Riding towards efficiency: Designing a comprehensive end-to-end database application for cab booking

Students Atharv Goel, Ashutosh Gera
Group 10
Instructor Prof. Vikram Goyal

Project Scope

Cab booking software is a tool that facilitates booking a cab ride for customers and managing a cab fleet for cab companies. It provides convenience for customers by eliminating the need to flag down a cab on the street or call a taxi company, allowing them to book a ride quickly. It also enables cab companies to manage their fleets more efficiently by providing real-time tracking of cabs, improving pick-up times and optimizing the use of resources. Additionally, it increases safety by allowing safe and traceable rides and enables customers to rate the driver and provide feedback. It also provides improved customer service by providing features such as fare calculation, push notifications and the ability to view past bookings.

The data we plan on storing would include:

- Data storage for customer information including:
  - Personal details such as name, contact information, and email address
  - Payment information such as credit card details and billing address
  - Trip history and preferences
- Data storage for driver information including:
  - Personal details such as name, contact information, and email address
  - Driver's license and vehicle registration information
  - Trip history and ratings
- Data storage for trip information including:
  - Pick-up and drop-off location
  - Trip distance and duration
  - Fare and payment details
  - Rating and feedback from customers

Functional Requirements

The client side of our cab booking app called **FloRide** (like Uber) will consist of two interactive web applications – one for demand (passengers) and one for supply (drivers), each with its own set of features.
There's also a need for the web-based admin dashboard, which will serve the role of a centralized backend office, to monitor the entire system and perform administrative actions.

For the **passenger client**, the user flow and functionalities are as follows:

- **User registration:** Users will first need to register on the platform by providing their personal information and creating a unique account.
- **Login:** Once the registration is complete, the user will be able to log in to their account using their email and password.

- **Booking a cab:** After logging in, the user will be able to book a cab by providing their pickup and drop-off location, and selecting the type of cab they would like to book. They will also be able to view the available cabs nearby.
- **Fare calculation:** After the user has provided their pickup and drop-off location, the system will calculate the fare based on the distance between the two points and the type of cab selected.
- **Payment gateway integration:** Once the fare has been calculated, the user will be prompted to make the payment through the integrated payment gateway.
- **Booking confirmation:** After the payment has been made, the user will receive a booking confirmation, including the details of the driver, vehicle, and estimated time of arrival.
- **Tracking the cab:** The user will be able to track the cab's location in real-time using the Google Maps API.
- **Rating and Review:** After the ride is completed, the user will be prompted to rate and review the driver.
- **Viewing past bookings:** The user will be able to view their past bookings on their profile page.
- **Canceling a booking:** The user will be able to cancel a booking before the cab arrives at the pickup location.

For the **driver client**, the user flow and functionalities are as follows:

- **Driver registration:** Drivers will first need to register on the platform by providing their personal information, vehicle information and creating a unique account.
- **Login:** Once the registration is complete, the driver will be able to log in to their account using their email and password.
- **Accepting bookings:** After logging in, the driver will be able to view the available bookings and accept them.
- **Tracking the booking:** The driver will be able to track the booking details, including pickup and drop-off location and fare.
- **Starting the ride:** Once the driver reaches the pickup location, they will start the ride by clicking a button on the app.
- **Tracking the ride:** The driver will be able to track the ride details, including distance travelled, time taken, and fare earned.
- **Completed the ride:** Once the driver reaches the drop-off location, they will complete the ride by clicking a button on the app.
- **Rating and Review:** After the ride is completed, the driver will be able to view the rating and review given by the user.
- **Viewing past bookings:** The driver will be able to view their past bookings on their profile page.

Admin Panel: An administrator will be able to access the admin panel to manage the system, including managing user accounts, monitoring and managing bookings, managing driver accounts, viewing statistics and reports, and making changes to the system's settings.

Technical Requirements

- For the front-end, we plan on using **HTML, CSS, and JavaScript** languages with the **React** library.
- For the back-end, we plan on using the **FastAPI** web framework for developing RESTful APIs in **Python**.
- We will use the **MySQL/MariaDB** DBMS to store and manage the data.