International Institute of Information Technology, Hyderabad
(Deemed to be University)

CS6.401 Software Engineering – Spring 2023

End Semester Examination

Max. Time: 3 Hr **Max. Marks: 80**

Instructions to the students

1. You are **allowed** to bring up to **2 sheets (A4 size, total of 4 pages including back-to-back)** of **handwritten notes**. You are not required to submit these notes along with your answer sheets.
2. Any form of printed/scanned materials, digital notes and photocopies are not allowed.
3. Borrowing notes from other students in the examination hall is prohibited.
4. Answers written with pencils won't be considered for evaluation
5. Please **read the descriptions** of the questions (scenarios) carefully. While answering please ensure that any assumptions made are clearly stated.
6. There are a total of twelve questions (some with sub questions). Please note that first eight questions are MCQs (requires **ONLY 1-2 lines of explanation)**, carries 2.5 marks each and you can select all that applies as answers for a given MCQ. Please refrain from writing long explanations for MCQ questions. Any explanations which is greater than 5 lines won't be considered for evaluations.

Good Luck

Congrats on being appointed as a consultant software architect at TukRide Private Limited!

"TukRide Private Limited" is a fictitious tuk-tuk ride services management company. India is famous for Tuk-Tuks (three-wheeler auto rickshaws) and it is also considered very people-friendly, especially in rural areas. Although today it's possible to get a tuk-tuk ride through established companies which provide four-wheeler taxi services, tuk-tuk drivers are very unhappy since the companies take a major share in the ride and the drivers are not getting many benefits. On the passenger side, the existing apps do not provide the flexibility of tuk-tuks which usually allows a passenger to have multiple stops between a source and destination.

The team from TukRide did an initial study on the feasibility of such a system to come up with a set of requirements. The high-level requirements are listed below:

- Users/passengers should be able to register and login to the application
- Users should be able to book rides by specifying their pickup and drop-off locations. Moreover, they should also be able to specify any intermediate stopping points and approximate waiting time at each point.
- The users should also be able to select the type of Tuk Tuk such as 3-seater one, 4-seater one or premier one (which will have facilities like magazines, some snacks, etc.).
- Once the users (passengers) request for a pickup, the system should be able to match the users with the available drivers based on the pickup, drop off locations and intermediatory drop points.
- The users should be able to able to see the details of the ride booked including driver details, vehicle details, etc.

- Drivers should be able to register to the platform with their vehicle. They should be able to update their availability and check their rider history, available balance and expected number of rides.
- User should be able to pay for the trip using multiple modes of payment including but not limited to: UPI, net banking, credit/debit cards, payment wallets, etc. The system should make use of some intelligence to provide pricing to different users based on their pickup locations, expected pickups, traffic, etc.
- Users should be able to rate the drivers and vice versa and further provide feedback to the system.
- For every ride matched, the price is dynamically calculated (surge pricing) during peak hours by considering the demand in each location, expected time of the trip, distance, traffic conditions, etc.
- The system should provide security ensuring that only qualified drivers are enrolled on the system and support mechanisms like real-time tracking of the passenger and driver to ensure that passenger safety is guaranteed
- Passengers should also be able to talk to support personnel in the event of any issues or concerns through SMS, Emails or even chatbots.

TukRide has engaged a team of approximately 20 software engineers to develop the system. These engineers can be distributed to different teams based on the development style. The team identified some initial subsystems and started developing some parts of the Driver Management System (DMS) which allows drivers to register with the platform, manage their payments, and further it allows the system to keep track of the real-time location of the available drivers, etc. Recognizing that poor design practices are a common issue among companies struggling with maintenance, the company has hired you as an expert consultant software architect to review their existing design and provide guidance to the teams regarding the design of the overall system in general as well as certain subsystems.

As the consultant, you have been allotted 180 minutes to complete several tasks. You must complete these tasks within the allocated timeframe, as your services have been billed accordingly. Each task is assigned a specific number of points, and the cumulative score of all the tasks will determine your final payment (based on total points), with a maximum of 80 points.

