



UNIVERSITY
OF WOLLONGONG
AUSTRALIA



Project Progress Report

SkuPool

Team FYP20-S2-10

Lim Kang Wei 6276404
Jessica Soh Li Jing 6350021
Lim Kia Yin, Natalie 6276416

Supervisor

Mr Premarajan P

Project website

<https://skupool.wixsite.com/skupool>

Revision History

Title: Project Progress Report

Owner: FYP20-S2-10

Last Modified Date: 13 June 2020

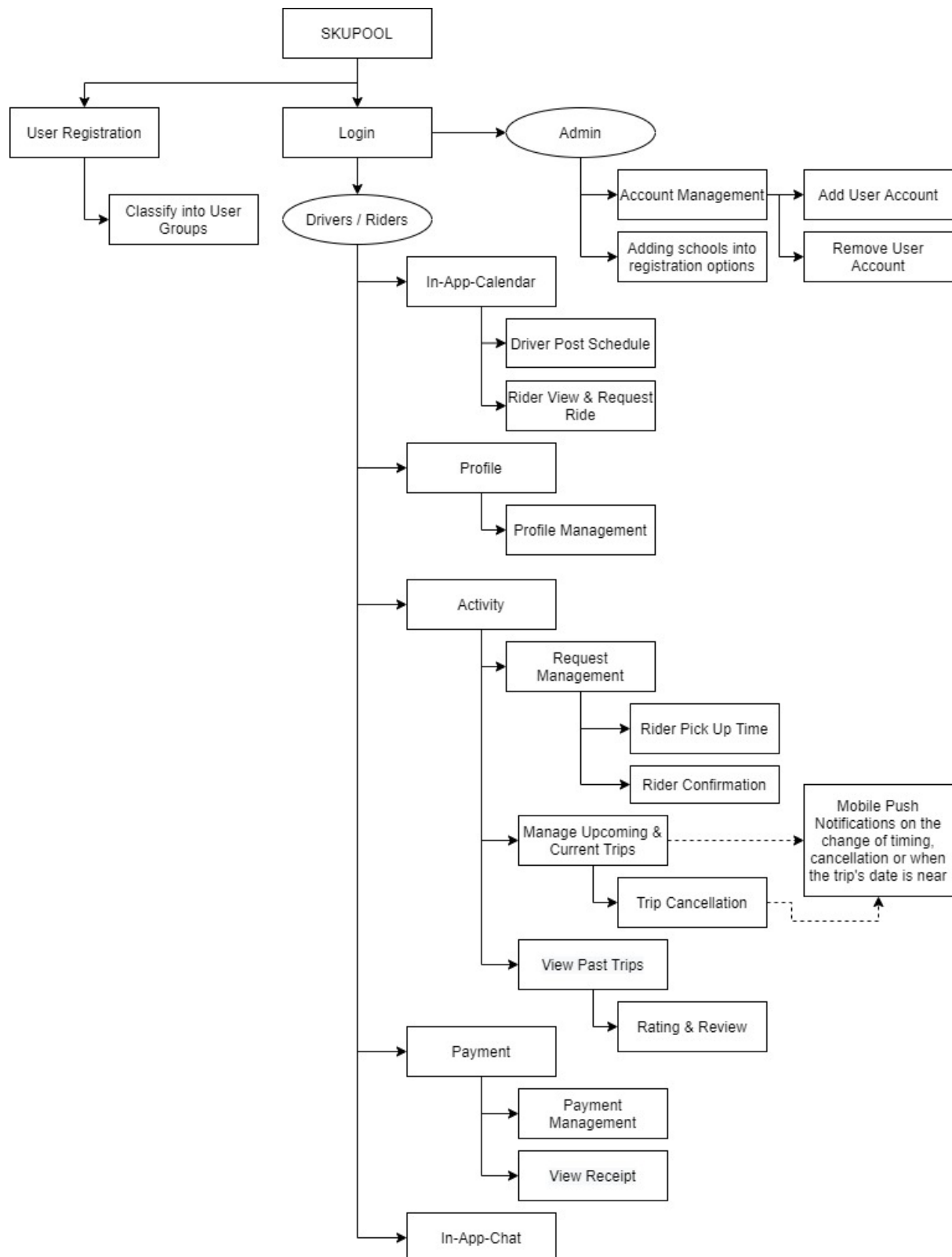
Version No.	Revised by	Description	Revised Date
1.0	Kang Wei	Created project progress report with layout and a rough content page	10/06/2020
1.1	Natalie	Added in Sequence and Activity Diagrams	12/06/2020
1.2	Kang Wei	Amendments in content	13/06/2020

Table of Contents

1	One-Page Product Summary	5
2	Project Overview	6
2.1	Project Description	6
2.2	Project Objectives	6
2.3	Learning Objectives	6
2.4	Target Audience	6
2.5	Business Model	7
3	Similar existing products	8
3.1	Grabhitch	8
3.2	RydePOOL	9
3.3	Gojek	10
3.4	UberPool	11
3.5	Cargo	12
3.6	Waze Carpool	13
4	Project Design	14
4.1	Overview	14
4.2	System Architecture	14
4.2.1	Functional Requirements	14
	Use Case Diagram	14
	Use Case Descriptions	15
4.2.2	Non-functional Requirements	32
4.2.3	Architectural Design	33
4.2.4	System Design	34
	Sequence Diagrams	34
	Activity Diagrams	41
4.2.5	Database Design	45
4.2.6	Data Flow Diagram for Booking Trips	46
4.3	User Interface Design	47
4.3.1	Overview	47
4.3.2	Login Page	47
4.3.3	Register Account Page	48
4.3.4	Menu Taskbar	49
4.3.5	Manage Profile	50
4.3.6	Manage Payment Details	51
4.3.7	In-App Chat	52
4.3.8	In-App Calendar	53

4.3.9	Driver Add Schedule	54
4.3.10	Rider's View of Calendar	55
4.3.11	Activity - View Requests	56
4.3.12	Rider's Request Page	57
4.3.13	Driver's Respond Page	58
4.3.14	Activity - View Upcoming / Past Trips	59
4.3.15	View Receipts	60
5	Project Implementation	61
5.1	Overview	61
5.2	Agile Development Methodology	61
5.3	Development Tools	61
5.3.1	Development Platform – React NativeA	61
5.3.2	Database Platform – Firebase	61
5.3.3	Product Website – Wix	62
5.3.4	Artwork – Wiz Logo Maker	62
6	Impact	63
6.1	Overview	63
6.2	Risk Analysis	63
7	Project Communication	64
7.1	Overview	64
7.2	Project Website Run-Through	65
8	Project Schedule (Gantt Chart)	68
9	Roles and Responsibilities	70
10	References	71

1 One-Page Product Summary



2 Project Overview

2.1 Project Description

Carpooling is the sharing of car journeys so that more than one person travels in a car, and preventing the need for having the others to drive to a location themselves. Understand that currently in the local market, there are already similar platforms like Grab, Gojek and Ryde, ready to be used. However, these applications being used are accessible to the public as a whole, therefore making it hard for one to get themselves a carpool ride to the school, since there might be others sharing the ride and heading to another location instead.

This project aims to develop a mobile application for Singapore Institute of Management (SIM) staff, faculty and students, with its to or from destination having to be SIM campus.

2.2 Project Objectives

Our project objective for the application is to allow people who need to share a car ride to the university to easily find each other and coordinate their pickup and travelling time. This in turn helps to utilise space in the parking area. Our application will act as a marketplace for both drivers and riders to use, being able to access upon successful registration. For our application, we are also aiming for a bigger market pool and attracting more people to use it. Our developed application will be suitable for both Android and iOS phone users.

2.3 Learning Objectives

There are a number of learning objectives that we have attained from this project. As a team, we are able to understand the proper flow of developing an application. Learning that communication is one of the key aspects in the project, not only between the team, but also with the other people who may be involved. The frequent meetings and discussions also give us exposure on presenting and selling our ideas. We also get to choose the proper development tools required to better cater to our application and why each development tool is well suited in our case. Apart from these, there are also other areas like project documentations and presentations, which might also be beneficial for us during our work in the future.

2.4 Target Audience

Instead of only users from SIM being able to use the application, we are opening up the use of it in selected schools as well. For example, Ngee Ann Polytechnic (NP) or even National University of Singapore (NUS). Our target audience would be school staff, faculty and students from the respective schools. Such that users from the same school will be classified under the same group so they can only see the drivers or riders from the same user group when using the application.

2.5 Business Model

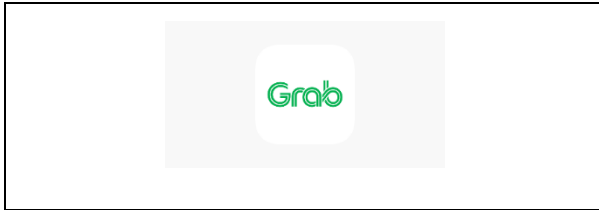
Instead of targeting only users from SIM, we are looking to expand the market to other schools as well, with schools being our domain target area. Such that users of the application will first be classified into groups when they register, based on the school that they select during registration. When using the application, they will then only be able to see other users in the same group, e.g. riders in the SIM group can only see SIM drivers' calendars.

We have then identified some of the following areas for our Business Model. With the main selling point being the drivers not requiring to set aside part of their earnings for the use of the app and also pushing the application for use by other schools. As a form of way to encourage more users on board and at the same time reducing the number of cars required to be in the individual's campus premises. Making it a win-win scenario for those travelling to campus, as well as on the school's end.

Area	Description
Customer relationships	Rating and Data based/Digital
Value Propositions	Reliable transport, Convenience, Additional Revenues, Low Cost
Cost Structure	Personnel required to maintain the app
Key Activities	Platform Development
Key Resources	Mobile Tech Platform, Network of Drivers
Channels	Mobile Apps (Android and iOS)
Customer Segments	Primary Riders - Students and School Employees Secondary Riders - People heading to school Drivers - Can only be students or school employees Able to toggle between being Driver or Rider
Revenue Streams	Mainly to provide additional revenues for drivers, as they won't be required to pay for using the app as well

3 Similar existing products

3.1 Grabhitch



GrabHitch is a social carpooling service that matches riders with a non-commercial driver going the same way. GrabHitch drivers are ordinary commuters who provide riders a lift to cover a portion of their petrol cost.

Functionalities:

- Ratings: Give ratings to drivers
- Reviews: Give ratings to drivers
- View Receipt: View payment transaction
- Driver Register Online: Register as driver online
- View & Manage Account: Drivers and riders can manage their own account profiles
- Help & Support: Help & Support from Grab Support when require
- Auto Best Match: Application able to give best match for driver base on the location they are going
- Choose Rider: Driver able to manually choose Rider
- Platform fee: Small charges to maintain the platform
- Compute cost: Application will compute the cost of ride base on the distance
- Map:
- Cashless Payment: Pay with credit card
- Save Credit Card Details: credit card information in application for future use
- Delete Credit Card Details: Remove credit card details when user wants to
- View upcoming trips: View upcoming trips that has been confirmed
- View history: View trips that has been completed
- Multiple pick up per trip: Driver able to pick up multiple riders on the way to destination
- Seats needed per pickup: Rider able to indicate how many pax for their pickup
- In-App Chat: A chat function within the application to allow conversation between Driver and Riders
- Rider Target Arrival Time: Rider able to indicate their targeted arrival time so Driver can plan the route and time
- Promo Code / Vouchers: Able to apply promo code for discount
- Save Addresses: Save address for future use
- Rider Location - Destination: Rider's upcoming trip for the day so Driver can find carpool service for last
- Profile Picture: Both Driver & Rider to upload profile picture of their own so their match could identify them and also for safety purpose
- Remarks When Posting Request: Rider able to add remarks when requesting for ride

3.2 RydePOOL



Ryde is our very own homegrown carpooling app, operating in Singapore, Malaysia and Hong Kong. Apart from RydePool, Ryde currently provide other services like RydeFlash, RydePET, RydeX, RydeXL, RydeHIRE, RydeSEND, RydeTAXI and RydeEXEC in Singapore.

Functionalities:

- Ratings: Give ratings to drivers
- Reviews: Give reviews to drivers
- View & Manage Account: Drivers and riders can manage their own account profiles
- Choose Rider: Drivers are able to choose the riders to pick up based on the rider's reviews and ratings on their profile
- View Trips: Riders can view upcoming trips/past trips
- In-App Chat: Drivers and riders can communicate within the chat
- Manage Addresses: Riders can save their addresses for convenience
- Map: Shows where the driver is currently at and the journey's runtime map
- Select number of seats: Riders are able to select how many seats they require for a trip; fares differ for the different number of seats selected
- Multiple pick-up per trips: Drivers are able to pick up more riders along the way, when heading to a supposed intended destination
- Pet-friendly: Pet-friendly option for riders whom might be bringing their pets out
- Price Adjustment: Riders are able to adjust the price higher but not lesser, to have a higher chance of getting a driver
- Payment Method: Credit Card, RydePay (Balance, Top-Up, Scan QR code)
- Settings: Add Home, Set Usual Leave Home Time, Add Work, Add Usual Leave Work Time
- Toggle between being a driver and passenger:
- Verification: Facebook, Mobile, Email, Business Email
- Invite Friends: Invite friends and get rewarded
- Favourites: Add favourite driver from the history page after completing a trip
- Help and Support: Includes FAQ and the necessary point of contact when support is required
- Remarks: Riders are able to include their additional comments or requests, for example, letting the Driver know that there will be wheelchair or even multiple luggage's.

3.3 Gojek

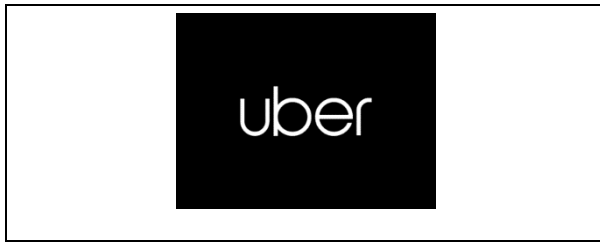


A multi-service platform first established in Indonesia, operating in Southeast Asian countries such as Vietnam, Singapore, Thailand, Philippines and Indonesia itself. Gojek provides many different services, ranging from Go-Food - a food delivery service - to Go-Car - a ride-hailing service for cars. As of now, only the Go-Car service is available in Singapore

Functionalities:

- Ratings: Give ratings to drivers
- Reviews: Give reviews to drivers
- View & Manage Account: Drivers and riders can manage their own account profiles
- Manage Payment Details: Riders can add/delete their credit card details
- View Trips: Riders can view upcoming trips/past trips
- In-App Chat: Drivers and riders can communicate within the chat
- Manage Addresses: Riders can save their addresses for convenience
- Map: Shows where the driver is currently at

3.4 UberPool



Uber is an American ride-hailing company that offers services including peer ridesharing, ride service hailing and food delivery. They currently operate in over 785 metropolitan areas worldwide.

Functionalities:

- View Driver Location: Real-time tracking of driver
- Choose Driver: Riders are able to choose their preferred drivers
- Multiple Payment Options: Riders are able to make payment using different methods
- Add Multiple Drop Off Points: The rider can have the driver make multiple stops
- Rating: Give ratings to drivers
- Review: Give reviews to drivers
- View Trip History: View the details of past trips
- In-App Chat/Call: Drivers and riders can communicate within the app
- Save Addresses: Riders can save the addresses of destinations they go to often
- Save Credit Card Details: Riders can add their credit card details
- Delete Credit Card Details: Riders can delete their credit card details
- Book Ride in Advance: Riders can either book a trip for now or later
- View/Manage Account: Riders/Drivers are able to view their account and manage their account

3.5 Carpo



Launched by a student from University of Maryland to help students to carpool to/from school. Target to provide students another mode of commute as well as reduce demand of car park slots.

