Course Project: CarPool Application (15%)

Introduction

Carpool is a rideshare application designed to facilitate rideshare service to customers. carpool can be used by students to share rides to or from AinShams campus. In this version of the application, we'll create a customized version for Faculty of Engineering Community at Ainshams University. We'll focus on ride to or from Abdu-Basha and Abbaseya square. Users must sign in with an active account @eng.asu.edu.eg to build a trusted closed community. Carpool has a revolutionary strategy in recruiting drivers and operating the service. Carpool is operated by students for students. To regulate the service in this pilot project, there will be only two destination points Gate 3 and 4 and one start ride time at 7:30 am from different locations and one return ride time at 5:30 pm from faculty of engineering campus. Customers who need a ride at 7:30 must reserve their seat before 10:00 pm previous day. Customers who need a ride from campus at 5:30 pm must reserve their seat before 1:00 pm same day.
You are free to design this application using Native Android app development (Java or Kotlin on Android Studio) or using cross-platform Flutter-based implementation.

App Description/Requirements:

In this project, you will develop a rideshare App. The minimum app requirements are:

- login page with sign up option using firebase authenticate. There should be a test account credentials with login information submitted with the app for testing
- List of available routes to and from ainshams campus. The list should be designed using recycler view.
- A cart page to review the order and make payments.
- Order History with tracking/status page
- Use Firebase real-time database for route and order status.
- Use Room database for profile data (or sqlite if you choose flutter)
- Payment and order tracking page.
- Web application for drivers to confirm orders and update status data. Order must be confirmed before 11:30 pm for morning ride and before 4:30 pm for afternoon ride.

A full list of the requirements is outlined in the marking rubric table below.

Project Submission Instructions

Submit two incremental versions of the project PDF reports to LMS.

a. The first milestone would be the full layout of the app design (XML or DART).
You should copy the XML or DART code in a Word file, add screenshots of the layout of all pages, and list the components used in the layout. Following that, convert the file into PDF and submit it on LMS.

b. The second milestone is the basic function of the application of sign-up/login page, available routes. The Java/Kotlin/Dart code should be copied in a Word file, add screenshots of the layout, and list the testcase scenarios. Following that, convert the file into PDF and submit it on LMS.

c. Final Delivery A full functioning application with all requirements as listed in the rubric. The Java/Kotlin/Dart code should be copied in a Word file, add screenshots of the layout, and list the testcase
scenarios. Following that, convert the file into PDF and submit it on LMS.

File submission

- Upload the final project report as a PDF along with the project workspace to LMS.
- Create a 2-minute video demo of the application and post it on the course channel
- Demonstrate your app to your lab/course instructor

Marking Rubric

Your mark will be determined by the level of functionality you achieve for your app

Component	Marks	Description
Milestone 1	25	All bare basic requirements of the first milestone. A full layout of the app design in XML/Dart must be submitted on time.
Milestone 2	25	All requirements of the second milestone, including fully functioning sign-up/login page, available routes.
User Interface quality	50	A fully functioning application. login page with sign up option using firebase - authenticate. There should be a test account credentials with login information submitted with the app for testing List of available routes that contains at least 10 routes. - .The list should be designed using recycler view .A cart page to review the order and make payments - Order History with tracking/status page - Use Firebase real-time database for routes and order - .status Use Room database or SQlite for profile data - .Payment and order status page - Web application for restaurants to confirm orders and - update status data.
TOTAL	**100**	