Software Requirements Specification (SRS)

Project Title: RideGo

A simple way to community-based Rideshare.

Document Version: 1.0

Date: 03-01-2023

Prepared By:

Md Rokibul Hasan (011193040)

Md. Mansurul Haque (011202034)

Md Rajib Hossen (011191244)

Emam Hasan (011201302)

Muminul Islam (011202115)

Prepared To:

Dr. Suman Ahmmed

Asst. Professor, CSE Director

Center for Development of IT Professionals (CDIP)

United International University

Short Description

RideGO is a straightforward web application that allows student to share a ride from one location to another for a reasonable fee.

Research Papers

The Ride-Sharing Services in Bangladesh: Current Status, Prospects, and Challenges

Md. Nazmus Sakib*   Md. Hasan Mia
Department of Management (MBA in SIM), University of Dhaka, Bangladesh

Figure 5.6: Improvement and suggestions for better services.

In contrast to the public transportation the ride-sharing services usually cots twice or thrice as such the users consider that the price should be reduced to make it more popular and affordable for all. Furthermore, the drivers need to be more professional, to quality, considering customer feedback and handling the compliance more effectively. The decision-maker should extend the service to other cities also. Additionally, the apps and servers need to be strengthened to cover and ensure quality services within short.

1.1 From businesses perspective

1.1.1 Challenges of ride-sharing services:

The following challenges are revealed from the viewpoint of decision-makers and executives of various ride-sharing organizations including Uber, Pathao, Obhai, Obon, Amarbike, Texiwala, Gariwala, Sohoz Ride, Chalu, Shareamotorcycle.com, etc.

A Ride Sharing System for University Community

Benjamin Kommey*a1, Benjamin Asane Asanteb1, Elvis Tamakloec1
[1]Department of Computer Engineering, Kwame Nkrumah University of Science and Technology, 00233 Kumasi, Ghana
*abkommey.coe@knust.edu.gh; bcalculusaffairs@gmail.com; ctamakloe.elvis@gmail.com

Travelling, which involves moving or transporting from one place to another, has become an integral part of our lives as human beings. In our modern day, it has become impossible for one to stay for a long time without having to move from one place to another. People may move from one place to another for the purposes of work, school, trade, vacation, family reunion, just to name a few. With the evolution in the transportation sector, people can now rely on very powerful and sophisticated vehicles such as airplane, trains, cars, ships, and bicycles to get to their destinations faster and easier. In Ghana, the most common means of transportation is by car. Most people personally own cars and transport companies also offer services by providing long buses to people. For long distance travels, most people who do not have cars have to move from their homes and carry their luggage to the various stations to book seats. Car owners, who mostly drive with free seats available, also must incur huge fuel and toll costs alone. Huge vehicular traffic is caused on the roads since every car owner uses his own car with empty seats. The rate of carbon emissions into the atmosphere is also increased in the process. Finally, time which could have been used for productive work is wasted as commuters spend more hours on the roads than necessary.

A systematic literature review of ride-sharing platforms, user factors and barriers

Lambros Mitropoulos*, Annie Kortsari and Georgia Ayfantopoulou

Abstract

Aim: Ride-sharing is an innovative on-demand transport service that aims to promote sustainable transport, reduce car utilization, increase vehicle occupancy and public transport ridership. By reviewing ride-sharing studies around the world, this paper aims to map major aspects of ride-sharing, including online platforms, user factors and barriers that affect ride-sharing services, and extract useful insights regarding their successful implementation.

A strong relation was found between having ride-sharers among family/friends and colleagues, and engaging in ride-sharing [14, 33]. The tendency to adopt ride-sharing services is higher for multi-person households and households having more licensed drivers than vehicles [58]. The presence of children, elderly persons, or both, in the household is likely to have a negative effect on the adoption and frequency of use.

Online Articles

WHY IS RIDESHARING IMPORTANT?

Sharing a ride has numerous benefits such as reducing traffic congestion and parking demands. Ridesharing also helps to eliminate vehicle emissions and creates less stressful commutes.

Here are some of the benefits to ridesharing:

1. **Reduced carbon footprint** – Ridesharing significantly reduces the amount of emissions in the air. If you rideshare to and from work or school, you could cut your yearly emissions number in half, or even more.
2. **Use of HOV lanes** – The HOV lanes in the area are usually less congested than the other highway or parkway lanes, which means your commute time could be greatly reduced if you rideshare.
3. **More free time** – Riding in a carpool, vanpool or on transit allows you to check email, browse the Internet, catch up on social media, read a book or even take a quick nap.
4. **Saving money** – By ridesharing you can save money on parking and gas.
5. **Reduced traffic congestion** – Ridesharing takes vehicles off the road, which means less congested roads. As more people take part in ridesharing, overall traffic congestion could decrease.
6. **Decreased stress** – Not having to fight traffic and saving money on gas and parking can significantly decrease the amount of daily stress you have.
7. **Less wear and tear on your car** – By ridesharing, you put less mileage on your car therefore decreasing wear and tear each month.
8. **Employer benefits** – Alleviate parking shortages, boost employee recruitment and help the company be more "green" by promoting ridesharing among employees.

