

RIDE POOLING

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfilment of the requirements to award the degree of

Bachelor of Technology

In

Computer Science and Engineering

School of Engineering and Sciences

Submitted by

Monika Sree Srinivasan (AP21110011359)

Sai Chaitanya Kagita (AP21110011367)

Anantha Teja Dasari (AP21110011369)

Charishma Pothireddy (AP21110011376)



Under the Guidance of

Dr. Sanjay Kumar

SRM University–AP

Neerukonda, Mangalagiri, Guntur

Andhra Pradesh – 522 240

April, 2024

Certificate

Date: 30-04-2024

This is to certify that the work present in this Project entitled “**Ride Pooling**” has been carried out by **Monika Sree Srinivasan, Sai Chaitanya Kagitha, Anantha Teja Dasari, Charishma Pothireddy** under my supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor Technology/Master of Technology in **School of Engineering and Sciences**.

Supervisor

(Signature)

Prof. / Dr. Sanjay Kumar

Designation,

Affiliation.

Acknowledgment

I would like to express my special thanks to our mentor Dr. Sanjay Kumar for his time and efforts he provided throughout the year. Your useful advice and suggestions were helpful to me during the project's completion. In this aspect, I am eternally grateful to you.

We would like to take this opportunity to express my gratitude to all our group members. The project would not have been successful without their cooperation and input.

Abstract

Urban transportation systems grapple with many problems, among them congestion, environmental pollution and wastage of resources. Globally, ride pooling has emerged as a concept that could potentially address these issues by maximizing vehicle utilization while minimizing the number of vehicles on the roads. In this context, we present a systematic review which seeks to assess what current research says about ride pooling vis-à-vis its impacts, challenges and potential for transforming urban mobility.

The scope of this study covers articles from journals, reports and cases published between 2010 to 2024 inclusive; it also includes grey literature such as working papers or conference proceedings where appropriate. One major finding is that ride-sharing has great potential for improving city transport systems because it reduces traffic jams, cuts down greenhouse gas emissions and provides better access to underserved communities in terms of their transport needs. Additionally, shared mobility services can act as complements to traditional public transport modes by providing first/last mile connectivity solutions.

Nevertheless there are numerous challenges which may hinder wide spread adoption or effectiveness of shared economy platforms like Uber/Lyft et al., some examples include lack regulatory framework (legal vacuum), privacy concerns regarding data sharing between users/drivers/platforms; technological limitations especially when it comes down too matching supply with demand within reasonable time periods (real-time); need for coordination across different stakeholders – government agencies responsible for planning & implementing various modes infrastructure improvement works etc.; socio-economic impact assessment e.g., employment opportunities

Table of Contents

1. Introduction	viii
2.The Overall Description	ix
2.1 Product Perspective.....	ix
2.2 Login Features:.....	ix
2.3 User Characteristics.....	x
2.4 Constraints	xi
2.5 Assumptions and Dependencies:.....	xii
3.1 Interface Requirements	xv
3.2 Hardware Interface Requirements	xvi
3.3 Software Interface Requirements:	xvi
3.3.1 Operating System Compatibility:.....	xvi
3.4 Communication Interface Requirements	xvii
4. System Features :	xviii
5. Other Nonfunctional Requirements.....	xix
5.1 Performance Requirements	xix
5.1.1 Capacity.....	xix
5.1.2 Dynamic requirements.....	xix
5.1.3 Quality.....	xix
5.2 Software System Attributes	xx
5.2.1 Reliability	xx
5.2.2 Availability.....	xx
5.2.3 Security	xx
5.2.4 Maintainability.....	xx
6. Other Requirements	xxii
7. List of Diagrams	xxiii
7.1 Level-0 DFD.....	xxiii
7.2 Level-1 DFD.....	xxiv
7.3 Level-2 DFD.....	xxv
7.4 Data Dictionary.....	xxvii

7.5 Use Case Diagram	xxviii
7.6 ER Diagram.....	xxx
7.7 Class Diagram	xxxii
7.8 Activity Diagram	xxxiii
7.9 State chart Diagram.....	xxxiv
7.10 Collaboration Diagram	xxxv
8. Testing.....	xxxvi
8.1 Functional Testing	xxxvi
8.2 Path Testing.....	xxxvii
8.2.1 Signup ()	xxxvii
8.2.2 Login ()	xxxvii
9. Technology Stack.....	xxxix
10. Results.....	xl
11. Conclusion.....	xliv
12. Future Work.....	xlvi
13. Reference	xlvi

List Of Figures

1. Level 0 DFD
2. Level 1 DFD
3. Level 2 DFD
4. Use Case Diagram
5. ER Diagram
6. Class Diagram
7. Activity Diagram
8. State Diagram
9. Collaboration Diagram

List of Tables

1. Definitions, Acronyms and Abbreviations.
2. Data Dictionary.

1. Introduction

In this introduction, we will be looking into the software engineering principles that are used for making a collaborative ride sharing platform. We shall explore its architecture, functionalities and key stakeholders involved during deployment. The foremost point is that the structure of a ride sharing platform should be able to handle real time interactions among passengers and drivers. This requires creation of strong back-end systems that can process huge amounts of data including user preferences, location details as well as route optimization algorithms. Moreover, the system needs friendly interfaces both for riders and drivers which facilitate easy booking procedures payment processing communication throughout the entire pooling.

Basically from software engineering perspective success behind any urban transportation-based application like ridesharing apps lies on adoption current technologies to solve specific problems associated with this industry. For example machine learning algorithms come in handy when it comes to forecasting passenger demand streamlining driver routes or even dynamically adjusting pricing models so as to encourage more people take part. Furthermore integration geospatial information systems together with mapping APIs helps enable vehicle tracking in real-time efficient allocation of resources.

2.The Overall Description

2.1 Product Perspective

- a. Centralized platform in the Ride Pooling.
- b. Connects users.
- c. Offers:
 - User Registration
 - User Login

Aims to:

- Broader context of transportation services
- Offering an alternative to traditional modes of transportation
- Reduce the traffic and reduce environmental pollution.

2.2 Login Features:

- User Authentication: Secure login mechanism for all users (Users). Password encryption and secure authentication protocols ensure the safeguarding of user credentials.

1. Ride Posting:

- Details: Drivers provide information about their upcoming trips, including where they're departing from, their destination, the date and time of the trip, and the number of available seats in their vehicle.
- Convenience: By posting their rides, drivers offer the opportunity for others traveling along similar routes to join them, making their journey more cost-effective and sociable.
- Flexibility: Drivers can specify any preferences or requirements, such as preferred pickup/drop-off points, the willingness to make detours, or restrictions like no smoking or pets.

2. Ride Searching:

- Filters: Passengers can use filters to search for available rides that match their criteria, such as departure and destination locations, date, and time.

- **Options:** The platform presents passengers with a list of available rides that meet their search criteria, allowing them to compare options based on factors like departure time, route, and cost.

3. Booking Rides:

- **Seat Reservation:** Passengers can book seats on available rides directly through the platform, usually by selecting the desired ride and confirming their booking.
- **Confirmation:** Once a booking is made, both the driver and passenger receive confirmation notifications, detailing the trip's details, including pickup time, location, and any additional instructions.

Additional features and considerations:

- **Communication:** Platforms often include messaging functionality to facilitate communication between drivers and passengers, allowing them to coordinate details like pickup points or special requests.
- **Reviews and Ratings:** Users may have the option to leave reviews and ratings for each other after the trip, providing valuable feedback for future users and fostering a sense of accountability within the community.

2.3 User Characteristics

<i>Participant</i>	<i>Priority</i>
<i>User</i>	Medium
<i>Guest</i>	Low
<i>Admin</i>	High

Admin:

Admin supervises every user and revies the feedback from the rides. If there is any suspicious activity from a user, the admin can delete the user from logging in or posting the rides again.

User:

The user has the major role as he must post his rides, give detailed description about his ride locations along with the number of passengers he can take along with him. User can also book his ride to his interest or necessity.

2.4 Constraints

Regulatory Challenges:

- Ride-sharing services often encounter regulatory hurdles and must adhere to diverse local transportation laws, which vary significantly across regions.
- Manoeuvring through complex legal frameworks and obtaining requisite licenses can consume considerable time and resources.

User Safety:

- Prioritizing the safety of both passengers and drivers remains paramount. Conducting thorough background checks, implementing identity verification measures, and integrating safety features into the platform are imperative.
- Establishing robust protocols to handle emergencies and establishing a dependable support system are critical for bolstering user confidence.

Trust Building:

- Fostering trust among users stands as a cornerstone for ride-sharing platform success. This necessitates the establishment of a transparent and reliable rating, review, and feedback system.
- Addressing user concerns regarding personal security and privacy is essential to instill and sustain user trust.

Payment Transactions:

- Effectively managing secure and seamless payment transactions poses a significant challenge. This entails deploying robust payment gateways, ensuring data security, and accommodating various payment methods.
- Upholding transaction integrity and user confidence through meticulous payment processing is vital.

User Preferences:

- Catering to a broad spectrum of user preferences and requirements presents a complex challenge. This encompasses accommodating diverse vehicle types, ride durations, and comfort levels.
- Providing customizable options while upholding an intuitive user experience remains pivotal.

Legal Issues:

- Ride-sharing services confront legal challenges pertaining to liability, insurance, and contractual arrangements between drivers and passengers.
- Resolving disputes and navigating evolving legal landscapes constitute ongoing endeavors.

Adapting to Local Markets:

- Tailoring services to suit the specific needs and cultural intricacies of distinct regions is essential. This involves comprehending local transportation practices, preferences, and regulatory contexts.

Competition and Scalability:

- Competing with traditional transportation alternatives and other ride-sharing platforms demands continual innovation and enhancement.
- Managing scalability concerns, particularly during peak periods or market expansions, is indispensable for ensuring consistent user satisfaction.

2.5 Assumptions and Dependencies:

Database System:**Assumptions:**

- The ride-sharing platform is developed to operate on a specific database system, such as MongoDB.

Dependencies:

The platform relies on the compatibility and support of the chosen database system. Any changes in the database system may necessitate updates to the application to maintain functionality.

Concurrent User Access:

Assumptions:

Users accessing the ride-sharing platform may perform transactions simultaneously.

Dependencies:

To maintain data consistency and avoid conflicts, especially during financial transactions, the application requires mechanisms to handle concurrent user access effectively.

Performance Optimization:

Assumptions:

- While specific quantitative measures are not imposed, it is implied that the ride-sharing application functions are optimized for speed and memory usage.

Dependencies:

- Continuous monitoring and optimization of the application's performance are essential to ensure a smooth user experience, particularly as the user base grows.

Scope Expansion:

Assumptions:

- The scope of the ride-sharing platform is expected to increase considerably in the future.

Dependencies:

- The application will be designed with scalability and flexibility to accommodate future enhancements and features as the scope expands, ensuring it can adapt to evolving needs and demands.

Data Security and Privacy Compliance:

Assumptions:

- It is assumed that the platform employs robust encryption protocols to safeguard user information and transactional data.
- There is an assumption that the platform regularly conducts security audits and assessments to identify and address potential vulnerabilities.
- It is assumed that users trust the platform to handle their data responsibly and securely, fostering confidence in the service.

Dependencies:

- The platform must adhere to data security and privacy regulations to protect user information and transactional data. Any changes in data protection laws or cybersecurity requirements necessitate updates to the platform's security measures and policies. Mobile Device Compatibility.

3. External Interface Requirements

3.1 Interface Requirements

Ride pooling prioritizes a seamless and user-friendly interface to enhance user engagement and accessibility. The platform's user interface requirements include:

- **Intuitive Design:** The platform should feature a clean and intuitive interface that allows users to quickly understand how to request rides, join pools, and navigate through different sections such as profiles, payment options, and support. Clear icons and intuitive placement of features can aid users in easily accessing the functionalities they need.
- **Responsive Layout:** Given the widespread use of smartphones for accessing ride-hailing services, the interface should be optimized for various screen sizes. Whether users are accessing the platform from a desktop computer, tablet, or smartphone, they should have a seamless experience without sacrificing functionality or clarity.
- **Clear Navigation:** Navigation menus should be well-organized and prominently displayed, guiding users to key features like searching for rides, setting pickup and drop-off locations, selecting ride preferences, and managing their account settings. Labels and tooltips can help clarify the purpose of different buttons and options.
- **Accessible Features:** To ensure inclusivity, the platform should adhere to accessibility standards such as providing alternative text for images, enabling keyboard navigation support, and ensuring compatibility with screen readers. This allows users with disabilities to fully engage with the platform and access its services.
- **Interactive Elements:** Interactive elements like dropdown menus for selecting ride preferences, buttons for confirming ride requests, and forms for inputting pickup/drop-off locations should provide immediate feedback to users. Visual cues such as loading animations or success/error messages can enhance the responsiveness of these elements.
- **Rich Media Integration:** Integrating rich media elements such as driver profiles with photos, interactive maps showing real-time ride availability, and videos demonstrating safety features can enhance the

visual appeal and engagement of the platform. Visual content can also help users make informed decisions when selecting rides or drivers.

- **Security Measures:** Given the sensitive nature of personal and payment information involved in ride transactions, robust security measures are paramount. Implementing end-to-end encryption for communications, secure authentication processes like two-factor authentication, and compliance with data protection regulations ensure the confidentiality and integrity of user data, fostering trust and confidence among users.

3.2 Hardware Interface Requirements

Device Compatibility:

- The ride-sharing platform must ensure seamless compatibility across a variety of devices, including desktops, laptops, tablets, and smartphones. To achieve this, the platform will employ a responsive design approach, optimizing user experience across different screen sizes.
- Resource optimization is paramount to prevent excessive consumption of device resources and to ensure optimal loading times for multimedia content. The platform will employ efficient resource management techniques to deliver a smooth and responsive experience to users.

3.3 Software Interface Requirements:

- In order to ensure seamless functionality and interoperability with various devices and software components, the ride-sharing or pooling platform must adhere to the following interface requirements:

3.3.1 Operating System Compatibility:

- The ride-sharing platform should be compatible with major operating systems including Windows, macOS, and Linux to ensure accessibility across a wide range of devices.
- Consistent functionality should be maintained across different operating systems to provide a uniform user experience.

3.3.2 MERN Stack Integration:

- The ride-sharing platform should seamlessly integrate with MongoDB, Express.js, React, and Node.js components to leverage the strengths of the MERN stack for enhanced performance and scalability.
- Efficient integration of these components is essential to ensure smooth operation and optimal resource utilization within the platform.

Profile Management Interfaces:

1. User Profile:

- Easy-to-navigate profile pages with options for users to update personal information, vehicle details, and payment preferences.
- Customizable profile pictures and preferences to personalise the user experience.

2. Settings and Preferences:

- User-friendly account settings with options for notification preferences, language settings, and accessibility features.
- Password reset functionality with clear instructions.

3.4 Communication Interface Requirements

- Users shall be able to attach multimedia files, images, or links within discussions to share ride-related content and experiences.

4. System Features :

- **User Registration and Authentication:** Allow users to create accounts, log in securely, and manage authentication credentials.
- **Profile Management:** Enable users to update personal information, manage vehicle details, and set preferences.
- **Ride Confirmation and Management:** Allow users to confirm or decline ride pooling requests, view details of confirmed rides, and manage ride preferences such as music, temperature, and route deviations.
- **Safety and Trust Features:** Implement safety features such as user verification, driver ratings, and emergency assistance buttons to ensure a secure and trustworthy ride sharing experience. **Feedback and Ratings:** Enable users to provide feedback and ratings for ride pooling experiences, contributing to the overall quality and reliability of the platform.
- **Ride History:** Offer users access to their ride history, including details of past shared rides, distance travelled, and cost breakdowns.

Infrastructure and Technical Features:

Scalability and Performance:

- Scalable architecture leveraging cloud infrastructure to handle varying levels of user traffic.
- Performance optimization techniques to ensure fast loading times and responsive user interactions.

Reliable Backend Infrastructure:

- Utilization of Node.js for backend development to ensure a reliable and efficient server-side infrastructure.
- Integration with MongoDB for flexible and scalable database management.

Interactive Frontend Development:

- Frontend development using React.js for building dynamic and interactive user interfaces.
- Component-based architecture for modular and maintainable frontend code.

Security and Data Privacy;

- Robust security measures to protect user data, including encryption, secure authentication, and data access controls.
- Compliance with data privacy regulations such as GDPR to safeguard user privacy rights.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- The ride pooling platform shall be capable of handling concurrent user requests efficiently without significant performance degradation, even during peak usage periods.
- Response times for critical functions such as ride matching, route planning, and payment processing shall be optimized to ensure a seamless user experience.

5.1.1 Capacity

- The platform should handle a significant user load efficiently.
- Simultaneous access by a large number of users, including doctors and admins.

5.1.2 Dynamic requirements

- Dynamic adaptation to varying workloads and user interactions.
- Maintaining optimal performance during peak usage times..
- The search and filtering functionalities should be dynamic, allowing users to refine their searches based on changing criteria, such as arrival to destination.

5.1.3 Quality

- The primary objective of the Ride Pooling project is to develop a user friendly way or means of transportation avoiding the traditional means.

- **Consistency:** Adhere to established coding standards to maintain a consistent and uniform coding style. Consistency is a key aspect that promotes readability and ease of maintenance.
- **Thorough Testing:** Testing should encompass various scenarios to identify and rectify potential issues and to verify and validate all functionalities within the Chatbot platform.
- **Scalability:** The software should be scalable, accommodating potential increases in user base and data volume.

5.2 Software System Attributes

5.2.1 Reliability

- The platform shall strive for high availability, with minimal downtime and system failures.
- Robust recovery mechanism in case of system failures.

5.2.2 Availability

- High availability, allowing users to access the platform 24/7.
- Planned maintenance windows communicated in advance.
- Conduct regular maintenance during non-peak hours to minimize disruption to users.

5.2.3 Security

- The platform shall implement robust security measures to protect user data, including encryption of sensitive information during transmission and storage.
- User authentication and authorization mechanisms shall adhere to industry best practices to prevent unauthorized access to user accounts and personal information.
- Use strong encryption algorithms to secure user data during transmission and storage.

5.2.4 Maintainability

- The platform shall implement robust security measures to protect user data, including encryption of sensitive information during transmission and storage.

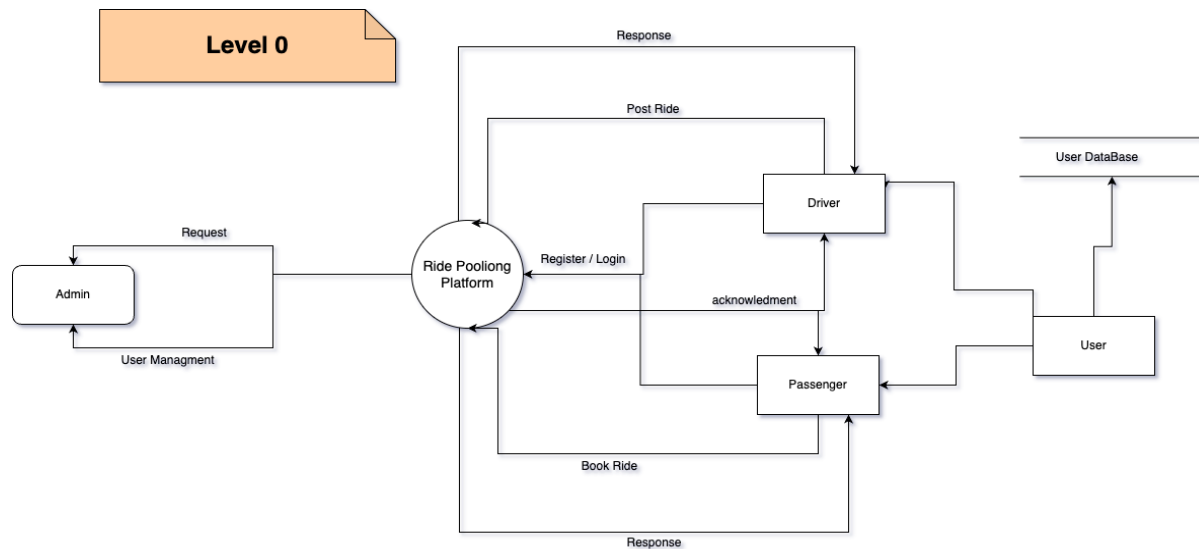
- Provide comprehensive and well-organized documentation to aid developers in understanding the system architecture and codebase.
- Utilize version control systems to track changes, enabling easy rollback in case of errors.
- Establishment of clear coding standards and guidelines to promote a consistent and maintainable codebase.

6. Other Requirements

- Appendix A: Glossary
- **HTTPS (Hypertext Transfer Protocol Secure)** - A secure version of the HTTP protocol used for secure communication over a computer network, providing encryption and authentication.
- **WebSocket** - A communication protocol that provides full-duplex communication channels over a single TCP connection, allowing interactive communication between a web browser and a web server.

7. List of Diagrams

7.1 Level-0 DFD



Entities:

- User: Posts rides.
- Book Ride: User can choose his rides from the available based on his preference.

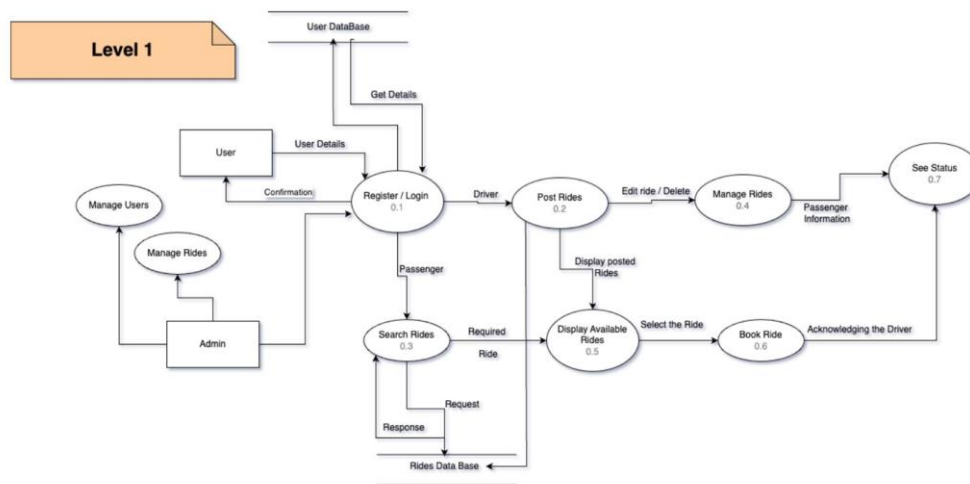
Data Flows:

- Book-ride: The user submits a book query to the application.
- Post-Ride: The User Can post his rides
- User: Fetching from the user database.

Processes:

- The level 0 DFD doesn't illustrate the specific processes that the chatbot performs.

7.2 Level-1 DFD



Processes:

- **Registration** (process): This process manages user registration. It takes user input for details and stores them in the Login Database.
- **Login** (process): This process validates the user's credentials upon login. It retrieves data from the Login Database and verifies it against the user-provided credentials.
- **Manage User Profile** (process): This process allows users to change their passwords. It presumably updates the Login Database with the new password.

Data Flows:

- **Login Credentials** (data flow): This data flow carries the user's login ID and password for validation. It goes from the User to the Login process.
- **Registration Details** (data flow): This data flow carries user data for registration. It goes from the User to the Registration process and likely includes items like username, password, and potentially other user profile information.

- **Updated Password** (data flow): This data flow likely carries the new password when a user updates their profile. It goes from the Manage User Profile process to the Login Database.
- **Update Rides:** The user who posted the rides can edit their respective rides.

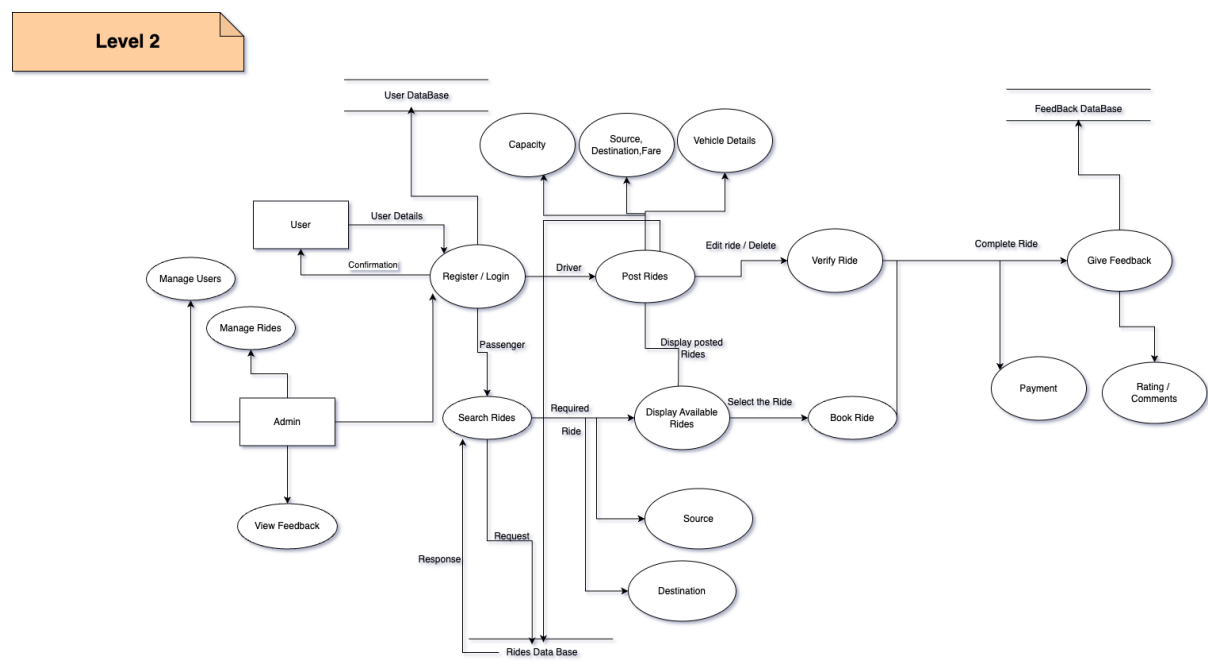
Data Stores:

- **Login Database** (data store): This database stores user login credentials.
- **Rides Database** (data store): This database stores doctor records.

External Entities:

- **User** (external entity): This is the person interacting with the Platform.

7.3 Level-2 DFD



Data Flows

- User enters registration/login details --> User Database
- User edits ride/cancels ride --> Manage Rides
- Driver posts ride details (including capacity, source, destination, fare) --> Manage Rides --> Rides Database

- Admin manages users --> User Database
- User searches for rides (specifying source and destination) --> Search Rides --> Rides Database
- Admin views feedback --> Feedback Database

Processes

- **Register/Login:** The system verifies the user's login credentials or creates a new user profile in the User Database.
- **Manage Users:** The admin can add, edit, or delete user information in the User Database.
- **Post Rides:** The driver submits ride details including capacity, source, destination, and fare. This information is stored in the Rides Database.
- **Manage Rides:** This encompasses functionalities such as editing or cancelling rides by the driver.
- **Search Rides:** The system searches for available rides based on the user's specified criteria (source and destination) and displays them to the user.
- **Book Ride:** The user selects a ride from the available options. The system verifies the ride and makes a booking.
- **Verify Ride:** (not shown in the data flow, but can be inferred) The driver verifies the passenger's identity before the ride begins.
- **Complete Ride:** Once the ride is complete, the system marks it as finished.
- **Rating/Feedback:** The user can provide feedback and a rating about the ride. This information is stored in the Feedback Database.
- **View Feedback:** The admin can view the feedback provided by users.

7.4 Data Dictionary

Data Entity	Description	Example Value
User ID	Unique identifier for a registered user	USR-5678
First Name	First name of the user	John
Last Name	Last name of the user	Doe
Email	Email address of the user	johndoe@example.com
Password	Securely stored password for the user	(not shown)
Phone Number	Phone number of the user	+1-123-456-7890
Payment Method	Payment method preferred by the user	Credit Card
License Plate	License plate number of the vehicle	ABC-123
Capacity	Number of seats in the vehicle	4
Trip ID	Unique identifier for a trip	TRP-5678
Pickup Location	Location where the trip starts	123 Main St
Dropoff Location	Location where the trip ends	456 Elm St
Scheduled Pickup Time	Time scheduled for pickup	2024-03-20 10:00
Actual Pickup Time	Actual time of pickup	2024-03-20 10:15
Scheduled Dropoff Time	Time scheduled for dropoff	2024-03-20 11:00
Actual Dropoff Time	Actual time of dropoff	2024-03-20 11:30
Trip Status	Status of the trip (e.g., Scheduled, Completed)	Completed
Booking ID	Unique identifier for a booking	BKG-7890
Booking Time	Time when the booking was made	2024-03-19 15:30
Number of Passengers	Number of passengers for the trip	2
Total Fare	Total fare for the trip	\$25.00
Comment	Additional comments about the trip	Great ride!

Table 2 Data Dictionary

7.5 Use Case Diagram



Actors

- **Passenger:** A person who uses the ride-pooling service to request rides.
- **Driver:** A person who uses the ride-pooling service to offer rides and transport passengers.
- **Admin:** A person who manages the system, including adding/editing users and viewing feedback.

Use Cases

Passenger

- **Register/Login:** The passenger creates a new account or logs in to an existing account.
- **Search Rides:** The passenger searches for available rides based on their specified criteria (source and destination).
- **Book Ride/Request Ride (assumed):** The passenger selects a ride from the available options and books it.

- View Ride History: The passenger sees a record of their past rides.
- Provide Feedback:** The passenger can provide feedback and a rating about their ride experience.

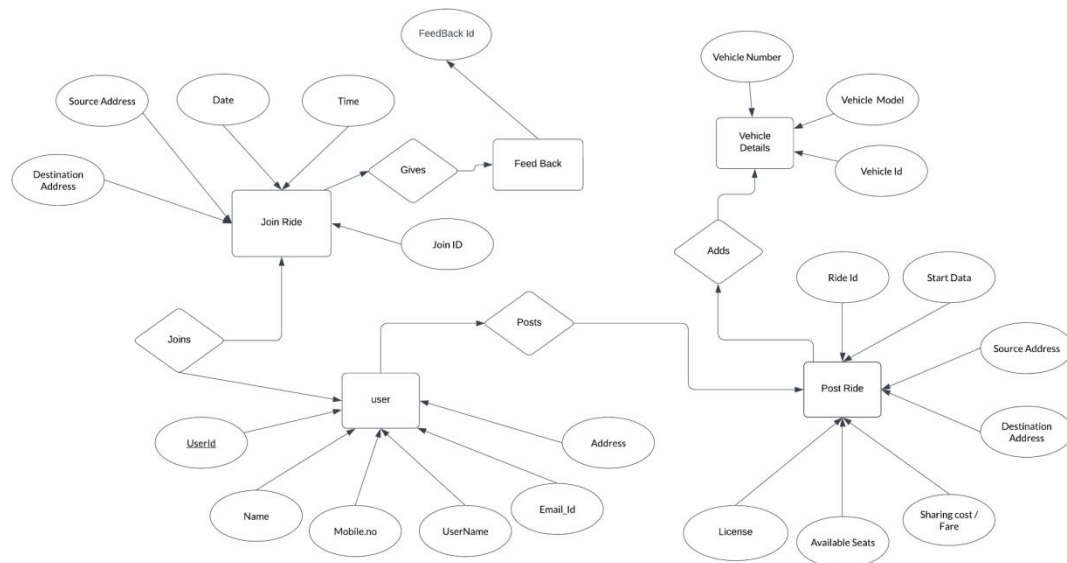
Driver

- Register/Login: The driver creates a new account or logs in to an existing account.
- Post Rides: The driver submits details about the rides they are offering, including capacity, source, destination, and fare.
- Manage Rides: The driver edits or cancels rides they have offered.
- Accept/Reject Rides: The driver chooses whether to accept or reject ride requests from passengers.
- View Ride History: The driver sees a record of the rides they have completed.

Admin

- Login/Register User (extends): The admin creates new user accounts or edits existing accounts (possibly for system maintenance).
- Manage User Management: The admin manages users and can potentially delete user accounts.

7.6 ER Diagram



The conceptual diagram you provided is similar to an Entity-Relationship Diagram (ERD) because it shows the high-level relationships between entities like users, rides, vehicles, and feedback. However, an ERD would typically use more specific terminology to depict entities, attributes, relationships, and cardinalities. For instance, the conceptual diagram shows "User" but an ERD might represent this as a table named "Users" with attributes like "UserID", "UserName", "Email", etc.

Entities

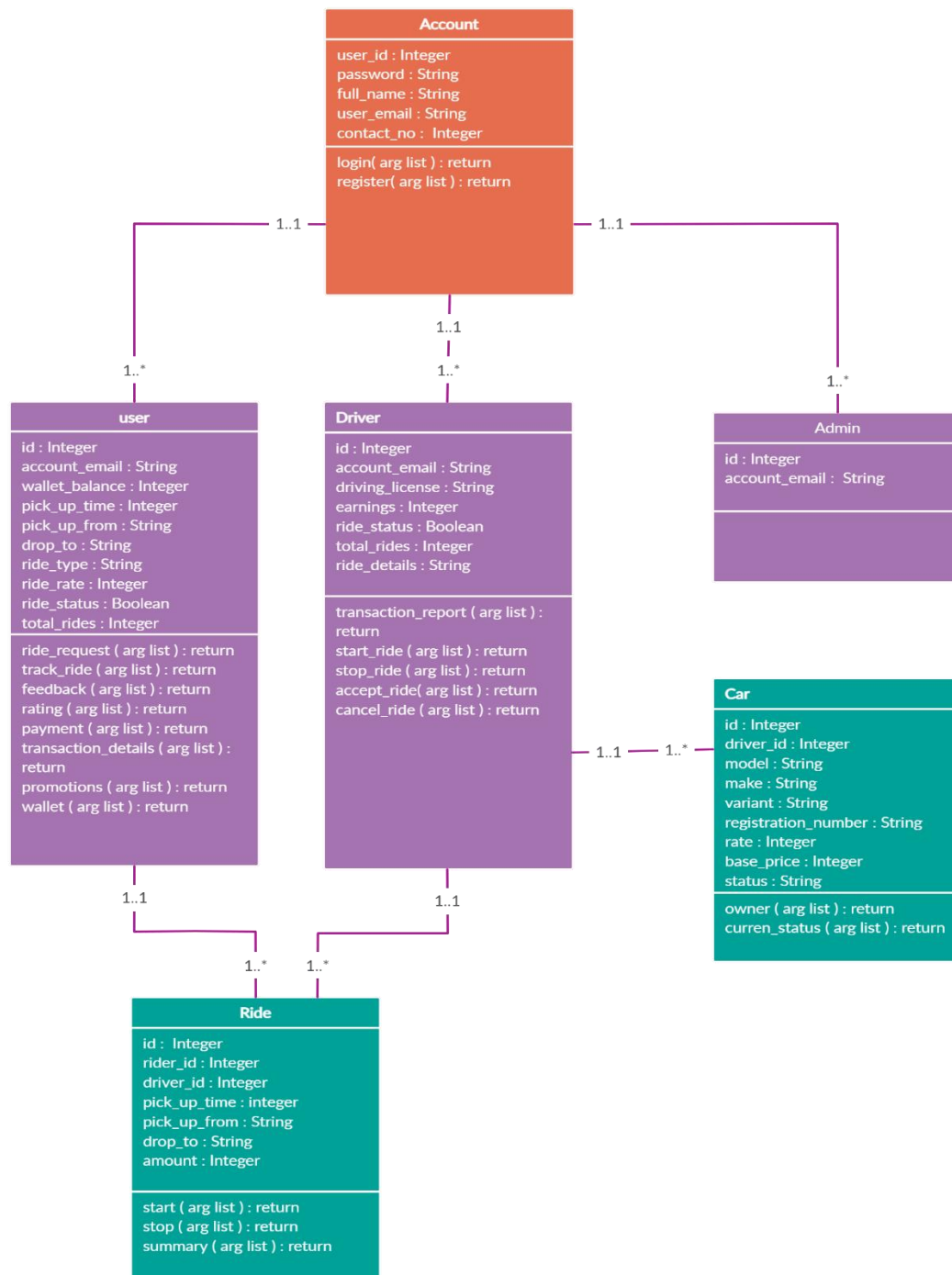
- **User**
- **Ride**
- **Vehicle**
- **Feedback**

Relationships

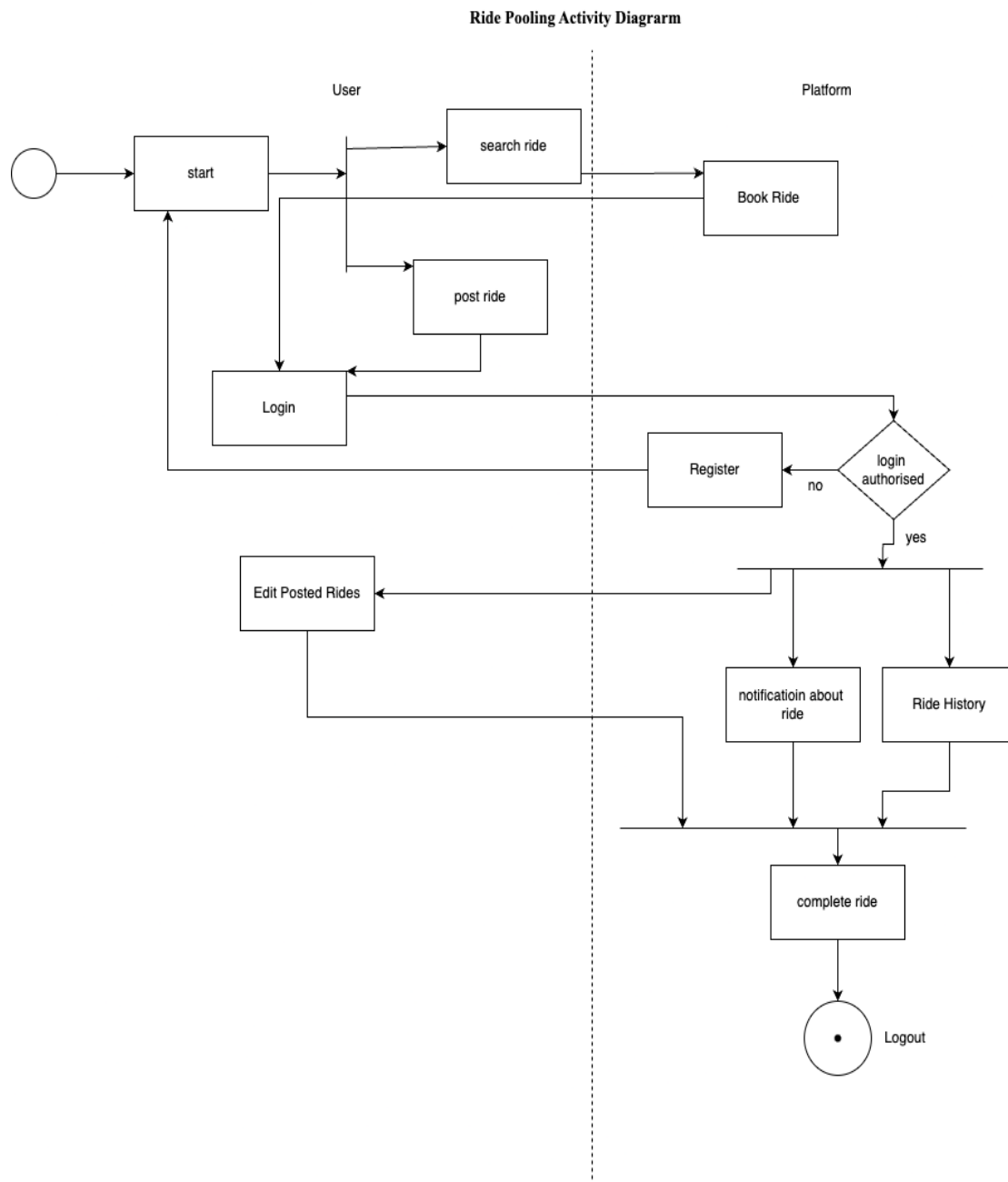
- **User:** A user can participate in many rides (one-to-many relationship). This is depicted by the line connecting "User" and "Join Ride" and another line connecting "Join Ride" and "Ride".

- **Ride:** A ride can involve one user (Passenger) and potentially multiple users (through "Join Ride") (one-to-many relationship). This is depicted by the line connecting "Ride" and "Join Ride".
- **Vehicle:** A ride is offered using one vehicle (one-to-one relationship). This is depicted by the line connecting "Ride" and "Vehicle Details".
- **Vehicle:** A vehicle can be used in many rides (one-to-many relationship). This is implied since multiple rides can connect to "Vehicle Details" through the "Posts" line.
- **User:** A user can provide feedback on one ride (one-to-one relationship). This is depicted by the line connecting "User" and "Feedback".
- **Ride:** A ride can have feedback from one user (one-to-one relationship). This is depicted by the line connecting "Ride" and "Feedback".

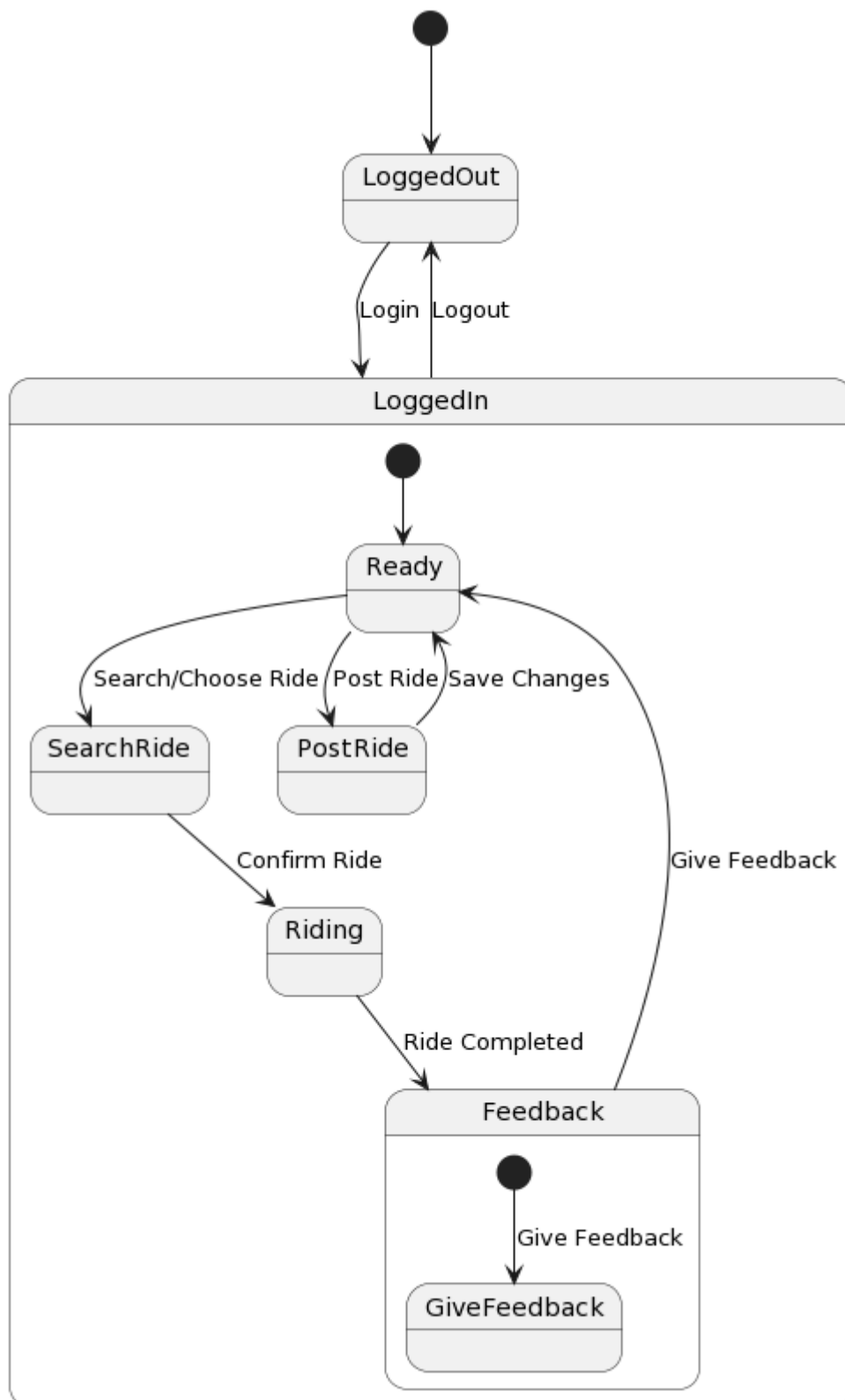
7.7 Class Diagram



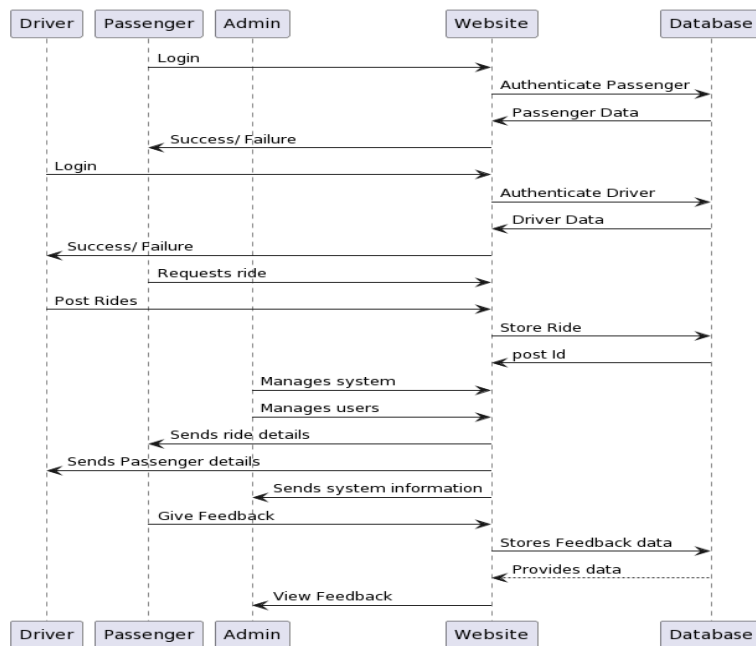
7.8 Activity Diagram



7.9 State chart Diagram



7.10 Collaboration Diagram



8. Testing

8.1 Functional Testing

Login Module:

Functional testing of the login modules ensures that users can successfully authenticate and access the Ride Pooling platform. The following test cases are executed to validate the functionality of the login modules:

- **Valid Credentials Test:**

Verify that users can log in successfully with valid username and password combinations.

- **Invalid Credentials Test:**

Confirm that users receive appropriate error messages when attempting to log in with invalid credentials.

- **Case Sensitivity Test:**

Test whether the system distinguishes between uppercase and lowercase characters in usernames and passwords during login.

- **Account Lockout Test:**

Validate that user accounts are locked out after a specified number of failed logins attempts to prevent unauthorized access.

- **Browser Compatibility:**

Test login functionality across different web browsers (e.g., Chrome, Firefox, Safari) to ensure consistency and compatibility.

Searching Module:

Test that users can input search queries for Rides based on criteria such as start location , end location, no of seats, date etc . Verify that the search functionality is prominently displayed and easily accessible from the user interface.

- **Search Filters:**

Validate that all filter options are functional and produce accurate results when applied individually or in combination.

- **Search Results:**

Confirm that search results are accurate and relevant, displaying matching Rides based on the input.

Error Handling:

Validate that appropriate error messages are displayed when users encounter issues such as invalid search queries, network errors, or database connectivity problems.

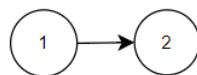
8.2 Path Testing

8.2.1 Signup ()

Pseudo code:

1. Enter User name, password, email
2. Submit the credentials

Control Flow Diagram:



Complexity: E-V+2

$$=1-2+2$$

$$=1$$

8.2.2 Login ()

Pseudo Code:

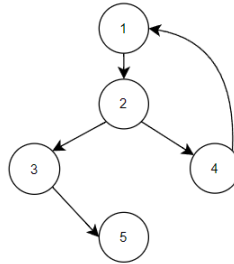
1. Enter loginID, Password
2. If (loginID == EnteredUserName && Password==EnteredPassword) then
3. print (“Login Successful”)
- 4.else

Print (“Login Failed”)

Goto 1

5.EndIf

Control Flow Diagram:



E: Number of Edges = 5

V: Number of Vertices = 5

Complexity: $E - V + 2$

$$= 5 - 5 + 2$$

$$= 2$$

9. Technology Stack

MERN Stack with React, Vite, and Express.js

Frontend (React with Vite):

- **React:** Provides the core library for building user interfaces with reusable components. It offers a declarative way to describe what you want to see on the screen.
- **Vite:** A development server and build tool for modern web applications. It leverages pre-bundling and hot module replacement (HMR) for a fast development experience. Vite excels in development speed and reduces the need for complex configuration compared to traditional bundlers like Webpack.

Backend (Express.js with Node.js):

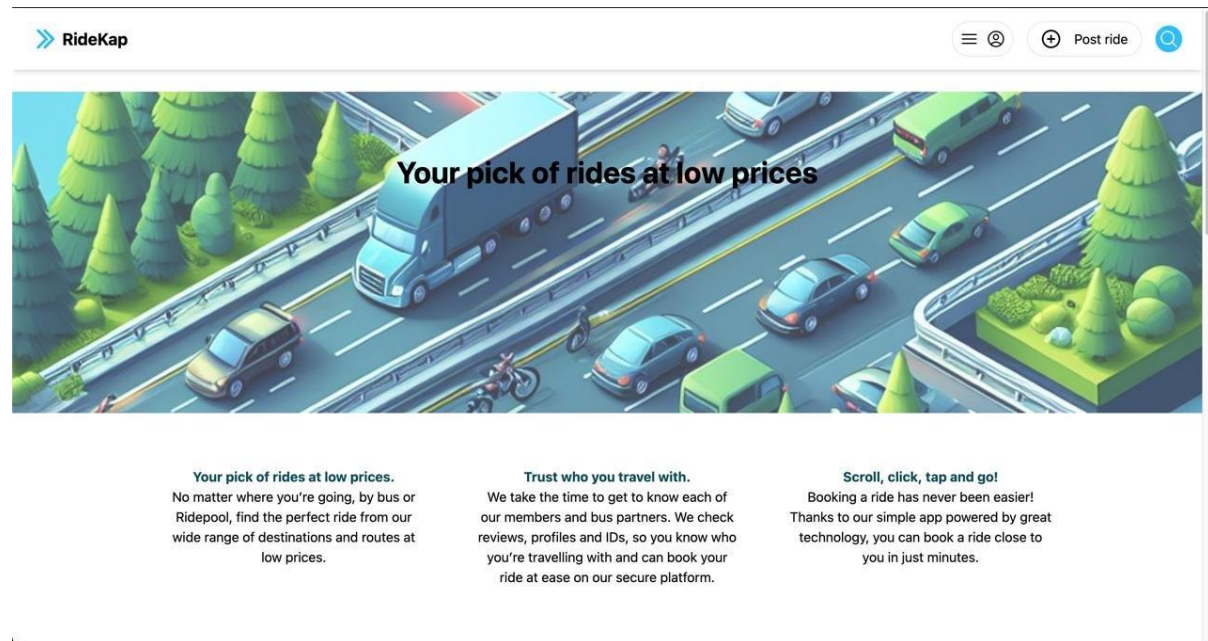
- **Express.js:** A flexible Node.js web framework for creating robust and scalable web applications and APIs. It simplifies server-side development by providing features like routing, middleware, and templating.
- **Node.js:** The runtime environment that executes JavaScript code outside of a web browser. It allows you to build server-side applications and APIs using JavaScript.

Database (MongoDB):

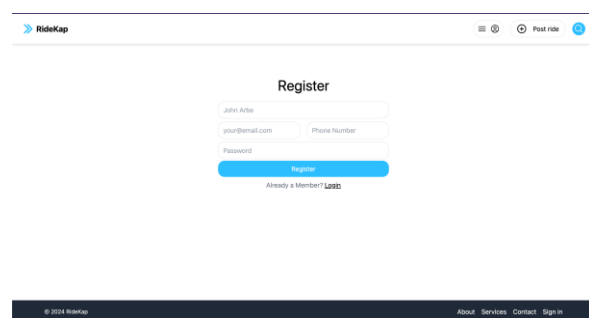
- **MongoDB:** A NoSQL document database that stores data in flexible JSON-like documents. It offers scalability and ease of use for storing and retrieving data for your application.

10. Results

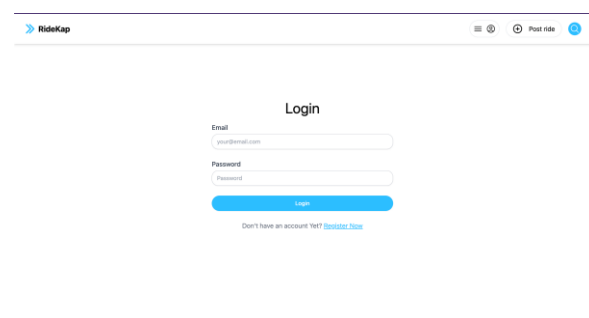
User Home Page - Before Login



Register Page:




Login Page




The above figures show the Register & Sign-In pages on User side.

User Login


Welcome, **chaitu**
Updated Profile Successfully

My Profile
My Joined Rides
My Posted Rides
All Rides



Full Name
 chaitu

Email
 chaitu@gmail.com


Mobile No.
 9848022332

EDIT PROFILE
CHANGE PASSWORD
LOGOUT

© 2024 RideKap
 About Services Contact

The users Login shows the details of the User who has logged in and can update his password and make edits in his profile.

Posted Rides:


Welcome, **teja**
teja
Post ride

Publish a Ride in Just Minutes

Create a Ride

From:
 To:

Available seats:
 Price:

Departure Time:

Arrival Time:

Car Details

 Ex: 2020 Silver Toyota Camry, license plate XYZ-1234

Publish Ride

Here you can post your rides that you are going to do.

Joined Rides :

Joined Rides

vijayawada to eluru

Date: 07/05/2024

Seats Booked: 1

₹200

Cancel Ride

Displays the joined rides to give feedback if the ride is done or cancel the ride if required.

Posted Rides:

RideKap

Welcome, teja
teja
Post ride
Search

My Profile
My Joined Rides
My Posted Rides
All Rides

Posted Rides

teja
₹1000
Edit Ride

Riding from vijayawada to guntur
Seats: 7
Departure Date: 2024-05-02 Time: 00:15:00
Arrival Date: 2024-05-02 Time: 20:45:00
Car: gtr

View Passengers
View Feedback

teja
₹3000
Edit Ride

Riding from kurnool to goa
Seats: 3
Departure Date: 2024-04-16 Time: 18:30:00
Arrival Date: 2024-04-25 Time: 18:30:00
Car: benz

View Passengers
View Feedback

© 2024 RideKap
About Services Contact

Here the user can check his Rides that he has posted and can edit them, view passengers and view feedback given by the user.

All Rides:

Posted Rides

CharishmaPothireddy

Riding from vijayawada to gudur

Seats: 1

Departure Date: 30/04/2024, 00:00:00

Arrival Date: 30/04/2024, 23:45:00

Car: BMW

Driver-email: charishma123@gmail.com

Driver-Phone No: 9392993951

₹2000

Ride Started

teja

Riding from vijayawada to guntur

Seats: 7

Departure Date: 02/05/2024, 05:45:00

Arrival Date: 03/05/2024, 02:15:00

Car: gtr

Driver-email: teja@gmail.com

Driver-Phone No: 9392993950

₹1000

Ride Started

To view and join the rides if needed.

Search Rides:

Leaving from

goa

dd/mm/yyyy

1

Search

Search Results

teja

Riding from kurnool to goa

Seats: 3

Departure Date: 2024-04-16 Time: 18:30:00

Arrival Date: 2024-04-25 Time: 18:30:00

Car: benz

Creator-email: teja@gmail.com

₹3000

This page shows the results of the rides the user has searched for according to his requirements

11. Conclusion

Ride sharing has emerged as a promising solution for the complex challenges of urban transportation. It offers a collaborative and effective alternative to traditional car/bike ownership, potentially benefiting passengers, drivers, cities, and the environment alike.

The power of car sharing lies in its ability to leverage technology and data. Ride-sharing platforms can utilize algorithms to optimize resource utilization. This translates to fewer cars on the road, leading to reduced traffic congestion and lower greenhouse gas emissions. Additionally, car sharing promotes resource efficiency by minimizing the number of idle vehicles, thereby lowering overall material and fuel consumption.

Beyond its environmental and resource-saving potential, car sharing fosters a sense of community. By connecting strangers on shared journeys, ride-sharing platforms create opportunities for social interaction and potentially new friendships.

However, car sharing is not without its challenges. Scaling services to accommodate a growing user base requires continuous platform adaptation and infrastructure development. Regulatory frameworks need to be established to address issues such as driver licensing, insurance coverage for ride-sharing activities, and vehicle safety standards.

12. Future Work

The Future of Ride Pooling: Innovation and Development

The future of ride pooling is brimming with exciting possibilities. Here's a breakdown of key areas where advancements can revolutionize the way we travel:

1. **Smarter Matching Algorithms:** Imagine getting matched with rides that perfectly align with your needs! Advanced algorithms will consider real-time traffic, your preferences (detour for coffee?), and even multi-modal options (train for part, then pool the rest). This translates to optimized matching, shorter travel times, and a more pleasant ride.
2. **Dynamic Pricing on Demand:** No more price surprises! Dynamic pricing models will adjust fares based on real-time supply and demand. Think surge pricing during rush hour, or lower fares in areas with fewer riders. This ensures affordability for riders while optimizing driver earnings, resulting in a win-win situation.
3. **Seamless Integration with Public Transit:** Imagine a world where your ride-pooling app seamlessly connects you to public transportation! Integration with buses and trains will provide you with first-mile/last-mile connectivity. This means coordinated schedules, combined fares, and optimized transfer points, leading to a smoother, multi-modal travel experience.
4. **Electric and Autonomous Vehicles:** Get ready for eco-friendly and driverless rides! The future involves electric vehicles reducing carbon emissions and operating costs. Autonomous vehicles, while still under development, hold the potential to further increase safety and efficiency by eliminating the need for human drivers.
5. **Accessibility for All:** Ride pooling shouldn't be a privilege! The future envisions services that cater to everyone. This includes providing wheelchair-accessible vehicles, offering subsidies for low-income riders, and implementing multilingual support. Everyone deserves a safe and reliable ride.
6. **Data-Driven Optimization:** Data is king! By leveraging data analytics and predictive modeling, ride-pooling companies can analyze user behavior, anticipate peak demand periods, and optimize vehicle deployment. This translates to improved service reliability, shorter wait times, and a more responsive system overall.

13. Reference

- <https://www.blablacar.in/>
- <https://react.dev/>
- <https://nodejs.org/docs/latest/api/>
- <https://fkhadra.github.io/react-toastify/introduction/>
- <https://expressjs.com/>
- <https://www.mongodb.com/docs/>
- <https://vitejs.dev/guide/>
- https://github.com/AnanthaTeja/Ride_kap_pooling