware of the courier until the order is assigned and accepted.
(iv) The courier directly communicates with the customer and the restaurant via the app once an order is confirmed and accepted by all parties.
(v) In addition, the courier has access to in-app navigation. This in-app navigation suggests the fastest route to the destination based on the current traffic conditions.

Q1 [Total 16 marks] Short questions on software engineering concepts.

a. [2 marks] Enumerate the main components for Foodgle system.

Answer:

b. [4 marks] Draw a high level architecture diagram of Foodgle. Reduce the coupling between the components.

Answer:

c. [4 marks] Explain the reasons for adding coupling among components.

Answer:

d. [2 marks] What is the type of architecture most suitable for designing Foodgle. Explain if this type of architecture does not match some of the requirements.

Answer:

e. [4 marks] Which SDLC process are you most likely to follow in developing Foodgle? Briefly explain what activities are planned in each phase of the chosen process.

Answer:

(*Blank page)*

Q2 [Total 14 marks] UML diagrams

You are now developing the **mobile application used by food couriers,** called *Foodgle-courier*. Assume that Foodgle-courier is always connecting successfully to the central Foodgle server, and that the interaction happens using API calls provided by the server.

a. [6 marks] Consider the following interaction between the courier (human user) and the Foodgle-courier (mobile app) for **accepting an order**:
   1. The courier logs in to Foodgle-courier to indicate his availability.
   2. Foodgle-courier determines the courier's location and simultaneously lets the courier know about the "hot spots"—locations where the demand for delivery is high.
   3. The courier can either choose to move to one of the high-demand locations and wait, or stay at his/her current location and wait for orders.
   4. On receiving information about the order, the courier accepts to deliver the order.
   5. Foodgle-server sends a one-time-password to the courier's registered hand phone number for authentication. The courier enters in Foodgle-courier app the one-time-password received from the server to authenticate himself/herself. Upon authentication, the order is considered accepted for delivery.
   6. After authentication, the courier can retrieve information about the restaurant and the customer.
   7. Once the food is delivered, the courier marks the order as completed.

Model this workflow of Foodgle-courier (mobile app) accepting an order using an appropriate UML diagram.

Answer:

b. [8 marks] At the server side of the Foodgle application, the information is modeled using classes.

Description: The courier can deliver up to two orders from a restaurant at once and not more. Delivery info includes information about the contact details of the customer that includes the address and phone number and the bill amount that the courier uses to verify the order with the customer. The courier can retrieve the restaurant info if necessary using the order, but not always necessary to do so.

The following code snippets are used to implement the above description:

```
class Courier extends Contact implements Payer {
    //… other implementation
    String courierID;
}

class Customer extends Contact implements Payer {
    //… other implementation
}

class Restaurant extends Contact implements Payee {
    //… other implementation
}

class Bill {
    Boolean pay(Payer payer, Payee payee, int amount) {
        //… other implementation
    }
}
```

```
class Delivery {
    Courier courier;
    Order order;

    Route getRoute(Contact customer) {
        //… other implementation
    }
}

class Order {
    Restaurant restaurant;
    Customer customer;
    Courier courier;
    Status status;
    Bill bill;
    List<Item> items;
    //… other implementation
}
```

```
interface Payer  {
    int getCardNumber();
}

interface Payer {
    int getBankAccount();
}

abstract class Contact {
    Address a;
    Phone phoneNumber;

    Address getAddress() {
        return a;
    }

    Phone getPhone() {
        return phoneNumber;
    }
}
```

Code 1: Class Structure

In addition to the classes in the code, include the following classes: `Item`, `Status`

Illustrate the class structure of **Code 1** using a suitable UML diagram. Show all associations as lines (arrows). Show attributes, methods, navigabilities, dependencies, and known multiplicities. Other optional notations can be used only when they add value.

Answer:

Q3 [Total 10 marks] Testing

Assume you have been asked to unit test the `isValidCourierID` method used in code. You can view the header comments, where you have the following information:

Courier identifier should be 3 to 6 alphanumeric characters long and start with a lower-case letter. Accepted alphanumeric characters are lower-case letters [a-z] and digits [0-9]. Other alphanumeric characters are not allowed.

a. [2 mark] What are the equivalence partitions for testing `isValidCourierID`? Briefly explain your answer.

Answer:

b. [8 marks] Give four test cases for testing `isValidCourierID`. Explain why you chose each test case.

Answer:

#	Courier ID	Explanation
1		
2		
3		
4		

---END of PAPER---