Benchmark Analysis

<u>UBER:</u>
Driver-Oriented Features:
- A passenger-rating system to identify difficult riders (and remove them from the platform if necessary)
- Special accommodations for hearing-impaired drivers
- Bonuses in certain cities based on the number of rides given
- Drivers can opt-out of UberPool, which can make for a better earnings strategy for drivers who aren't trying to get a bonus by giving more rides

Rider-Oriented Features:

- Scheduling rides in advance
- Opportunity to pay in cash in certain parts of the world where credit cards aren't as common (great for international expansion)
- Ability to split a fare between friends sharing a ride
- Multiple drop-off points for groups of riders

<u>lyft:</u>

Driver-Oriented Features:
- A separate app for drivers that includes driver-specific features
- Ability to mark the current ride as the last ride for the day so drivers don't get an unexpected request they feel compelled to accept
- Opportunity to earn tips from riders (Uber does this as well, but Lyft pioneered it)

Rider-Oriented Features:
- Shared-saver rides for discounts
- "One tap to ride" option for quick requests and pickups
- Drivers have to pass through background checks to ensure safety
- "Lyft Amp" device—a glowing Lyft sign—helps riders find their drivers at night

<u>pathao:</u>

Driver-Oriented Features:
- Drivers get paid daily
- 24/7 live support
- Complete flexibility to work when you want to work

Rider-Oriented Features:
- A variety of options for travelers
- "One tap to ride" option for quick requests and pickups

Comparison Chart

Product/Features	GPS and Location Determination	Booking Rides Ahead of Time	Cancellation Policies and Methods	Searching for a Driver	Receiving Real-time Notification	Only University/ College/ School Service	Extra Charge	Live Chat
Uber	Yes	Yes	Yes	Yes	Yes	No	Yes	No
Pathao	Yes	No	Yes	Yes	Yes	No	Yes	No
MUV	Yes	No	Yes	Yes	Yes	No	Yes	No
O Bhai	Yes	No	Yes	Yes	Yes	No	Yes	No
Shohoz	Yes	No	Yes	Yes	Yes	No	Yes	No
RideGo	Yes	Yes	Yes	No	Yes	Yes	No	Yes

Gap Analysis

We have read a bunch of articles related to our project . We found some information and gaps among them.
1. There is no known platform for Student only Ride Sharing Service.
2. Some of the service are more costly.
3. There is no such type of "Live Chat" option with all user.

Features List
Those are
-Student Verification.

-Auto Suggestion Searching.

-Live chat support with all user.

-Rate/Feedback.

-Live Location Sharing

-Pick Up Point

-Booking Rides Ahead of Time

-Online Payment.

StakeHolders
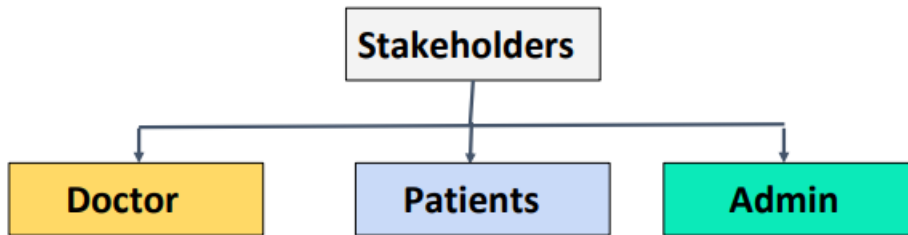
Rider Features

Those are –

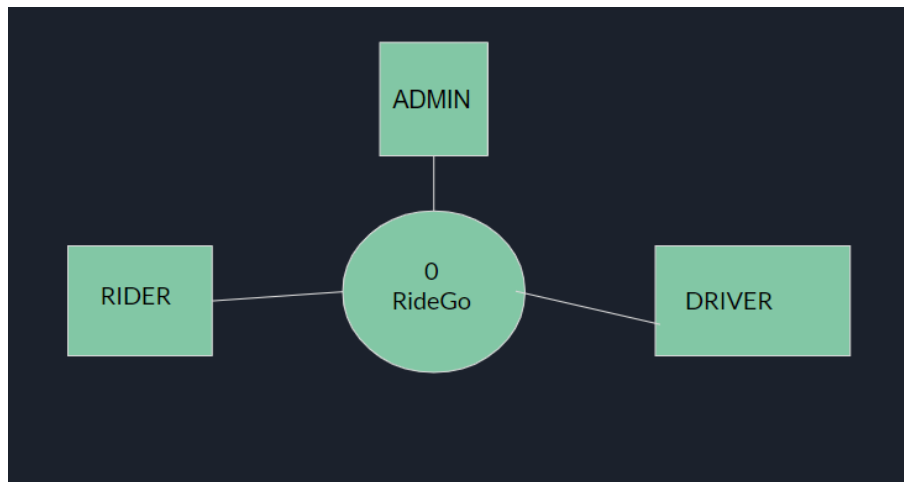1.   Student Verification.
2.   Auto Suggestion Searching.
3.   Live chat support with all user.
4.   Rate/Feedback.
5.   Live Location Sharing.
6.   Booking Rides Ahead of Time
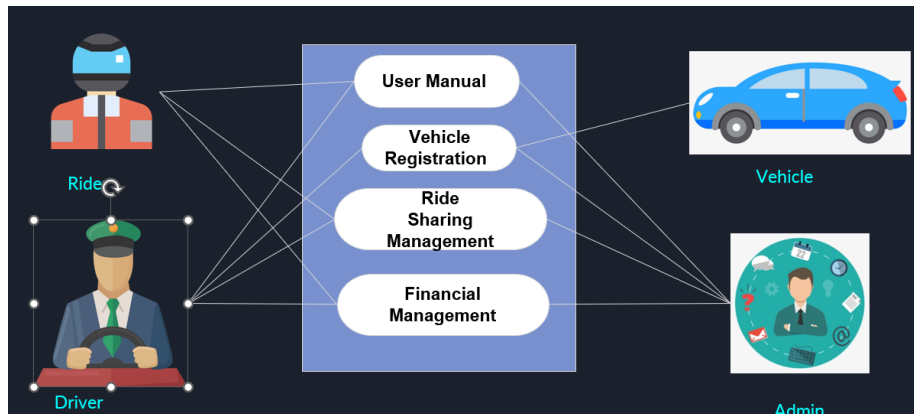7.   Online Payment.

