



American International University-Bangladesh (AIUB)

Department of Computer Science

Faculty of Science & Technology (FST)

Ride Sharing Management System

A Software Requirement Engineering Project Submitted
By

Semester: Spring_22_23		Section:	Group Number:	
SN	Student Name	Student ID	Contribution (CO1+CO2)	Individual Marks
6.	Kazi Maruf Ahmed Alif	19-40044-1		
12.	Sumaiya Nabi Namira	19-40790-2		
33.	Asif Sharif Akash	20-42647-1		
37.	Maharabur Rahman Apu	20-42675-1		

The project will be Evaluated for the following Course Outcomes

Evaluation Criteria	Total Marks (50)	
Introduction, Format, Submission, Defense	[10 Marks]	
System Overall Description & Functional Requirements	[10 Marks]	
System Quality Attributes and Project Requirements	[10 Marks]	
UML and E-R Diagram with Data Dictionary	[10 Marks]	
UI/UX Prototyping	[10 Marks]	

Software Requirements Specification

for

<Ride Sharing Management System>

Version 2.0 approved

Prepared by < Kazi Maruf Ahmed Alif ,

Sumaiya Nabi Namira,

Asif Sharif Akash,

Maharabur Rahman Apu>

<American International University-Bangladesh>

<29th April, 2023>

Table of Contents

Revision History	3
1. Introduction.....	4
1.1 Purpose.....	4
1.2 Document Conventions.....	4
1.3 Intended Audience and Reading Suggestions	4
1.4 References.....	5
2. Overall Description	5
2.1 Product Perspective.....	5
2.2 Product Functions	6
2.3 User Classes and Characteristics.....	6
2.4 Operating Environment.....	7
2.5 Design and Implementation Constraints	7
2.6 User Documentation	7
3. System Requirements.....	8
3.1 System Features	8
3.2 Non-Functional/Quality Requirements	14
3.3 Project Requirements	16
4. Design and Interface Requirements	18
4.1 UML Diagrams	18
4.2 Data Dictionary	22
4.3 UI/UX Design Specification	23

Revision History

Name	Date	Reason for Changes	Version
Maharabur Rahman Apu	27/04/2023	Updated Cross reference of the Functional Requirement	1.1
Asif Sharif Akash	29/04/2023	Updated the simple diagram	1.5
Kazi Maruf Ahmed Alif	29/04/2023	Updated UI/UX design	2.0

1. Introduction

1.1 Purpose

This document contains software requirements of “Ride Sharing Management System” and it shall be labeled as version 1.0. The purpose of a ride-sharing with another passenger project is to provide a platform that enables individuals to share rides with others who are traveling in the same direction. By connecting drivers and passengers who are going to the same location or along the same route, the project aims to reduce transportation costs, alleviate traffic congestion, and decrease the carbon footprint of transportation. Additionally, ride-sharing with another passenger can provide social and environmental benefits, as it can help people to make new connections, reduce their individual impact on the environment, and promote a more sustainable and efficient transportation system. By making it easier for individuals to share rides, the project seeks to provide a convenient, affordable, and eco-friendly transportation option that benefits both passengers and drivers. The product describes the whole subsystem as a passenger takes a ride and if possible shares a ride with another passenger.

1.2 Document Conventions

The major headings are labeled such as

1. Introduction
2. Overall Description

Font style: Times New Roman and **Font Size:** 18

Sub-headings are labeled with points such as

- 1.1 Purpose
- 1.2 Document Convention

Font style: Times New Roman and **Font size:** 14

Font size 12 is used for paragraphs. Headings and sub-headings are highlighted as bold. Justify is used as alignment.

1.3 Intended Audience and Reading Suggestions

The intended audience for a ride-sharing with another passenger project would be anyone who needs to travel from one place to another and is interested in sharing a ride with others who are going in the same direction. This could include commuters, students, travelers, and anyone else who is looking for an affordable and convenient transportation option.

1.4 References

[1] V. Vuchic, Transportation for Livable Cities, Journal of Transportation Engineering, vol. 2, pp. 5-12, 2017.

Available link:

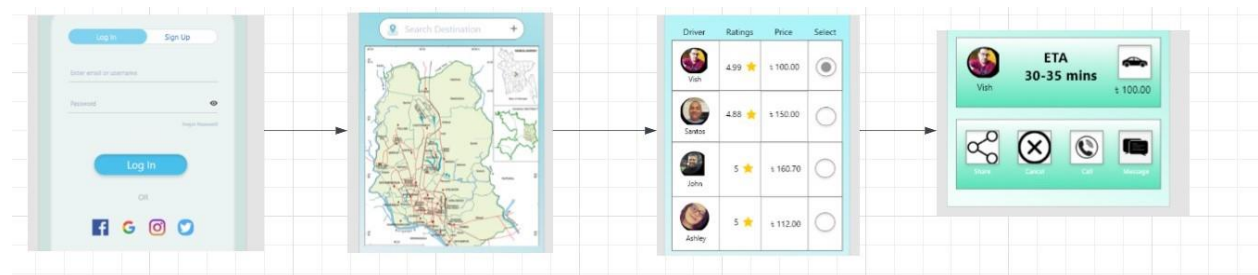
<https://t4tutorials.com/ride-service-management-system-project-like-uber-and-cream/>

https://www.researchgate.net/publication/359609541_Web_Application_based_Ridesharing_Your_Transportation_in_Jakarta

