FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGIES

FALL 2023

FINAL EXAMINATION

COURSE TITLE:	Intro To Mobile App
COURSE CODE:	
INSTRUCTOR:	Serh Jeff
DATE:	
DURATION:	3 weeks

INSTRUCTIONS:

- ✓ ***Credit is given for legibility, clarity of expressions and use of relevant illustrations.***
- ✓ ***Clearly write your registration number on each answer sheet used***
- ✓ ***Answer all Questions without missing a step***

SECTION A: COMPULSORY QUESTIONS

The questions require a deep understanding of mobile app development concepts and problem-solving skills. Keep in mind that this is just a sample, and you can modify it according to your needs:

Question 1:
You are tasked with developing a ride-sharing mobile app that allows users to request rides, track drivers in real-time, and handle payments. Describe the key architectural components and design considerations you would take into account when building this ride-sharing app.

Question 2:
A user of your ride-sharing app reports that they are experiencing difficulties in finding available drivers during peak hours. Outline the steps you would take to optimize the driver matching algorithm and improve the user experience in such situations.

Question 3:
You are developing a ride-sharing app that requires secure and seamless payment transactions. Explain the payment processing flow and the security measures you would implement to protect user payment information.

Question 4:
Imagine you are building a ride-sharing app that operates in areas with limited or intermittent internet connectivity. Discuss the strategies you would employ to ensure the app functions offline, allowing users to request rides and track their progress without relying on a constant internet connection.

Question 5:
Your ride-sharing app needs to handle surge pricing during high-demand periods. Explain the logic and algorithms you would use to determine surge pricing rates and communicate them to users effectively.

Question 6:
A user reports that they are experiencing slow performance and delays when requesting a ride in your app. Describe the steps you would take to identify and address the performance issues, considering factors such as network latency and server response times.

Question 7:
You are developing a ride-sharing app that operates in multiple cities and countries. Explain the strategies and considerations you would follow to implement localization and provide a seamless user experience across different languages, currencies, and cultural norms.

Question 8:
Discuss the importance of user feedback and ratings in a ride-sharing app. Explain how you would incorporate user feedback into app improvements and maintain driver and passenger ratings to ensure a reliable and trustworthy service.

Question 9:
Your ride-sharing app needs to integrate with external mapping and navigation services to provide accurate directions and estimated arrival times. Explain the process of integrating these services into your app and the considerations for ensuring smooth and reliable navigation.

Question 10:
Describe the process of testing and deploying a ride-sharing app to ensure its quality and performance. Discuss the different types of testing you would conduct, such as functional testing, performance testing, and user acceptance testing, and the tools you would use to automate the testing process.

Here are the steps to answer the ride-sharing app scenario within a two-week exam period:

Week 1:

Requirements Gathering and Analysis:

Review the given specification document for the ride-sharing app.

Identify the main features, functionalities, and non-functional requirements.

Analyze the user registration, ride request, driver management, real-time tracking, payment processing, and review/rating system.

Design and Planning:

Design the overall architecture of the ride-sharing app, including the client-side and server-side components.

Plan the database schema to store user data, ride requests, driver information, payments, and ratings.

Define the APIs and data structures needed for communication between the client and server.

Implementation:

Develop the user registration and authentication features, including account creation and secure login.

Implement the ride request functionality, allowing users to enter pickup and drop-off locations, estimate fares, and display nearby available drivers.

Develop the driver management system, enabling drivers to register, accept or reject ride requests, and update their availability.

Integrate real-time tracking using mapping and location APIs to track the driver's location and display it to the passenger.

Implement the payment processing flow, integrating secure payment gateways and supporting multiple payment methods.

Develop the review and rating system, allowing users to rate drivers and provide feedback after completing a ride.

Week 2:

Testing and Debugging:

Perform unit testing of each implemented feature, ensuring their correctness and proper functionality.

Conduct integration testing to verify that different components of the app work together seamlessly.

Test the app's performance, security measures, and offline functionality.

Debug and fix any issues or bugs identified during testing.

Documentation:

Document the architecture, design decisions, and implementation details of the ride-sharing app.

Create user guides or instructions on how to use the app's features and functionalities.

Provide clear documentation on the APIs and data structures used within the app.

User Acceptance Testing:

Present the implemented ride-sharing app to users (instructors or fellow students) for testing and feedback.

Collect feedback on the app's usability, functionality, performance, and any additional requirements.

Address any necessary changes or enhancements based on user feedback.

Finalize and Submit:

Make any final adjustments or improvements based on user feedback.

Ensure the code adheres to coding standards and is well-documented.

Package and submit the completed ride-sharing app, along with the source code and documentation.

Remember, the two-week timeline is a constraint for the exam scenario, and you may need to prioritize certain features or make adjustments based on the available time. It's essential to focus on understanding the requirements, designing a scalable architecture, implementing core functionalities, and ensuring thorough testing and documentation within the given timeframe.

ALL THE BEST