Functionalities:

- Rating: Give ratings to drivers
- Reviews: Give ratings to drivers
- Driver Register Online: Register as driver online
- View & Manage Account: Drivers and riders can manage their own account profiles
- Choose Driver: Rider able to choose driver base on the trip information Driver posted
- Platform fee: Small charges to maintain the platform
- Compute cost: Application will compute the cost of ride base on the distance
- Cashless Payment: Pay with credit card
- Save Credit Card Details: Save credit card information in application for future use
- Delete Credit Card Details: Remove credit card details when user wants to
- Driver Availability for upcoming days: Driver upload availabilities of the upcoming days base on his schedule so Rider can plan and request for their trip early
- Rider Availability: Rider upload availability so Driver able to pick Rider and make plan before actual day
- Driver Location - Destination: Driver upcoming trip for the day so Rider can find carpool service for last minute trip
- Fixed Single-Side location (to/from SIM): Location is either to or from school
- Profile Picture: Both Driver & Rider to upload profile picture of their own so their match could identify them and also for safety purpose

3.6 Waze Carpool



Waze is first best known as a crowdsourced live-mapping app. It then launched its first carpool app in San Francisco during 2016, and subsequently made available in the US, Mexico, Brazil, and Israel.

Functionalities:

- Ratings: Give ratings to drivers
- Reviews: Give reviews to drivers
- View & Manage Account: Drivers and riders can manage their own account profiles
- View Trips: Riders can view upcoming trips/past trips
- Manage Payment Details: Riders can add/delete their credit card details
- Choose Rider: Drivers are able to choose from the pool of Riders, for example, having to know which destination they are heading to first
- Multiple pick-ups per trips: Drivers are able to pick up more riders along the way, when heading to a supposed intended destination
- Calendar: A calendar-like page that let riders know when the drivers are driving, see who's going their way and stay on track with the reminders
- Payment Method: Credit Card, Cash
- In-App Chat: Drivers and riders can communicate within the chat
- Manage Addresses: Riders can save their addresses for convenience
- Map: Shows where the driver is currently at and the journey's runtime map
- Help and Support: Includes FAQ and the necessary point of contact when support is required

Remarks: Riders are able to include their additional comments or requests, for example, letting the Driver know that there will be wheelchair or even multiple luggage's.

4 Project Design

4.1 Overview

In this section, it will cover the designs of our product using the various diagrams and our thought processes.

4.2 System Architecture

4.2.1 Functional Requirements

Use Case Diagram

The Use-Case Diagram describes how the product is envisioned to be used from the point of view of the different end-user roles.



Use Case Descriptions

Name: Give Ratings	ID:#001
Stakeholders & Goals: Rider wants to give driver a rating	
Description: Rider can give rating to driver after the journey	
Actors: Rider	
Triggers: Rider must choose to give rating after the journey	
Normal Flow: <ol style="list-style-type: none">1. The rider ends a journey2. The system prompts the rider to give a rating/review to the driver3. The rider chooses to give a rating4. The rider selects a rating from 1 to 5 stars5. The system displays a thank you message to the rider6. End	
Sub-Flow: None	
Alternate: <ol style="list-style-type: none">1. The rider ends a journey2. The system prompts the rider to give a rating/review to the driver3. The rider chooses not to give a rating4. End	

Name: Give Reviews	ID:#002
Stakeholders & Goals: Rider wants to give driver a review	
Description: Rider can write a review for a driver after the journey	
Actors: Rider	
Triggers: Rider must choose to give review after the journey	
Normal Flow: <ol style="list-style-type: none">1. The rider ends a journey2. The system prompts the rider to give a rating/review to the driver3. The rider chooses to write a review4. The rider writes their review5. The rider submits the review6. The system displays a thank you message to the rider	

7. End
Sub-Flow: None
Alternate: <ul style="list-style-type: none">1. The rider ends a journey2. The system prompts the rider to give a rating/review to the driver3. The rider chooses not to give a review4. End

Name: Choose Driver	ID: #003
Stakeholders & Goals: Rider wants to select a driver	
Description: Rider is able to select their preferred driver	
Actors: Rider, Driver	
Triggers: None	
Normal Flow: <ul style="list-style-type: none">1. The system displays the main page2. The rider selects the carpool calendar3. The system displays the carpool calendar with the driver schedules4. The rider selects a driver5. The rider submits a request6. End	
Sub-Flow: None	
Alternate: None	

Name: Add Remarks When Requesting Ride	ID: #004
Stakeholders & Goals: Rider wants to notify the driver of any special requests	
Description: Rider can add a remark when requesting for a ride	
Actors: Rider	
Triggers: Rider must request for a ride	

Normal Flow:

1. The system displays the carpool calendar
2. The rider selects a driver
3. The rider adds a remark to the request
4. The rider submits the request
5. End

Sub-Flow: None**Alternate:** None**Name:** View Driver Availability for Upcoming Days**ID:**#005**Stakeholders & Goals:** Rider wants to view driver availability in the upcoming days**Description:** Rider can check driver's availability in the system**Actors:** Rider**Triggers:** None**Normal Flow:**

1. The system displays the list of drivers
2. The rider selects a driver
3. The system displays their schedule in the upcoming days
4. End

Sub-Flow: None**Alternate:** None**Name:** View Carpool Calendar**ID:**#006**Stakeholders & Goals:** Rider wants to view the driver's schedule**Description:** The rider is able to view the driver's schedule in the carpool calendar**Actors:** Rider**Triggers:** None

Normal Flow:

1. The system displays the main page
2. The rider selects the carpool calendar
3. The system displays the carpool calendar along with the schedule of all the drivers
4. End

Sub-Flow: None**Alternate:** None**Name:** Save Addresses**ID:**#007**Stakeholders & Goals:** Rider wants to save their frequently address**Description:** Rider is able to save their frequently use address for convenience**Actors:** Rider**Triggers:** None**Normal Flow:**

1. The system displays the rider's profile
2. The rider chooses to save address
3. The system displays the save address page
4. The rider enters the address they want to save
5. The rider saves the information
6. The system displays the save address page with the newly saved address
7. End

Sub-Flow: None**Alternate:** None**Name:** Make Payment**ID:**#008**Stakeholders & Goals:** Rider wants to make payment for a journey**Description:** Rider is able to make payment for any journeys**Actors:** Rider**Triggers:** The rider's journey has ended

Normal Flow:

1. The rider reaches their destination
2. The system prompts the rider for payment
3. The rider selects their saved credit card
4. The rider submits the payment
5. The system processes the payment
6. End

Sub-Flow: None**Alternate:**

1. The rider reaches their destination
2. The system prompts the rider for payment
3. The rider enters their credit card details
4. The rider submits the payment
5. The system processes the payment
6. End

Name: Save Payment Details**ID:**#009**Stakeholders & Goals:** Rider wants to save their payment details**Description:** The rider is able to save their payment details for convenience**Actors:** Rider**Triggers:** None**Normal Flow:**

1. The system displays the main page
2. The rider selects the payment option
3. The system displays the payment page
4. The rider selects card
5. The system displays the rider's current payment options
6. The rider selects save credit card details
7. The rider enters their credit card details
8. The rider saves the page
9. The system displays the rider's current payment options with the newly saved credit card details
10. End

Sub-Flow: None**Alternate:** None

Name: Delete Payment Details	ID:#010
Stakeholders & Goals: Rider wants to delete their payment details	
Description: The rider is able to delete their payment details	
Actors: Rider	
Triggers: The rider must have previously saved payment details	
Normal Flow: <ol style="list-style-type: none">1. The system displays the main page2. The rider selects the payment option3. The system displays the payment page4. The rider selects card5. The system displays the rider's current payment options6. The rider selects delete credit card details7. The rider selects the credit card to delete8. The rider saves the page9. The system displays the rider's current payment options without the deleted credit card details10. End	
Sub-Flow: None	
Alternate: None	

Name: View Upcoming & Current Trips	ID:#011
Stakeholders & Goals: Rider/Driver wants to view any upcoming trips	
Description: The rider/driver is able to view any upcoming and also current trips	
Actors: Rider, Driver	
Triggers: The rider/driver must have an upcoming trip booked	
Normal Flow: <ol style="list-style-type: none">1. The system displays the main page2. The rider/driver chooses activity3. The system displays the activity page4. The rider/driver selects upcoming trip5. The system displays a list of upcoming trips6. The rider/driver selects a trip7. The system displays the details of the selected upcoming trip	

8. End
Sub-Flow: None
Alternate: None

Name: View History	ID:#012
Stakeholders & Goals: Rider/driver wants to view past trips	
Description: The rider/driver is able to view any past trips	
Actors: Rider, Driver	
Triggers: The rider/driver must have made at least one trip before	
Normal Flow: <ol style="list-style-type: none"> 1. The system displays the main page 2. The rider/driver chooses activity 3. The system displays the activity page 4. The rider/driver selects history 5. The system displays a list of past trips 6. The rider/driver selects a trip 7. The system displays the details of the past trip 8. End 	
Sub-Flow: None	
Alternate: None	

Name: Use In-App Chat	ID:#013
Stakeholders & Goals: Rider/Driver wants to communicate with each other	
Description: Rider/Driver is able to send messages to each other using the in-app chat	
Actors: Rider, Driver	
Triggers: The rider needs to have booked a trip with the driver/ The driver needs to have trip booked by the rider	

Normal Flow:

1. The system displays the main page
2. The rider/driver selects chat
3. The system displays a list of chats with riders/drivers from upcoming rides
4. The rider/driver selects a chat
5. The system displays the in-app chat room between the rider and driver
6. The rider/driver sends a message
7. The message is displayed in the chat
8. End

Sub-Flow: None**Alternate:** None**Name:** View Receipt**ID:**#014**Stakeholders & Goals:** Rider/Driver wants to view a receipt from a trip**Description:** Rider/Driver can view a receipt from a past trips**Actors:** Rider, Driver**Triggers:** The rider/driver needs to have made a trip before**Normal Flow:**

1. The system displays the main page
2. The rider/driver selects payment
3. The system displays the payment page
4. The rider/driver selects receipt
5. The system displays a list of receipts from the rider's/driver's past trips
6. The rider/driver chooses a receipt
7. The system displays the receipt for the chosen trip
8. End

Sub-Flow: None**Alternate:** None**Name:** View Account**ID:**#015**Stakeholders & Goals:** Rider/driver wants to view their account/profile**Description:** The rider/driver is able to view their account details/profile

Actors: Rider, Driver
Triggers: None
Normal Flow: <ol style="list-style-type: none">1. The system displays the main page2. The rider/driver selects profile3. The system displays the account profile4. End
Sub-Flow: None
Alternate: None

Name: Manage Account	ID: #016
Stakeholders & Goals: Rider/Driver wants to manage their account details	
Description: The rider/driver is able to manage their account details	
Actors: Rider, Driver	
Triggers: None	
Normal Flow: <ol style="list-style-type: none">1. The system displays the main page2. The rider/driver selects profile3. The system displays the account profile4. The rider/driver selects manage account5. The system allows the rider/driver to manage their account details6. End	
Sub-Flow: None	
Alternate: None	

Name: Edit Profile Picture	ID: #017
Stakeholders & Goals: Rider/Driver wants to edit their profile picture	
Description: The rider/driver is able to edit their profile picture	

Actors: Rider, Driver
Triggers: None
Normal Flow: <ol style="list-style-type: none"> 1. The system displays the main page 2. The rider/driver selects profile 3. The system displays the account profile 4. The rider/driver selects manage account 5. The system allows the rider/driver to manage their account details 6. The rider/driver chooses to edit profile picture 7. The system prompts the rider/driver to choose a profile picture 8. The rider/driver selects a profile picture 9. The rider/driver saves changes 10. The system displays the account profile with the edited profile picture 11. End
Sub-Flow: None
Alternate: None

Name: Indicate Pickup Location	ID:#018
Stakeholders & Goals: Rider wants to indicate their pickup location	
Description: The rider is able to indicate their pickup location	
Actors: Rider	
Triggers: The rider needs to make a request to a driver	
Normal Flow: <ol style="list-style-type: none"> 1. The system displays the main page 2. The rider selects the carpool calendar 3. The system displays the carpool calendar 4. The rider selects a driver 5. The rider makes a request 6. The rider indicates their pickup location in the request 7. The rider submits the request 8. End 	
Sub-Flow: None	
Alternate: None	

Name: Choose Rider	ID:#019
Stakeholders & Goals: Driver wants to choose a rider	
Description: The driver is able to choose which rider to accept	
Actors: Driver	
Triggers: The driver needs to have received a request from a rider	
Normal Flow: <ol style="list-style-type: none">1. The system displays the main page2. The driver selects activity3. The system displays the activity page4. The driver selects requests5. The system displays the list of requests received by the driver6. The driver chooses a request7. The driver views the request details8. The driver indicates pick up time9. End	
Sub-Flow: None	
Alternate: <ol style="list-style-type: none">1. The system displays the main page2. The driver selects activity3. The system displays the activity page4. The driver selects requests5. The system displays the list of requests received by the driver6. The driver chooses a request7. The driver views the request details8. The driver selects reject9. End	

Name: Register Online	ID:#020
Stakeholders & Goals: User wants to register account online	
Description: User is able to register via the application	
Actors: User	
Triggers: None	
Normal Flow:	

<ol style="list-style-type: none">1. The system displays the main page2. The user selects Registration3. The system displays the registration form4. The user fills in the registration form5. The user submits the form6. The system displays a confirmation message7. End
Sub-Flow: None
Alternate: None

Name: Post Carpool Calendar	ID:#021
Stakeholders & Goals: Driver wants to post their schedule	
Description: The driver is able to post their schedule in the carpool calendar	
Actors: Driver	
Triggers: None	
Normal Flow: <ol style="list-style-type: none">1. The system displays the main page2. The driver selects the carpool calendar3. The system displays the carpool calendar4. The driver chooses to post schedule5. The system displays a page for the driver to enter their schedule details6. The driver enters their schedule details7. The driver submits the schedule8. The system updates the carpool calendar with the new schedule details9. End	
Sub-Flow: None	
Alternate: None	

Name: Compute Cost	ID:#022
Stakeholders & Goals: Application platform has to be able to compute the cost of the trips	
Description: The application platform is able to compute the cost of the trips	

Actors: Platform
Triggers: There must be requests made by riders in the system
Normal Flow: <ol style="list-style-type: none">1. Driver/Rider create a trip2. System compute the cost base on distance and platform fee3. End
Sub-Flow: None
Alternate: None

Name: Indicate Pick Up Time	ID: #023
Stakeholders & Goals: Driver to indicate pick up time	
Description: Driver will indicate pick up time if they are comfortable with the pick up location indicated by Rider.	
Actors: Driver	
Triggers: There must be requests made by riders in the system	
Normal Flow: <ol style="list-style-type: none">1. The system displays the main page2. The driver selects activity3. The system displays the activity page4. The driver selects requests5. The system displays the list of requests received by the driver6. The driver chooses a request7. The driver views the request details8. The driver selects indicate pick up time9. End	
Sub-Flow: None	
Alternate: None	

Name: View Pickup Time	ID: #024
Stakeholders & Goals: Rider wants to view the pickup time	
Description: The rider is able to check the pickup time indicated by the driver	
Actors: Rider	
Triggers: The driver needs to have accepted a request from the rider and indicated a pickup time	
Normal Flow: <ol style="list-style-type: none">1. The system displays the main page2. The rider selects activity3. The system displays the activity page4. The rider selects requests5. The system displays a list of requests made by the rider6. The rider selects a request7. The system displays the information about the request, including the pickup time indicated by the driver8. The rider confirms the request9. The system displays the information about the confirmed trip10. End	
Sub-Flow: None	
Alternate: None	

Name: Switch To Rider	ID: #025
Stakeholders & Goals: Driver wants to switch to Rider interface	
Description: User that chose As Driver at the start wants to switch to Rider interface	
Actors: Rider, Driver	
Triggers: None	
Normal Flow: <ol style="list-style-type: none">1. The system displays the main page2. The user clicks back button3. The user clicks As Rider4. End	

Sub-Flow: None
Alternate: None

Name: Switch To Driver	ID:#026
Stakeholders & Goals: Rider wants to switch to Driver interface	
Description: User that chose As Rider at the start wants to switch to Driver interface	
Actors: Rider, Driver	
Triggers: None	
Normal Flow: <ol style="list-style-type: none">1. The system displays the main page2. The user clicks back button3. The user clicks As Driver4. End	
Sub-Flow: None	
Alternate: None	