2. Overall Description

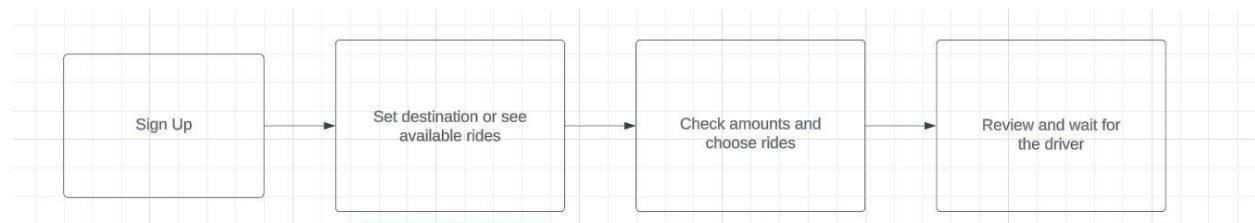
2.1 Product Perspective

The product is brand-new and complete. The product's main goal is to give passengers a ride when there is already one in the vehicle. Although it guarantees to offer a comparable level of service, because it can be shared with other passengers, it places a greater emphasis on routes and destinations. The product guarantees the utilization of fewer vehicles and resources for two or more services.



Since this product is essentially an open source online ride service, it offers drivers the chance to register their vehicles online. Drivers can register any sort of car using this product and then start looking for the closest pick-up and drop-off locations. All types of vehicles will be supported by this app. Drivers that experience business losses can remedy their problems with this software. This software works in all cities because there is no central management office. When it notifies the driver that a passenger has requested anything, this program also offers a text-to-speech function. The introduction of open source software will boost its market worth.

2.2 Product Functions



There will be various functionalities in this product, but the four major functions are as subsequent. There are functions in a ride share application to sign up, share the location of the ride or provide the destination, choose ride, and review the driver.

- Every software or product requires login in order for a regular user to access the application. Therefore, the user must sign up for our product using his or her username and password.
- Knowing the location to which the user must travel is necessary in order to set a destination for a place for a user. Therefore, in this scenario, the user must set the location using a map to the destination.
- This product's main components are ride sharing and rider selection. A user can check whether there are any open rides for the specified destination along the same route after setting the destination, and then apply to join those rides.
- The act of waiting for the drivers and evaluating them is a function of the user because it signifies that the request for the ride has been granted and allows the user to determine whether the passenger enjoyed the ride or not.

2.3 User Classes and Characteristics

In this product there are three users respectively. There will be Passenger, Navigator and Driver.

Passenger:

- First user and most important user
- The passenger can login to the product
- Search nearby routes and check available rides
- Set the destination
- Check reviews of the drivers and bonuses
- Give feedback
- Contact the driver
- Pay bill by cash or Bkash

Driver:

- Second user and less important than passenger
- Check shortest routes
- Checks vehicle condition

- Safely travels the passenger to the destination
- Receives payment
- Checks for update

Navigator:

- Third user and can be said the one controls the whole system
- Checks and verify the user
- Checks on employee
- Manage the whole system

2.4 Operating Environment

Operating System: iOS & Andriod.

Specification:

- Should be compatible with iPhone, iPad, and iPod touch. Application should be downloaded from iOS's official App Store.
- Should be compatible with phones and tablets having Android 10 or later. Application should be downloaded from Google Play Store.

2.5 Design and Implementation Constraints

Ride-sharing services have the ability to connect passengers with comparable timetables and itineraries, which might be very advantageous for both individual users and the city as a whole. This is a difficult undertaking, though, because consumers' requests are unknown in advance and only become available just before departure.

The design is efficient and optimal. This is achieved by dividing the system into two components:

- 1) **The constraint satisfier:** Constraint satisfier takes as input the spatio-temporal constraints of drivers and passengers and provides feasible (driver, passenger) pairs in real time.
- 2) **The matching module:** Matching module takes as input the feasible pairs and provides a maximum cardinality matching of drivers and passengers.

2.6 User Documentation

The ride sharing management system will be given with the user documentation. It could contain:

User guides: Details regarding the features and instructions for use will be supplied.

Tutorials: Images and lessons will be provided with reference to the functionalities.

Notes: Bugs, modifications, and updates each edition's documentation information

Guidelines: A brief summary of the ride sharing management system installation.

Troubleshooting: Information about potential problems that some users might encounter.