Driver Features

Those are –

1.   Live Location Sharing.
2.   Pick Up Point.
3.   Able to cancel any Rider.
4.   Can choose any Payment method.

Context Diagram

Use Case Diagram:

Use Case Descriptive Form:

Use Case 01: Ride Sharing Management

<u>Use case 01:</u>

Ride Sharing Management

<u>Primary Actor:</u>

Rider

<u>Stakeholders and interests:</u>

Rider:

　　　Wants to successfully book a ride depending on his/her requirements.

Driver:

Client is encouraged to book a ride through this website and to select his or her vehicle.

Admin:

Oversees all the rental bookings.

Need to make sure that journey become safe and enjoyable both for driver and client.

Pre-Condition:

- The driver needs to give details about the vehicle so that the client can choose his/her required ride.
- To confirm a reservation, an advance fee must be paid.

Post-Condition:

- The rental details remains saved when Rider or driver is logged out.
- The changes are made in the database.
- Rider can see the booked vehicle in their account.
- Rider and driver both can cancel the booking at any time and the admin can authorize the cancellation.
- The reservation fee is non-refundable.

Main success Scenario:

- Client goes to the web site.
- Customer creates a new account or logs into their old account using their username and password.
- Client navigates to the available vehicle page from the home page.
- Clients find the destination and chooses a vehicle of his/her choice and clicks the book button.

Use Case 02: Financial Management

Use case 02:

- Financial Management

Primary Actor:

- Driver

Stakeholders and interests:

1. Rider:
   - Wants to successfully pay the payment.
2. Driver:
   - Wants to get the due payment he deserve.

3. Admin:
    - Oversees all the payment process.
    - Need to make sure that there is all kind of option Rider and Driver needed.

        Pre-Condition:
        - The Driver reached the destination.
        - There will be some option to chose for the rider to pay.
        - To confirm the payment.

    Post-Condition:
        - The payment details remains saved when Rider or driver is logged out.
        - The changes are made in the database.
        - Rider can communicate with the drive about payment.
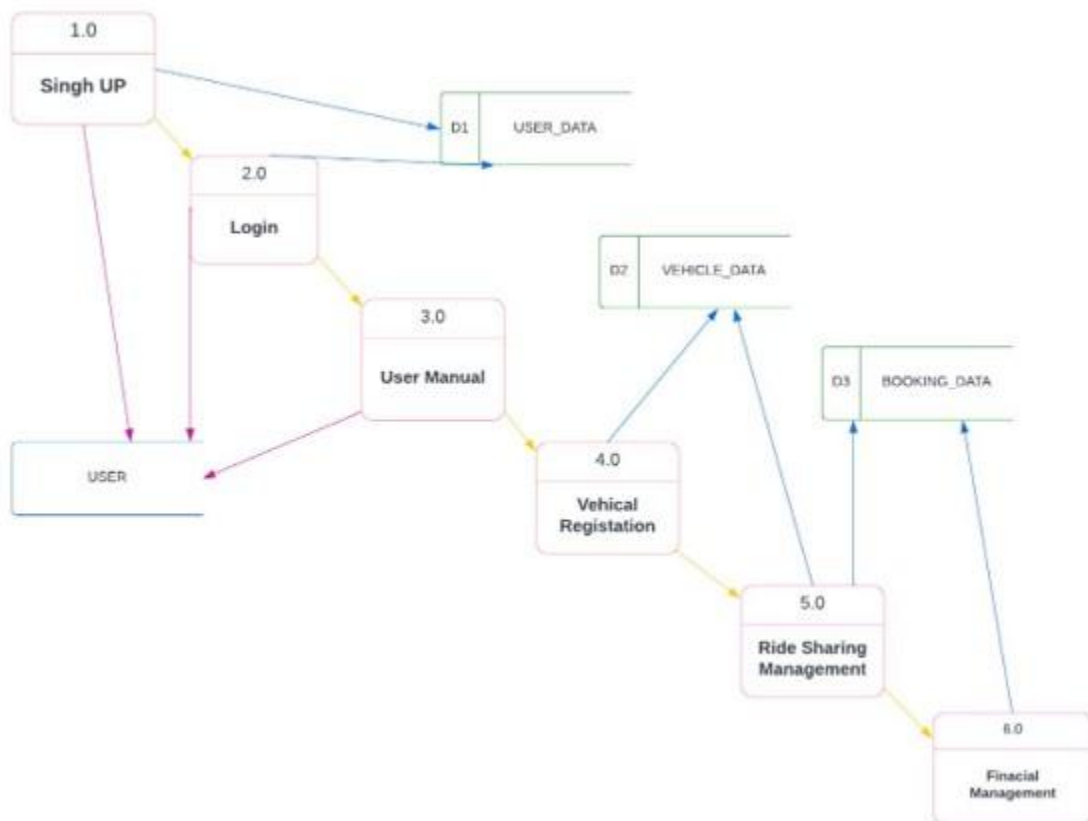   - The reservation fee is non-refundable.

Main success Scenario:

- The Rider book a ride.

- Driver reach the reachable destination where Rider want to go.

- The Rider will pay dues to the Drive.

- If online payment not working then Rider can pay hand cash.

