Group 2 Assignments

by

Jake Gebeline, Nick Katzenberger, Danny Moss

Stevens.edu

March 19, 2024

© Jake Gebeline, Nick Katzenberger, Danny Moss
Stevens.edu
ALL RIGHTS RESERVED

Group 2 Assignments

Jake Gebeline, Nick Katzenberger, Danny Moss
Stevens.edu

Table 1: Document Update History

Date	**Updates**
02/5/2024	Initials: JG, NK, DM • Created Overleaf Project and Added Chapter 1 and 2
02/19/2024	Initials: JG, NK, DM • Added User Requirements - Chapter 3
02/24/2024	Initials: JG, NK, DM • Added Final Project Mission Statement - Chapter 4
02/24/2024, 3/11/2024	Initials: JG, NK, DM • Added Use Cases and User Stories Chapter 5, Figures (5.1, and 5.2); and Tables 5.1, 5.2, 5.3, 5.4

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

– Jake, Danny, Nick

1.1 Team Member Introduction

Jake Gebeline - Hello my major is software engineering and I am currently a junior. I have some experience through a internship with the FAA (Federal Aviation Administration) working on some of the software that is used in and out of airports. I enjoy playing video games and dealing with tech such as creating and building PC's and tech like that. My availability for the most part is free the only exception to this would be I am not free on Tuesday or Thursday until after 6 pm.

Danny Moss - Hello World! My name is Danny Moss and I am a software engineer attending Stevens Institute of Technology. I am from New Jersey and I am a third-year student. I am a huge fan of video games and law! My love of computer programming started during the pandemic. Before, the only coding language I was familiar with was Scratch.mit.edu. The first language I ever truly became fluent in was Python, from there I was obsessed with it and started learning a wide variety of languages. From popular ones, like java and C#, to the lesser-known languages like Kotlin.

I am especially interested in the world of Artificial Intelligence and its future in our industry. I also would like to have a better understanding of how a computer can learn to think and behave like us. At that point, this is where we can start to consider when AI is alive. After all, we are nothing more than computers powered by french fries. You can probably tell I also like Philosophy.

Nick Katzenberger - I am a 3/4 Software Engineering major studying with a minor in Quantitative Finance. I know a variety of programming languages, and always find it interesting that functionally, they all work very similarly. I'm interested in watching and practicing MMA, and I like playing video games. Over the summer I will be working as an IT intern at Merck!

Chapter 2

Vision and Scope for Messaging App

– Jake, Danny, and Nick

2.1 Business Requirements

2.1.1 Background

Cellular communications business that focuses on rural areas. The project aims to enable more people to communicate with each other for emergencies and casual conversation. We chose this project because it involves useful concepts that can the group can learn. For example, the project involves how different devices communicate, which is relevant to all of us because we all use the messaging app Discord.

2.1.2 Business Opportunity

This project can be utilized to research more efficient ways to compress and transfer data and lose the least amount of data. Potentially, other software can be researched and used to upscale large files to match the original quality of the sent file. We can also use this project to further our understanding of a particular programming language, and learn how similar applications utilize online communication to function. This will ultimately enable the team to fix errors when they arise, and possibly duplicate our work for a different project. Potentially, we could sell this project to a company that will use our work in an already existing application.

2.1.3 Business Objectives

Allow 2 or more devices to be able to communicate with one another over a low bandwidth to make it easier for them to communicate in low coverage area. We aim to have our project be able to share data consisting of primarily text and images. We want to provide users with swift and convenient performance in low bandwidth areas.

2.1.4 Success Metrics

A good success metric that upper level management could use in order would be how many messages are being sent over a certain amount of time. This would show the developers and people involved in the project if people are using their app or not. Plus, this will allow you to have a quantifiable value to see performance go up or down.

2.1.5 Vision Statement

This app would be to allow friends and family to be able to communicate with text message no matter the bandwidth of their environment.