3. System Requirements

3.1 System Features

1. Sign Up

Functional Requirements (FRs)

- 1.1 The software shall allow users to sign up the application for booking a ride, offer a ride and/or to access any of the functionality of this application.
- 1.2 After installation when user open the application, by default user should be ask for User Name and Password. A TCP connection should be established between Phone and server (client-Server). Only one connection should be allowed per device (i.e., through the same device/phone user cannot create multiple login)
- 1.3 If user has not created account yet, a provision of sign up the account should be provided. Also, a provision to reset the password should be provided on login screen only via 'Forgot Password' feature.
- 1.4 After filling all the parameters, while clicking on 'Sign Up', user should ask for the profile verification. A four-digit OTP should be sent to user's mobile number/email address as per user's preference. Only numeric values should be generated in OTP. OTP should be unique number for a login device.
- 1.5 OTP needs to be verified by server through Database. On successful login, below screen should be display. Hyperlink at 'Login' should redirect user to login page.
- 1.6 In case of wrong OTP, an error message should be display at screen along with 'Resend' option. On access of 'Resend', again a new OTP should be sent to user's preferred mode (mobile number/email address).
- 1.7 In such case, if user receives first OTP after clicking on 'Resend' then also do not allow to proceed further. Consider that OTP as invalid OTP for user.

Priority Level: High

Precondition: The user should have installed the application on their device.

Cross-references: 2.1, 3.1

2. Software Login

Functional Requirements (FRs)

- 2.1 The application shall allow users to login with their given username and password.
- 2.2 The login credentials (username and password) will be verified with database records.
- 2.3 If the login successful, the home page of the user account will be displayed.
- 2.4 If the username and/or password has been inserted wrong, the random verification code will be generated and sent to the user's email address by the system to retry login.
- 2.5 If the number of login attempt exceed its limit (3 times), the system shall pop up a window named forgot password!

Priority Level: High

Precondition: The user should have signed up for the application.

Cross-references: 1.1, 3.1, 4.1

3. Forgot Password

Functional Requirements (FRs)

- 3.1 The application shall redirect the user to his/her registered email account after clicking the pop-up window named forgot password!
- 3.2 The link should be valid till next 24 hours of the email's sent time. Also, link should be expired once user successfully reset the password.
- 3.3 When user access the link then user should be redirected to the application's change password screen. Here, user needs to provide a new password and confirm new password (Password's validation should be as per mentioned above in Sign Up screen).
- 3.4 Once user set the password, a message of successful password reset should be display along with the notification at mobile number and email address. Also, that reset password link should expire now as user has successfully accessed the link.
- 3.5 After setting a password along with the successful message, user will be provided a link to re-direct at the 'Login' page.
- 3.6 While accessing the hyperlink, user should be navigated to the login page where user needs to provide a user name and new set password for the application login.

Priority Level: High

Precondition: The user should have confirmed his/her password again by logged in.

Cross-references: 2.4, 4.1

4. Home Page

Functional Requirements (FRs)

4.1 Here, after providing User Name & Password and clicking upon 'Login', user should be redirected to 'Home' page.

4.2 From this screen, user can access two features:

- Book a Ride
- Offer a Ride

Priority Level: High

Precondition: The user must have logged in with their valid user id and password

Cross-references: 1.1, 2.1, 7.1

5. See nearby available cars

Functional Requirements (FRs)

5.1 The system should allow users to search for available cars based on their current location. The system should use the user's device location or allow the user to input their location to filter available cars in their area.

5.2 The system should provide real-time updates on the availability of nearby cars. It should show which cars are available for rent, how long they will be available, and the distance to the car.

5.3 The system should allow users to filter available cars by different criteria, such as car type, price, and availability time.

Priority Level: High

Precondition: The user needs to turn on the live location feature.

Cross-references: N/A

6. Set Destination

Functional Requirements (FRs)

6.1 The system should allow users to search for their desire location.

6.2 The system should provide location-based suggestions for popular destinations in the area, such as landmarks, tourist attractions, and restaurants.

6.3 The system should provide a map view that displays the destination location and allows the user to view the route to their destination.

6.4 The system should calculate the best route to the user's destination based on their current location, traffic conditions, and mode of transportation.

6.5 The system should provide alternate route options in case of traffic or road closures.

Priority Level: High

Precondition: user have to turn on live location

Cross-references: N/A

7. Book a Ride

Functional Requirements (FRs)

- 7.1 This feature should available for traveler's (who wants to travel in the car).
- 7.2 The system should allow the user to select the desire car from the list of cars.
- 7.3 While user accessing the feature the user should be asked to provide some information regarding pickup point and destination point/ end point (mapped with google map).
- 7.4 User can select Date and time in case of he/she wants to pre-book the car. Pre-book is allowed in advance of 1 month only. User can't select a date which duration is greater than 1 month.
- 7.5 User can choose the car size like (4-seater, 6-seater, 8-seater) that's depends on his/her needs.

Priority Level: High

Precondition: The user must have logged in with their valid user id and password and the user's location and destination should be available.

Cross-references: 4.1, 5.1, 6.1, 9.1

8. Bid the Ride amount

Functional Requirements (FRs)

- 8.1 After showing all the available cars the user can bargain with the drivers for their fair amount.
- 8.2 In this case which driver is agree with the amount than he/she can select the vehicle for the ride.

Priority Level: Medium.

Precondition: The users should select the car to bid the amount.

Cross-references: N/A

9. Share the ride with other users

Functional Requirements (FRs)

- 9.1 This is the main focusing feature of this application, customers can share the ride with others customers if the path of the both customers are similar.