Activity Diagram

Swimlane Diagram:
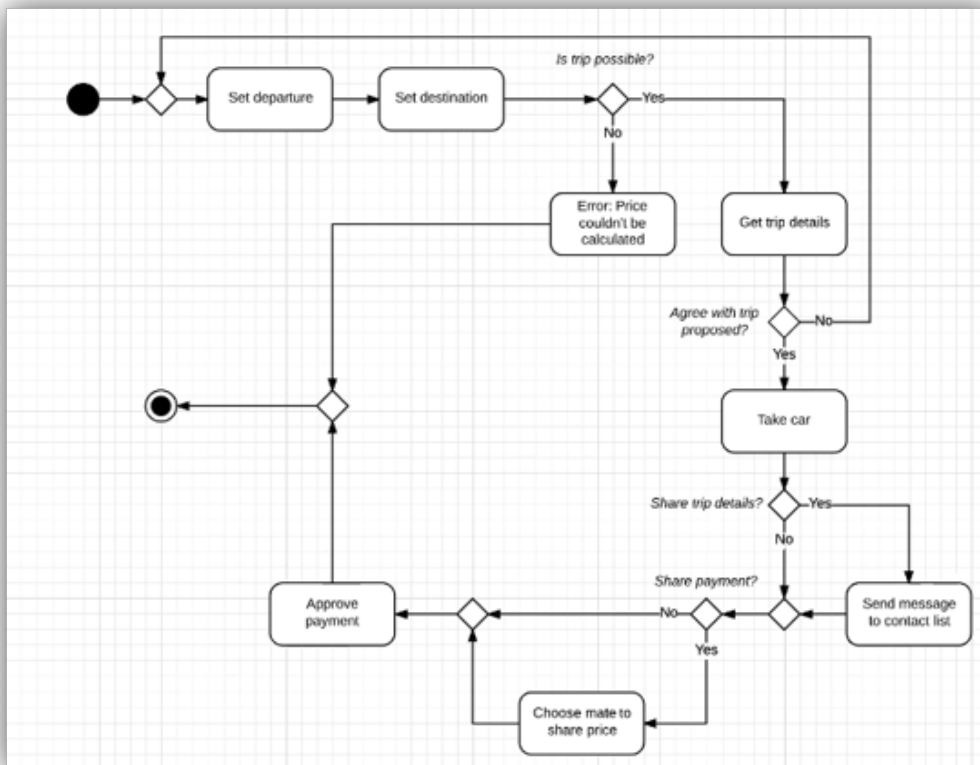
Data Flow Diagram

Class Diagram:

What is Class Diagram?

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

What is Class:

Class are a blueprint or a set of instructions to build a specific type of object. It is a basic concept of Object-Oriented Programming.

How we can identify who are the candidates as a class?

To identify the candidates as a class it needs to be physical objects, or a group of objects that are tangible and devices with which the application interacts. examples- books, cars, pressure sensors etc.

What is attribute?

Attribute represents the structure or the information of a class. It can be identified by any of these three ways

- Examining class definition
- Studying requirements
- Applying domain knowledge

What is method?

A method is a procedure associated with a message and an object. One of the most important capabilities that a method provides is method overriding which allows the same name to be used for multiple different kinds of classes.

Class Diagram Description:

- Here we use some class for describing our ride sharing website like as –User, Login, Driver, Admin, Vehicle, Payment, review etc.
- First of all we need get information from user for knowing who using this website. Hence we need login information for checkup. So that if login information is correct then user can easily login.
- Another class is admin class which used to control the full system.
- In driver class reserve all driver related information.
- All payment related information is reserved in payment class.

Class Diagram:
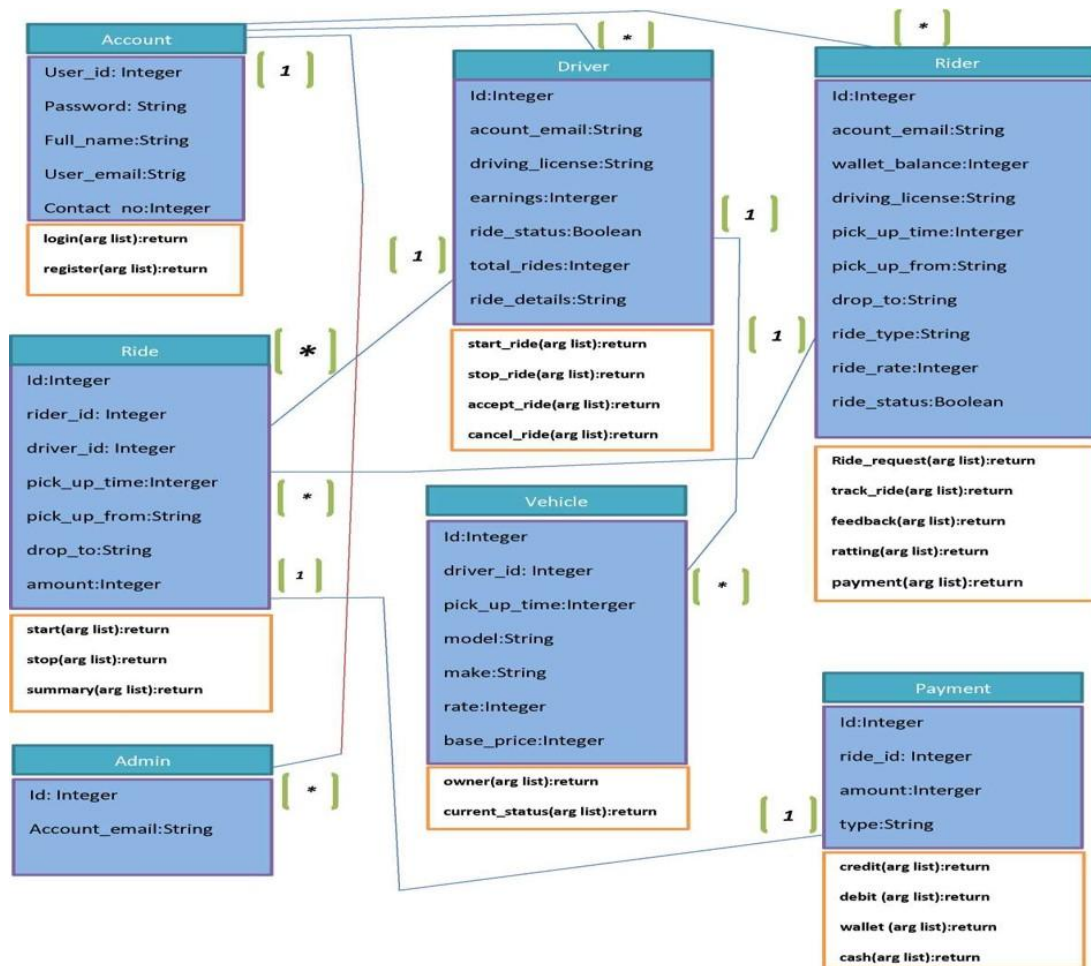
State Diagram:

A state diagram is the graphical representation of a state machine. State diagrams show a behavioural model consisting of states, state transitions and actions. State diagrams are irreplaceable when describing systems and system behaviour, for example electronic components such as control units for driver assistance systems in vehicles. State diagrams are the ideal way to model object life cycles. State diagrams enable us to describe the behaviour of objects during their entire life span. On a short we can say that State diagrams make the system behaviour visible.

State:

A state is a set of values that describes an object at a specific point in time, and it represents a point in an object's life in which it satisfies some condition, performs some action, or waits for something to happen.

Event:

An event is something that takes place at a certain point in time and changes a value(s) that describes an object, which in turn changes the object's state.

Basic Components of State Diagram:

1. **Initial state** – We use a black filled circle represent the initial state of a System or a class.

Figure – initial state notation

2. **Transition** – We use a solid arrow to represent the transition or change of control from one state to another. The arrow is labelled with the event which causes the change in state.

Figure – transition

3. **State** – We use a rounded rectangle to represent a state. A state represents the conditions or circumstances of an object of a class at an instant of time.

Figure – state notation

4. **Fork** – We use a rounded solid rectangular bar to represent a Fork notation with incoming arrow from the parent state and outgoing arrows towards the newly created states. We use the fork notation to represent a state splitting into two or more concurrent states.

Figure – a diagram using the fork notation

5. **Join** – We use a rounded solid rectangular bar to represent a Join notation with incoming arrows from the joining states and outgoing arrow towards the common goal state. We use the join notation when two or more states concurrently converge into one on the occurrence of an event or events.

6. **Self transition** – We use a solid arrow pointing back to the state itself to represent a self transition. There might be scenarios when the state of the object does not change upon the occurrence of an event. We use self transitions to represent such cases.

Figure – self transition notation

7. **Composite state** – We use a rounded rectangle to represent a composite state also. We represent a state with internal activities using a composite state.

Figure – a state with internal activities

8. **Final state** – We use a filled circle within a circle notation to represent the final state in a state machine diagram.
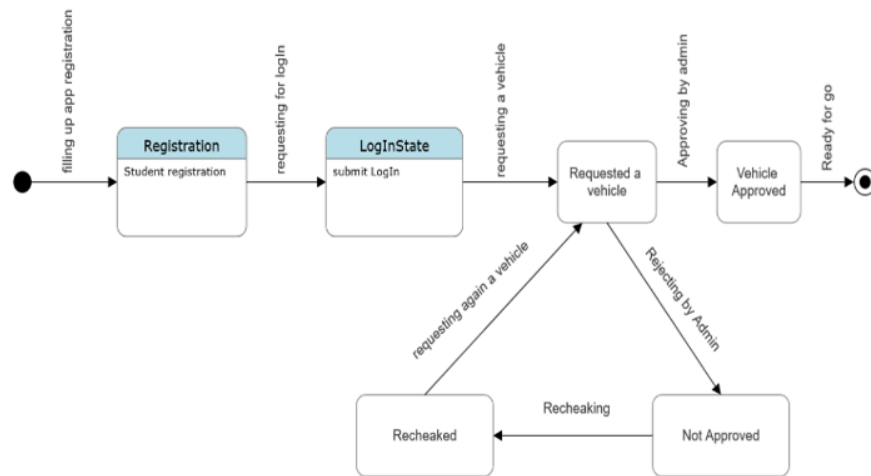

Figure – final state notation

Steps to draw a state diagram –

- Identify the initial state and the final terminating states.
- Identify the possible states in which the object can exist (boundary values corresponding to different attributes guide us in identifying different states).
- Label the events which trigger these transitions

References–

- ❖ https://cloud.smartdraw.com/editor.aspx?credID=-43352121&depoId=40565734&flags=128
- ❖ System Analysis and Design 5th Edition Alan Dennis, Barbara Haley Wixom, Roberta M. Roth
- ❖ https://www.geeksforgeeks.org/unified-modeling-language-uml-statediagrams

State Diagram:

State Diagram Description:

Initially order placed from user and go to the state named unprocessed order, then checked if order is rejected then go to reject state and finish that order but if order is accepted then go to accepted order state. After that, if everything ok then go to order fulfill state otherwise go to pending state. Hence, when got everything ok then go to pending state to order fulfill state. Then they complete the ride and go to payment state for collecting payment. After completing payment state go to finish state for complete the rideshare system.