2.1.6 Business Risks

The most threatening thing for this application would be competition with similar apps. Our main concern is that the low bandwidth requirement will limit our functionality and speed of the application. The low bandwidth could prevent certain things getting to the client on the other end. This may be caused by a file being too large to send, and possibly too many texts at once being sent.

2.1.7 Assumptions and Dependencies

One assumption for this project would be that people would leave their messaging app that they currently use and move to this one. If no one would be willing to transfer apps with the people that they talk to this project would have no reason to exist. A dependency for this project would have to do with the bandwidth of the app. With this low of a bandwidth we would have to limit the size of each text, have reduced delivery speed, and the types of messages you could send such as like videos, photos, and stuff like that would all have to be modified or compressed.

2.2 Scope and Limitations

2.2.1 Major Features

The app will have text and image based communication that is less than 10 Mbps. The app will have users communicate between at least 3 devices at a time. The app will reduce compression of data sent using an up scaling software in order to maintain quality of data. The app will communicate over distances of 5 miles with message delivery time of under 20 seconds at maximum.

2.2.2 Scope of Initial Release

The group hopes to have the first public release by the end of the semester. We hope to complete all of the major features within our initial release. Our first release will focus mainly on functionality of the program to demonstrate that a messaging app with a low bandwidth can be made within a timely manner. Any bugs or logic errors that would prevent the features of the app from working would be corrected by the time of the initial release.

2.2.3 Scope of Subsequent releases

The group aims to provide subsequent releases that will fix any bugs that affect the app's graphical or functional performance. This will improve the user's experience by fixing unappealing visuals that do not reflect the team's vision for the application. Alternatively, subsequent releases may include more features that improve the diversity of features in the app that will make the user experience simpler and easier to use. Other changes in future releases may include cosmetic options that allow users to change how the app looks or sounds. This will make our project more appealing to people as most messaging apps do not allow much customization of the UI by the user.

2.2.4 Limitations and Exclusions

The app will most likely upon initial release, have a limit to the file size of items shared between users in order to provide the same quality of functionality as promised in our Major Features section 2.2.1. Our project may perform slower than competing messaging applications such as WhatsApp because of the limited bandwidth we intend to work with. The application may also be limited to only text and file sharing instead of live calls because the reduced bandwidth will not be able to support an appealing quality of sound or video on a real time basis. Our project may have to exclude Apple devices because of the long process to make an appearance on the App Store, which the timeline for the project cannot include. Therefore, it is likely that this app may only work on other Operating systems like Android, Windows, or Linux. On the other hand, our project could be hosted on a website to work around these limitation. However, this would sacrifice it's status as an application.

2.3 Business Context

2.3.1 Stakeholders and Stakeholder Conflicts

User
Advertisers: Users want to be able to block ad trackers. Advertisers want to tailor ads to maximize engagement

Developers: Users want a complicated function, like sorting images. Developers want to focus on easier tasks

Investor: Users don't want to pay for a premium feature. Investors want to make a profit

Server Managers: Server managers might shutdown services to cut cost. Users want a more versatile app

ISP: Internet Service Providers might not want large images being sent over their servers. Users want their images to be as premium as possible

Advertisers

Developers: Advertisers might want to add new types of advertisements. Developers don't want to worry about reworking the system to support it.

Investors: Investors might want to have ads as many places as possible, advertisers might not be comfortable with advertising on certain social media posts.

Server Managers: Advertisers want to keep records of user engagement, Server managers might not want to endure additional storage costs.

ISP: Advertisers might want to have larger videos for advertisements, ISP don't want to improve servers to handle them

Developers

Investors: Investors might want to reduce cost by not conducting academic research or removing benefits. Developers want to feel like they are working at a clean and comfortable environment

Server Managers: Server Managers want the developers to update the data to take up less space. Developers might want to focus on other projects and features.