9.2 There will be a function name share ride, if the customer clicks the button there will be the collection of rides that are ongoing so they can select their rides with their preferred location.

Priority Level: Medium.

Precondition: The user should have booked a ride and selected the option to share ride with others.

Cross-references: 7.2, 7.3, 7.5

10. Ride information & Booking

Functional Requirements (FRs)

10.1 While clicking on the ride info.

10.2 There will be the amount to be paid after the ride.

10.3 There will be the start time and the approximate end time.

10.4 Driver info is also will be there that what is the name of the driver, how many rides did he complete, driver's average ratings.

10.5 The fare should be shown here. If passenger is not available at the pickup point as per mentioned time/Driver has to wait for a long time. (Driver can add the extra charge for the same)

10.6 The list of passenger's information should be available (for shared rides)

10.7 The approximate pickup time should display after calculation (For the time calculation, the kilometers should be derived as per Google MAP & according to that the tentative time should be display)

Priority Level: High

Precondition: The user must have booked a ride.

Cross-references: 7.2, 7.5

11. Payment Mode

Functional Requirements (FRs)

11.1 Three options should be provided to user for payment mode as per mentioned above.

11.2 Customer can pay the fare to the driver by (Cash/ Mobile banking/ Card) these modes.

Priority Level: High

Precondition: The user must have booked a ride and provide the payment details.

Cross-references: 10.2

12. Offer a Ride

Functional Requirements (FRs)

12.1 Applicable for drivers who wants to make their car available for pooling.

12.2 Drivers can offer a ride while providing the details of Start Point, End Point & Preferred Time (Drivers should be provided a provision to enter a preferred time/time duration).

Priority Level: High

Precondition: The user should have logged in to the application.

Cross-references: 4.1, 13.1

13. Rider share Car details

Functional Requirements (FRs)

13.1 While accessing 'Offer a Ride' feature, if driver is accessing application 1st time and if the license information has not been found for the driver, then driver should be redirected to car info page.

13.2 Driver/rider should fill all the parameters given in the car info page.

Priority Level: Medium.

Precondition: The user must have logged in to their account and offered a ride.

Cross-references: 12.1

14. Rewards & Offer

Functional Requirements (FRs)

14.1 User shall be able to see the offers available for him/her.

14.2 User can gather all the points he/she gets from every ride, later this coin can be used for discount based on the number of points.

Priority Level: High.

Precondition: The user must have completed at least one ride.

Cross-references: 7.3

15. Edit Profile

Functional Requirements (FRs)

15.1 A parameter of a user with filled values should be available in the screen. Also, user should be provided a provision to change the password & upload a photo.

Priority Level: Low.

Precondition: The user must log into their account.

Cross-references: 2.1

16. Change Password

Functional Requirements (FRs)

16.1 While clicking on the change password button then it will redirect to a page where users have to provide current password, new password & confirm new password.

16.2 Then the system will auto logout.

16.3 User have to login again with his/her new password.

Priority Level: Medium.

Precondition: The user must log into their account.

Cross-references: 2.1, 3.1

17. Logout

Functional Requirements (FRs)

17.1 User will be logout from the system.

Priority Level: Medium.

Precondition: The user must log into their account.

Cross-references: 2.1, 4.1, 7.1

3.2 Non-Functional/Quality Requirements

QA1: Usability: The system should be easy to use for both users and drivers. The user interface should be intuitive and user-friendly, and the system should provide clear instructions for drivers on how to accept and complete rides.

Priority Level: High.

Precondition: N/A

Cross-references: User testing reports, load testing reports.

QA2: Performance: The system should be able to handle a large number of ride requests and transactions within an acceptable time frame. For example, the system should be able to process and confirm ride requests within 30 seconds.

Priority Level: High.

Precondition: N/A

Cross-references: Performance testing reports, load testing reports.

QA3: Availability: The system should be available to users and drivers 24/7. The system should have a high availability rate of at least 99.9% with a maximum downtime of 1 hour per month.

Priority Level: High.

Precondition: N/A

Cross-references: High availability architecture, uptime and downtime reports.

QA4: Reliability: The system should be reliable and minimize the risk of failures or errors. For example, the system should be able to recover from failures within 5 minutes and have a mean time between failures (MTBF) of at least 10,000 hours.

Priority Level: High.

Precondition: N/A

Cross-references: Testing reports, debugging reports, MTBF metrics.

QA5: Security: The system should be secure and protect sensitive user data. The system should comply with industry standards such as PCI DSS and have proper authentication and access control mechanisms in place.

Priority Level: High.

Precondition: N/A

Cross-references: Security testing reports, security guidelines and standards.

QA6: Maintainability: The system should be easy to maintain and update. The system should have proper documentation and be designed with modularity in mind, so that updates and changes can be made without disrupting the overall system functionality.

Priority Level: Medium.

Precondition: N/A.

Cross-references: Maintenance reports, update reports, modular design documentation.