1. As a starting step, the team described the architecture of the existing driver management system (DMS). DMS has been designed and implemented as one web service using Java. It provides different features like enrolment of drivers/riders to the TukRide system, verification of drivers, keeping track of driver payments and location of drivers. The team describes that of late they are finding it difficult to debug, change parts of the codebase and release new features. Having just known this. What could be some potential issue(s) in your understanding? (2.5 points)

   a. Tight coupling between different objects
   b. DMS is built as a monolithic system
   c. DMS is built following a service oriented architectural style
   d. Every member of the team needs to know entire code base to make changes
   e. All of the above

   Describe the reason in 1-2 lines

2. Once you gathered some high-level knowledge about the DMS, you wanted to take the next step in understanding if there are some issues at the code level. You noticed that a function that takes care of driver verification has a signature of the form:

   verify_driver (String driverId, String driverName, String licenceId, String stateName,..)
   {
   //logic to verify the driver details
   }

   The verification logic is having a high cohesion. Despite that, you clearly noticed some code smells and you wanted to suggest some refactoring. What do you suggest to the teams (2.5 points):

   a. This code has a smell of the type "Long Method" and one of the ways to refactor is through "Extract Method"
   b. There are many parameters resulting in "Long parameter list" smell and one way to refactor is to "Preserve Whole Object"
   c. This code is classic example of "Feature Envy" and the refactoring is "Move Method"
   d. The parameters are primitives mostly and hence this suffers from "Primitive obsession" and one way to refactor is to "Use objects" instead of primitives.
   e. All of the above

   Describe the reason in 1-2 lines

3. As the next step, you went through some of the classes and their relationships, you noticed that there is a class Vehicle that acts as a parent class for different types of child classes - 3-seater, 4-seater and premier. Then for each of the child classes, you noticed a protected attribute, tukData of type TukDetails which itself is a class sharing a composite relationship with each of the child classes (3-seater, 4-seater and premier) and further contains all the details of the respective tuktukModel. You are certain that there is code smell(s) but you are not sure what smell(s) it is (2.5 points)

   a. This is an example of a hierarchy smell
   b. The fact that there is a protected attribute indicates an encapsulation smell
   c. There is an abstraction smell here related to the fact that unnecessary classes are created
   d. The smell here is due to broken modularization
   e. None of the above

   Describe the reason in 1-2 lines

4. In the code base of DMS, there is a DriverManager class which creates an instance of the driver object. In your opinion, does it violate some design principle? (2.5 points)

   a. Driver objects can be aggregated or composed to form DriverManager hence it does not violate the creator principle
   b. This violates Information expert principle
   c. This in principle creates high coupling between DriverManager and Driver
   d. This does not violate any design principle and it just shares a composite relationship
   e. None of the above

Describe the reason in 1-2 lines

5. As you completed an initial assessment of the DMS, you have been asked about the architecture of the overall system. As an architect, you plan to use the IEEE 42010 framework to identify stakeholders, concerns and corresponding views and viewpoints. You decided to use the 4+1 view model. Whom do you think the development view will help and what shall such a view provide (2.5 points).

    a. It will help the developers and it will describe the structural composition through component diagrams
    b. It will help the developers and it will describe the structural composition through class diagrams
    c. It will help the architects and it will describe the structural composition through deployment diagrams
    d. It will help the architects and it will describe the structural composition through sequence diagrams
    e. It will help the business users and it will describe the structural composition through use case diagrams

    Describe the reason in 1-2 lines

6. Now that the views and viewpoints have been decided, you informed the team that it's time for some design decisions. However, few of the team members were not clear with the notion of design decisions. What do you think best describes the design decisions? (2.5 points)

    a. Anything is an architectural design decision, even changing a function call
    b. Software architecture is nothing but a collection of significant design decisions
    c. Architectural design decisions represent those decisions that can have a significant impact on the system
    d. Architectural design decisions can be captured using lightweight Architectural decision records (ADR)
    e. All of the above

    Describe the reason in 1-2 lines.