Name: Trip Cancellation	ID:#027
Stakeholders & Goals: Rider/Driver wants to cancel a trip	
Description: Rider/Driver is able to cancel a trip	
Actors: Rider, Driver	
Triggers: Rider/Driver must have an upcoming trip	
Normal Flow: <ol style="list-style-type: none">1. The system displays the main page2. The rider/driver selects activity3. The system displays the activity page4. The rider/driver selects upcoming5. The system displays the list of upcoming trips6. The rider/driver selects a trip7. The system displays the information about the trip	

<ol style="list-style-type: none"> 8. The rider/driver chooses to cancel the trip 9. The system displays a confirmation message 10. The respective party receives a mobile push notification on the cancellation 11. End
Sub-Flow: None
Alternate: None

Name: Account Management	ID:#028
Stakeholders & Goals: Admin of the application to maintain the system user accounts	
Description: Admin is able to remove or add user account	
Actors: Admin	
Triggers: None	
Normal Flow: <ol style="list-style-type: none"> 1. Log in with admin account 2. The system displays admin main menu view 3. Click on "User Accounts" 4. Click "Add User" 5. Enter the user's information 6. Click "Save" 7. End 	
Sub-Flow: None	
Alternate: <ol style="list-style-type: none"> 1. Log in with admin account 2. The system displays admin main menu view 3. Click on "User Accounts" 4. Click "Remove User" 5. Select the user account to be removed 6. Click "Save" 7. End 	

Name: Add / Remove school selections	ID:#029
Stakeholders & Goals: Admin able to add or remove schools from the selections	
Description: To be able to have more schools available for selection during registration	
Actors: Admin	
Triggers: None	
Normal Flow: <ol style="list-style-type: none">1. Log in with admin account2. The system displays admin main menu view3. Click on "School Selections"4. Click "Add School"5. Enter the name of the school6. Click "Save"7. End	
Sub-Flow: None	
Alternate: <ol style="list-style-type: none">1. Log in with admin account2. The system displays admin main menu view3. Click on "School Selections"4. Click "Remove School"5. Select the school to be removed from selections6. Click "Save"7. End	

4.2.2 Non-functional Requirements

Performance and Scalability

- The mobile application that is developed for the use of carpooling has to be accessible at all times
- It should have up-to-date information of the available rides as put up by the drivers
- The app should undergo a beta test run before the release to ensure that the application runs smoothly
- One of the prominent features of this application is having our fixed location being SIM campus

Portability and Compatibility

- The mobile application has to be compatible and work in both Android and iOS
- Minimally Required Operating System:
 - Android Version 8.0 and above
 - iOS Version 11.0 and above
- The mobile application must be made available and can be downloaded from both google play store and iOS app store
- The average mobile application size for Android is 11.5MB and as for iOS, it is 34.3MB as per the figures in Year 2020

Reliability, Availability and Maintainability

- The mobile application should be displayed on full screen across all mobile devices
- The developers developing the mobile application should try out the application from the perspective of a user so as to fix any bugs and also make improvements to areas they feel can be improved

Security

- Application should be up to date at all times, in order to match the security aspects area
- Application must be made secure, such that user's credit card details and account information are securely protected
- Must have at least framework and infrastructure to match the security standards

Localization

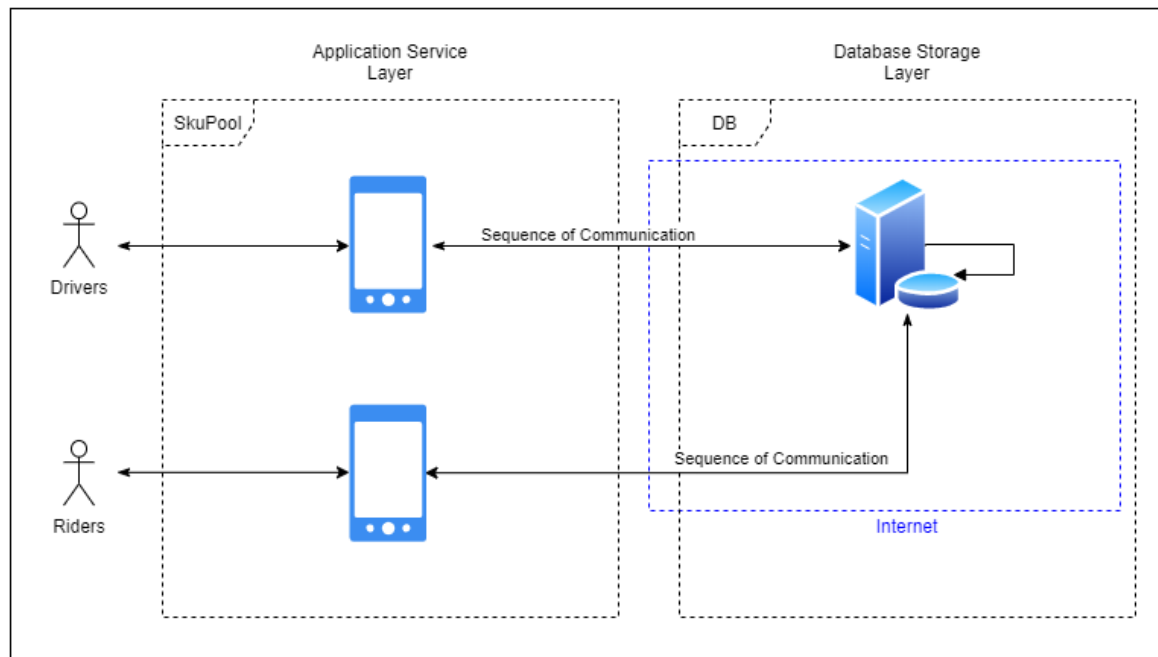
- The User Interface should display English as the main language, and there if there are going to be other languages in the application, it should be able to support

Usability

- The application should be user friendly and easy to use
- User Experience has to be to the extent that the user feels at ease while using the application and doesn't have to figure out everything before using

4.2.3 Architectural Design

This section illustrates a high-level overview of our system, which mainly consists of the communication between the drivers and riders on the application layer. It also illustrates that network layer existed throughout most of the process as internet is required in order to relay the communication to the database storage. Whereby all communications in the form of data are stored in the database and subsequently queried from there via the application layer from the driver's and rider's perspective respectively.

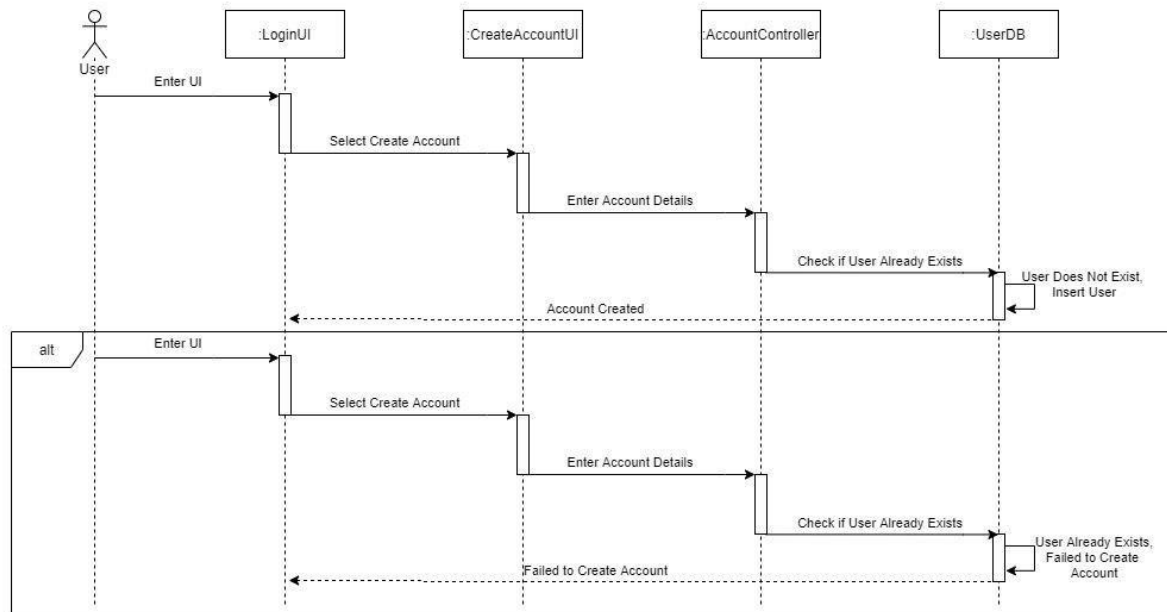


4.2.4 System Design

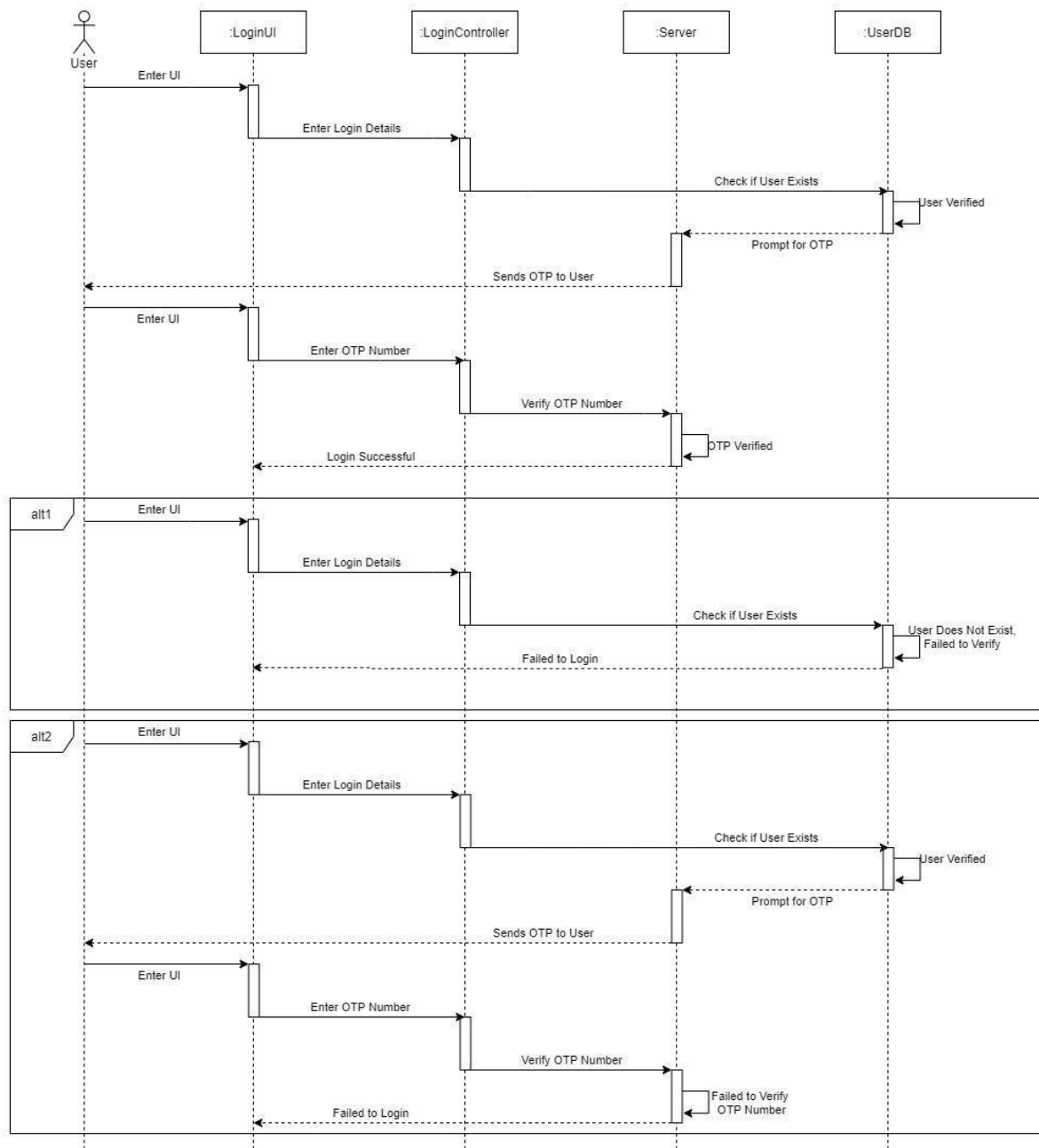
This section illustrates the process of defining the components, modules, interfaces, and data of our system to satisfy the specified requirements. It demonstrates the processes, practices, models and methodologies used to develop the system in the form of diagrams mentioned in this section.