3.3 Project Requirements

- **Requirement gathering Tools:** The system developer needs JIRA tools to perform managing, gathering, documenting the requirements for this software.
- **Prototyping tool:** These tools help in creating quick mockups and prototypes of the software to get feedback from stakeholders. Examples of such tools include Balsamiq, Axure, and Sketch.
- **Project management Tools:** The system developer needs smartsheet, Microsoft project to managing the overall project, including scheduling, resource allocation etc.
- **Testing tools:** The system developer needs selenium tools in perform testing activities before delivering the final product.

Constructive Cost Model

Software project type: Organic; [p=1.05]
 Coefficient<Effort Factor> = 2.4
 So, P = 1.05 and T = 0.38
 SLOC = 25000 Lines
 $\text{Persons-months, PM} = \text{Coefficient<Effort Factor>} * (\text{SLOC} / 1000)^P = 2.4 * (25000/1000)^{1.05}$
 $= 29.37$
 Development time, DM = 2.50 * (PM) ^T
 $= 2.50 * (29.37)^{0.38}$
 $= 9.03 = 10 \text{ months}$
 $= 1600 \text{ Working hours in total (Per week 40 hours)}$
 Required number of people, ST = PM/DM
 $= 29.37/10$
 $= 2.937 = 3 \text{ people}$

Budgeting

Developer/Tester salary of 10 months:
 Per employee salary per month = 40000 Taka = 400 Taka per hour
 Total salary = $400 * 1600 = \mathbf{6,40,000 \text{ Taka}}$

Requirement analysis:
 Required time = 1 month = 25 working days = 200 working hour
 Requirement analysis persons per hour salary = 250 Taka
 Total requirement analysis salary = $250 * 200 = \mathbf{50,000 \text{ Taka}}$

Transportation cost: **15,000 Taka** (Approximate)
 Hardware expense: **1,20,000 Taka** (Approximate)
 Rent expenses:
 Total in 10 months = **1,65,000 Taka** [Per month = 15,000 Taka]
 Total utilities in 10 months: **15,000 Taka** (Approximate)
 Maintenance (Till 4 months after delivery):
 Cost per hour = 1,200 Taka
 Total estimated time needed for maintenance = 40 hours
 Total estimated maintenance cost = $1,200 * 40 = \mathbf{48,000 \text{ Taka}}$

Project manager's salary of 10 months:
 Per month salary = 40,000 Taka
 Total salary = $40,000 * 10 = \mathbf{4,00,000 \text{ Taka}}$

Accountant's salary of 10 months:
 Per month salary = 12,000 Taka
 Total salary = $12,000 * 10 = \mathbf{1,20,000 \text{ Taka}}$

Total expense: $6,40,000 + 50,000 + 15,000 + 1,20,000 + 1,65,000 + 15,000 + 48,000 + 4,00,000 + 1,20,000 = \mathbf{1,573,000 \text{ Taka}}$

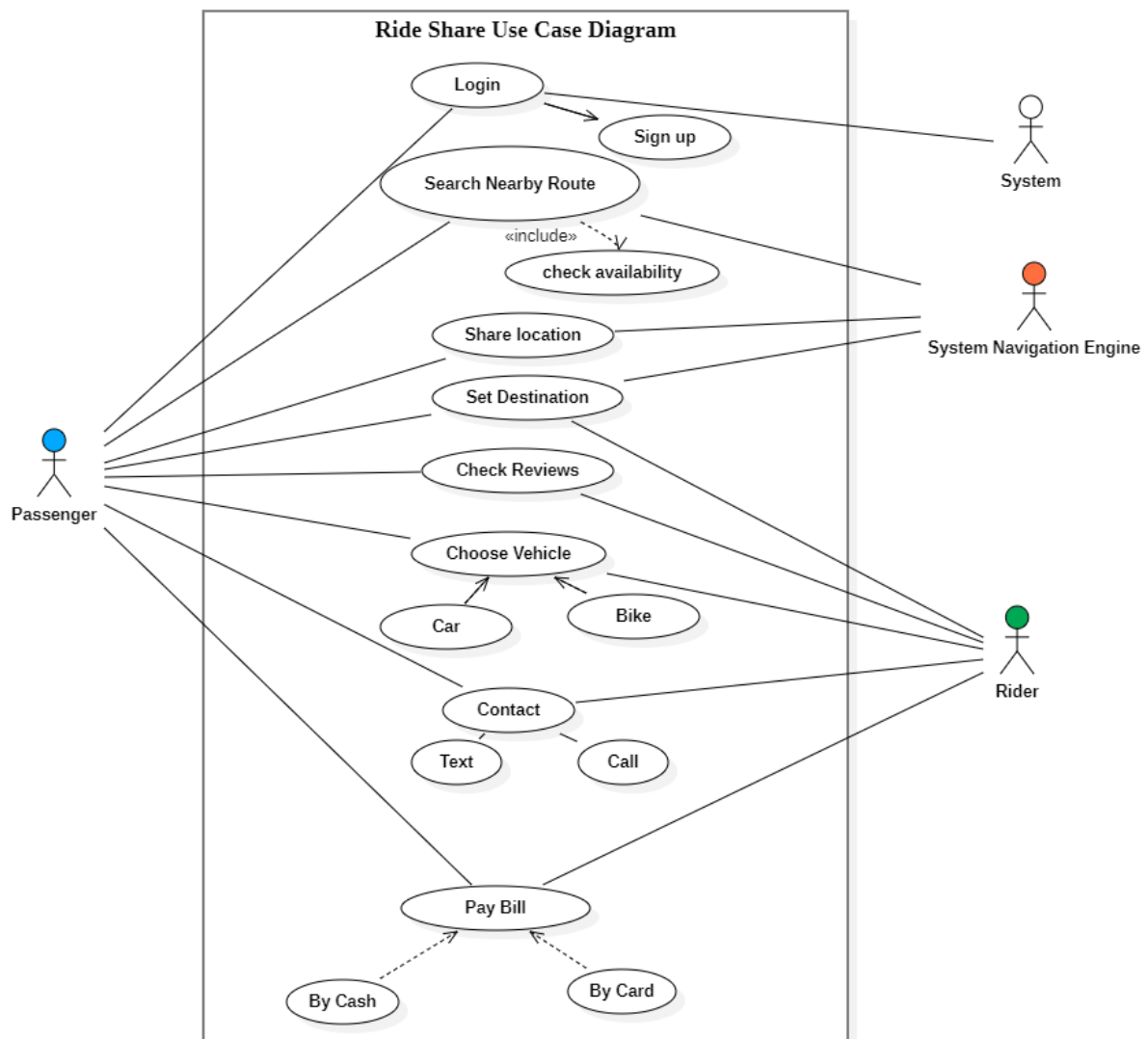
Profit: 25% of total expense = $1,573,000 * 25\% = 3,93,250 \text{ Taka}$

