

Project Requirement Document

NIKHIL (2022321)

January 24, 2023

1 Project Scope

Our project implements the various essential backend functionalities of a cab booking system, while at the same time providing a desirable and easy user experience making it a fully functioning application. These functionalities will include ride booking, driver and vehicle management, payment handling, integration of maps and various alerts and notification generation as and when required while ensuring that the backend architecture is able to handle requests from multiple users at the same time.

2 Technical Requirements

2.1 MySQL (Relational Database Management System)

MySQL is a free and open-source relational database management system. We plan on using it for managing all the data storage and retrieval needs such as bookings & payment management, vehicle & dispatch management etc.

2.2 Django (Python-based Backend Framework)

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. We plan on using it to develop supporting functionalities in our project such as user profile management. Django facilitates easy interfacing with MySQL and provides secure maintenance of websites without extra effort.

2.3 React (JavaScript-based Frontend Library)

React is a free and open-source JavaScript library for building user interfaces based on UI components. We plan on using it to design the user interface of the cab booking system which is easy to use and provides convenient access to all the functionalities we plan to implement.

3 Functional Requirements

3.1 Bookings Management

Functionalities to book a ride, view current booking details and history, along with options to cancel the ride or view driver details.

3.2 Payment Management

Dummy payment processing functionality. Options for online payment (only upon completion of which the booking will be confirmed) and cash (once the trip has been completed)

3.3 Driver & Vehicle Management

Functionalities to assign the nearest available driver to the user, and options to view driver/vehicle details and track their location.

3.4 Dispatch Management

Functionalities to estimate trip times and compute the shortest path between two points using traditional shortest path algorithms like Dijkstra.

3.5 Scalability & Performance

Design the back-end in a way that's easy and efficient to scale to higher volumes of requests in the future.

3.6 Integration with Maps

Functionalities to integrate geolocation services to plot the location of the driver and passenger, along with a highlighted path showing the route.

3.7 Push Notifications

Simple alerts to notify the user about key events such as "Booking confirmed", "Payment successful", "Driver arrived" etc.

3.8 Security by design

Functionalities to ensure the security of sensitive data such as user credentials and other personal information.