Sequence Diagrams

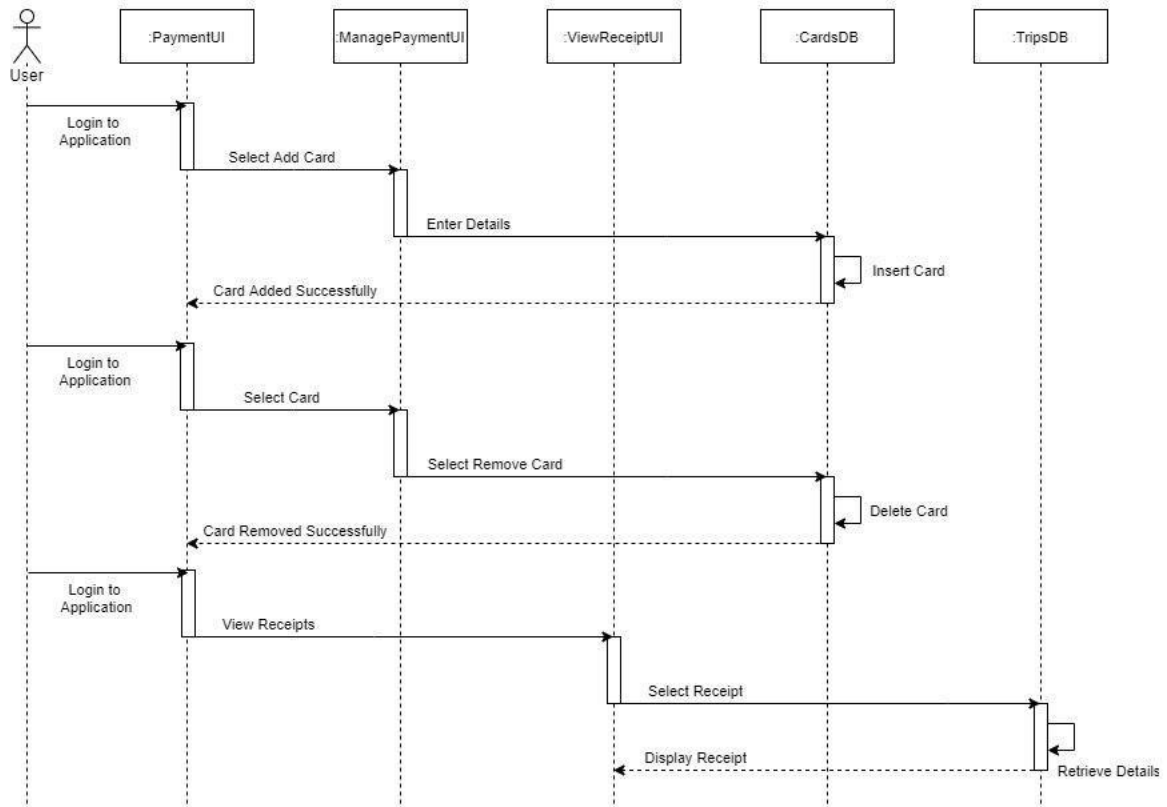
Create Account



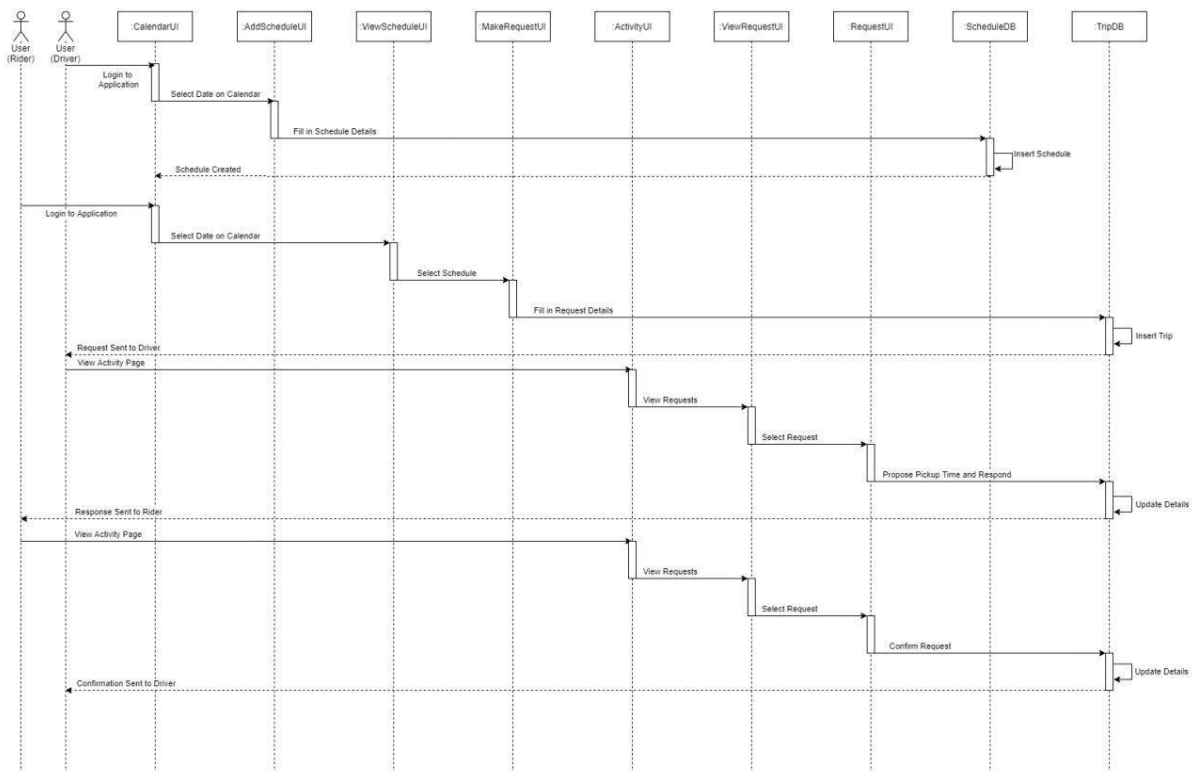
Login

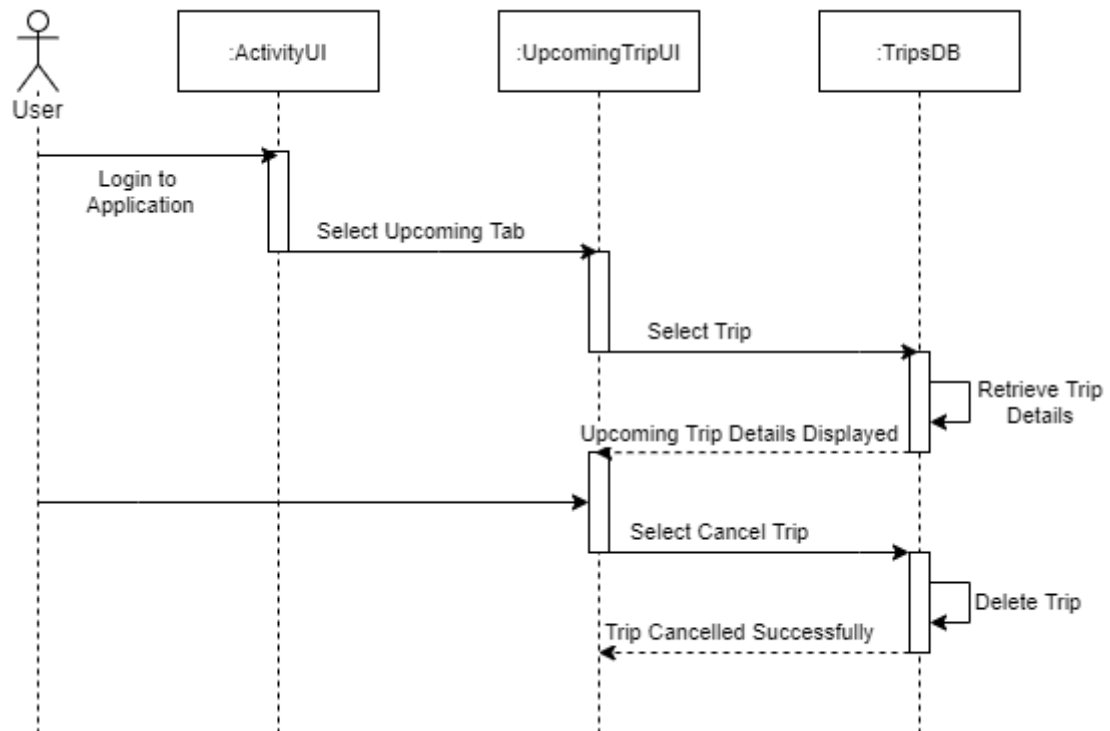


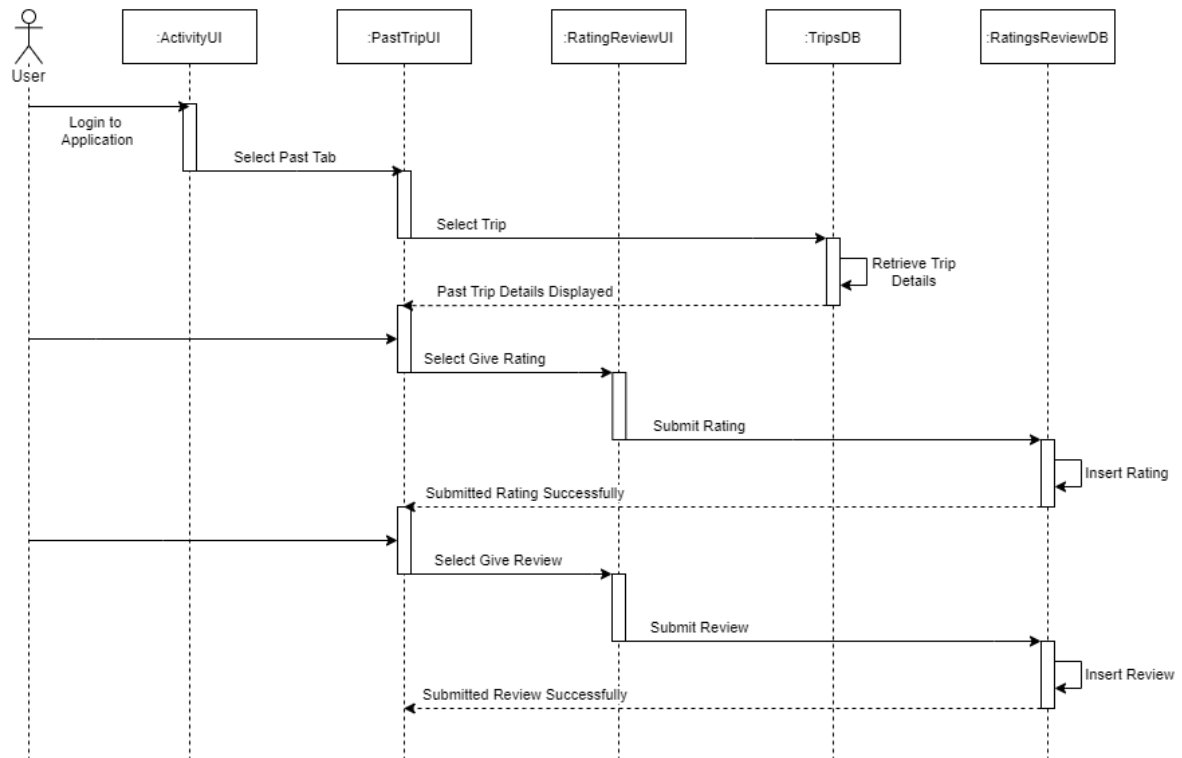
Payment Management

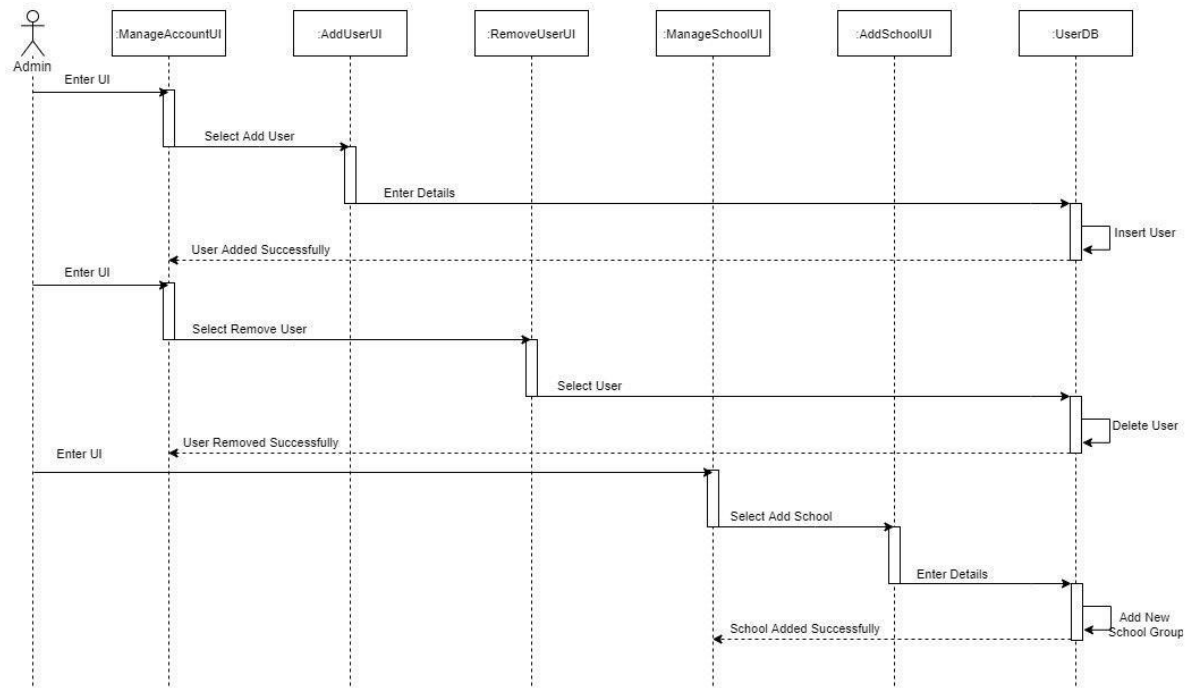


Posting Schedules and Booking Trips



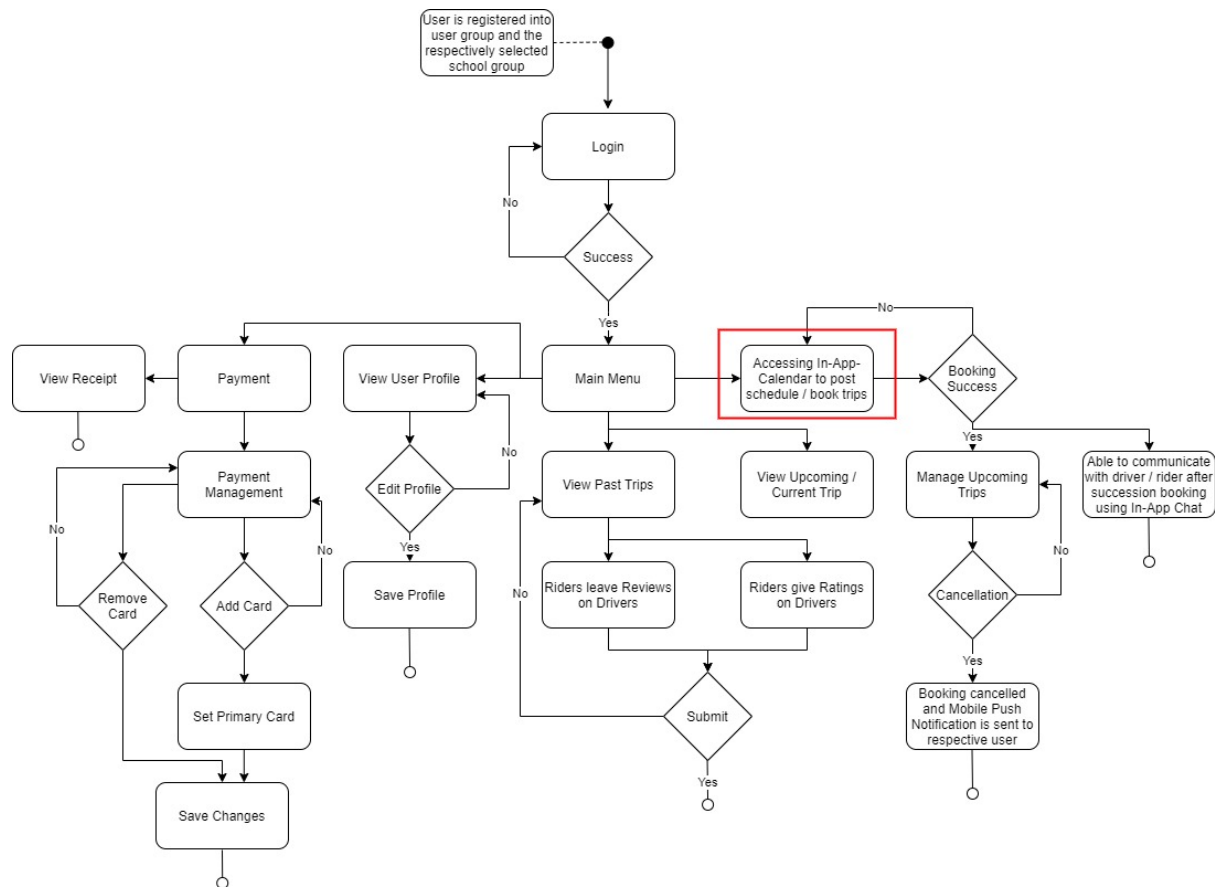
Manage Upcoming / Current Trips

Viewing Past Trips and Posting Ratings/Reviews

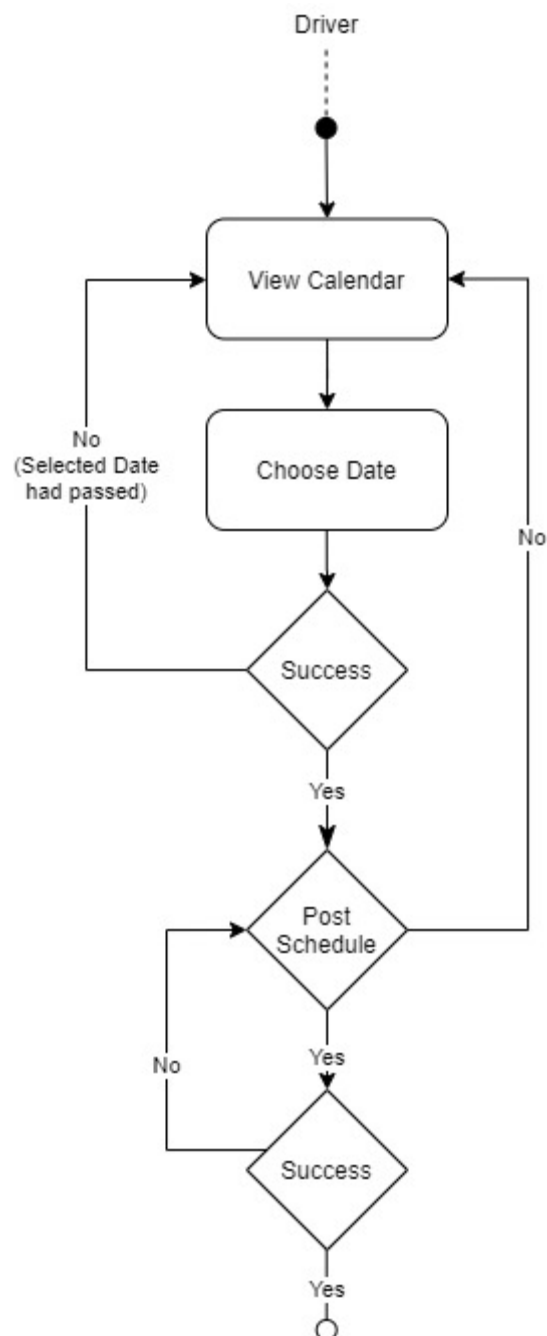
Admin User's Sequential Flow

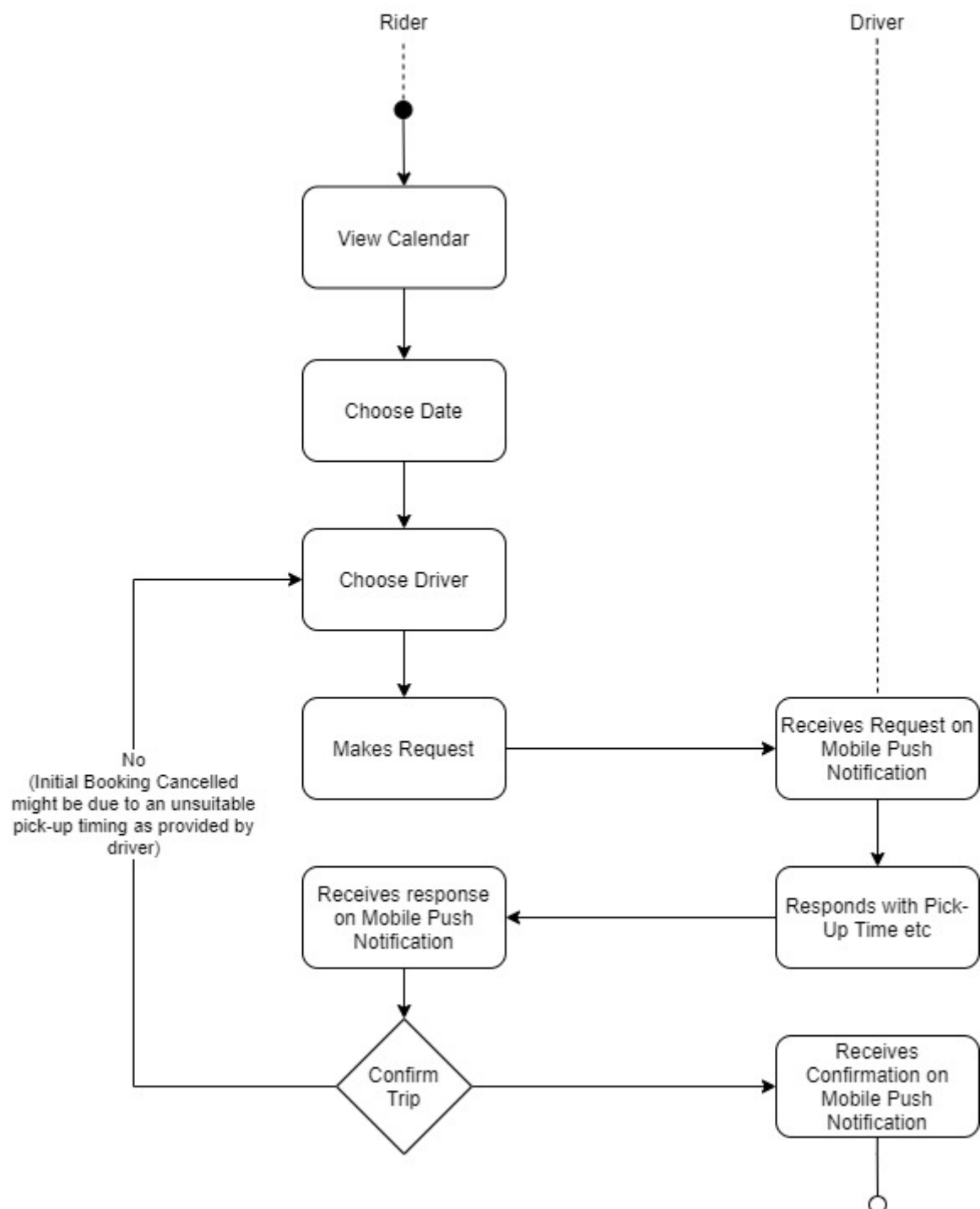
Activity Diagrams

Users – Driver's and Rider's

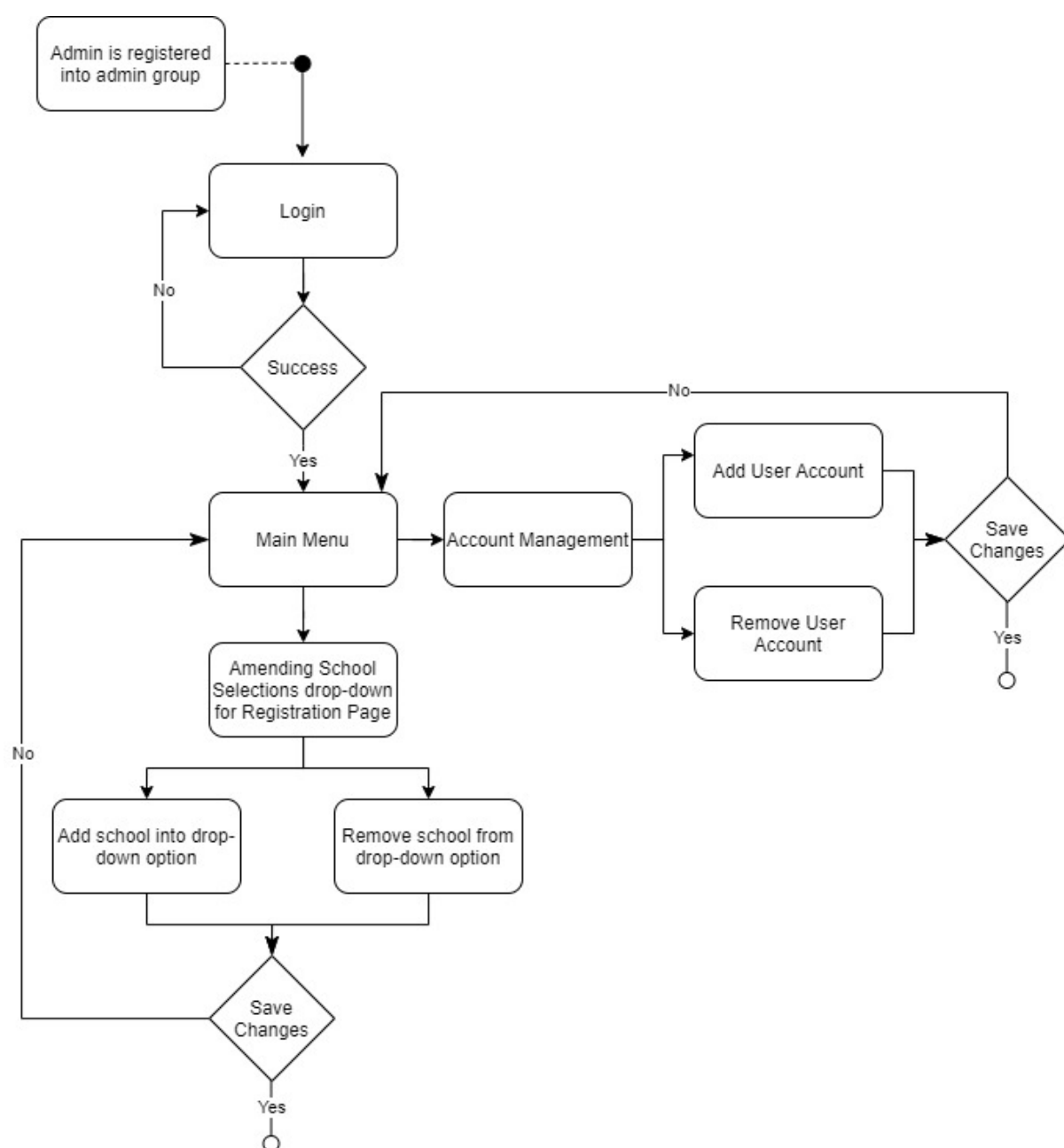


The activity “Accessing In-App-Calendar to post schedule / book trips”, as indicated in red in the above diagram, will be further illustrated in blow-up diagrams during the subsequent portions as there are a series of activities during this specific activity.

Blow-up Diagram of “Accessing In-App-Calendar to post schedule / book trips”**Driver’s Posting of Schedule**

Blow-up Diagram of “Accessing In-App-Calendar to post schedule / book trips”**Rider’s Booking of Trips**

Admin Users

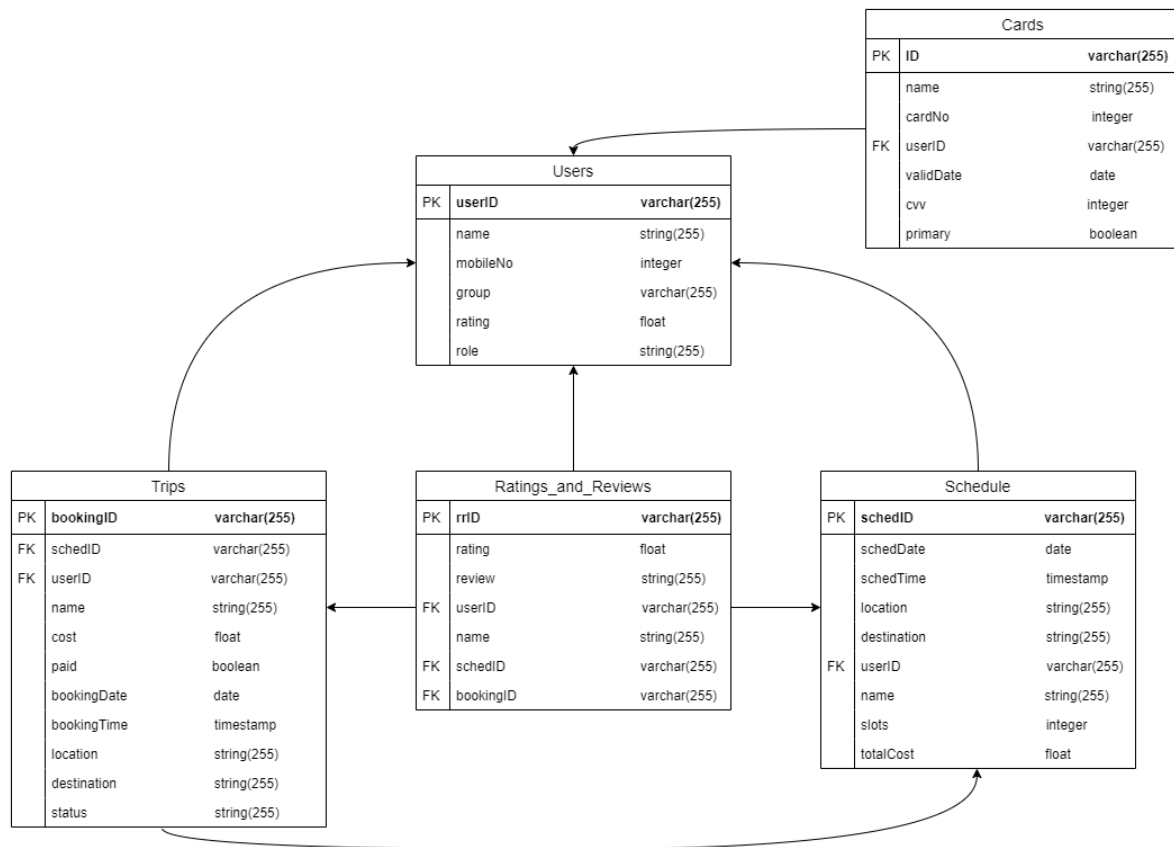


4.2.5 Database Design

Overview

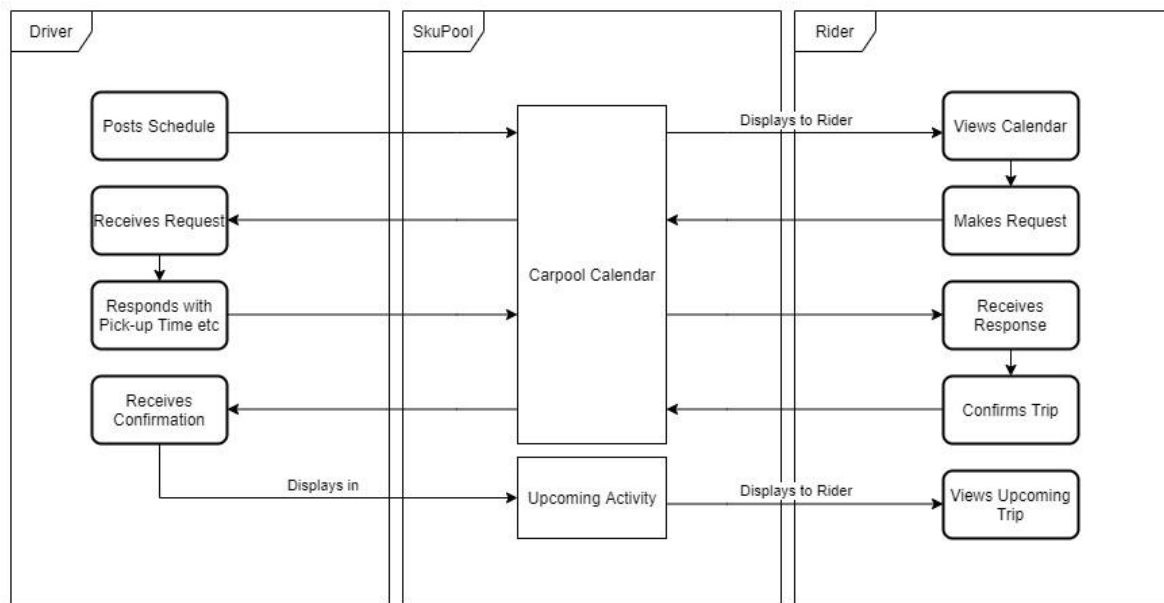
This section describes how the database design of our application is illustrated.

Database Schema Design



