1.1 Introduction

In recent years, the rise of ride-sharing systems has revolutionized the way people commute and travel. A ride-sharing system is a modern and efficient approach to transportation, designed to connect passengers with drivers using technology platforms. By leveraging the power of smartphones, GPS tracking, and online payment systems, ride-sharing has become a convenient and cost-effective alternative to traditional taxi services and public transportation.

The concept is simple: passengers can request a ride through a dedicated mobile application, specifying their current location and desired destination. Nearby drivers, who are registered and approved by the ride-sharing company, receive the request and have the option to accept or decline it based on their availability and location. Once a driver accepts the request, the passenger can track the driver's location in real-time and receive an estimated time of arrival.

Ride-sharing systems have several advantages over traditional transportation methods. They offer greater flexibility and convenience, as passengers can request a ride at any time and from virtually any location with internet access. The system's efficiency minimizes waiting times, making it a more time-saving option for daily commuters and travelers alike.

Moreover, ride-sharing promotes a sense of community and shared responsibility as passengers often share rides with others heading in the same direction, reducing the number of vehicles on the road and decreasing traffic congestion and environmental impact.

Safety is a significant focus in the ride-sharing industry. Companies implement strict background checks and safety measures to ensure that both passengers and drivers feel secure during their journeys. Additionally, users can rate their experience after each ride, contributing to a transparent and accountable community.

1.2 Purpose

The basic purpose of a ride-sharing system is to provide a convenient and efficient means of transportation for individuals by connecting riders with drivers through a digital platform or mobile application. This system aims to address various transportation needs and improve the overall commuting experience for users. The key purposes of a ride-sharing system include:

- Accessibility: Enhancing accessibility to transportation options by connecting riders with a pool of available drivers, making it easier for users to book rides at their preferred time and location.

- Cost-effective: Offering a cost-effective alternative to traditional taxi services by optimizing routes, allowing riders to share the cost of the trip with others heading in a similar direction.

- Reduction of traffic congestion: By encouraging ride-sharing and pooling, the system aims to reduce the number of vehicles on the road, contributing to alleviating traffic congestion in urban areas.

- Environmental impact: Promoting a greener approach to transportation by maximizing the occupancy of vehicles, resulting in a reduction of carbon emissions and overall environmental footprint.
- 5. Seamless user experience: Providing a user-friendly, intuitive mobile application or web platform that allows riders to easily request rides, view estimated fares, track their drivers in real-time, and make cashless payments.

- Safety and reliability: Implementing safety measures such as driver background checks, vehicle inspections, and rating systems to ensure a reliable and secure experience for both riders and drivers.

- Flexibility for drivers: Offering a flexible earning opportunity for drivers, enabling them to work on their own schedules and utilize their personal vehicles to earn extra income.

- 8. Integration with public transit: Integrating with public transit systems to complement existing transportation options and provide seamless multimodal travel for users.

1.3 Scopes

- The scope of a ride-sharing system encompasses all the features, functionalities, and services that the system will provide to users and stakeholders. Here are the key aspects of the scope for a ride-sharing system:

- User Registration and Authentication: The system should allow users to register and create accounts using their email, phone number, or social media profiles. Proper authentication mechanisms should be implemented to ensure the security of user data.

- Ride Request and Matching: Users should be able to request rides through the mobile app or website, specifying their pickup and drop-off locations. The system should match riders with nearby drivers based on various factors such as location, distance, and driver availability.

- Driver Registration and Verification: Drivers interested in providing rides should be able to sign up and register with the platform. The system should verify their identity, driving license, and car documents to ensure safety and compliance.
- Real-Time Tracking: Users should be able to track the location of the assigned driver in real-time on the app, providing transparency and enhancing safety.

- Fare Calculation: The system should calculate the fare for each ride based on factors like distance, time, and surge pricing (if applicable). The fare should be displayed to the user before they confirm the ride.
- Payment Gateway Integration: The ride-sharing system should integrate with a secure payment gateway to process cashless payments. Users should be able to pay through credit/debit cards, digital wallets, or other payment methods.
- Reviews and Ratings: After each ride, users and drivers should have the option to rate and review each other, helping to maintain accountability and improve the overall quality of the service.
- Customer Support: The system should provide a customer support mechanism to address any issues, complaints, or queries raised by users or drivers.

- Multi-Platform Support: The ride-sharing system should be accessible across various platforms, including iOS, Android, and web browsers, to cater to a wide range of users.

- Trip History: Users and drivers should have access to their trip history, enabling them to review past rides and transactions.

- Security and Privacy: The system should implement robust security measures to protect user data and transactions from unauthorized access or breaches.
- Admin Panel: The system should include an administrative panel to manage users, drivers, and oversee system operations. Admins should have the ability to review and moderate user-generated content, resolve disputes, and handle system settings.
- Geographical Coverage: The scope should define the regions and cities where the ride-sharing service will be available.
- 14. Legal and Regulatory Compliance: The system should adhere to local transportation regulations, licensing requirements, and tax laws in the regions it operates.