Sequence Diagram:

- A sequence diagram represents the scenario or flow of events in one single use case. The message flow of the sequence diagram is based on the narrative of the particular use case.
- A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.
- Represent the details of a UML use case.
- Model the logic of a sophisticated procedure, function, or operation.
- See how objects and components interact with each other to complete a process.
- Plan and understand the detailed functionality of an existing or future scenario.
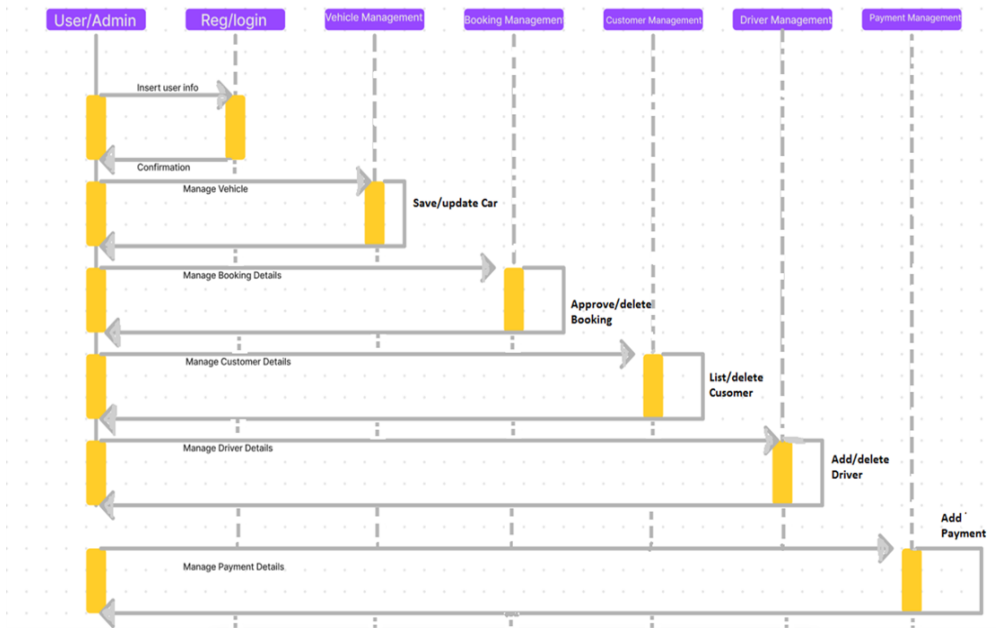
CRC Diagram:

- CRC Diagram is a class responsibility collaboration Diagram It represents each classes responsibilities and with which class it collaborates to complete it's responsibilities.
- Each CRC card has three components:
- On top of the card, the class name
- On the left, the responsibilities of the class
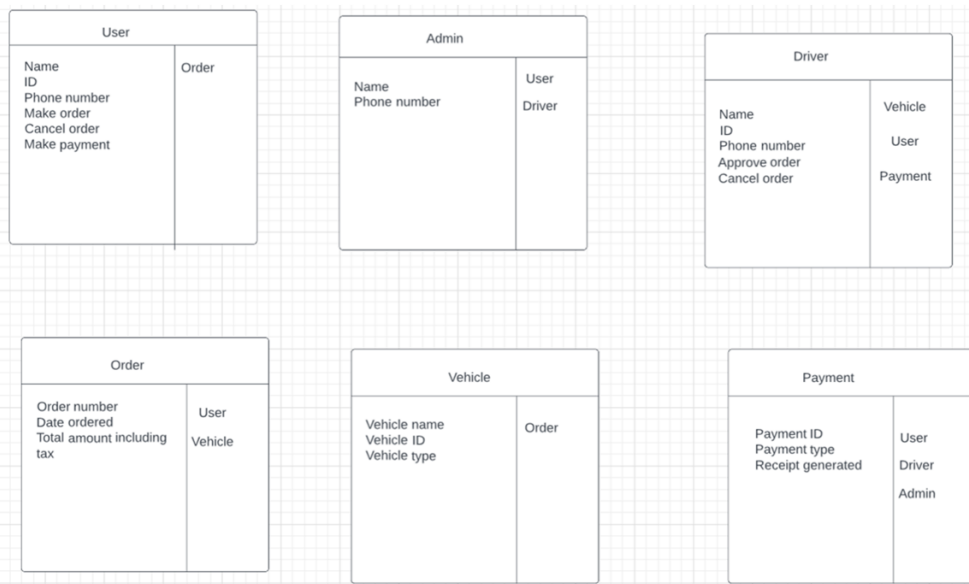
Deployment Diagram:

What is Deployment Diagram:

A UML deployment diagram is a diagram that shows the configuration of run-time processing nodes and the components that live on them. Deployment diagrams are a kind of structured diagrams used in modeling the physical aspects of a system or a platform. It shows how the end-users are accessing the system and how the system is serving the service. It only covers the basic structured view of the system without going into details.

Where and when to use Deployment Diagram:

- When we need to show, with what the newly added system will interact or integrate with.
- How large or vast the system needs to be.
- Who/what will interact with the system.
- What kind of protocols does the system need to go through.
- What kind of HW/SW does the system need to function properly.
- Who/What can access which part and how'd they access it.

Deployment Diagram
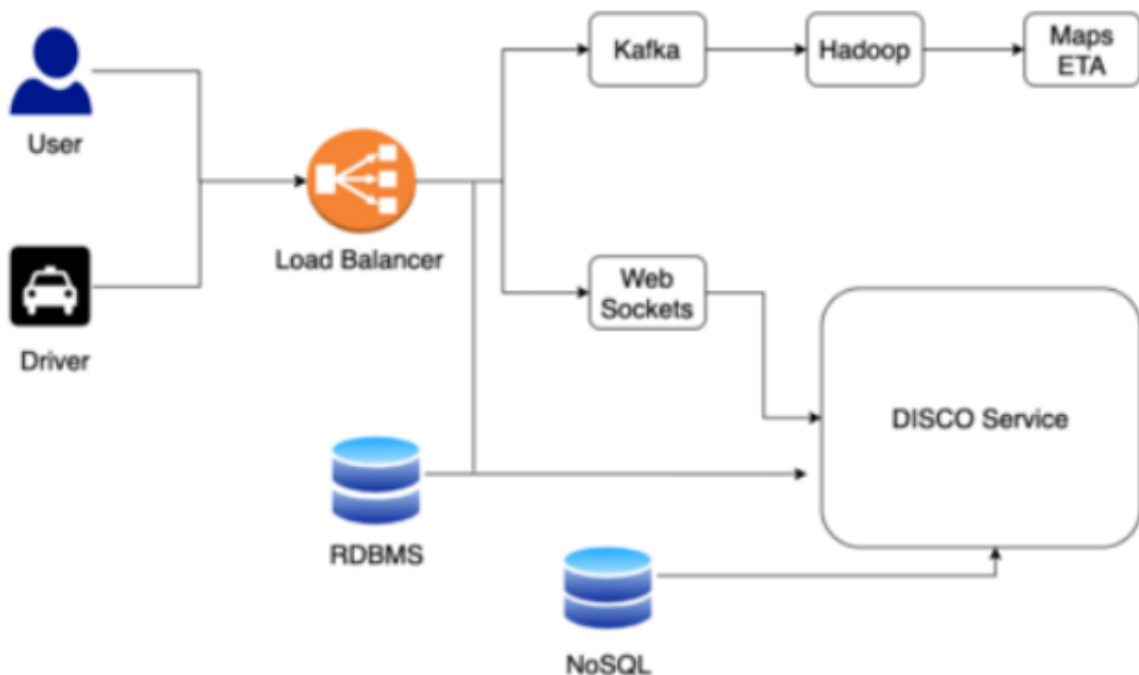
Financial feasibility:

Cash Flow Method-

	Month 1	Month 2	Month 3	Month 4	Month 5
Expense	90$	130$	140$	150$	120$
Revenue	40$	90$	150$	220$	280$
Cash Flow	(50) $	(40) $	10 $	70 $	160 $
Accumulating Cash Flow	(50) $	(120) $	(110) $	(40) $	120 $

Expense:
Month 1: Server Cost
Month 2: Server Cost + Promotion Cost
Month 3: Server Cost + Collaboration Cost
Month 4: Server Cost + Maintenance Cost
Month 5: Server Cost

Revenue:
Month 1: Ads Revenue
Month 2: Ads Revenue
Month 3: Ads Revenue + Sponsorship
Month 4: Ads Revenue + Sponsorship
Month 5: Ads Revenue + Subscription

NPV Method-

	Month 1	Month 2	Month 3	Month 4	Month 5
Expense	90$	130$	140$	150$	120$
Revenue	$40	$90	$150	$220	$280
Expenses (NPV)	$90	$128.93	$137.69	$146.31	$116.08
Revenue (NPV)	$40	$89.26	$147.53	$214.59	$270.86
Cash Flow (NPV)	$(50)	$(39.67)	$9.84	$68.28	$154.78
Accumulating Cash Flow (NPV)	$(50)	$(89.67)	$(79.83)	$(11.55)	$143.23

Interest Rate: 10% (Yearly)
            0.83% (Monthly)

SWOT Analysis-

Strength	Weakness
•Enthusiast about Ride sharing platform. •Can spend long time on this project. •Small team size.	•Lack of knowledge among the team member. •Low fund.
Opportunity	**Threats**
•Community based Sharing platform. •Revenue increasing proportional to member increasing.	•Limited no of clients as new app. •Too much similar app available.

Feasibility Report:

Pros(Project View)

- Specially based on a community for using this platform. So the community will get a premium Ride Sharing Platform.
- Revenue increases when member of the platform increase.
- There are lots people who are enthusiast about Ride sharing platform. So there is a possibility of getting good number of user.
- Can spend long time on this project because lots of feature to develop.
- Small team size so easy to handle.

Cons(Project View)

- Lack of knowledge among the team member about doing large scale project.
- Low fund for doing the project. So it will be impossible to make it available on a large scale.
- Limited no of clients because we have no marketing team for the product.
- It's a direct threat that there are lots of similar million dollar app and project available. It may not be possible for us to survive in the field.

Project Outsourcing

- Can get good quality-based Software.
- Project could be well structured.
- Higher Cost

Project Insourcing

- Software Quality would be average.
- Small member. So, low cost.

Counclusion

Due to budget constraints, it is recommended that our ride sharing project utilize the insourcing option rather than outsourcing. This will allow us to better manage costs and ensure that the project remains within budget. While outsourcing may offer some potential cost savings, the added complexity of managing an external vendor may make it more challenging to maintain control over the budget. As a start-up, it is important that we carefully consider cost-cutting measures in order to ensure the long-term financial viability of the project. Therefore, we recommend pursuing the insourcing option as a means of maximizing efficiency and minimizing expenses.

UI Design:

UI Design Definition-

- A user interface is the part of the system with which the users interact.
- The user interface design defines the way in which the external entities will interact with the system and the nature of the inputs and outputs that the system accepts and produces.

UI Design Principle Guidelines:

1. Three Golden Rules of User Interface Design
   a. Place Users in Control
   b. Reduce Users' Memory Load
   c. Make the Interface Consistent
2. Norman's Design Principles (1988)
   a. Visibility: Can I see it?
   b. Feedback: what is it doing?
   c. Affordance: Is it self descriptive?

Design:
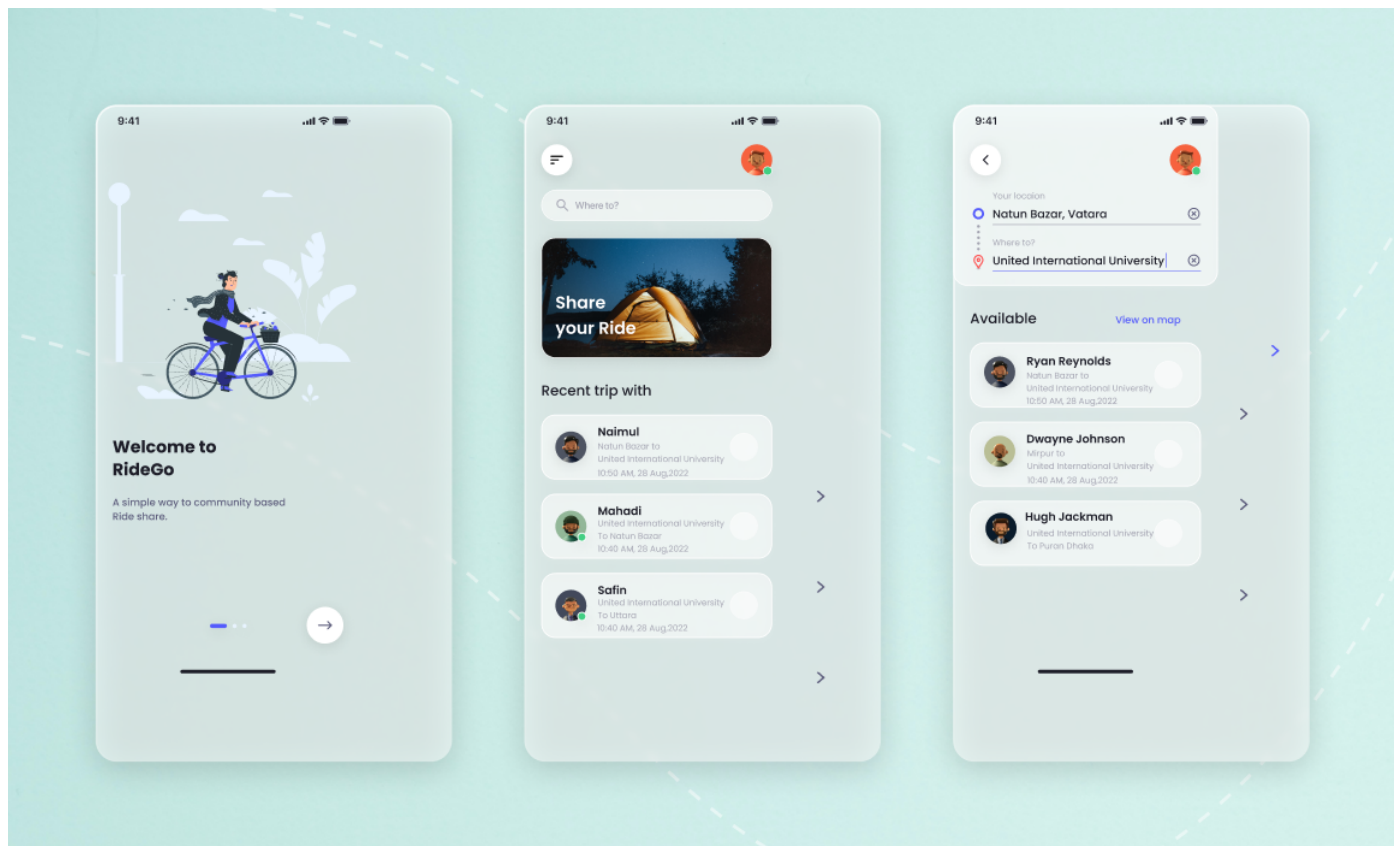
Conclusion:

In conclusion, the RideGO service presents a unique opportunity for developing companies to provide a valuable service to both drivers and users. While there are certainly challenges to be faced in building a successful and reputable organization, we believe that our service has the potential to thrive in today's market. By offering affordable and high-quality transportation services, we aim to meet the needs of a wide range of individuals and communities. Through careful planning and attention to the factors that contribute to success and challenges, we believe that RideGO can succeed in creating positive and mutually beneficial relationships with our drivers and users. In a time when access to reliable and affordable transportation is increasingly important, we believe that RideGO has the potential to make a significant impact and improve the lives of many.

Group Member	Task
Md Rokibul Hasan (011193040)	Use Case Descriptive form, Research Paper, Comparison table, Features list fixation, Feasibility Analysis, Financial Feasibility, SWOT Analysis, UI Design
Md. Mansurul Haque (011202034)	Online Article, Gap Analysis, Benchmark product, State Diagram.
Md Rajib Hossen (011191244)	Data Flow Diagram, Benchmark Product, UI Design Principle, CRC Diagram.
Muminul Islam (011202115)	Activity Diagram, Swimlane, Benchmark Product. Feasibility Report - Project Outsourcing, Project Insourcing, Conclusion,Sequence Diagram
Emam Hasan (011201302)	Context Diagram, Use Case Diagram, Research Paper, Feasibility Report - Pros and Cons(Project View),Class Diagram.