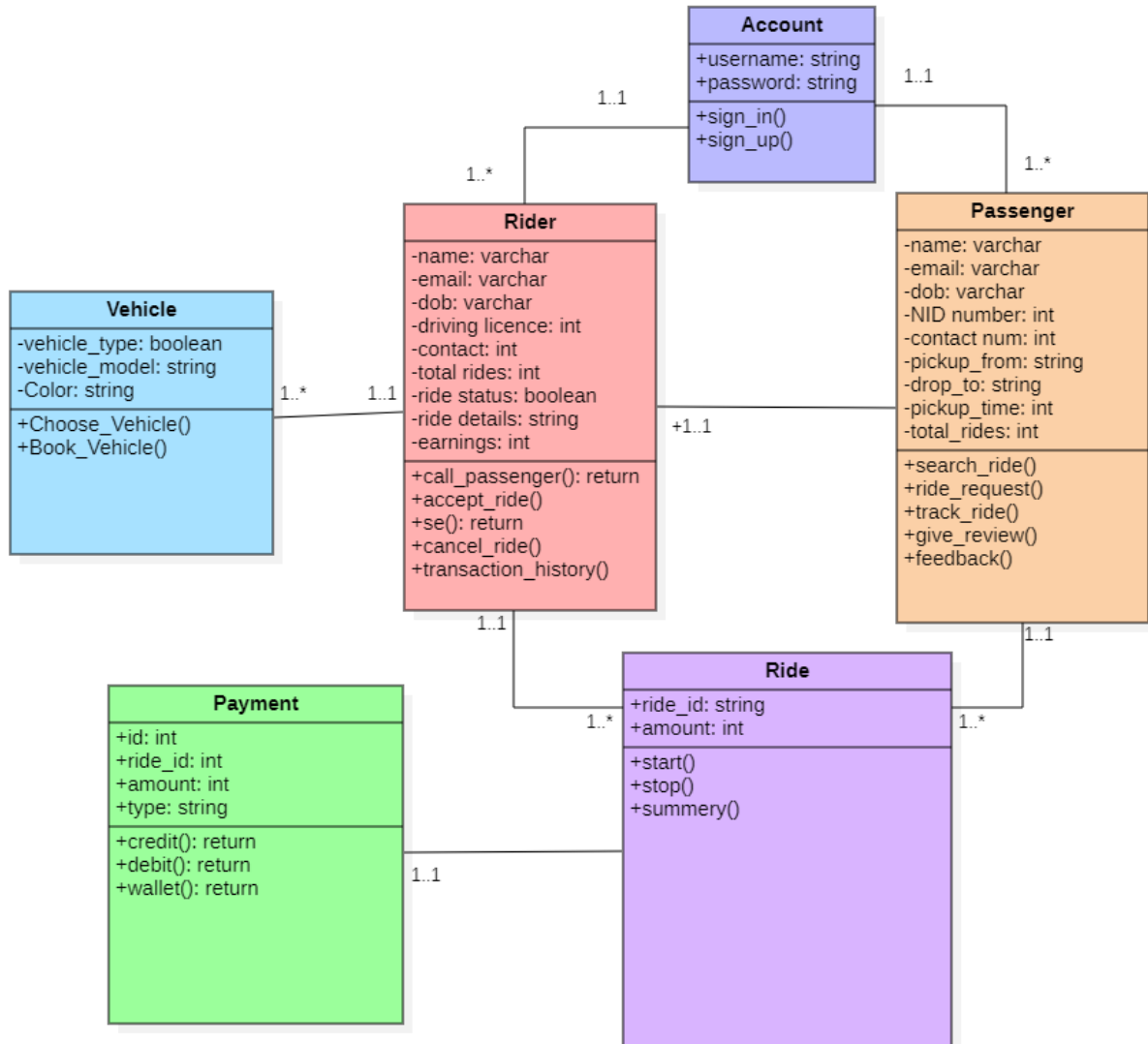
Total budget: $1,573,000 + 3,93,250 = \mathbf{1,966,000 \text{ Taka}}$

