CS3203 Test 2 (2020/21 Sem 1)

**[Section III]**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Name** | **Student No.** | **Group No.** |
| **1** |  |  |  |
| **2** |  |  |  |
| **3** |  |  |  |

**Declaration:**

**We contributed equally for this section.**

**We did not contribute equally for this section.**

*If your group select this option, the teaching team will be in touch with you soon to gather details on individual contributions.*

**Duration: 40 minutes Max Marks: 40**

1. Answer ALL questions in this section as a group. Submit only ONE (1) set of answers at the end of the test.
2. You can type / handwrite your answers on a hardcopy / softcopy of the question paper or other sheets of paper.
3. **Create a PDF** **file** containing **your answers** and **the required information as shown in the box above**.
   1. **Name your PDF** using your group number (i.e., T01A, T14B).
   2. Ensure that the required information (as shown in the box above) are clearly indicated in your file.
   3. Ensure that there are **FIFTEEN (15) pages or less** in your PDF. Any answers written beyond page 15 will **NOT** be marked. You can consider deleting pages containing the questions.
   4. Ensure that there are **NO landscape pages** in your PDF. If you wish to include diagrams in landscape mode, rotate the diagram clockwise by 90 degrees.
   5. Ensure that your answers are legible when printed on A4 paper.
4. **Upload the PDF** by **27 Oct (Tue), 6:10pm** on LumiNUS Files 🡪 Student Submission 🡪 Test 2 Section III.
5. Failing to submit your answers to the correct LumiNUS folder by the stated deadline means you will get **0** marks for Section III. Late submissions are **NOT** allowed.
6. Only the most recent submission will be graded.
7. Address any questions you might have to the host of the Zoom session.
8. Concise, point-wise answers are accepted and preferred over long-paragraph answers.

|  |  |  |
| --- | --- | --- |
| Question | Max | Marks |
| 1 | 6 |  |
| 2 | 20 |  |
| 3 | 14 |  |
| Total | 40 |  |

# Component Design, API specification, Component Interactions

## Context and Requirements

The recent advance of deep learning and computer vision algorithms has lowered the barrier to entry for AI-based software systems. An enterprising jewellery chain has tasked your team to develop an application that captures and analyses audience traffic coming to its stores. The application is used by the store officers exclusively (i.e., user = store officer) and requires the following features:

1. Application should allow video recording via in-store cameras and perform face recognition on the recordings.
2. Application should allow users to create mappings between the face IDs generated by face recognition and the customer information in the database.
3. Application should 1) allow users to select whether they would like to be notified when an important customer (e.g., the ones who have made a lot of purchases in the past) re-visits a store and gets recognized by the application and 2) send the notifications accordingly. (Note: There could be multiple officers in the same store, each of whom is in charge of a different set of important customers.)
4. Application should allow users to view the profile and purchase history of a customer. In this way, the officers can offer better experience to a customer accordingly.

A possible architectural design of the application is as follows:

**User Interface**

**Front-end**

**User Manager**

**Camera Server**

**Database**

**Customer Manager**

* **User Interface**: This is the UI for the officers to access the system.
* **Front-end**: This is the main logic component for handling and routing requests from different components. Notably, this component helps the Camera Server to store information into the database. It also monitors the information in the database and issues notifications appropriately by consulting the User Manager.
* **User Manager**: This is the component for managing the user profiles (e.g., the name of the officer and the store in which the officer works), the user authentication information, as well as the information on which officer should receive notifications on which customer. It has direct access to the database for the information it manages.
* **Customer Manager**: This is the component for managing customer specific information such as the profile and purchase history of customer. It has direct access to database for the information it manages.
* **Camera Server**: This component serves as a portal for retrieving the frames of the recordings captured by the cameras. It also continuously runs face recognition locally and sends information (e.g., the face IDs, the store in which the camera is located, and the timestamp) to the database through the Front-End.
* **Database**: This component is a storage for the information mentioned above.

You may assume that the Camera Server uses a (perfectly accurate) face analysis toolkit. For every frame in the recoding, it returns a list of IDs for the faces that appear in that frame. The IDs are unique for different faces but consistent for the same face.

For details not specified in the write-up, you can make your own assumptions but do remember to state them clearly in your answers.

## Questions

1. **[6 marks]** Comment on the given design with respect to 2 design principles of your choice. For each design principle, state whether it is followed or not followed in the given design and justify your answer.

**Consider the following scenario for Questions 2 and 3:**

An important customer comes into a store and is detected by the application. The officer who works in that store and is in charge of that customer receives a notification about this customer’s visit. The officer logs in to the dashboard to view this customer’s profile and purchase history.

1. **[20 marks]** Design the abstract APIs related to the given scenario. For each API operation:
2. Write the header (i.e. the name of the operations, the parameters and the return type).
3. Write a brief description of what the operation does.
4. Annotate clearly which component the API is on.
5. [Optional] Create a label for the operation and use it in the sequence diagram in Question 3.

The quality of API matters. Try to make your operations and their descriptions as clear, unambiguous, simple and concise as you can.

1. **[14 marks]** Draw a sequence diagram to capture the sequence of interactions related to the given scenario. You may break your sequence diagram into multiple smaller ones for the ease of drawing.

The following pages are left blank intentionally. Please write your answer from this page onwards.

Question 1.

Question 2.

Question 3.