ISP: ISP might want the developers to optimize the program to take up less bandwidth. The developers might want to take up more bandwidth using new features like video upload to improve the quality of the app.

Investors

Server Managers: Investors might want to sell the data on the servers for additional revenue. Server Managers might not want to associate with it and have their reputation harmed

ISP: Investors might want ISP to upgrade to 5g for their services. The ISP might not have the capital to build new infrastructure, and does not see much reason to for a lack of other customers

Server Managers

ISP: the ISP might want large data transfers to be handled over local and physical hard-drives being transported. The Server Managers may not want to pay for the transportation fees.

ISP

2.3.2 Project Priorities

The number one priority for the messaging app would be to get the bandwidth to the maximum capacity it can be. Since our bandwidth will already be limited, getting it to the maximum limit that we can is very important so that people can send bigger and longer messages. Another priority of this program would be to have the UI look nice. If the UI looks awful and doesn't function easily and smoothly than people are not going to want to use it.

2.3.3 Deployment Considerations

The first consideration that we would have to make is that even if we advertise well at first some people may not use the app due to the fact that they don't know how it works yet. It may take people awhile to figure out how it works and if they want to switch from their old messaging app to this one. Another consideration that we need to take into account would be if someone has issues with the app we need to make sure that we have help desk people that know how the app works. This will make sure that when people are having a hard time figuring out how the app works we have people who know it like the back of their hand to help.

Chapter 3

User Requirements

– Jake, Danny, Nick

3.1 Description - Course Registration System

This is just a simple course registration system. The students should be able to make a profile, search for courses, declare and change majors and minors, communicate with professors and other students and review professors. The facilty should be able to make a profile, open and close classes, transfer studetns, add prerequisites and Delete accounts. The administrators have the ability to Access all accounts, override course Requirements, take inquires from students, generate a transcript and make annoucements. For Constants: Students, Faculty should only have one account each. Administrations are allowed to have multiple (Individual and Group Account). Students cannot have more than one class at the same time. Teachers can teach more than one class at a time that are under the same education (I.E SSW 555 and CS 555). Facilitators can register for multiple classes without problem as they might view the progress of one class one week and another on a different week. Students, Facilitators and Administration are not allowed to schedule anything on the weekend or national holidays.

3.2 5 Requirements - Students

Table 3.1: Student Requirements

	Requirement 1	Requirement 2	Requirement 3	Requirement 4	Requirement 5
Name	Make a profile	Search for Courses	Ability to declare and change minor and major	Communication with Teachers and Students	Review Professors
Need	Needed	Needed	Needed	Not Needed	Not Needed
Priority	High	Medium	High	Low	Low
Stability	High	Medium	High	Low	Low
Source	Data can be changed, but not removed	Course selection may change over time, add or remove classes	Major and minor may change at a student's decission, but the selection must stay active indefinitely after declaring	Service must be functional and stay the same after deployment	Reviews need to stay the same after posting, and not be overwritten by others
Clarity	Data universally known	Courses need to be accurately labeled and coursework should match the title of the course	All students declare a major	Would be similar to how emails work	Similar to Yelp
Verifiability	Ensure users can add and change information	Do test searches for specific uploaded courses	Have a test student declare and change a major and minor	Send test messages between different accounts to measure functionality	Have an account make a test reviews of multiple professors
Other Attributes	Photo is optional	Includes teacher information	Double major/Double minor option	Allowing for image upload	Editing of posts and student interaction with other posts. Profanity filter