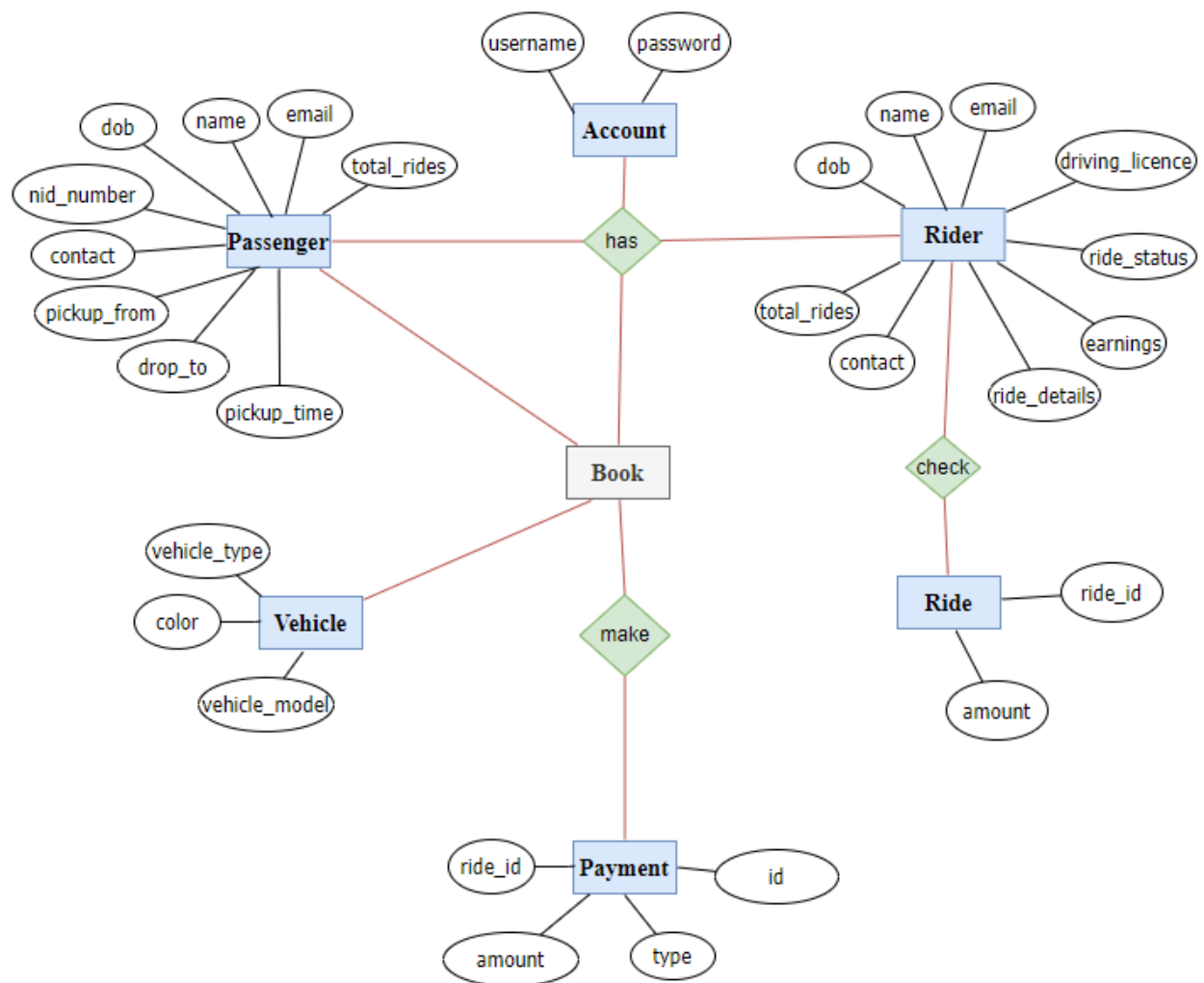
4. Design and Interface Requirements

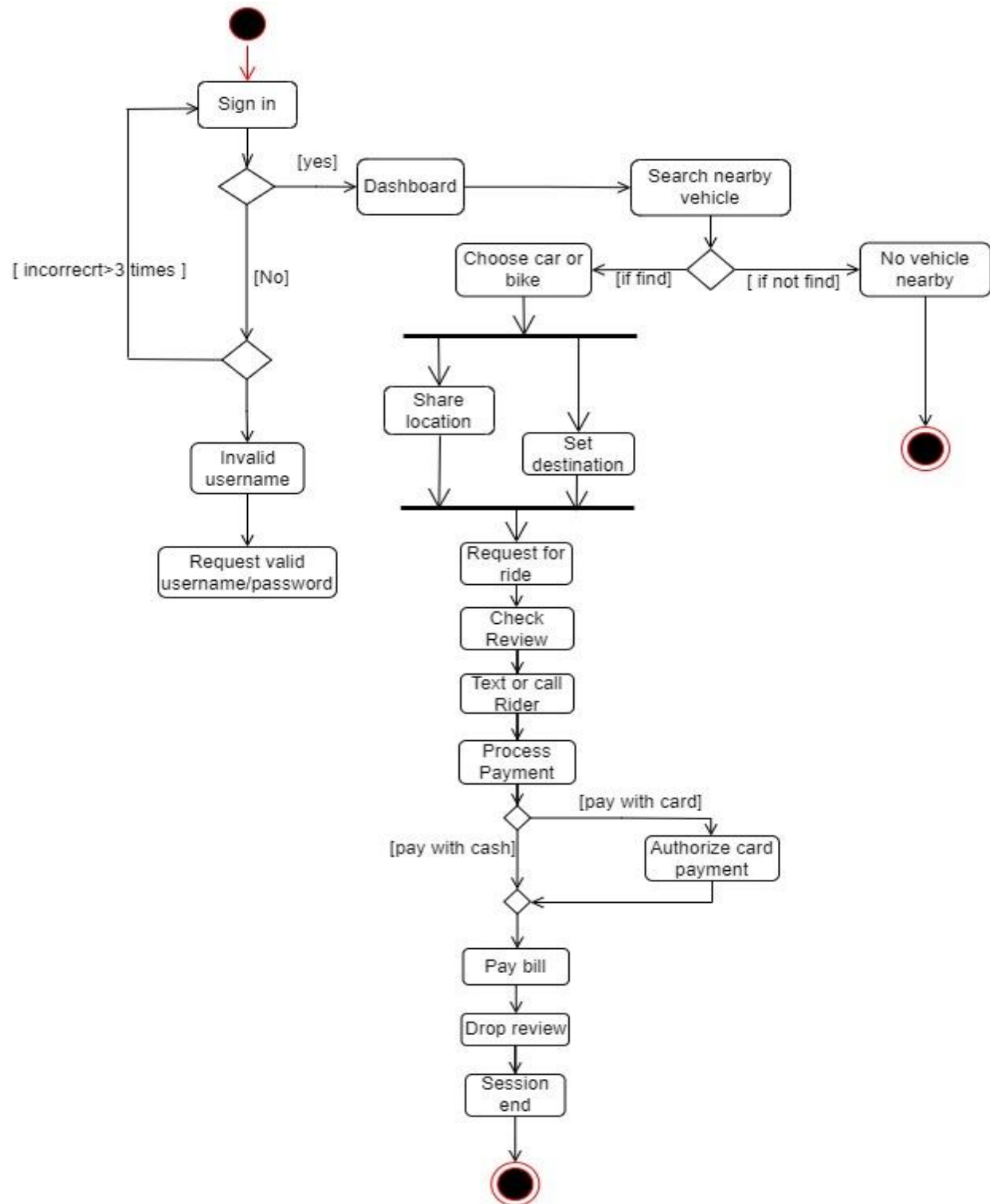
4.1 UML Diagrams

Use Case Diagram:



Class Diagram:

E-R Diagram:

Activity Diagram:

4.2 Data Dictionary

Passenger Class:

Entity	Attribute	Type/Size	Validation	Key
Passenger	Name(30)	Varchar	Required	
Passenger	email(15)	Varchar	Required	
Passenger	dob(8)	Varchar	Valid Date	
Passenger	Password(8)	String	Required	
Passenger	NID number(10)	Integer	Valid Nid Number	
Passenger	contact_num(11)	Integer	Valid Contact	
Passenger	pickup_from(10)	String		
Passenger	drop_to(10)	String		
Passenger	pickup_time(6)	String		
Passenger	total_rides(10)	int		

Rider Class:

Entity	Attribute	Type/Size	Validation	Key
Rider	Name(30)	Varchar	Required	
Rider	email(15)	Varchar	Required	
Rider	dob(8)	Varchar	Valid Date	
Rider	Driving licence(10)	Integer	Valid lic Number	
Rider	contact_num(11)	Integer	Valid Contact	
Rider	Total rides(5)	Integer		
Rider	Ride status(2)	Boolean		
Rider	Ride details(15)	String		
Rider	Earnings(6)	Integer		

Vehicle Class:

Entity	Attribute	Type/Size	Validation	Key
Vehicle	Vehicle_type(2)	Boolean	Required	
Vehicle	Vehicle_model(8)	String	Required	
Vehicle	color(8)	String	Required	

Account Class:

Entity	Attribute	Type/Size	Validation	Key
Account	Username (20)	Boolean	Required	
Account	Password(8)	String	Required	

Payment Class:

Entity	Attribute	Type/Size	Validation	Key
Payment	id(4)	Integer		primary
Payment	Ride_id(8)	Integer		primary
Payment	amount(4)	Integer		
Payment	Type(4)	String		

Ride Class:

Entity	Attribute	Type/Size	Validation	Key
Ride	Ride_id(5)	String		primary
Ride	Amount(4)	Integer		

4.3 UI/UX Design Specification