

# uniRide

*A Ride-sharing Application*

Aya Benjelloun & Nour El Houda El Iamani

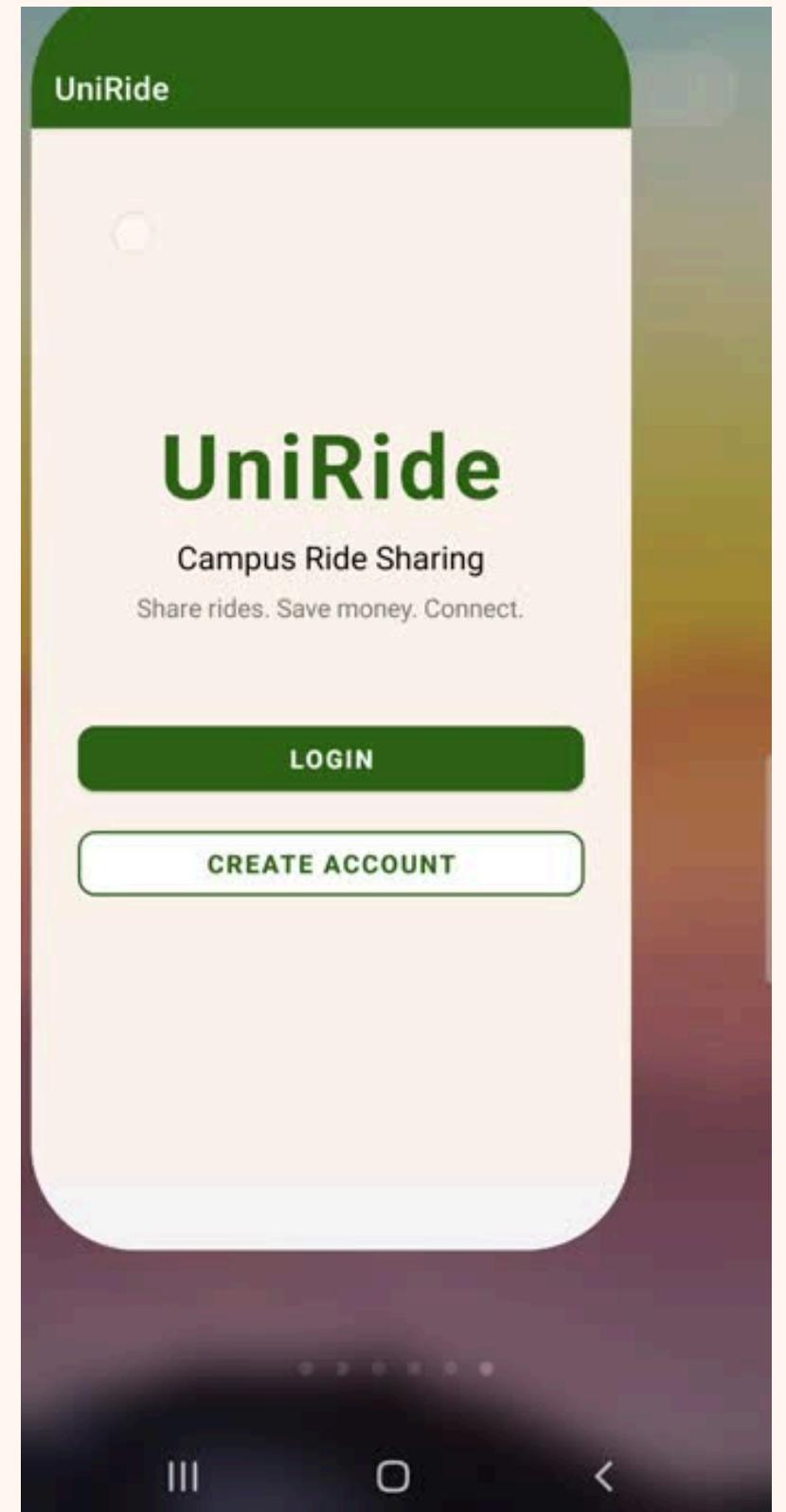


# What is UniRide ?

- Mobile ride-sharing app for university communities
- Connects drivers and riders within campus ecosystem
- Safe, verified, and cost-effective transportation
- Real-time booking

Target Users:

✓ Students ✓ Faculty ✓ Staff



# Core Features

## Driver

- Create ride offers with flexible scheduling
- Set available seats and pricing
- Track earnings in real-time
- Manage ride status

## Rider

- Browse available rides
- Book seats instantly
- Track ride progress live
- Rate and review drivers

# **Advanced Capabilities**

- **Dual Ride Types (Inside ,Outside Campus)**
- **Dynamic Fare Calculation**
- **Driver verification system**
- **Admin oversight and control**

# Feasibility Study

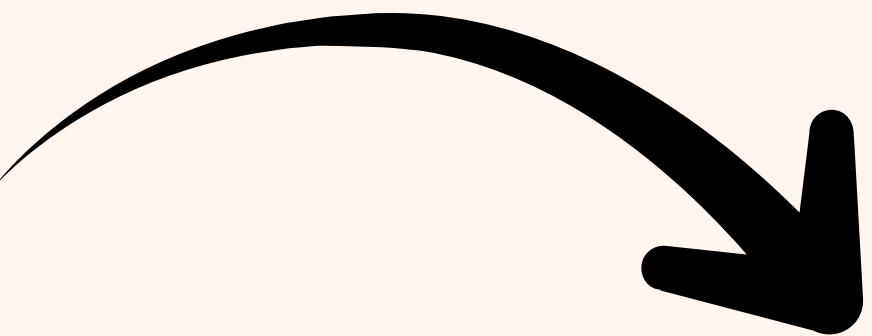
## Market Research

- Survey of 80+ university students and faculty
- Analysis of existing transportation solutions
- Gap identification in current systems

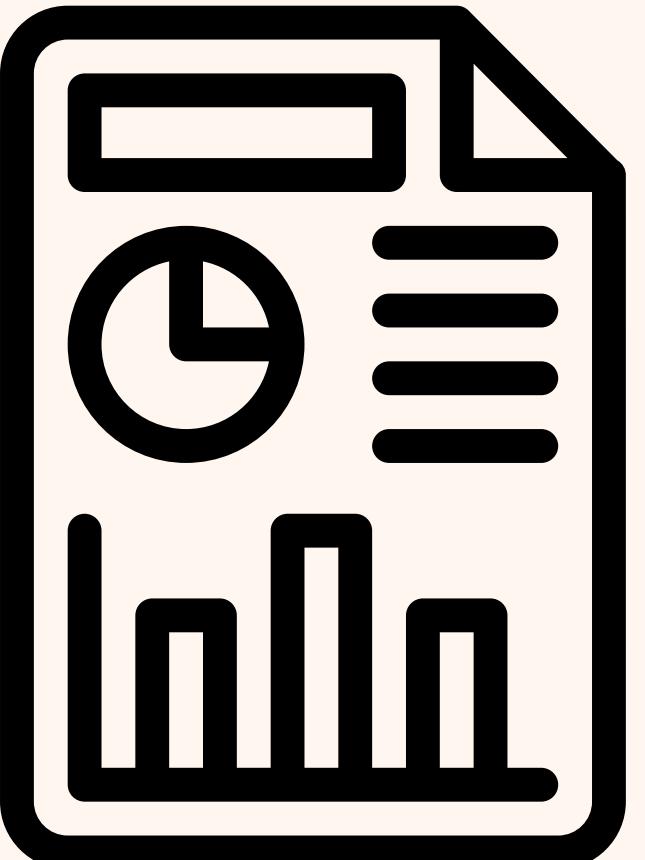
## Competitive Analysis

- Study of existing ride-sharing platforms
- Identification of university-specific requirements