3.3 5 Requirements - Faculty

Table 3.2: Faculty Requirements

	Requirement 1	Requirement 2	Requirement 3	Requirement 4	Requirement 5
Name	Make a teacher profile	Allow faculty to open/close classes their teaching	Allow faculty to transfer students	Add prerequisites	Delete Account
Need	Needed	Needed	Needed	Needed	Not Needed
Priority	High	High	Medium	Medium	Low
Stability	Data must stay and be accessible	Classes can change based on which they are teaching	Service relies on account information of students to be reliable	Must ensure that equivolent classes can also satisfy the requirment	Make sure any references to the deleted accounts don't cause the site to crash
Source	Institution	Teachers	Institution and application	Professors	Professors
Clarity	Data known	Data subject to change	Similar to managing people in a group	Names of classes are easy to find	Just like deleting an account on a social media platform
Verifiability	Allow users to log in and create account with their information	Add and close classes making sure it works	Able to be tested by adding and dropping test accounts from specific classes.	A test run by both teacher and student	Deleting the profile, and then another professor verifing the account is gone
Other Attributes	Photo optional	Easy to use button	A batch edit feature to save time	Add link to prerequisite info	Having a way to recover the account/ Saving the information on the account

3.4 5 Requirements - Registrar

Table 3.3: Registrar Requirements

	Requirement 1	Requirement 2	Requirement 3	Requirement 4	Requirement 5
Name	Access to all accounts	Override Course Requirements	Take in inquiries from Students	Generate a Transcript	Make announcement
Need	Needed	Needed	Needed	Needed	Not Needed
Priority	High	Medium	High	High	Low
Stability	Data must be carefully maintained to prevent data loss	Requirments can change, and other classes can satify them.	Statements must be stored in a single location	GPA calculations must not have a 0/0 error	Must not result in a system crash for all the students contacted
Source	Students, Facility and others that communicate with the system	Faculty and Student Requests	Students and IT support	Registrar	Registrar
Clarity	Information easy to aquire	Data can be subject to change	It's just like sending an email	Class info easy to obtain	Must have all the accounts created
Verifiability	Cross referencing mass info with individual accounts	Student files a request, and the facilty reviews the case	Student sends request, Facilty responds	Student sends transcript, Employer confirms the transcript	Make anoucement, and ensure professors got the message
Other Attributes	Sorting, statistics between majors.	Send a message if the request was denied	Allow for image upload	Choosing different styles and color styles	Allow for different types of announcement

3.5 3 Specific Constraints

Table 3.4: Constraints

	Constraint One	Constraint Two	Constraint 3
Name	Number of Accounts	Classes Simultaneously	Weekend and Holiday Availablility
Students	One Per Student	Max 1 Class at a time	None
Professors	One Per Professor	Multiple classes in the same classroom	None
Administration	At least one, can be more	Unlimited	None

3.6 Conflicts Between Pairs

Table 3.5: Conflicts

Group	Conflict
Student and Student	Two students might want to take the same class, but the class registration might only have enough room for one. The best way to fix this would be to increase the capacity or allow oe student to take another course requirement in the meantime
Student and Facilitator	A student might want to look at the syllabus of a class before registering. However, the professor might be busy and isn't available to respond to their email. The best solution would be to provide the syllabus on the course page
Student and Registrar	The student might want to advocate for their major's requirements to be changed, while the Registrar will argue against it. The best way to avoid this would be to provide the students with a complete list of classes they can take before they declare the major
Facilitator and Facilitator	Two Facilitators might want to teach the same class at the same time period with the rest of their schedule being filled. The best idea to avoid this would be to provide a more flexible schedule with the other classes they must teach
Facilitator and Administration	A facilitator might not have enough enrolled students to teach a class. The administration might argue that their skill are better used in other regards. The best way to avoid this is to have less open time periods for the class so the groups are naturally consolidated.
Administration and Administration	Two administration heads might have different opinions on the number of students allowed per class. The best way to deal with this is to ask the professors how many students they are able to teach before a noticible grade drop is visible.

Chapter 4

Final Project Mission Statement

– Jake, Danny, and Nick
Ctrl Finders

4.1 Project Description

Our project aims to develop a better search engine for the Windows 11 operating system that has more functionality than the current File Explorer. This project will utilize the File Explorer format to iterate through files, however the UI will be vastly improved over the original. Some features that our program will add is the ability to search while filtering for upper and lowercase letters, be able to search files by type, and search for terms in quotations found in text based documents. However, our main focus will be to reduce search times for finding specific items.