It's important to note that the scope can be expanded or customized based on specific business requirements and market needs. However, defining a clear scope is crucial to ensure the successful development and implementation of the ride-sharing system.

1.4 Document Conventions:

Particulars	Font Types	Font Size
Main Heading		
Sub-Heading		
Body		

1.5 References:

- http://www.chat.openai.com
- http://www.carpool.in/etiecatte.html
- http://www.vdocuments.mx
- http://www.wikipedia.org/carpool

2.Overall Description

2.1 Product Perspective

- Market Positioning: The ride-sharing app should define its unique selling proposition (USP) to differentiate itself from competitors in the market. Whether it focuses on affordability, reliability, convenience, or other factors, the positioning should resonate with the target audience.

- User-Centric Design: The app should prioritize user experience and be designed with an intuitive interface to ensure ease of use for both riders and drivers. Features like clear navigation, minimal steps for ride booking, and real-time tracking contribute to a positive user experience.

- Multi-Platform Accessibility: To cater to a broader audience, the app should be available on multiple platforms, such as iOS and Android, and have a responsive web version for users who prefer to book rides from their desktops.

- Seamless Integration: The ride-sharing app may integrate with other popular services, such as mapping and navigation tools, to enhance the overall user experience and ensure efficient route optimization.

- Real-Time Updates: The app should provide real-time updates to users and drivers regarding ride status, estimated arrival times, and any changes to the trip, ensuring transparency and reducing uncertainty.

- Driver-Centric Features: To attract and retain drivers, the app should offer features such as driver earnings visibility, flexible working hours, and efficient trip allocation algorithms to optimize their earnings.

- Scalability: The app's architecture should be designed to handle a large volume of users and rides, especially during peak hours or in regions with high demand.

- Data Analytics and Insights: The app can leverage data analytics to gather insights into user behavior, driver performance, popular routes, and other patterns, helping in decision-making and improving the overall service.

- Partnerships and Collaborations: The ride-sharing app might consider forming partnerships with local businesses, events, or public transportation services to provide additional benefits to users and promote its service.

- Regulatory Compliance: The app must comply with local transportation regulations, ensuring that drivers and vehicles meet the required standards and safety measures.

- Marketing and Promotion: Effective marketing strategies should be employed to attract new users and drivers, including referral programs, promotional offers, and targeted advertising.

- Continuous Improvement: The app should have a roadmap for continuous improvement, with regular updates, bug fixes, and feature enhancements based on user feedback and changing market dynamics.

- By considering these aspects from the product perspective, a ride-sharing app can deliver a comprehensive and user-friendly experience to both riders and drivers, ultimately contributing to its success and growth in the competitive transportation industry.

2.2 Product Functions:

The goride functions will support the following functionality:

- Login – logging onto the web server
- Register - Register in the goride
- goride search – search other members participating in the ride
- Create goride – create carpooling for special events
- Join goride – join others ride.
- View goride detail – view detail of the one-time event schedules.
- Delete goride detail - delete own created ride

2.3 User Classes and characteristics

.

Participant	priority
User	Normal
Guest	Medium
Admin	High

Admin: He will check the information given by the guest user & provide him user-id and password. He has to maintain the priorities given by the car owners regarding the person for which they are comfortable in journey. He can provide new password if user wants.

User: The guest can see the schedule to different places with search option by providing the corresponding date and time and he can register on the website by providing the required details. But he cannot join the carpool and create carpool without getting registered

2.4 Operating Environment:

- The computer must be linked up to a server loaded with the Database. The users will require for peripherals a mouse and a keyboard.
- It should be compatible with all the well known and widely used web browsers like Internet Explorer, Mozilla Firefox, Google Chrome, and Opera.

2.5 Design and Implementation Constraints

- Its front end should be implemented using PHP, HTML and Java-script.
- The back end should be managed by the MYSQL.
- It will be using the Andromeda server provided by NITC.
- The Backup and recovery functionalities will be constrained by the servers and the data.

- Storage facilities provided by Andromeda.

2.6 User Documentation

- This website will be a user – friendly and created in such way that the person having little or no knowledge of the website can use this; therefore no user manual will be required.
- Other queries will be solves in FAQ'S.

2.7 Assumptions and Dependencies

It is assumed that the user should familiar with computer &having internet connection in the system. • It is assumed that all information given by the user is correct regarding personal & scheduling information. • Our system mainly depends on the users & highly affected when the information given for pooling is not followed.

3. External Interface Requirements

3.1 User Interfaces

On the main page, the guest can check the carpool schedule and will be given an option to login or register as a new user. The registered user will be a given the option to edit their profile, perform a carpool search, create a car pool, view others carpools, join car pool, and delete his own carpool. While creating carpools, users can establish their own preferred origin and end destinations places along with his own choice of partners he wants in his ride sharing journey. When joining the car-pool we can join according to the description given by the car owner i.e. after matching source-destination places, timings & other preferences.

3.2 Hardware Interfaces

- The user should have end systems (computers or laptops).
- The hardware interfaces (such as network connectivity) will be managed by Internet Service Provider.