## Technical Assessment



## Feasibility report



---

# Result Of Feasibility Study

## Cost Concerns

- 76% find current transport options too expensive

## Underutilized Capacity

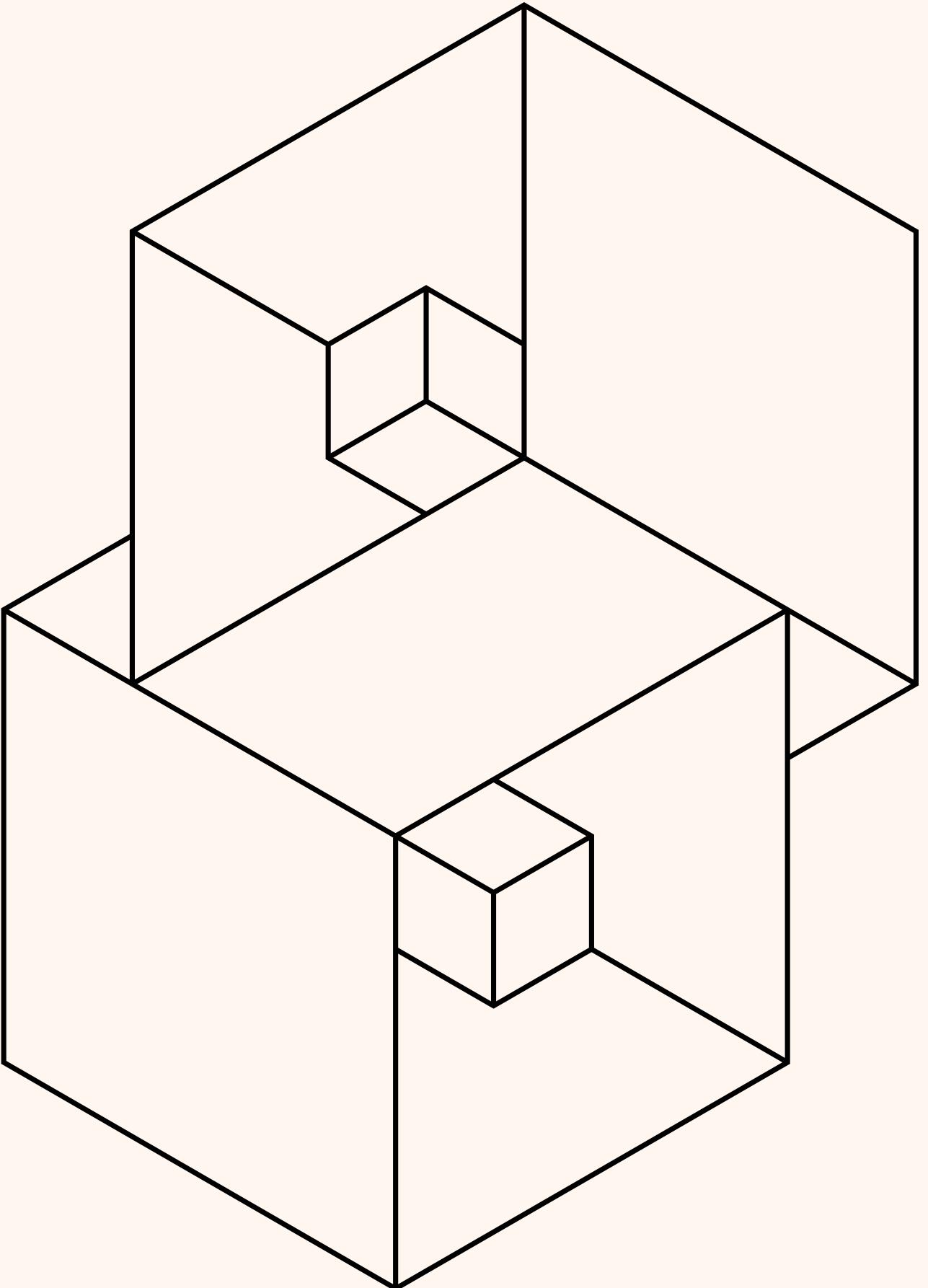
- 64% of student drivers travel alone
- 99% Willing to share rides for extra income

## Scheduling Issues

- 68% report difficulty finding rides at odd hours
- Campus shuttles have limited schedules

## Safety Priority

- Trust within campus community is high



# Technologies Used

## Frontend Layer:

- Android SDK (Java , xml layout)

## Backend Layer:

- Firebase
- Firestore database
- Firebase Authentication - User management

## Maps & Messaging:

- Google Maps API,Google Places API,Distance Matrix API
- WhatsApp Integration

## Development Tools:

- Android Studio IDE
- Git & GitHub

# SRS Document

01

## List of App Requirements:

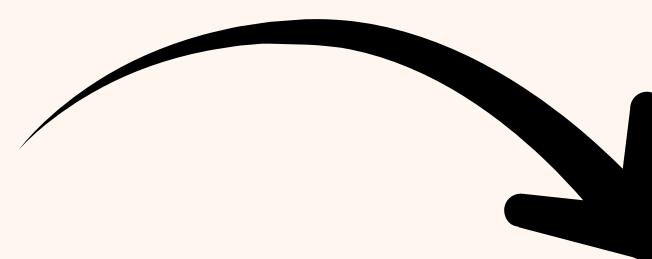
After :

- Requirement Elicitation
- Analysis & Specification
- Implementation & Validation

02

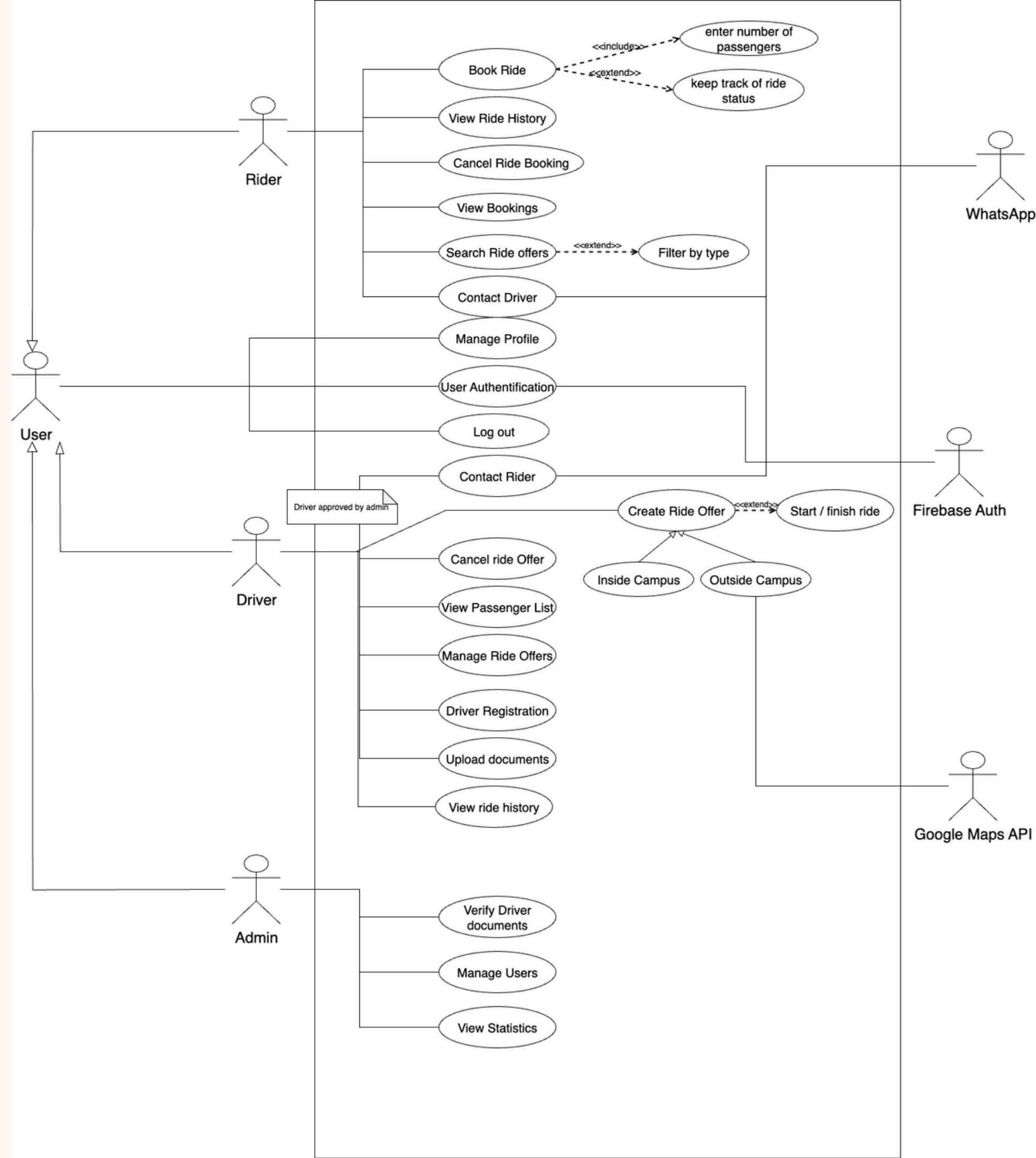
## UseCase Diagram:

- Rider
- Driver
- Admin
- Firebase Backend
- Google Maps API
- WhatsApp

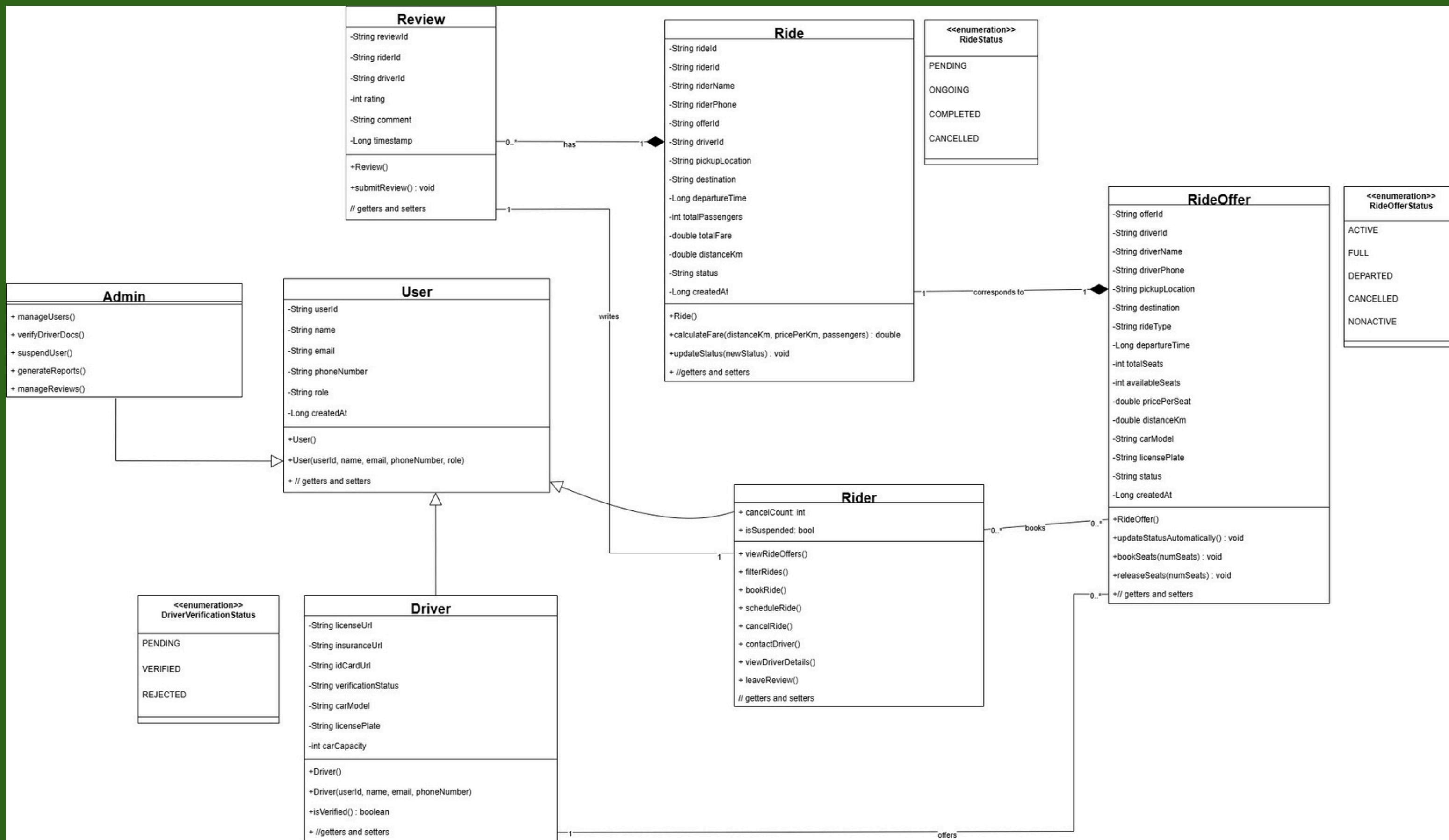
 SRS



# Use Case Diagram



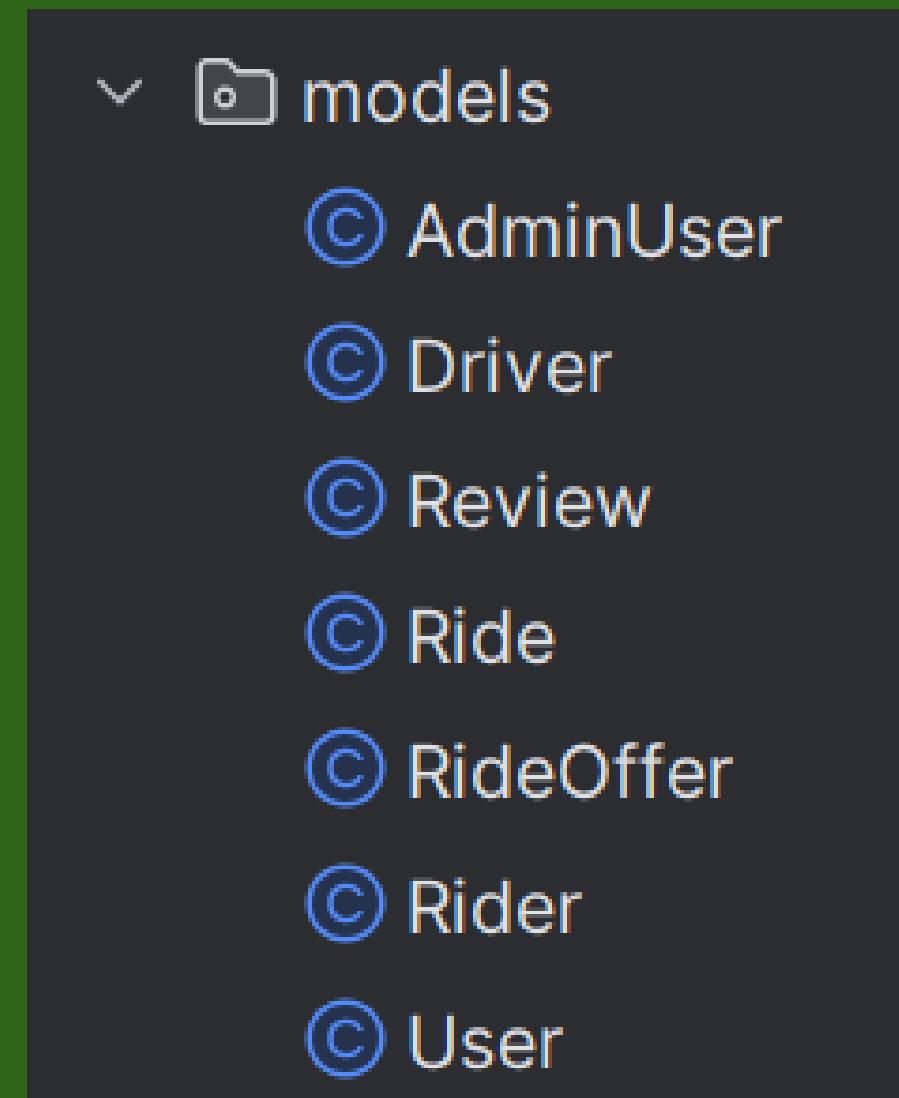
# Class Diagram



# Models

## Core Classes:

- User - Base class
- Driver - Extends User
- Rider - Extends User
- AdminUser - Extends User
- RideOffer - Available rides
- Ride - Booked rides
- Review



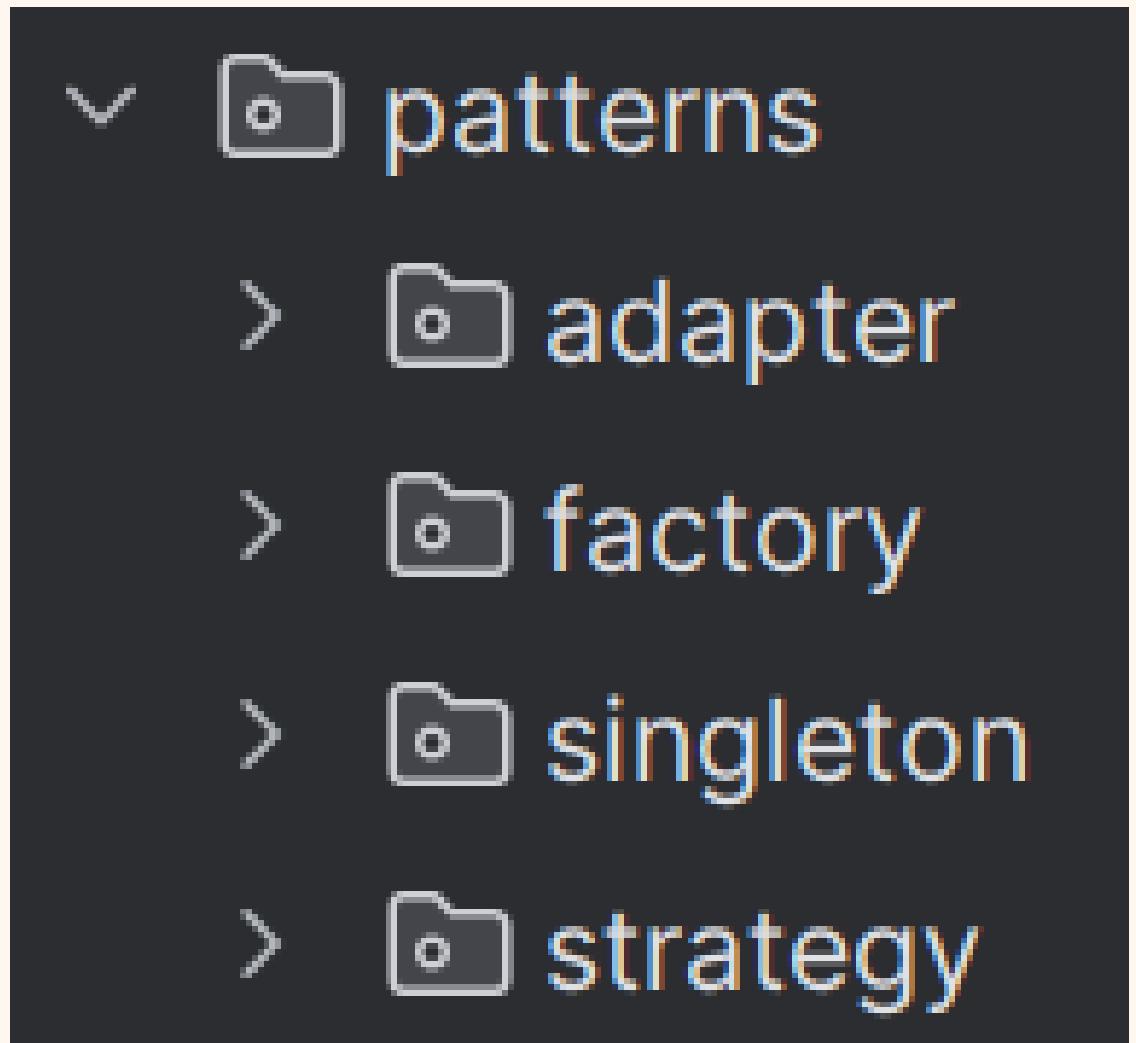
# Design Patterns

Singleton → One session, one database connection

Factory → Automatic user type creation

Strategy → Flexible fare calculation

Adapter → Clean Firebase integration



# Singleton

```
/*
 * Singleton Pattern: Manages user session throughout the app
 */
public class UserSessionManager {

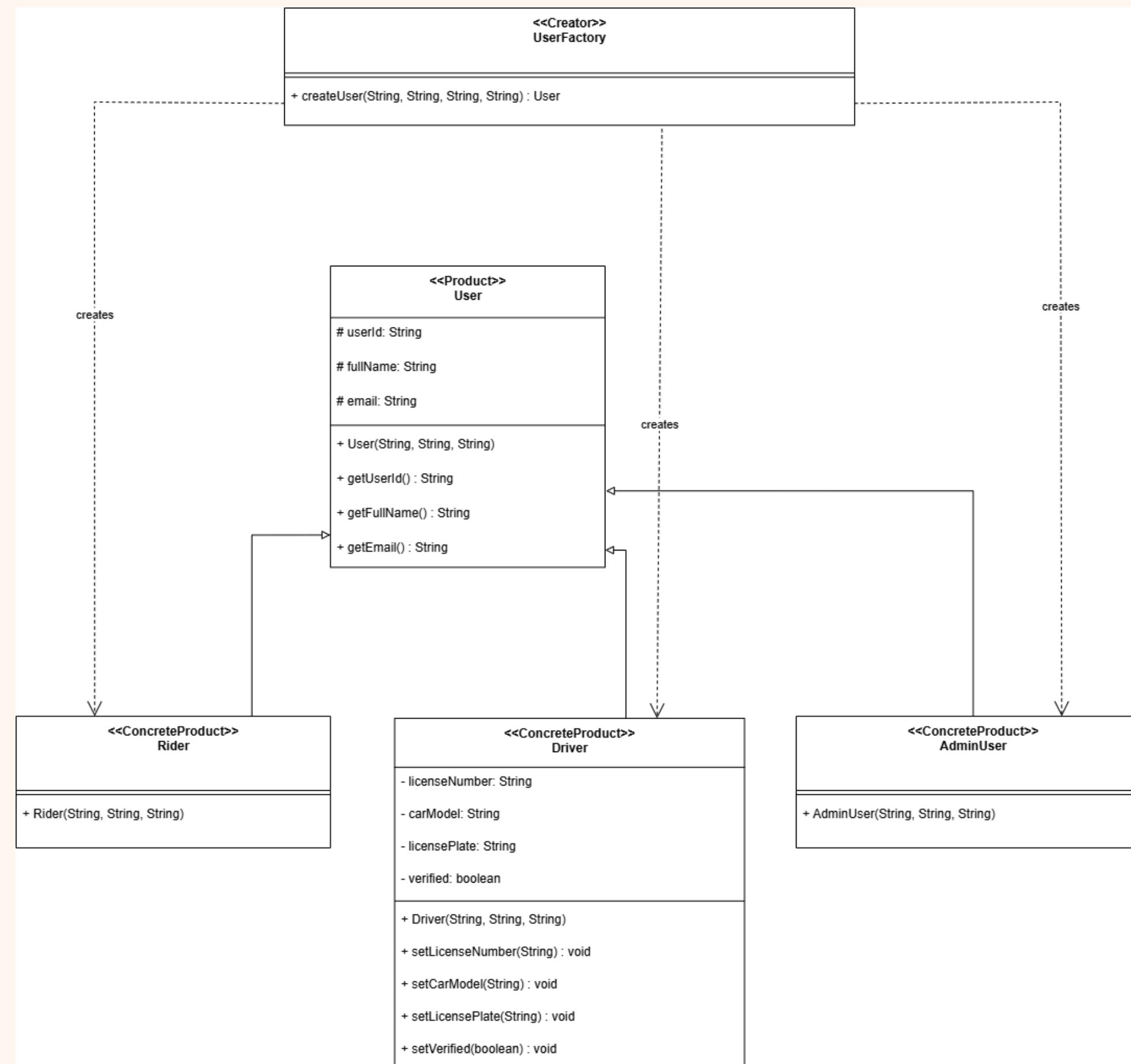
    3 usages

    private static UserSessionManager instance;
    public static synchronized UserSessionManager getInstance() {
        if (instance == null) {
            instance = new UserSessionManager();
        }
        return instance;
    }
}
```

# Singleton

```
/**  
 * Singleton Pattern: Manages all Firebase instances  
 * Ensures only one instance of Firebase services exists  
 */  
  
public class FirebaseManager {  
  
    3 usages  
  
    private static FirebaseManager instance;  
    public static synchronized FirebaseManager getInstance() {  
        if (instance == null) {  
            instance = new FirebaseManager();  
        }  
        return instance;  
    }  
}
```

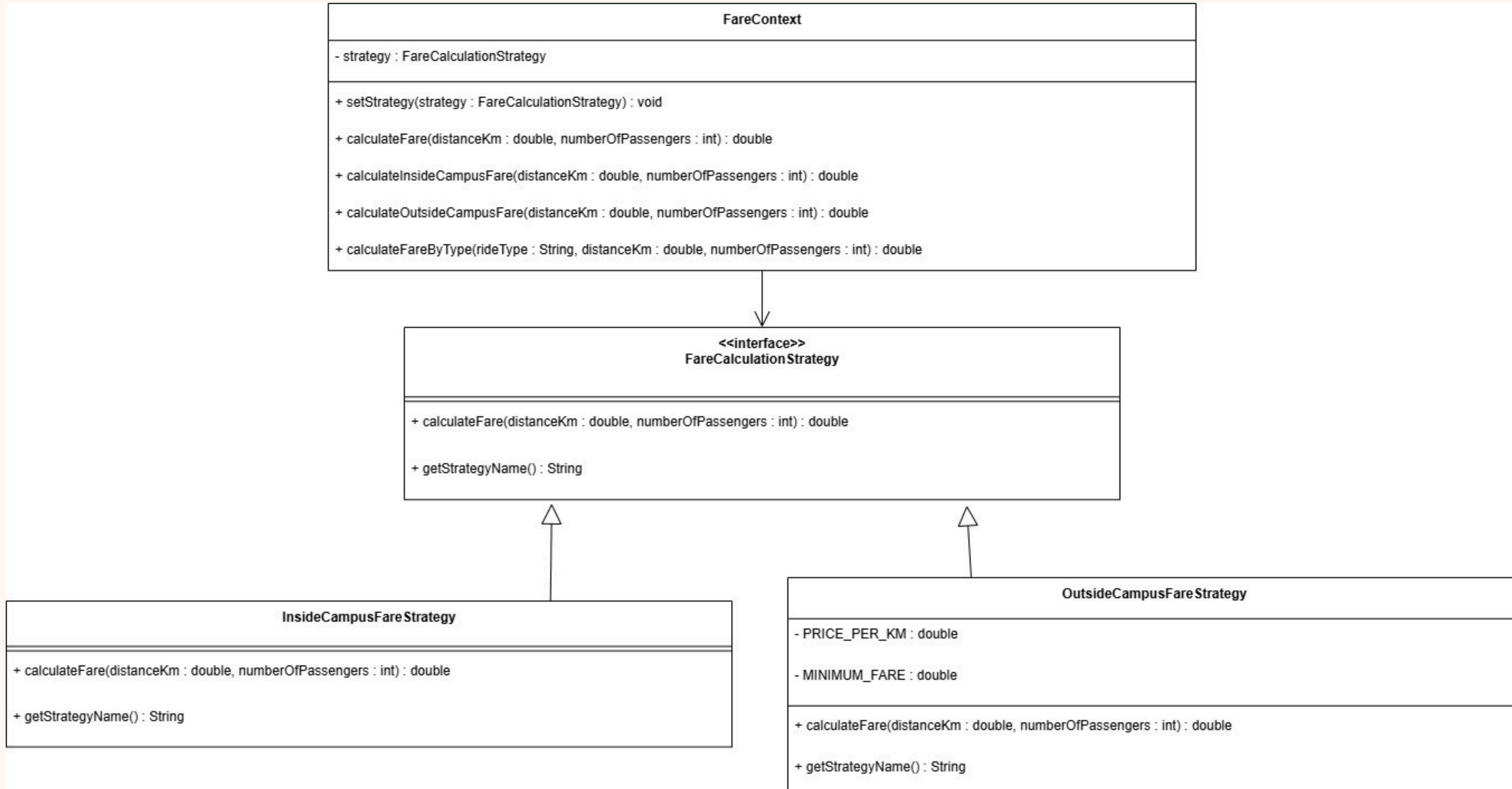
# Factory



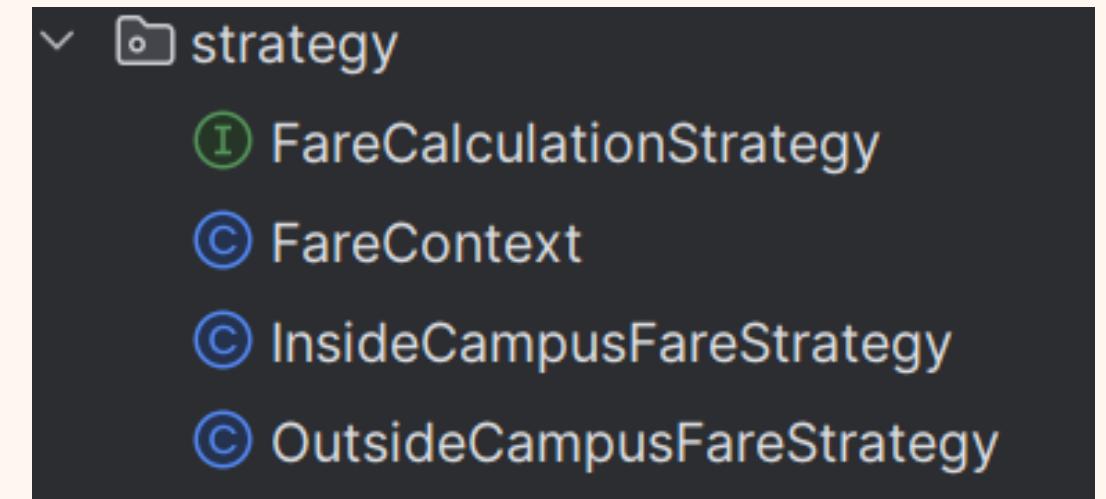
# Factory

```
/**  
 * Factory Pattern: Creates appropriate User subclass based on role  
 */  
5 usages  
public class UserFactory {  
    3 usages  
    public static User createUser(String role, String userId, String fullName, String email) {  
        if (role == null) {  
            throw new IllegalArgumentException("Role cannot be null");  
        }  
        switch (role.toLowerCase()) {  
            case "rider":  
                return new Rider(userId, fullName, email);  
  
            case "driver":  
                return new Driver(userId, fullName, email);  
  
            case "admin":  
                return new AdminUser(userId, fullName, email);  
  
            default:  
                throw new IllegalArgumentException("Unknown role: " + role);  
        }  
    }  
}
```

# Strategy

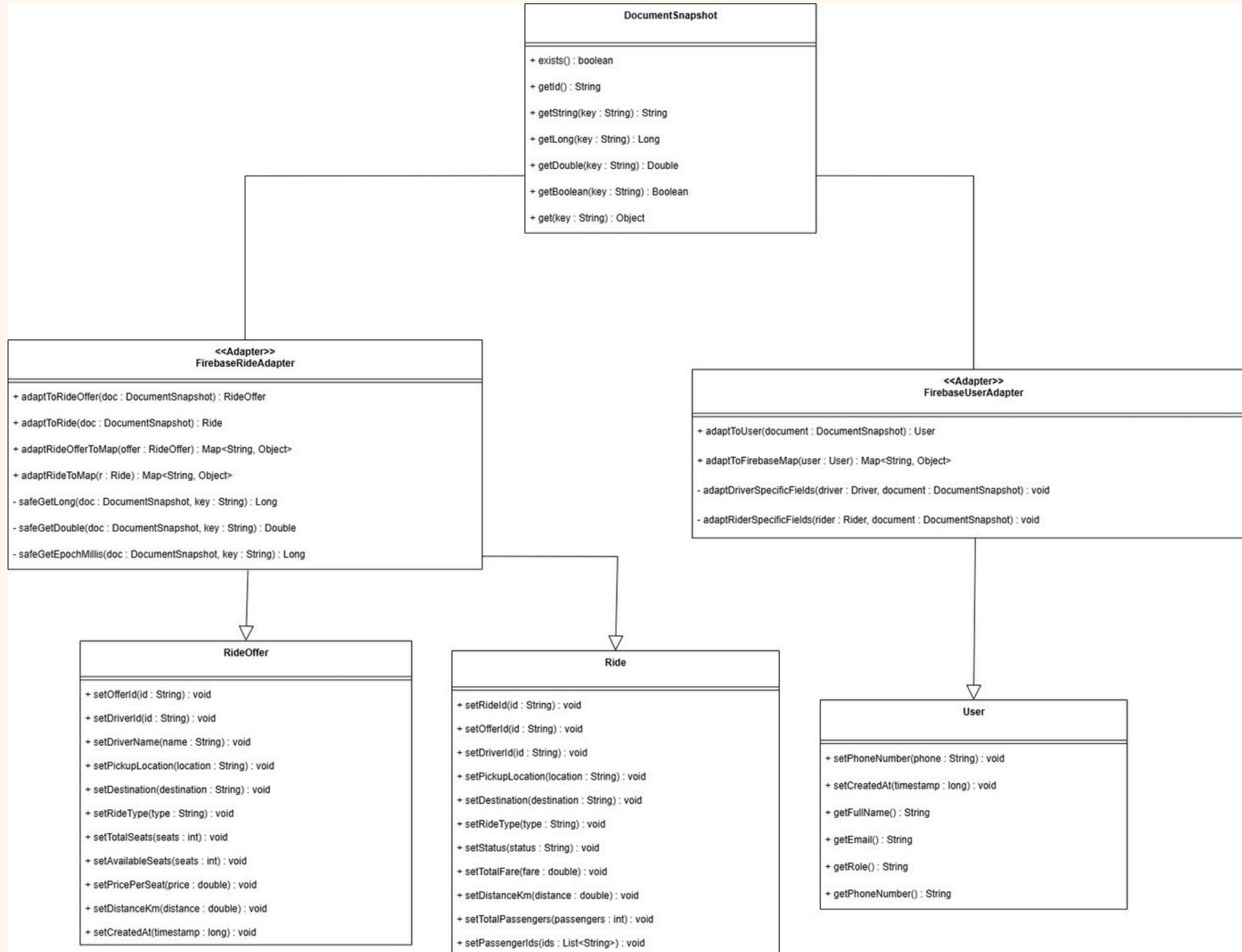


# Strategy



```
/**  
 * Strategy Pattern: Interface for fare calculation strategies  
 */  
4 usages 2 implementations  
public interface FareCalculationStrategy {  
    1 usage 2 implementations  
    double calculateFare(double distanceKm, int numberOfPassengers);  
  
    no usages 2 implementations  
    String getStrategyName();  
}
```

# Adapter



# Adapter

```
/**  
 * Adapter between Firestore DocumentSnapshot and app models  
 * - Handles numeric timestamps (Long) and Firestore Timestamp objects  
 * - Sets offerId from doc.getId()  
 */  
public class FirebaseRideAdapter {  
    9 usages  
    public static RideOffer adaptToRideOffer(DocumentSnapshot doc) {  
        if (doc == null || !doc.exists()) return null;  
        try {  
            RideOffer offer = new RideOffer();  
            offer.setOfferId(doc.getId());  
        } catch (Exception e) {  
            Log.e("FirebaseRideAdapter", "Error adapting document to RideOffer", e);  
        }  
    }  
}
```

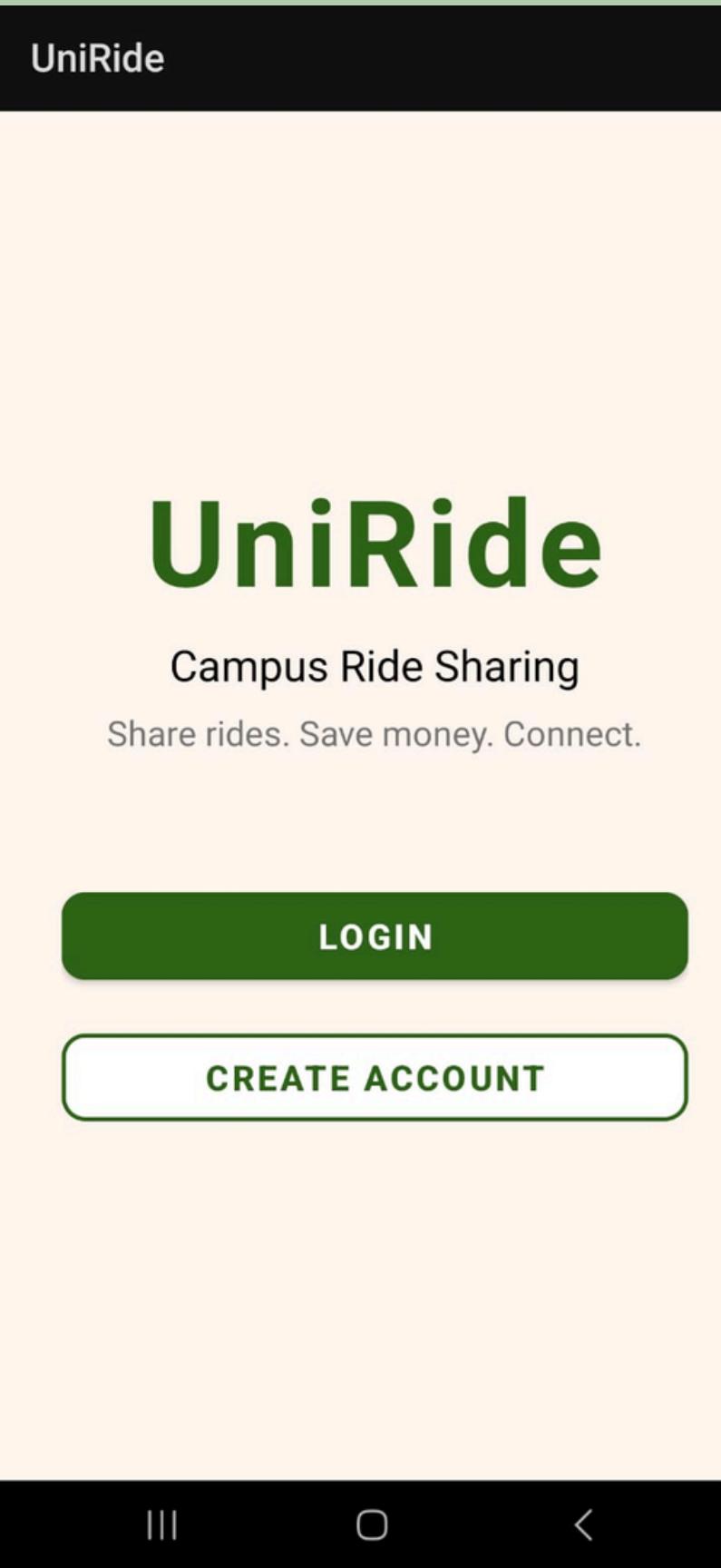
# Adapter

```
public class FirebaseUserAdapter {  
    public static Map<String, Object> adaptToFirebaseMap(User user) {  
        Map<String, Object> map = new HashMap<>();  
        map.put("fullName", user.getFullName());  
        map.put("email", user.getEmail());  
        map.put("phoneNumber", user.getPhoneNumber());  
        map.put("role", user.getRole());  
        map.put("createdAt", user.getCreatedAt());  
    }  
}
```

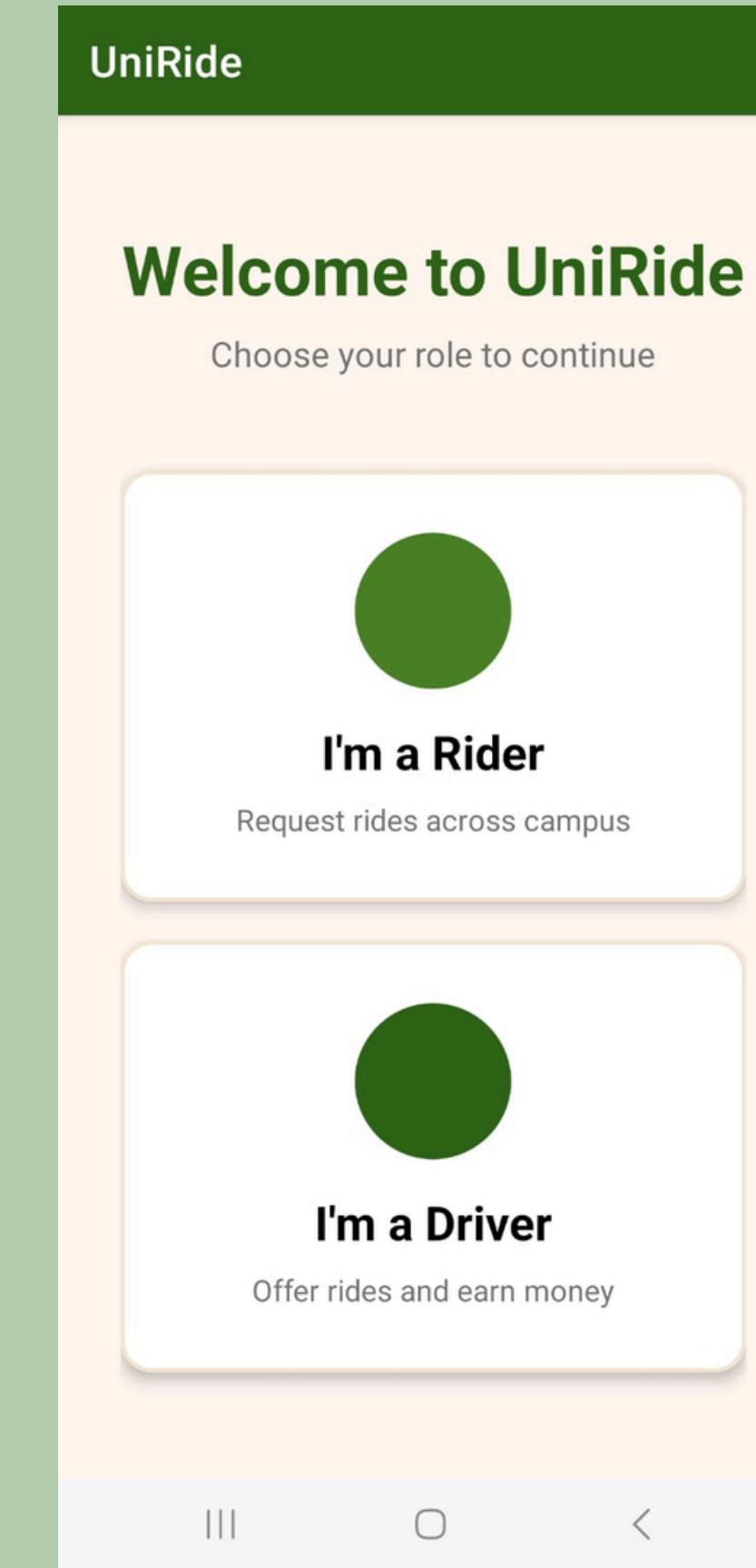


Demonstration

# Welcome Page



# Role Selection



# Creating Driver Account ...

UniRide

## Create Account

Register as Driver

Full Name

Email

Phone Number

Password

Confirm Password

Driver Information

License Number

Car Model

License Plate

REGISTER

Already have an account? [Login](#)

III    O    <

UniRide

Phone Number

Password

Confirm Password

Driver Information

License Number

Car Model

License Plate

REGISTER

Already have an account? [Login](#)

III    O    <

# Driver Uploading Documents

The screenshots illustrate the process of a driver uploading documents for verification:

- Step 1: Driver License**  
The app displays a large upload icon for the driver license. A modal overlay titled "Select Image Source" offers options: "Take Photo", "Choose from Gallery", and "Cancel".
- Step 2: ID Card**  
The app displays a large upload icon for the ID card. A modal overlay titled "Select Image Source" offers options: "Take Photo", "Choose from Gallery", and "Cancel".
- Step 3: Insurance Document**  
The app displays a large upload icon for the insurance document. A modal overlay titled "Select Image Source" offers options: "Take Photo", "Choose from Gallery", and "Cancel".
- Step 4: Document Preview**  
The app shows a preview of the uploaded documents:
  - 2. ID Card:** A Moroccan ID card for Zaineb El Alami, born 05/12/1983, valid until 22/07/2029.
  - 3. Insurance Document:** An image of a car key fob and a smartphone displaying a car insurance contract.
- Step 5: Submission Confirmation**  
The app displays a confirmation message: "Documents Submitted! Your documents have been submitted for verification. You will be notified once they are reviewed by our admin team." A green "OK" button is visible.
- Step 6: Verification Status**  
The app shows the driver's profile with a green "PENDING VERIFICATION" status badge. It also displays a message: "No ride offers found with matching status".

# Log in as Admin

UniRide

## UniRide

### Welcome Back

Login to continue

Email

admin@uniride.com

Password

.....



**LOGIN**

[Forgot Password?](#)

[Don't have an account? Register](#)



# Admin approving Driver

Admin Dashboard

Total: 1 | Drivers: 1 | Riders: 0 | Pending: 1

All All

**driver one**  
driver1@gmail.com  
Registered: Nov 30, 2025

PENDING

VIEW DETAILS

DRIVER

UniRide

**driver one**  
driver1@gmail.com  
Registered: Nov 30, 2025  
Toyota Corolla • 55-5257-1  
License: DR12345

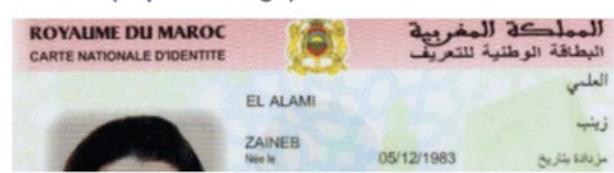
Status: pending

Documents

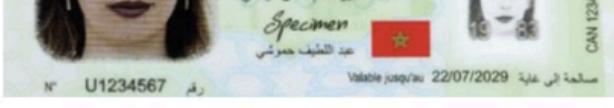
Driver License (Tap to enlarge)



ID Card (Tap to enlarge)



UniRide



Insurance (Tap to enlarge)