4.2 Mission Statement

To provide users with a more efficient and convenient way to find files on their system.

Chapter 5

Use Cases and User Stories

– Jake, Danny, and Nick

5.1 Use Case Diagram 1: Logistics

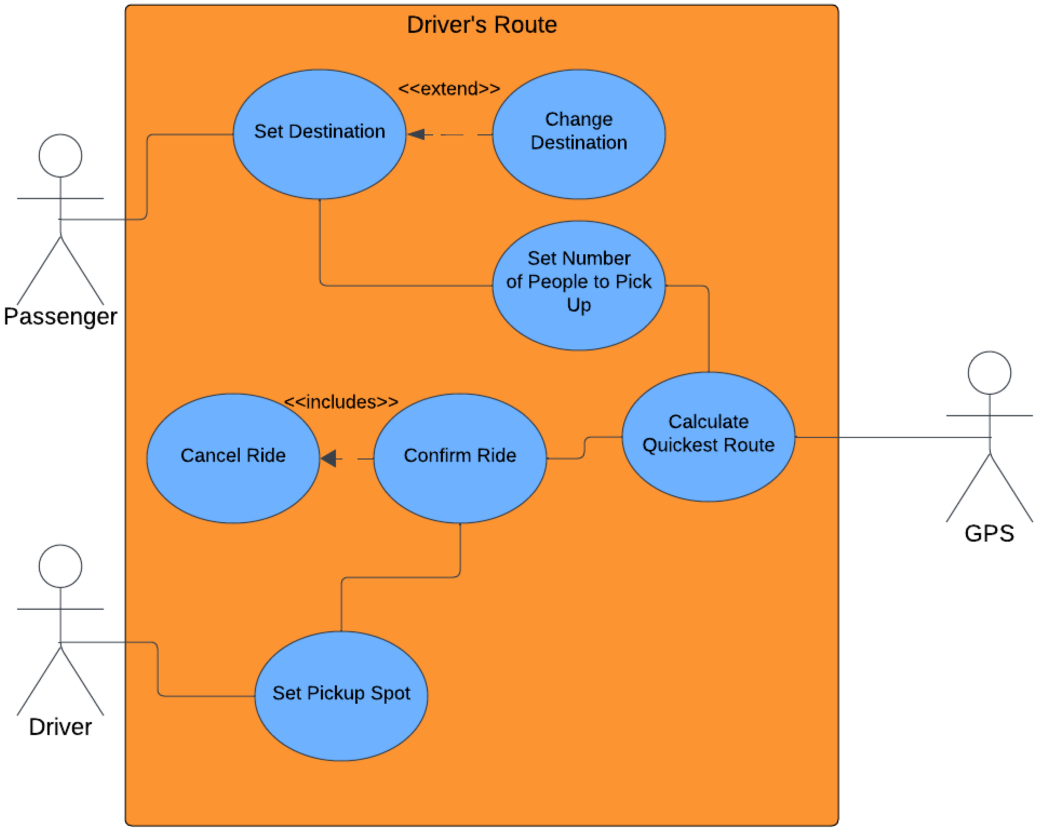

Figure 5.1: Use Case 1 Diagram

5.2 Use Case 2 Diagram: Payment

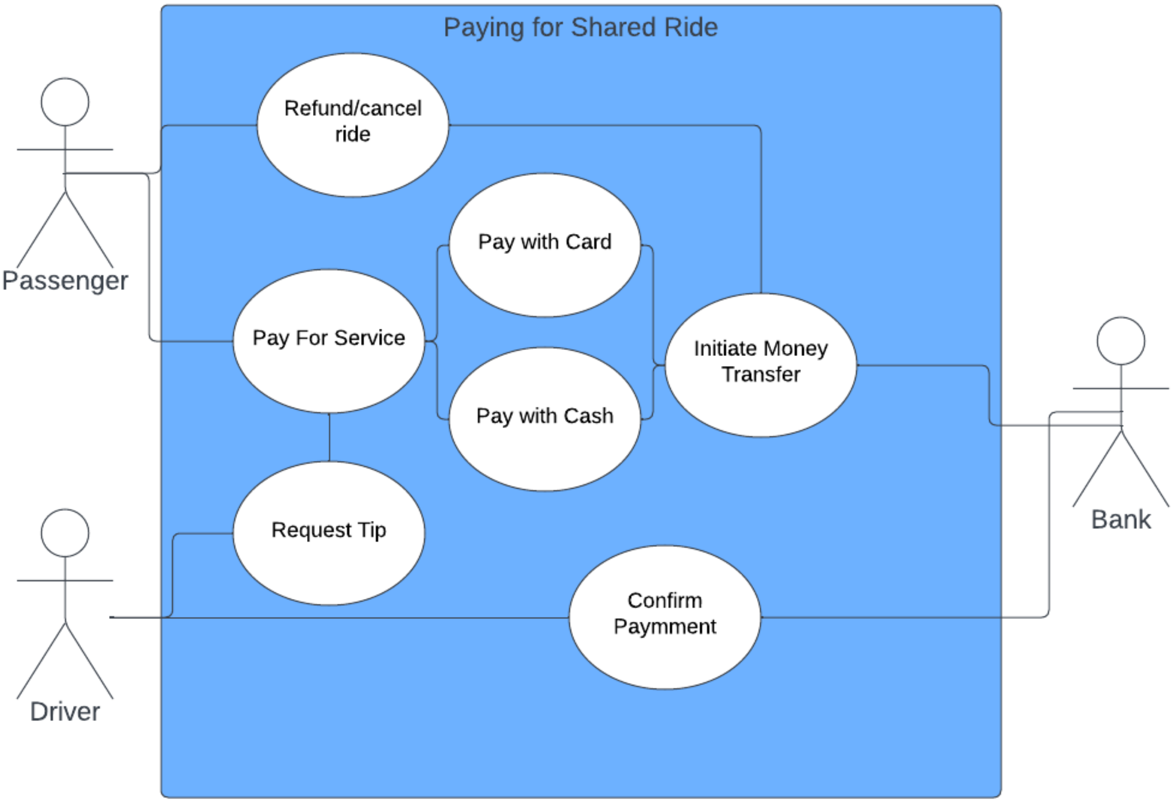

Figure 5.2: Use Case Diagram 2 for Payment

5.3 Use Case 1 Table: Logistics

Table 5.1: Use Case 1

Name	Driver’s Route
Brief Description	The driver has to know where they are going to pick up the customer, where to drop them off, and the quickest route to go
Actors	Driver, Customer, Satellite
Basic Flow	Driver obtains location, driver heads to location and picks up customer, Drives to destination provided by GPS
Alternate Flows	User cancels ride, driver gets lost, extra time for traffic, and driver cancels ride
Pre-Conditions	Customer needs service, needs to request a ride to desired location, and customer gets dropped off in correct spot
Post-Conditions	Driver drops off customer, driver gets paid, and driver can now pick up next customer
Special Requirments	Route Redirection if road is closed due to construction, flooding... ect. Driver wants to avoid tolls along the route Driver only will drive a certain distance before having to go home

5.4 Use Case 2 Table: Payment

Table 5.2: Use Case 2

Name	Payment to Driver
Brief Description	The customer has to pay the driver for his service using funds from a bank or credit card
Actors	Driver, Customer, Bank
Basic Flow	Customer requests transfer from bank to pay driver, Bank authorizes transaction, Driver recieves cash
Alternate Flows	Customer requests refund OR Driver requests tip, