International Institute of Information Technology, Hyderabad

(Deemed to be University)

CS6.401 Software Engineering – Spring 2024

End Semester Examination

Max. Time: 3 Hr

Max. Marks: 75

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 thirteen questions (some with sub questions). Please note that first ten questions are MCQs (requires **ONLY 1-2 lines of explanation**), carries 2.5 marks each. Each MCQ may require you to pick more than one answer. **Select all that applies.** 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 OneForAll Private Limited!

"OneForAll Private Limited" (OFA) is a fictitious innovative media aggregation platform that consolidates access to various OTT and audio streaming services. Today, we need to subscribe to different streaming providers to watch the content offered by different OTT platforms (e.g., Amazon Prime, Disney+, etc.). The same is the case with music streaming platforms like Spotify, Apple Music, etc. To this end, the goal of OFA is to provide a single subscription platform to view and listen to content from multiple sources, eliminating the need for separate subscriptions. The platform is intended to offer a unified, seamless entertainment experience while addressing specific user and administrative needs.

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

- Users should be able to register and sign in to the platform
- Users should be able to specify their preferences for movies, series, and music genres, and the system should support the curation of content from various streaming services accordingly.
- The users can subscribe to the platform to access the content through one of the subscription modes. The platform shall support multiple modes of subscription:
  - Premium Subscription: Offers unlimited access to all content on the platform for a monthly fee.

  - Base Subscription: Allows users to rent or purchase content individually. Prices are determined by the content providers. In this mode, each movie, TV series, or song will have a price that the users can pay either for rent or for purchase.
  - Free: Allow users to watch any free content that is provided by any of the OTT platforms.
- Users should have a payment wallet to add funds. Moreover, they should be able to make payments through credit cards, debit cards, UPI and net banking.
- Users will be able to rate the content by providing ratings and feedback. This will also be used for recommendation. Further, the user should be able to create their own watchlist and playlist.
- The OFA system should be able to handle different types of workloads (some users may be interested in live streaming sports/concerts, some movies/music, and some popular series, and each can be from different providers).
- The OFA system should support the processing of a continuous inflow of data without storing any sensitive user information, adhering strictly to privacy laws and regulations.
- Since this platform will be used by millions of users across the globe in different time zones, the platform should support high availability.
- The OFA system should provide recommendations to the user based on their preferences and usage history. The recommended content can be from multiple platforms.
- The system should support only up to a maximum of two devices per user to maintain security and manage bandwidth more effectively.
- The OFA system will provide support for an LLM-based chatbot, which users can leverage to learn about movies and reviews, summary of stories, details of songs, etc.
- Users should also be able to talk to support personnel in the event of any issues or concerns through SMS, emails or even chatbots.
- The system should provide easy means for admins/maintenance team to get access to the different analytics with respect to platform use like number of users, response time, server utilization, throughput, etc.

OFA has engaged a team of approximately 25 software engineers to develop the system. Further, the company plans to expand to about 100 software engineers in the next year. The engineers can be distributed to different teams based on the development style. The team identified some initial subsystems. The team has also started working on two of the subsystems: i) Content Management System (CMS) allows the OTT/music providers to upload content, describe the content, add subtitles or lyrics, add rent value, etc. Further, it also allows users to manage the content to watch or listen to and ii) Streaming and Delivery Management System (SDMS). The role of this subsystem is to stream content from various OTT/music platforms to the user devices by performing some quality optimizations. Every time the user prefers to watch a movie or a live stream or listen to some music from any provider, the SDMS is responsible for delivering the content to the user's device.

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 75 points.

1. As a starting step, the OFA team described the architecture of the Streaming and Delivery Management subsystem (SDMS). SDMS was developed as a Java web service that provides different functionalities. However, lately, the team has started noticing that the performance of the SDMS is not really up to the mark, as sometimes there are latency issues, sometimes some components are failing, the throughput is going down, etc. To this end, the team is deliberating on the use of a tactic: "ping/echo" to periodically check if there are some issues. What is your opinion on the same (2.5 points):

   a. This tactic would address all the major performance issues
   b. The team needs to add tactics like heartbeat to check availability of components.
   c. The team needs to use tactics like "Limit access" to components to improve performance
   d. Along with ping/echo the team needs to use tactics for Resource demand and resource management

   Describe the reason in 1-2 lines

2. As you were inspecting the code of the SDMS, you noticed that there is one function that is responsible for delivering the streaming content by identifying the corresponding OTT platform and then invoking the API of the respecting platform. The high level code looks like the following:

```
public void generate_stream (String ottPlatform, String ContentName, String contentId)
{
        // Logic to perform some preprocessing
        if (ottPlatform.equals("p1"):
        {
            //invoke the API of p1, perform the quality optimization and return the stream
        }
        else if (ottPlatform.equals("p2"):
        {
            //invoke the API of p2, perform the quality optimization and return the stream
        }
        // conditions for rest of the OTT platforms

}
```

   The function is highly cohesive. 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. The parameters are primitives mostly and hence this suffers from "Primitive obsession" and one way to refactor is to "Use objects" instead of primitives.
   b. There are many parameters resulting in "Long parameter list" smell and one way to refactor is to "Preserve Whole Object"
   c. This code has a smell of the type "Long Method" and one of the ways to refactor is through "Extract Method"

d. The function also is in a God class and the class may be suffering from a design smell related to "Broken Modularization"

e. All of the above

Describe the reason in 1-2 lines

3. The CMS development team has provided you with the class diagram of the system. As you were going through the class diagram, you notice that there are two disconnected classes, *Content and Subtitle* but you also notice that they share some relationship. What relation do you think needs to be established? (2.5 points).

   a. Generalization
   b. Association
   c. Aggregation
   d. Composition
   e. None of the above

   Describe the reason in 1-2 lines

4. The CMS allows content to be added by OTT platforms based on the language and genre. CMS provides an API that OTT platforms use to connect their content. CMS internally has different classes for each genre and language. However, each time they need to add the content, the OTT providers have to create objects of genre, language, and content. This has resulted in a lot of conditional statements on the client side where there is a condition for creating the content object based on genre and language. The team wanted to check with you if there is a better way of handling it (2.5 points)

   a. Use factory design pattern
   b. Use a combination of strategy and adapter pattern
   c. Use strategy pattern to create based on conditions
   d. Use builder pattern to address the issue
   e. None of the above

   Describe the reason in 1-2 lines

5. As you were moving around, you noticed that the team working on the architecture of the overall system is requesting your support. The team follows the IEEE 42010 framework to come up with an architecture. They require your support in identifying the different diagrams that can be used to frame the concerns of stakeholders, which mainly include: i) C1: cost of the overall system; ii) C2: overall system performance and iii) C3: the high-level interactions between subsystems; In your opinion what are some of the diagrams (views) that could capture these concerns (2.5 points)

   a. A use case diagram or the context diagram of c4 model can address all concerns
   b. C1 and C3 can be captured through class diagrams and C2 through component diagrams
   c. C1 requires deployment diagram whereas C2 and C3 can be captured using sequence, component and use case diagrams.

d. C1 can be captured using sequence diagram, C2 through sequence and C3 using use case diagrams

e. C3 can be captured using component diagrams whereas C1 and C2 through deployment and use case diagrams respectively

Describe the reason in 1-2 lines

6. Now that the views and viewpoints have been decided, it's time for some design decisions. The organizational owners of the OFA system are very much concerned about security as well as performance but at the same time wants to be mindful of the cost. One important challenge that you have been asked to address is to come up with a solution for handling the storage of the different content that is being streamed from various OTT platforms. In your opinion what is the optimal way to handle this? (2.5 points)

    a. Cache all the content being streamed and store it locally in OFA on-prem servers
    b. Store all the content in a cloud owned by third party provider
    c. Only cache the most streamed content and store it in OFA on-prem server
    d. Only cache the most streamed content and store in cloud owned by third party provider
    e. Don't store any content, keep fetching it from APIs of OTT and music sharing platforms as and when needed

    Describe the reason in 1-2 lines.

7. The OFA team wants to have a recommendation system built for generating content recommendations. The recommendations can be songs, movies, series, etc, based on the user activity. The team plans to take in data from different sources, including overall user ratings, overall user feedback, individual user interests, and external sources like IMDB. The OFA team wants to check with you about how to build such a workflow and keep it automated. Which of the key architectural patterns do you suggest? (Please note: there may be multiple patterns required; you need to select only the most important one as per your opinion) (2.5 points)

    a. Use pipes and filter architectural pattern
    b. Make use of a blackboard pattern with all independent process acting as knowledge source and black board keeps track of overall process
    c. Leverage microservices pattern
    d. Use a layered architectural pattern
    e. None of the above

    Describe the reason in 1-2 lines

8. The OFA team wants to ensure high availability for paid customers, especially premium customers. The team is also fine with Free users being offered less availability. To this end, the team is planning to come up with an availability percentage to be added to their service level agreement. The team is fine with serving premium customers with a downtime of a maximum of 6 minutes in a year,(whereas for free, even a downtime of up to 9 days is acceptable.)What will be your recommendation? Also, take into consideration that offering high availability may have an impact on the cost (2.5 points):

a. Five nines for premium and three nines for free users
b. Six nines for premium and five nine for free users
c. Four nines for premium and five nines for free users
d. Five nines for premium and two nines for free users
e. Five nines for premium and five nines for free users

Describe the reason in 1-2 lines

9. In order to develop the architecture of the chat system which helps users, the team is trying to identify one or more Architecturally Significant Requirement (s) (ASR) that needs to be considered. Which of the following do you suggest the team to consider? (2.5 points)

   a. The chat application should allow users to track their conversation history
   b. The chat application should only allow authenticated users to access the chat interface
   c. The chat application should support concurrent users
   d. The chat application should provide a responsive UI
   e. The chat application should support streaming responses to users (like chatGPT)

   Describe the reason in 1-2 lines

10. Being an integrated streaming service provider, the company is expecting a large user base. At present the company is deciding to use a Service oriented approach where some of the subsystems like Content Management System, Payment Management Systems are built as separate monolithic services and in future it can be migrated when necessary. What are your thoughts on the organizational decision? (2.5 points)

   a. This will result in a lot of technical debt
   b. This is a correct decision given the current size of the organization
   c. The company should have selected a microservice approach
   d. The company can think of this as a technical credit and handle later
   e. All of the above

   Describe the reason in 1-2 lines

11. The Subscription and payment system team is contemplating the use of a combination of publish-subscribe and microservices for design and implementation. For instance, users can choose their subscription, and the choice made will be sent as a message to a topic of a message broker. This will be subscribed to by the subscription microservice, which will then trigger the payment microservice to handle the payment through another message over a topic and so on. In this manner, each microservice involved in this system will be decoupled from each other and will communicate over message queues (e.g., message brokers like Kafka or RabbitMQ serve as integration points). However, some members prefer to create subscription and payment subsystems as a monolithic system following a layered pattern. As an architect, it's your responsibility to explain to the team which works better for the given scenario (15 points)

   a. Use appropriate diagrams to represent the possible architecture of the subscription and payment subsystem while using each of the patterns. Feel free to use UML or C4Model for the diagrams (5 points)

b. Which pattern do you prefer and why? Explain briefly considering different quality attributes and their trade-offs (5 points)

c. Considering the architectural pattern selected above, what are some of the design patterns that you intend to use to realize such a system? (5 points)

12. As the OFA platform is expected to expand its user base globally, there is a significant need to enhance the performance, availability, and security of its streaming and content delivery subsystem (SDMS). The subsystem is crucial for maintaining high-quality user experiences during peak demand times, such as during new releases or live events. The technical team is considering various architectural tactics to address these challenges, and your expertise is required to formulate a strategic approach. Can you help the team by answering the following questions? (10 points)

a. What are the key challenges (at least two per quality) that the SDMS might face in terms of each of the quality attributes: i) performance; ii) availability; and iii) security given that SDMS is a monolithic system (3 points)

b. Briefly describe the architectural tactics that you would recommend for addressing the above mentioned challenges (at least one per quality attribute). Create a high level diagram (box and arrow) of how the tactics will be used in the system design. (7 points)

13. The OFA team is looking forward to your support in coming up with an architecture for the Content Management (CMS) and User Notification subsystems (UNS). The goal of the CMS is to provide support for functionalities that allow OTT providers to add their content by providing various metadata information. Further, CMS allows OTT providers to add live-streaming content. On the other hand, the role of UNS is to provide real-time notifications to the users on the arrival of new content, completion of payment, pending payments, availability of any live stream matches, etc. (just like any notification system). You can assume different interactions between CMS and UNS as well as with other subsystems of the OFA System. Also, take into consideration that CMS has already been developed as a monolith (you may choose to rearchitect if you feel it is necessary). 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 Architecture Decision Records (ADRs) for documenting design decisions. You can use any standard template (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!!!