Verification Actions

APPROVE

REJECT

Driver Reviews

Total Reviews: 0

Average Rating: 0.0 ★

No reviews yet

# Driver can now offer rides

**UniRide**

## Welcome driver one!

Manage your ride offers

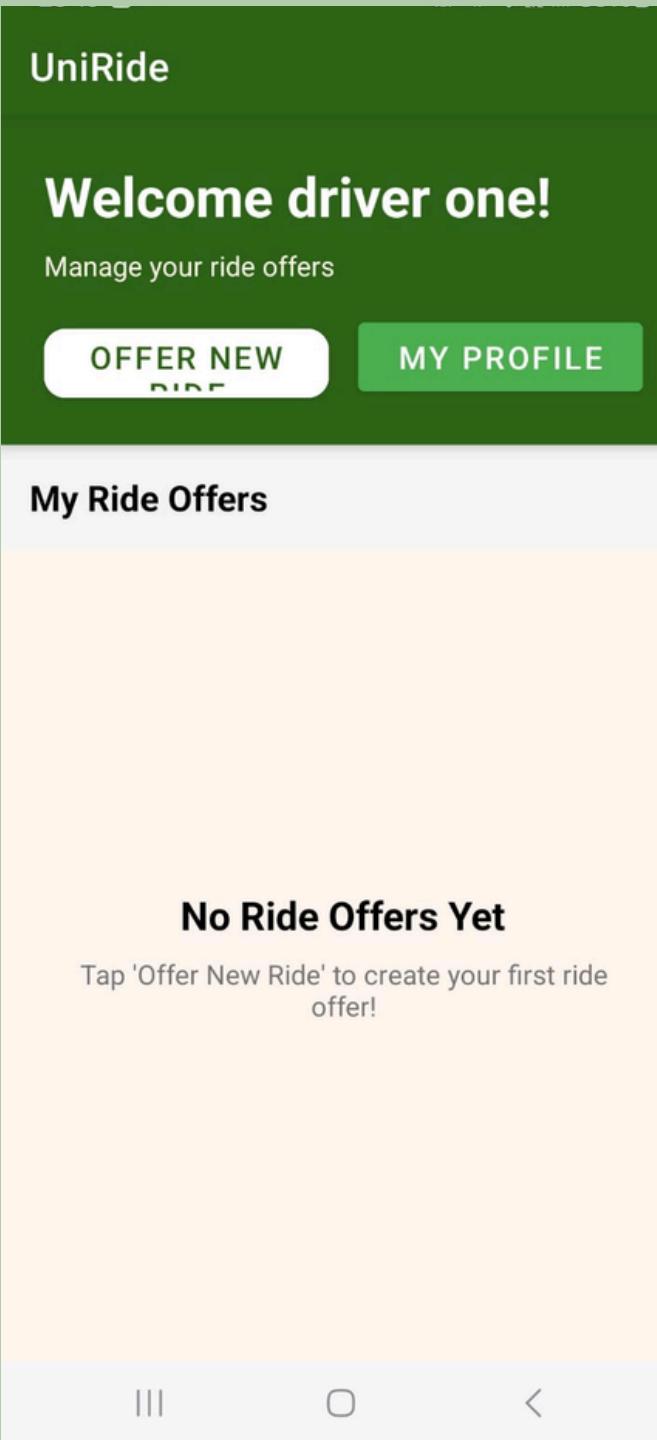
**OFFER NEW RIDE** **MY PROFILE**

### My Ride Offers

**No Ride Offers Yet**

Tap 'Offer New Ride' to create your first ride offer!

||| ⌂ <



**UniRide**

## Create Ride Offer

Ride Type

INSIDE\_CAMPUS

### Select Campus Locations

Pickup Location

R8

Destination

R8

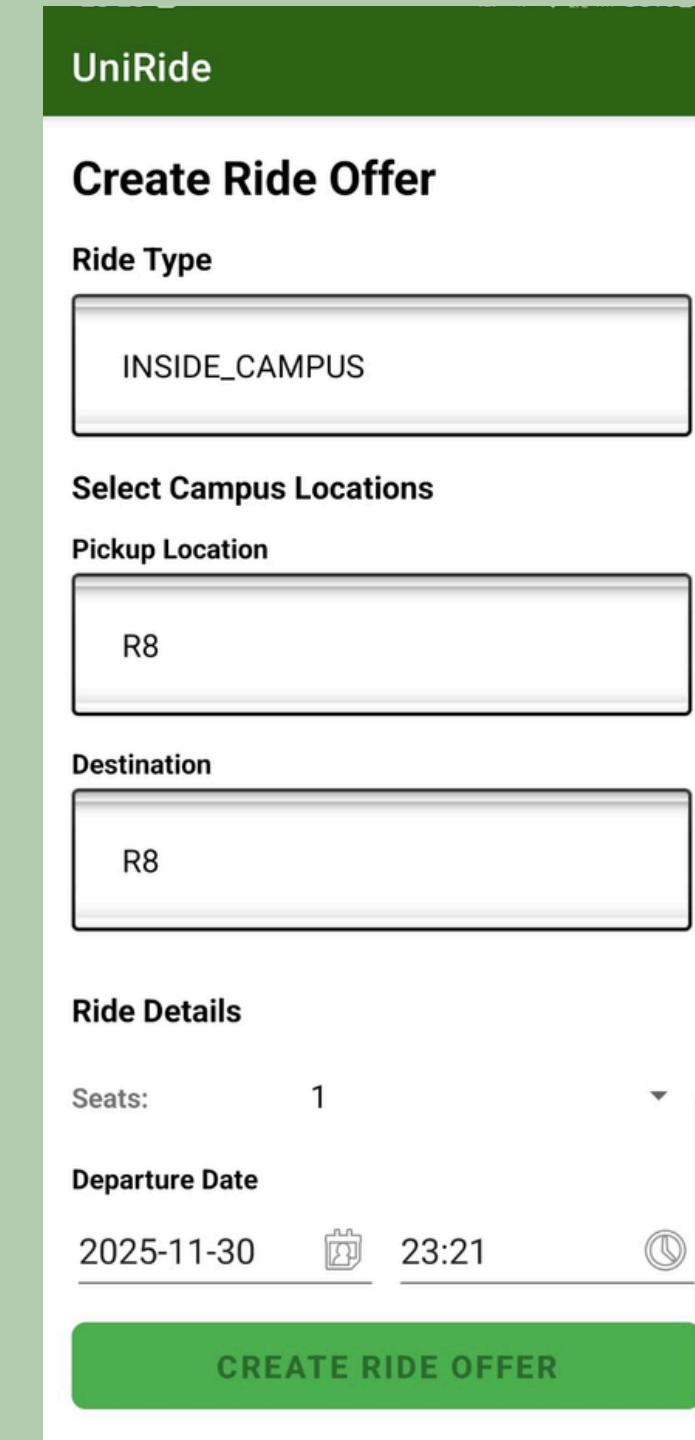
### Ride Details

Seats: 1

Departure Date

2025-11-30 23:21

**CREATE RIDE OFFER**



**UniRide**

## Create Ride Offer

Ride Type

OUTSIDE\_CAMPUS

### Search Location

Search for a location



**Select Locations on Map**

Pickup: Tap 'Select Pickup' to begin **SELECT**

Destination: Select pickup first **SELECT**

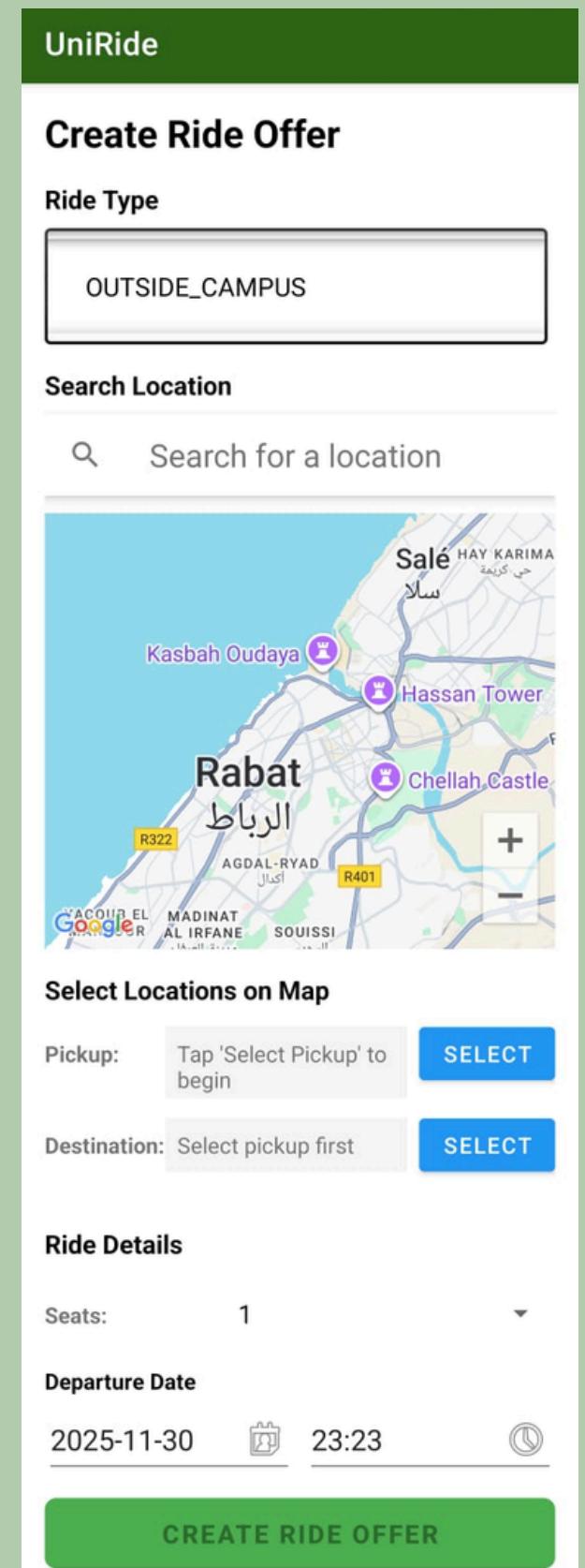
### Ride Details

Seats: 1

Departure Date

2025-11-30 23:23

**CREATE RIDE OFFER**





Live demonstration

# Rider

Email

Password

# Driver

Email

Password

**Thank you !**

Nour El Houda El Iamani & Aya Benjelloun