7. The TukRide team wants to have an approval system built for driver enrolment. As a first step, the driver and vehicle details are collected and the approval happens by the TukRide support team, which is then sent to the DV process which verifies the documents of the driver and vehicle such as ID details, licence information, vehicle registration details etc. Further, this is then sent to a BG process which does a background verification to verify if there are some pending cases or open issues on the driver/vehicle. Finally, the TukRide Admin approves the driver, and the enrolment process is complete. The TukRide team wants to check with you about how to build such a workflow and keep it automated. What implementation mechanism do you suggest? (2.5 points)

    a. Build a monolithic subsystem for the approval and follow chain of responsibility pattern
    b. Use pipes and filter architectural pattern

c. Make use of a blackboard pattern with all independent process acting as knowledge source and black board keeps track of overall process
d. Use a layered architectural pattern where each approval can be one one layer
e. None of the above

Describe the reason in 1-2 lines

8. Being a ride service application, the company has an availability requirement of 99.999 % per year. What shall be the total down time of the system in a year? (2.5 points)

a. Approximately 1 day
b. Approximately 6 minutes
c. Approximately 10 minutes
d. Approximately 5 minutes
e. None the above

Describe the reason in 1-2 lines

9. The TukRide system plans to use data analytics to generate surge pricing and discounts. For surge pricing, the idea is to take in as input, the location, expected number of passengers, expected set of destinations, etc to generate an estimate. For discount, the idea is to use number of trips booked in the past as a parameter. Most of the team members are contemplating the use of blackboard pattern for implementing the data analytics subsystem. However, some members prefer to use the pipe and filter pattern. As an architect it's your responsibility to explain to the team which works better for the given scenario. (10 points)

a. Use appropriate diagrams to represent the possible architecture of the data analytics subsystem while using each of the patterns (5 points)
b. Which pattern do you prefer and why? Explain briefly considering different quality attributes and their trade-offs (5 points)

Note: A typical data analytics pipeline can consist of multiple steps starting from collection, processing, storage, analysis and visualization

10. The team wants you to check an important aspect related to the design of the ride booking subsystem. Since there may be 1000s of users using the application in parallel for requesting pickups, the team wants to go for an event-driven architecture to handle bookings from different users. However, they are not sure about the topology to be used for designing the ride-booking subsystem (15 points)

a. Identify the components or services that make up such a subsystem (4 points)
b. What are some of the quality requirements that you would consider in such a scenario (4 points)
c. Describe using diagram the topology to be used to arrange the components or services and why do you feel this is the topology to be used? Feel free to use box and arrow diagrams (7 points)

11. One of the central aspects of the TukRide system is real-time location tracking which keeps track of the real-time location of the passengers. Further, this information needs to

be sent to many other components (performing functionalities such as surge price prediction, trip time estimation calculator, ride matching, etc). The team decided to use Java for implementing this and they are thinking about using an observer pattern which can leverage the notion of subscription to send location information periodically to the objects of the corresponding components. In essence the idea is to treat the involved components as blocks in a larger monolith. The team is expecting to track at least 1000 users per second. Given this idea: (10 points)

a. what is your opinion on the scalability and performance of the system? Explain briefly in 3-5 lines (3 points)
b. Do you suggest some changes in terms of low-level and high-level design? Explain briefly with the help of diagrams (UML class diagram for low level and UML component or C4 component for high level) Focus only on the key classes/components. (7 points)

12. The company wants your support in coming up with an architecture for the Ride Matching and Ratings & Review subsystems. You can assume different interactions between the two subsystems as well as with other subsystems of the TukRide System. Since the DMS already exists as a monolithic system, either you can rearchitect that system or you can integrate the existing DMS with other parts. Can you come up with an architecture by going through the following process (25 points):

    a. List down the key functional and non-functional requirements (3 each) for the system (3 points)
    b. Identify the key stakeholders and concerns. At least 4 stakeholders (3 points)
    c. What are the tactics and patterns that you intend to use and why? Just brief explanation would suffice (3 points)
    d. Mention at least 2 key design decisions that you will consider for these subsystems. Make use of ADR for design decisions (5 points)
    e. Sketch the component and deployment diagrams providing a brief explanation of the overall architecture. Feel free to use UML or C4Model for the diagrams (7 points)
    f. What could be some design patterns that can be used for low level implementation? For each design pattern provide 1-2 lines explanation on how you think it could be used (4 points)

Now that you have completed the task, it's time to wait to know your final points!!!