The Database Schema Design, as seen above, illustrates the relationship between the **Users** table with the other tables. It will establish association between the tables, with the UserID being the connection point used. The reason why we segregate these other tables, **Cards**, **Trips**, **Schedule**, **Ratings_and_Reviews**, and create them separately, is because there might be multiple entries relating to the same user. Therefore, it will be more efficient when it comes to querying the data when the tables are separated accordingly.

4.2.6 Data Flow Diagram for Booking Trips



This data flow diagram illustrates the main data flow of our application, with it being the driver posting schedule to the rider booking of trips, through mainly the In-App-Calendar and the other application functionalities.

4.3 User Interface Design

4.3.1 Overview

This section illustrates the mock-up User Interface (UI) of SkuPool. The UI design shown in this section is subjected to review and may vary from the final GUI of the end product.

4.3.2 Login Page

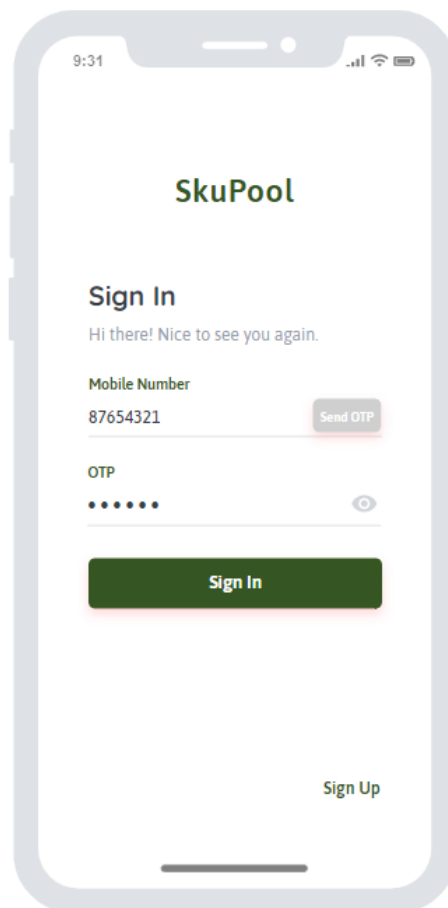
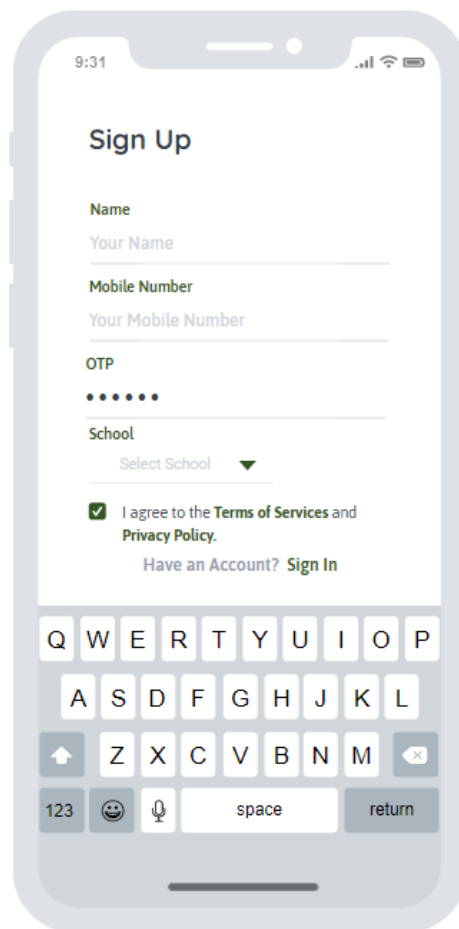


Figure 1: Login Page for Users

As seen in Figure 1, users who have not registered for their account will have to do so by clicking on the 'Sign Up' button as located at the bottom right of the login page. Registered users are required to sign in using their registered mobile number and an OTP which will be sent to the provided mobile number. No password is intended as OTP will be in place instead for logging in.

4.3.3 Register Account Page



The image shows a mobile application interface for account registration. The screen is titled "Sign Up". It contains several input fields: "Name" with a placeholder "Your Name", "Mobile Number" with a placeholder "Your Mobile Number", "OTP" with a placeholder "•••••", and "School" with a placeholder "Select School" and a dropdown arrow. Below these fields is a checkbox labeled "I agree to the Terms of Services and Privacy Policy." and a link "Have an Account? Sign In". A virtual keyboard is visible at the bottom of the screen.

Figure 2: Account Registration Page

All first-time users of the application are required to register for an account. First-time users are required to enter all the mandatory fills in the account registration page as shown on Figure 2. That includes an OTP that will be sent to the provided mobile number for verification purposes. In the event that the user did not receive the OTP, they can request for another one again after 60 seconds. They are also required to choose from a list of schools from the drop-down option. This will then be their desired destination or pick-up point for the SkuPool application and they will only be able to interact with users from the same school.

4.3.4 Menu Taskbar

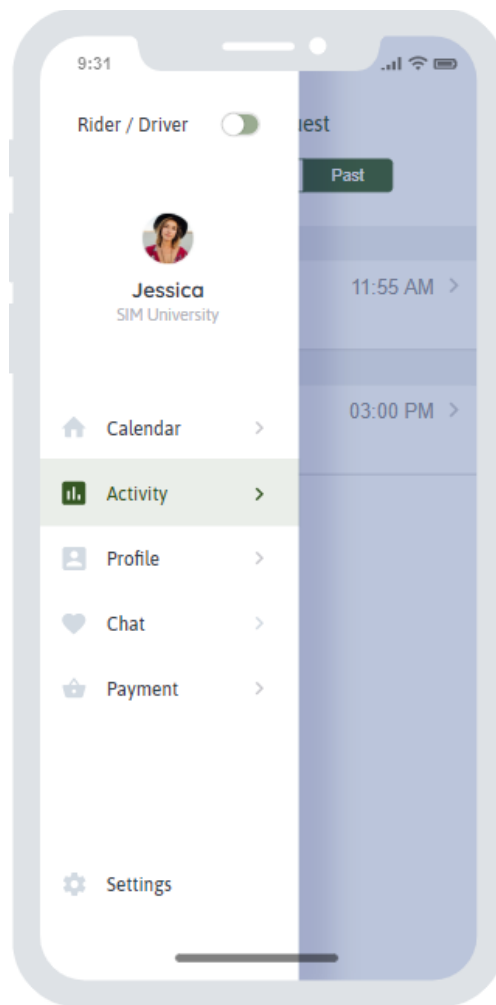


Figure 3: Menu Taskbar

Upon successfully login to the application, they will be able to access the menu page with the features such as User's Profile Information, In-App Calendar, Activity, Profile, In-App Chat, Payment and Settings (Figure 3). Users can also toggle between the Driver and Rider role. These features will then navigate to the respective pages.

4.3.5 Manage Profile

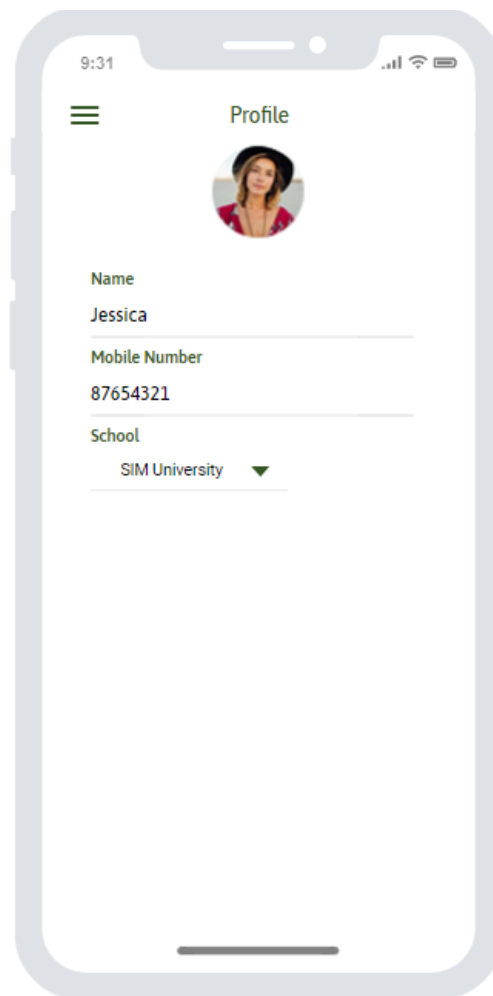


Figure 4: Profile Management

Users are able to update or edit their individual profile respectively as shown in Figure 4. They are also able to change the School option that they have chosen during the registration.

4.3.6 Manage Payment Details

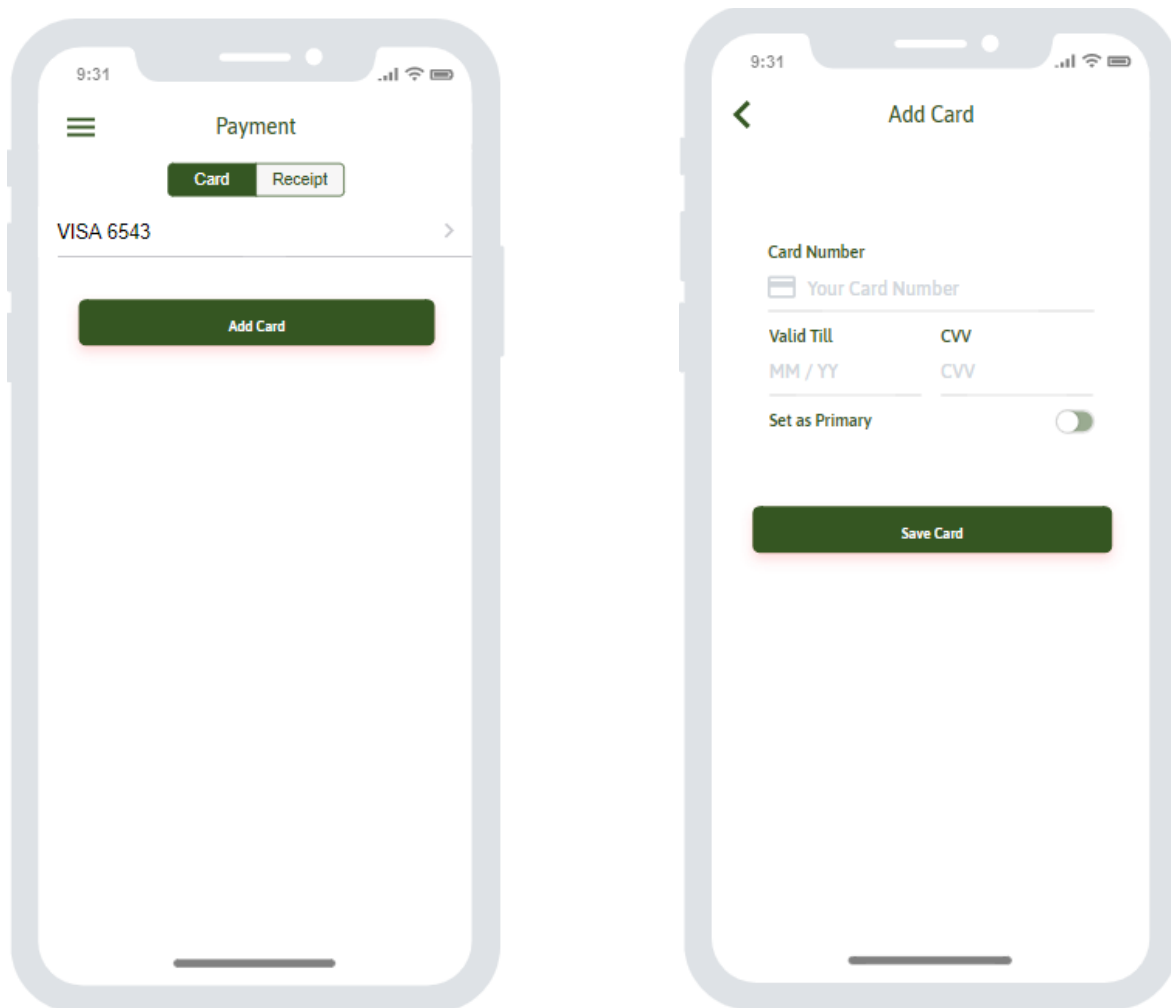


Figure 5 & 6: Managing Payment Details

The user is able to manage their payment details at the Payment page. They can view the cards that were saved previously, and also add/delete cards. To add a new card, they can click on 'Add Card'. After entering their card details, they can click 'Save Card', and the card will appear in their list of cards in the Payment page.

4.3.7 In-App Chat

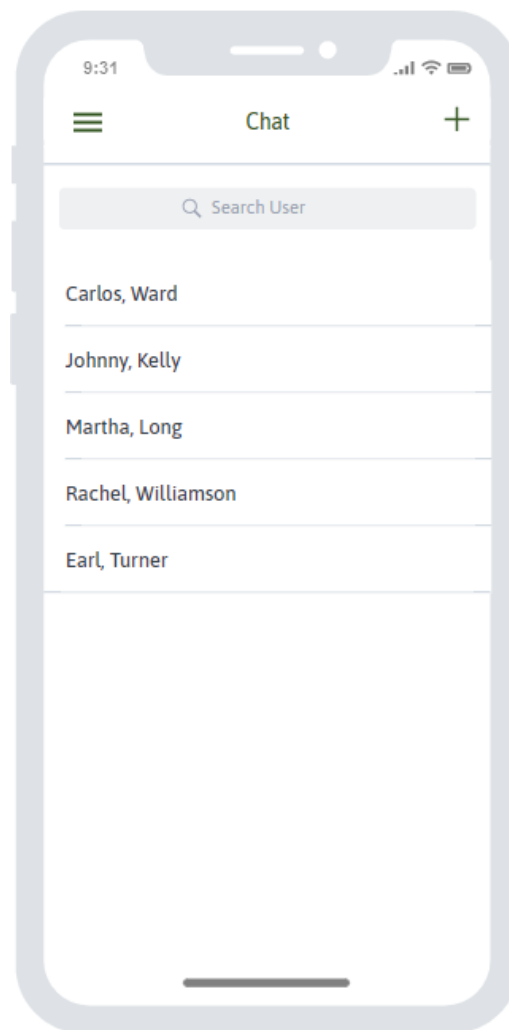


Figure 7: In-App Chat

The users can utilise the In-App Chat to communicate with other users in the same group. This will also serve as the point of communication between riders and drivers when required during the booked trip. They can only chat with users whom they have carpooled with before or made requests to.

4.3.8 In-App Calendar

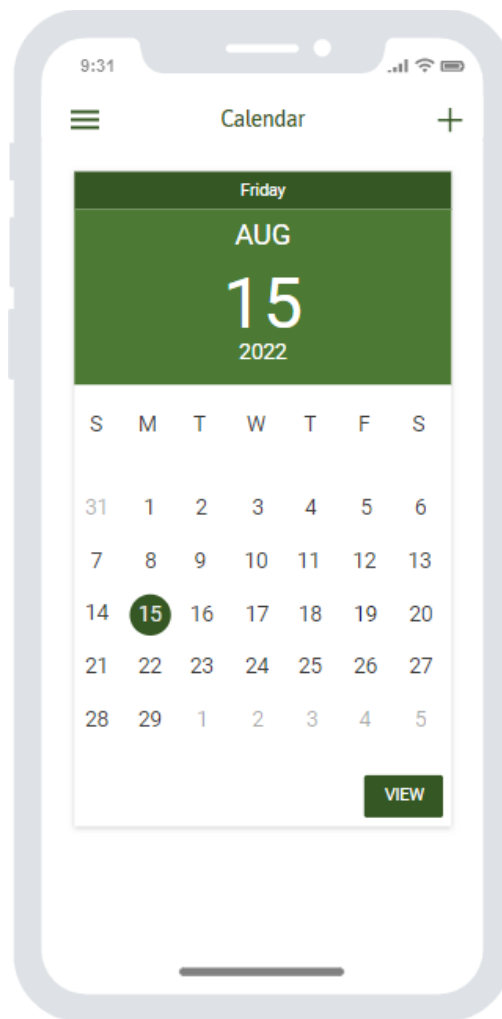
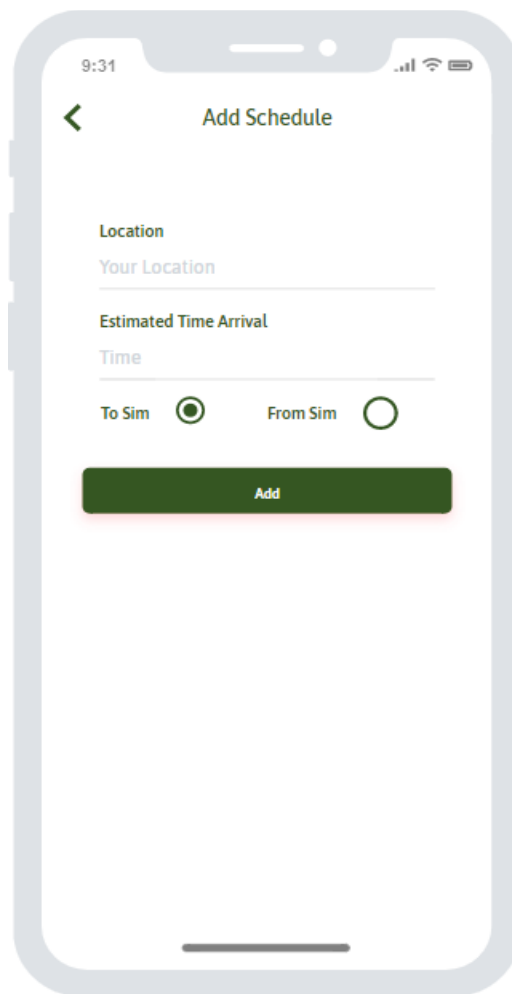


Figure 8: In-App Calendar

Users are able to view the In-App Calendar, where drivers can post their schedules and riders can look at the drivers' schedules and make requests to the drivers. Drivers can click on the '+' sign at the top right corner to add their schedules, and riders can click on any date and click 'View' to view the drivers' schedules for that day. This allows riders to have quicker and convenient access to the driver's schedule for booking on the day of their intended trip.

4.3.9 Driver Add Schedule



9:31

< Add Schedule

Location

Your Location

Estimated Time Arrival

Time

To Sim ☒ From Sim ☐

Add

Figure 9: Adding Schedule

When a driver wants to add their schedule to the calendar, they have to fill in the location they are heading to/from, estimated time of arrival, and select whether they are going to SIM or going to their designated location from SIM. After filling in the information, they can click 'Add' and the schedule will be posted on the calendar for riders to see.

4.3.10 Rider's View of Calendar

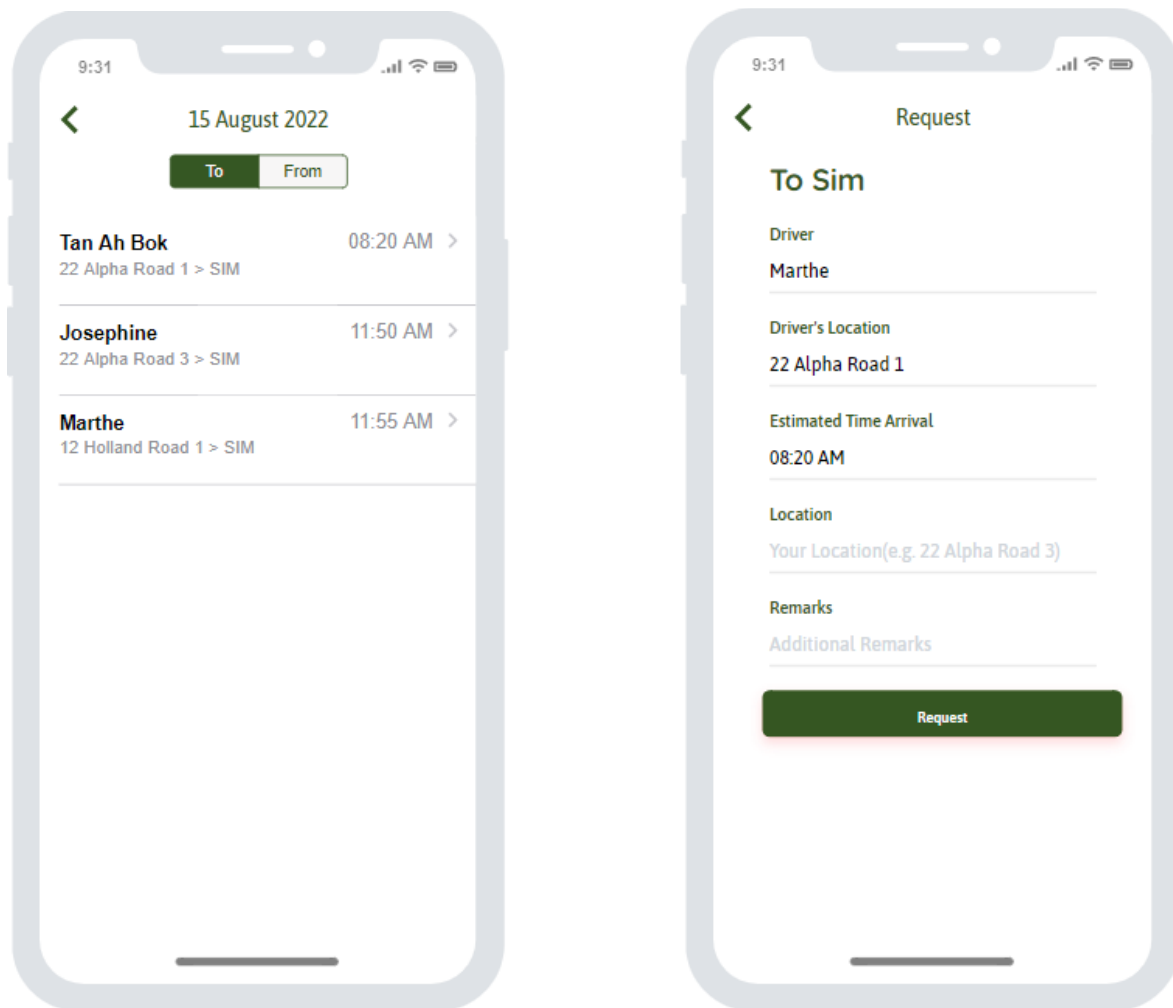


Figure 10 & 11: Rider viewing calendar and making request

When the rider chooses to view a date in the calendar, they will see a page (Figure 10) displaying the drivers available for that day. After selecting a driver, they will be brought to the request form (Figure 11), where they will proceed to fill in their pickup location and any remarks if applicable. They can then click on 'Request' and the request will be sent to the driver.

4.3.11 Activity - View Requests

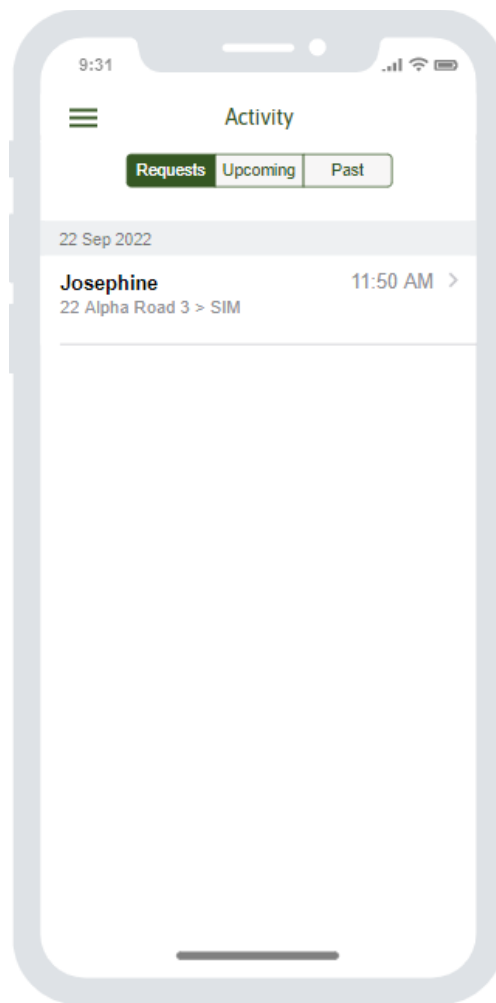
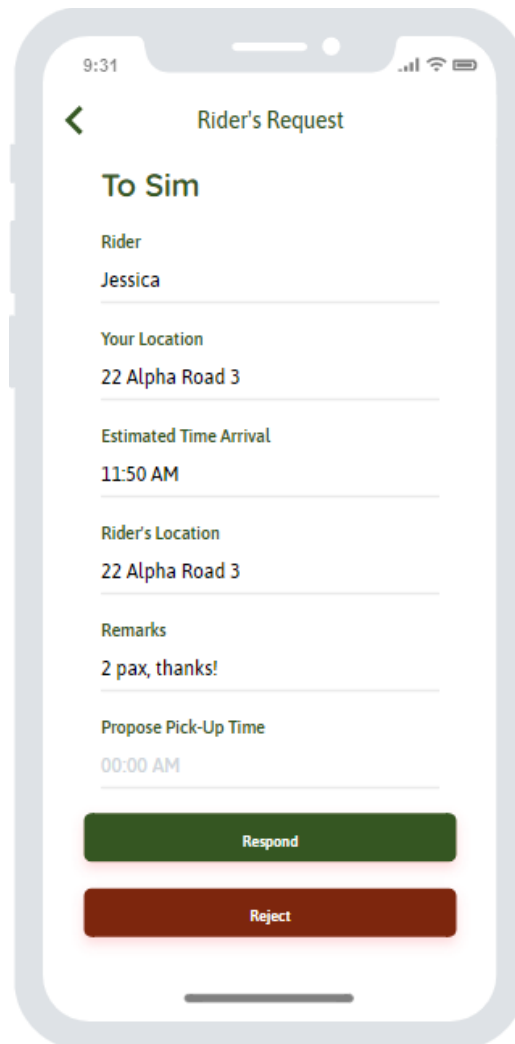


Figure 12: Viewing requests

In the Activity page, the driver is able to view the requests they have received from riders. They will be able to view any pending requests, upcoming requests, and past requests. The date and time for the request are displayed, and they will be able to see more details if they click into the request (As seen in Figure 13 on the next page).

4.3.12 Rider's Request Page



9:31

Rider's Request

To Sim

Rider
Jessica

Your Location
22 Alpha Road 3

Estimated Time Arrival
11:50 AM

Rider's Location
22 Alpha Road 3

Remarks
2 pax, thanks!

Propose Pick-Up Time
00:00 AM

Respond

Reject

Figure 13: Responding to a rider's request

After a rider has made a request to a driver, the driver receives the request and is able to view the details of the request. The request shows the rider's name, rider's location, estimated time of arrival, the driver's location, and any remarks the rider has made. The driver will then enter a pick-up time and select 'Respond' if they decide to accept the rider's request. Else, they can choose to 'Reject' the request.

4.3.13 Driver's Respond Page

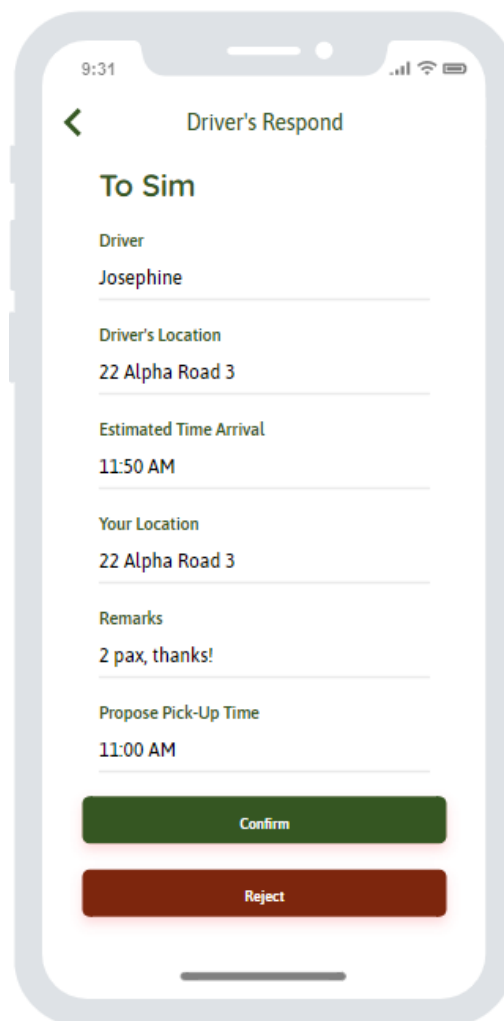


Figure 14: Rider viewing the driver's response to their request

After a driver has responded to a rider's request with a pick-up time, the rider will receive the response from the driver. The rider can then view the proposed pick-up time and choose to either 'Confirm' or 'Reject' the driver's response. If the rider chooses to confirm, the booking will be made, else it is cancelled.

4.3.14 Activity - View Upcoming / Past Trips

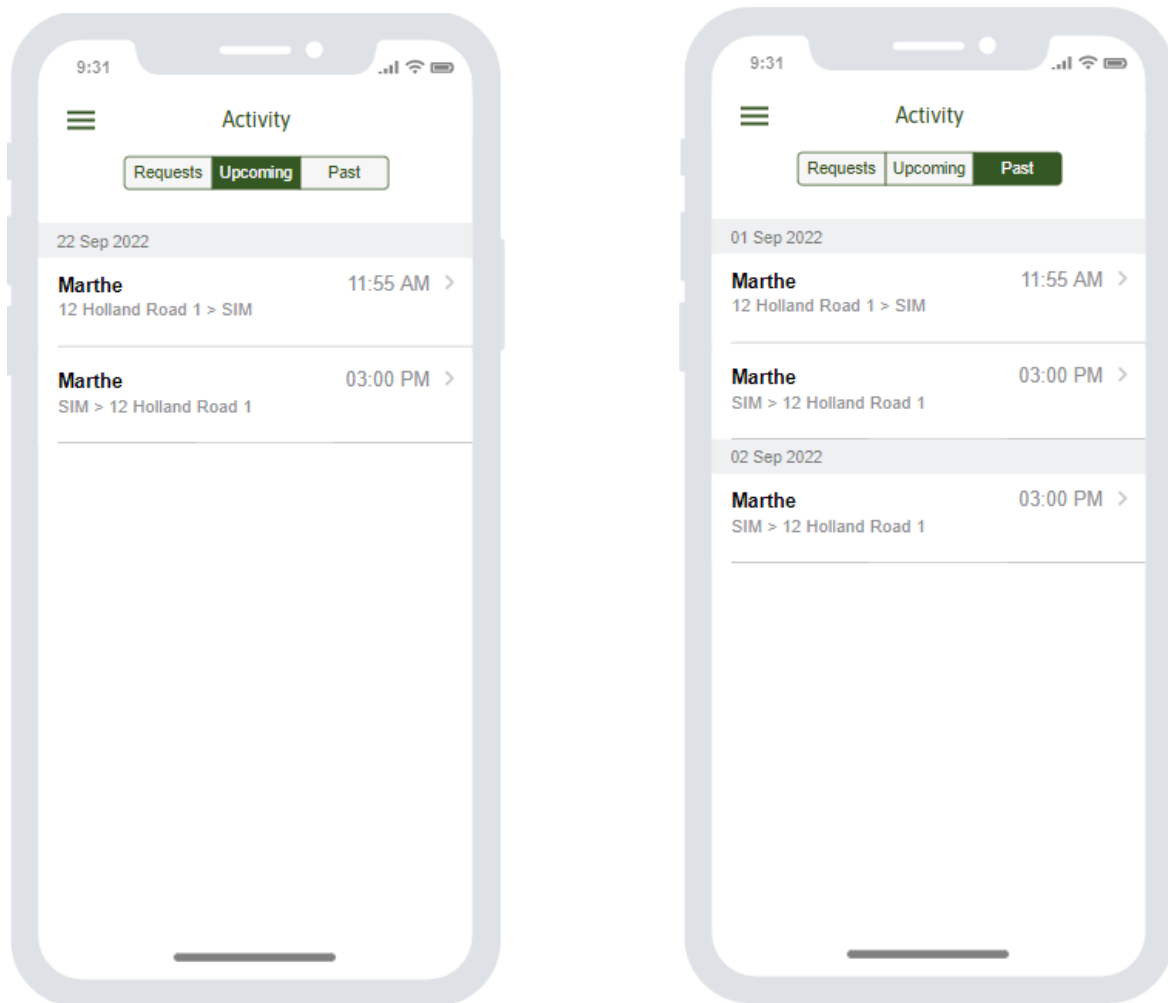


Figure 15 & 16: Viewing upcoming and past trips

Users are able to view their upcoming and past trips in the Activity page. There are tabs on the page for the user to navigate between Requests, Upcoming Trips (Figure 15), and Past Trips (Figure 16). The page will display an overview of the list of trips, and the user can select a trip to view further details of the trip.

4.3.15 View Receipts

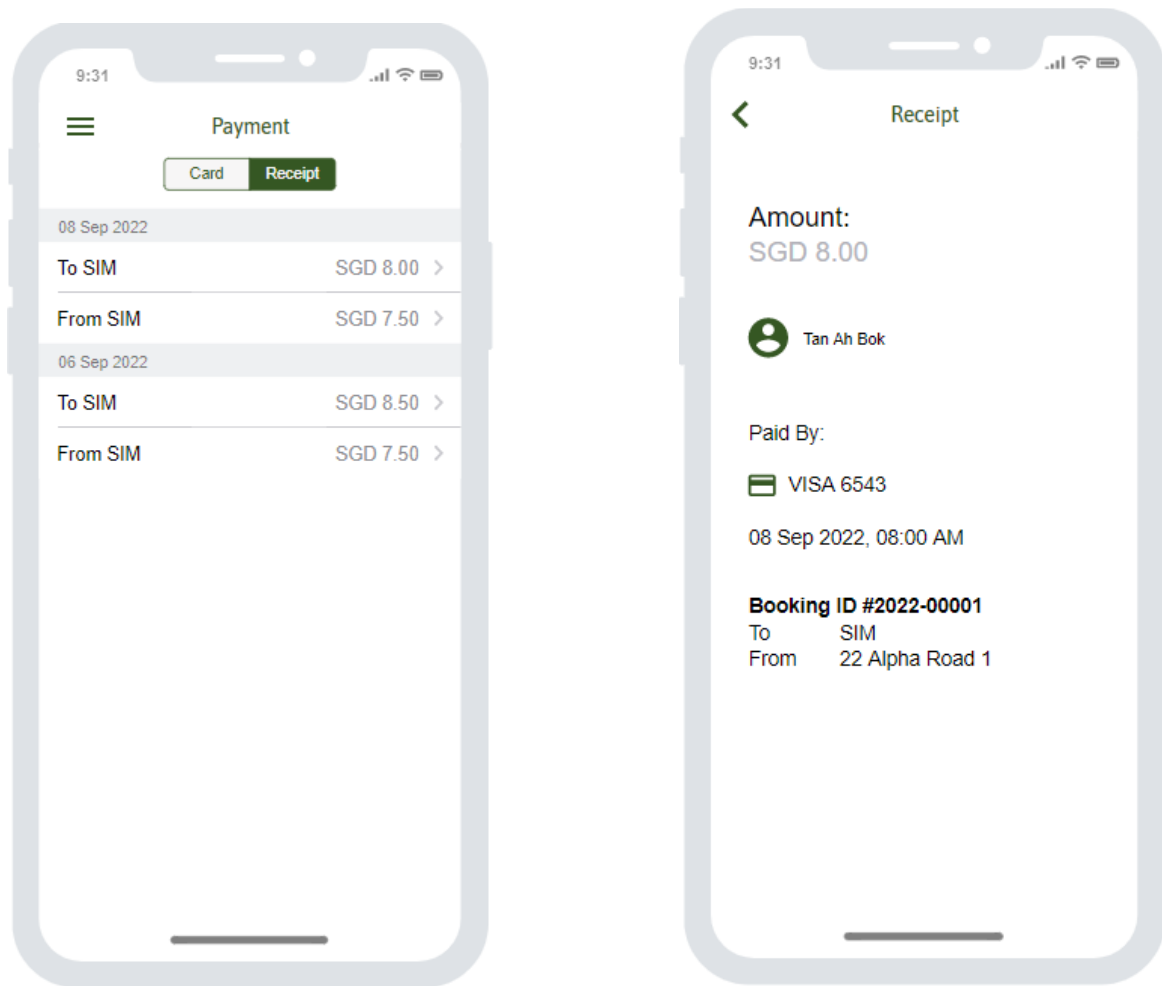


Figure 17 & 18: Viewing receipts

Users are able to view their receipts from past trips on the Payment page. They can navigate to the 'Receipt' tab where they will be shown an overview of all their receipts (Figure 17), grouped by date. Selecting a receipt will bring the user to a page (Figure 18) where the full details of the receipt is displayed.

5 Project Implementation

5.1 Overview

In this section, it describes how the project is implemented during the development time.

5.2 Agile Development Methodology



We have chosen Agile Development Methodology for our project. It works by having an iterative cycle of product release after every iteration. Functionalities of the product will be added on after every cycle and they are implemented based on the importance of every function. Progressively, shareholders are able to see the progression of the product and such that, if there's a case of requirement changes needed during the development process, our team is able to access the change and make the necessary changes. Therefore, Agile Development Methodology being the chosen methodology that is well suited for our product development.

5.3 Development Tools

5.3.1 Development Platform – React NativeA

We decided to use React Native as our development platform for the application as it is a platform that uses the commonly used Java Language and being robust for cross-platform application development. The codes can be reused to lessen the time and work involved in the implementation of the application. It comes with ready-made components that allows rapid development and modification of our program.

5.3.2 Database Platform – Firebase

Firebase is well suited to support both iOS and Android platform services. The main reason as to why we have chosen this is it is a perfect fit with our development platform React Native. Such that it will be easier to integrate the data between our developed application and databases.

5.3.3 Product Website – Wix

Wix is a free-to-use platform that allows users to develop interactive and creative websites to cater to one's marketing needs. It provides a list of available templates, with drag-and-drop components, and also tutorials as to how to implement certain components. Therefore, we find Wix to be extremely user friendly and suitable for our project especially during conceptualisation and incubation stages, when need to update the website from time to time.

5.3.4 Artwork – Wiz Logo Maker

Our project logo is created and developed using Wiz Logo Maker, which provides the necessary capabilities and functions for us to customize our logo according to our needs. The tool itself is free-to-use and blend in to our project website, since we used Wix to develop our product website.

6 Impact

6.1 Overview

In this section, we will be identifying and analysing potential issues that could negatively affect the usage of our application. This is to help prevent and resolve any issues that might arise in the future. At the same time, having to take extra precautions on those Risk Categories with Medium and High-Risk Rating.

6.2 Risk Analysis

Risk Category	Identified Risk	Risk Elements		
		Impact	Probability	Risk Rating
Market	The requirements of customers must be addressed and the need to take the current similar application platforms in the market as competitors	M	L	L
Communication	Between the client and developer should be consistent. Ensure no design errors and miscommunication	M	L	M
Resources	Platform (Compatible with both Android and iOS)	L	L	L
Technical	App should be optimized to fit the screen of all devices and ensure that it is free of bugs at all times	L	L	L
User	The needs of all the customer to be analysed and interpreted correctly or else the end product can't be delivered up to standard	M	L	M
Maintenance	Always keep the application system up to date	M	L	H
Security	Security must not be compromised and all user's information must be secured	M	M	H

Risk Rating Key	Low
	Medium
	High

7 Project Communication

Project Website: <https://skupool.wixsite.com/skupool>

7.1 Overview

The project website comprises of our team members details, brief introduction of our product and also the documentations for the project milestones. All the project documentations can be downloaded and viewed locally under the “**Project Documentations**” tab. Whereas our Project Meeting Minutes and Weekly Individual Reflective Diary can be accessed on the website itself under the “**Project Minutes**” and “**Project Diary**” respectively. All our documentations and progress updates are updated there on a regular basis, such that it can be accessible by the relevant project related personnel simply.

On top of the project website, our team make use of Google Drive as one of the communications means that we used to share and access our files. As every member is tasked on different task, there is always a need for a platform to access and update our respective part. Be it source codes, discussion topics or documentations, our team members can be in line on the same page since everyone is able to access the latest edit at any point in time. Making it easier for us to post our discussion topics and consolidate our documents.

During this period, when most of the face-to-face discussion are not possible. We used Discord for our internal group meeting and Zoom for our group meeting with the supervisor. As having a proper communication serves as a vital aspect in project work, this helps to relay messages between the team clear and establishing a stronger bond of teamwork.

7.2 Project Website Run-Through

The website might subject to changes during the Final Year Project Phase 2 as more details on our developed application may be included.

For a start, when accessing the project website, it will bring us to the home page. It is simple and user-friendly to begin with, having the brief introduction on our project, as well as, our team members information located on the bottom of every webpage.



Project Documentations tab, where all our Project Progress Documentations are located.



[Home](#) | [Project Documentations](#) | [Project Minutes](#) | [Project Diary](#) | [Contact us](#)

Project Documentations

Project Requirements Documentation



Software Requirements Documentation



Technical Design Review

Project Minutes tab, where all our Project Meeting Minutes are documented.



[Home](#) | [Project Documentations](#) | [Project Minutes](#) | [Project Diary](#) | [Contact us](#)

Project Meeting Minutes

Project Meeting Minutes 11

👁 4 Write a comment

Project Meeting Minutes 10

👁 5 Write a comment

Project Meeting Minutes 9

👁 11 Write a comment

Project Meeting

Project Meeting

Project Meeting

Project Diary tab, where all our Weekly Individual Reflective Diary are documented.



[Home](#) | [Project Documentations](#) | [Project Minutes](#) | [Project Diary](#) | [Contact us](#)

Reflective Diary



Week 1 (4 Apr 2020 - 10 Apr 2020)



Week 2 (11 Apr 2020 - 17 Apr 2020)



Week 3 (18 Apr 2020 - 24 Apr 2020)



Week 4 (25 Apr 2020 - 1 May 2020)



Week 5 (2 May 2020 - 8 May 2020)



Week 6 (9 May 2020 - 15 May 2020)



Contact us tab, it will navigate users to the Contact information portion on the home page.



[Home](#) | [Project Documentations](#) | [Project Minutes](#) | [Project Diary](#) | [Contact us](#)



Navigation to the Contact information portion.

CONTACT

Project Assessor: Mr Sionggo Japit
Email: sjapit@uow.edu.au

Project Supervisor: Mr Premarajan Ponnambath
Email: premrajan_p@yahoo.com

Lim Kang Wei (Group Leader)
Email: kwlim018@mymail.sim.edu.sg

Jessica Soh Li Jing
Email: jljs0h002@mymail.sim.edu.sg

Lim Kia Yin, Natalie
Email: kynlim@mymail.sim.edu.sg

8 Project Schedule (Gantt Chart)

Abbreviations:

Name	Abbreviation
Kang Wei	KW
Jessica	J
Natalie	N

	Represents normal timeline
	Shows timeline in Sprints (2 weeks a sprint)

Projected Timeline Overview (Full)

	Week 1 4 Apr - 10	Week 2 11 Apr - 17	Week 3 18 Apr - 24	Week 4 25 Apr - 1	Week 5 2 May - 8	Week 6 9 May - 15	Week 7 16 May - 22	Week 8 23 May - 29	Week 9 30 May - 5	Week 10 6 Jun - 12	Week 11 13 Jun - 19	Week 12 20 Jun - 26	Week 13 27 Jun - 3	Week 14 4 Jul - 10	Week 15 11 Jul - 17	Week 16 18 Jul - 24	Week 17 25 Jul - 31	Week 18 1 Aug - 7	Week 19 8 Aug - 14	Week 20 15 Aug - 21
Skupool																				
Project Planning																				
Scope Planning	KW, J, N																			
Resource Planning	KW, J, N																			
Scheduling	KW, J, N																			
Research																				
Requirements Gathering from client		J, N																		
Literature Review		J, N																		
Research on similar existing platforms			KW, J, N																	
Identifying functionalities for our application				KW, J, N																
Functionalities - Functional, Non-Functional						KW, J, N														
Develop Prototypical																				
Project Website																				
Project Name and Logo		KW, J, N																		
Design layout of Website		KW																		
Working on Website			KW																	
Testing of Website				KW																
Launch of Website																				
Develop Carpool																				
Rough User-Like Diagrams and Descriptions				N																
Design Layout																				
Program Overview Layout								KW, J												
Design								KW, J												
User Interface Design								KW, J												
Database Design								KW, J												
Program Implementation																				
User Interface Creation									KW, J			KW, J		KW, J		KW, J		KW, J		
Class and Functions									KW, J			KW, J		KW, J		KW, J		KW, J		
Database Creation									KW, J			KW, J		KW, J		KW, J		KW, J		
Linking of Classes, Functions, UI and										KW, J			KW, J		KW, J		KW, J		KW, J	
Program Testing																				
Different Tests to be done										N			N		N		N		N	
Documentation to compile for that sprint										KW, J, N		KW, J, N		KW, J, N		KW, J, N		KW, J, N		
Launch of Application																				
Launch every end of 2 weeks sprint										KW, J, N		KW, J, N		KW, J, N		KW, J, N		KW, J, N		
Final Product and Presentation																				
Launch of Final Application and																				KW, J, N
Project Requirements																				
Documentation				KW, J, N																
Project Progress								KW, J, N												
Project Prototype									KW, J, N											
Final Documentation																KW, J, N				
Final Presentation																				KW, J, N

Projected Timeline Overview (Clearer View)

	Week 1 4 Apr - 10 Apr	Week 2 11 Apr - 17 Apr	Week 3 18 Apr - 24 Apr	Week 4 25 Apr - 1 May	Week 5 2 May - 8 May	Week 6 9 May - 15 May	Week 7 16 May - 22 May	Week 8 23 May - 29 May	Week 9 30 May - 5 Jun
Resource Planning	KW, J, N								
Scheduling	KW, J, N								
Research									
Requirements Gathering from client		J, N							
Literature Review		J, N							
Research on similar existing platforms		KW, J, N							
Identifying functionalities for our application				KW, J, N					
Functionalities - Functional, Non-Functional, Security						KW, J, N			
Develop Promotional Project Website									
Project Name and Logo		KW, J, N							
Design layout of Website		KW							
Working on Website		KW							
Testing of Website			KW						
Launch of Website			KW						
Develop Carpool Application									
Rough Use Case Diagrams and Descriptions				N					
Design Layout									
Program Overview Layout Design								KW, J	
User Interface Design								KW, J	
Database Design								KW, J	

	Week 10 6 Jun - 12 Jun	Week 11 13 Jun - 19 Jun	Week 12 20 Jun - 26 Jun	Week 13 27 Jun - 3 Jul	Week 14 4 Jul - 10 Jul	Week 15 11 Jul - 17 Jul	Week 16 18 Jul - 24 Jul	Week 17 25 Jul - 31 Jul	Week 18 1 Aug - 7 Aug	Week 19 8 Aug - 14 Aug	Week 20 15 Aug - 21 Aug
Program Implementation (2 weeks sprint)											
User Interface Creation	KW, J		KW, J		KW, J		KW, J		KW, J		
Class and Functions Creation	KW, J		KW, J		KW, J		KW, J		KW, J		
Database Creation	KW, J		KW, J		KW, J		KW, J		KW, J		
Linking of Classes, Functions, UI and Databases		KW, J		KW, J		KW, J		KW, J		KW, J	
Program Testing											
Different Tests to be done		N		N		N		N		N	
Documentation											
Documentations to compile for that sprint		KW, J, N		KW, J, N		KW, J, N		KW, J, N		KW, J, N	
Launch of Application											
Launch every end of 2 weeks sprint		KW, J, N		KW, J, N		KW, J, N		KW, J, N		KW, J, N	
Final Product and Presentation											
Launch of Final Application and Presentation											KW, J, N

	Week 3 18 Apr - 24 Apr	Week 4 25 Apr - 1 May	Week 5 2 May - 8 May	Week 6 9 May - 15 May	Week 7 16 May - 22 May	Week 8 23 May - 29 May	Week 9 30 May - 5 Jun	Week 10 6 Jun - 12 Jun	Week 11 13 Jun - 19 Jun
Project Requirements Documentation	KW, J, N								
Project Progress Report				KW, J, N					
Project Prototype Slides							KW, J, N		
Final Documentation									
Final Presentation Slides									

	Week 12 20 Jun - 26 Jun	Week 13 27 Jun - 3 Jul	Week 14 4 Jul - 10 Jul	Week 15 11 Jul - 17 Jul	Week 16 18 Jul - 24 Jul	Week 17 25 Jul - 31 Jul	Week 18 1 Aug - 7 Aug	Week 19 8 Aug - 14 Aug	Week 20 15 Aug - 21 Aug
Project Requirements Documentation									
Project Progress Report									
Project Prototype Slides									
Final Documentation	KW, J, N								
Final Presentation Slides							KW, J, N		

9 Roles and Responsibilities

Name	Roles / Responsibilities
Lim Kang Wei	Team Leader, Documenter, Developer, System Integrator
Jessica Soh Li Jing	Developer, System Integrator, Analyst
Lim Kia Yin, Natalie	Documenter, Tester, Analyst

10 References

- *Agile Methodology* [online], Available at: <https://www.digite.com/agile/agile-methodology/> [Accessed 28 Apr 2020]
- Nyma Malik (February 02, 2020), *React Native: Pros and Cons* [online], Available at: <https://citrusbits.com/react-native-pros-and-cons/> [Accessed 28 Apr 2020].
- Aman Mittal (December 5, 2018), *Integrating Firebase with React Native* [online], Available at: <https://blog.jscrambler.com/integrating-firebase-with-react-native/> [Accessed 6 May 2020]
- Sung Cho (Feb 28, 2017), *Source Code Protection: What Is It and Why You Need It* [online], Available at: <https://blog.se.works/source-code-protection-what-is-it-and-why-you-need-it> [Accessed 16 May 2020]
- Robert C. Seacord, Robert Martin (Jun 15, 2010), *MITRE CWE and CERT Secure Coding Standards* [online], Available at: <https://www.us-cert.gov/bsi/articles/knowledge/coding-practices/mitre-cwe-and-cert-secure-coding-standards> [Accessed 16 May 2020]
- *10 Best Mobile APP Security Testing Tools In 2020* [online], Available at: <https://www.softwaretestinghelp.com/mobile-app-security-testing-tools/> [Accessed 16 May 2020]