3.3 Software Interfaces

- The system is not specific for any particular operating system.

- We will be using HTML, Java script and PHP to build the front end of our Database.
- Mainly consist of usernames, passwords, emails, address, and schedule of car-pooling.
- We will use MYSQL to manage the database on our server.
- PHP will be used to connect to MYSQL.

3.4 Communications Interfaces

- HTTP protocol is used for transferring data between the server and the client.
- We will be using HTTP for establishing connection between user and database.
- We will be using PHP along with SMTP authentication to send and receive Email.

4. **System Features**

1.1.1 Description and Priority

Register option is provided on the home page of the website. Guest visiting the website can register them self by clicking on the register button after this user redirected to the register page. A register user can only join and create carpool. It’s priority 7.

1.1.2 Stimulus/Response Sequences

When the user is redirected to the register page he has to provide the information to fill the required register form. The information given by the user will be check from the database and if the information found correct a will be displayed successfully register. Error can occur if the user already registered. Or the information given by him isn’t matched with the database.

4.1 Search

4.2.1 Description and Priority Search option is displayed in the home page on the browse. Any guest (Not- registered) can also access the search option. It’s directly connected with the database. Its priority is 6.

4.2.2 Stimulus/Response Sequences User can search the site by giving his required origin and destination places. User can give only those places which are given drop down menu in the search option. After selection of places the user has to click on the search button. After this options of carpooling will be displayed on the screen.

4.3 Login

4.3.1 Description and Priority This option will on the home page of the website. It allows the user who have been registered before to get access to his own profile. User can also request for the new password in case he forgot the password. It’s priority 9

4.3.2 Stimulus/Response Sequences

The user has to provide the user name given to him at the time he registered and the last updated password to login to access his profile after clicking on the login button. To get a new password user has to provide the username/email id. Error can occur if the user name and password didn’t match, and the user will be redirected to login home page.

4.4 Create Car-pool

4.4.1 Description and Priority Only registered user and having car can create carpool . he has to provide the information regarding the origin and destination place along with the timing. Its priority is 8.

4.4.2 Stimulus/Response Sequences

For creating a car pool the user is redirected to the create carpool page, there he has to provide the information to fill the required form like origin and destination places, timing. His description, etc. after this a if the carpool created then message will be displayed.

4.4.3 Functional Requirements

REQ - 1:- Name TYPE: - String
REQ - 2:- Description TYPE: - String
REQ - 3:- Seat TYPE: - Numeric
REQ - 4:- Origin TYPE: - String
REQ - 5:- Destination TYPE: - String
REQ - 6:- Start timing TYPE: - Time
REQ - 7:- End timing TYPE:- Time
REQ - 8:- Vehicle no. TYPE: -Alphanumeric

4.5 Join Car-pool

4.5.1 Description and Priority

Only registered user can join the car pool. He has to provide the information regarding the origin and destination place in the search box. If the user has already joined car pools an option will be provided to leave that car pool. Its priority is 8.

4.5.2 Stimulus/Response Sequences

After giving origin and destination place in join pool search box, the user is redirected to the join carpool page, where the search result is displayed according to the information given in the search box. He can join any of the carpool showing in the search result by clicking the join button, according to the time and description for the journey given by the person who created the carpool. After clicking on the leave button the user is no more member of that car pool.

4.6 View Car-Pool

4.6.1 Description and Priority

When the user click on the view button given on his profile page the user is redirected to the page where he can see the pools created by him and joined by him. On going to the page showing the pools created by him, a option is provided for him to Delete the car pool. On going to the page showing the pools joined by him, a option is provided for him to leave the pool. Its priority is 8.

4.6.2 Stimulus/Response Sequences

In the page showing the carpools created by the user. The user can delete that carpool if he desired so. All member who are already joined that pool are being automatically deleted. In the page showing the carpools joined by the user. The user can leave the pool.

5.0 Other Nonfunctional Requirements

5.1 Performance Requirements

A proper internet connection is needed for the users using this website and the user should be user friendly with computer and the user interface of the website should be easy

5.2 Safety Requirements

If some unauthorized person get access to the site he can damage the site, therefore, the system shouldn't allow the user to access, until he provides correct username and password.

5.3 Security Requirements

- Only administrator has the access to update and delete the database.
- Guest user can only search for carpool to join or delete a car pool he should be registered.
- The system shouldn't allow the user to access, until he provides correct username and password.
- If the user request for a new password the password will be send to his mail id.

5.4 Software Quality Attributes

- A healthy internet connection having a good speed is to be use to get a better response time.
- This website is available 24*7.
- This website site is easy to use; it is being made, keeping in mind that the use has a little knowledge of the computer and website.

5.5 Business Rules

The user who created carpool has complete access over the car pool and on the members joining the carpool. He has the contact detail of all the persons who often join his carpool. In case the website is crashed he can directly contact with them.

6 Other Requirements

- Licensing Requirements - The usage of CAR-POOL MANAGEMENT is restricted to only who has the license.

- The system will try to retain the integrity of data • Access Reliability - The system shall provide 100% access reliability.