

# uRide

Agamdeep Singh 2021306, Shobhit Pandey 2021287

## **Project Scope:**

Our project aims at making a cab booking system which

1. Allows intracity travel to customer
2. Allows for ride sharing in intracity travel
3. Allows intercity travel to customer
4. Implements the checking of available drivers in real time [simulation]

Our project aims to develop a cab booking system that allows customers to book intercity and intracity travel, as well as the option for ride sharing. The system will be able to check the availability of drivers in real-time [simulation], ensuring that customers can quickly and easily book a ride allowing handling of multiple users (and drivers) at the same time. It would be handling ride booking and payment details and also incorporating maps for ease of use. The system will be user friendly, adaptable and scalable (could be used in a variety of regions). Additionally, the system will be designed to be secure and reliable, ensuring that customer data is protected at all times.

## **Tech Stack :**

### **Relational Database Management System : [MySQL]**

Creating different tables for customer, drivers, their distance preferences, their ratings and implementing real-time updation to ensure ride availability and actively taking feedback from the users to maintain a rating database that allows for better rated drivers and users to get preference over lower rated users. Also maintaining fields to keep track of drivers currently in a ride so that they can efficiently be assigned their next ride.

### **Backend Development : [Flask, Python]**

Flask will facilitate us to interface the database with the front-end and ensure the smooth functioning of the service. Flask is a micro web framework written in Python which doesn't have a lot of boilerplate code and makes it easier to maintain web-apps.

### **Frontend : [HTML,JS,CSS]**

Create an interactive interface which lets the customer order rides with ease. This will update based on the completion of rides, user ride requests and driver availability and also give

provisions to provide feedback which in turn would be used in the backend for availability of rides.

## **Technical Requirements:**

### **Unique profile identifiers(primary keys):**

These would ideally be the user's phone number or email address and will be unique for each user

### **Names:**

These could be the same for various users.

### **Profile Rating:**

These would be accessible through the unique profile identifiers and would be assigned to each profile

### **Wallet information:**

Each user would have a wallet attached to their profile.

## **Functional Requirements:**

### **Ride booking:**

Functionality which allows booking a cab which will assign a driver to the user based on the driver's availability, rating, distance preference and other factors.

### **Map Integration:**

Users will be able to search for the location they want to go to and set their destination based on the results or manually choose it on a map. This can then be used along with the information about the drivers and the users to get a good driver-rider match.

### **Service Distribution:**

It lets the user choose between intercity and intracity travel. The difference lies in the driver availability and the number of passengers a ride can have in intracity and intercity travels respectively.

### **Driver preference:**

Our system targets the most efficient distribution on the basis of the distance that the driver prefers to travel. It aims at allotting the driver with the closest preferred distance to a customer who wants to travel the actual given distance from one location to another.

**Feedback System:**

The feedback from users is stored and used by the system to provide them with the driver of the best possible rating available ensuring a positive user experience.

**Basic Framework:****Efficiency:**

Data will be organised with preference to required parameters so as to take minimum time to allot a customer a cab upon booking.

**Similar to real-time:**

So as to show how the database is updated in real-time there will be hardcoded delays which can help with simulating the database and backend.

**Intracity[many to one and one to one]:**

The cabs and the customers in intracity could be many to one depending on the people who want to go from one location to another. This is an additional feature, users can have their personal rides as usual.

**Intracity[one to one]:**

Depending on the distance that the driver has to go and whether a driver is available or not the customer will be allotted a vehicle to go from one city to another.

**Cardinality of set of entities:**

Below is the list of entities that will be present in the database:

1. Cities
2. Customers
3. Drivers
4. User ratings
5. Account Balances

*(we will be adding on to this list as we progress through this project)*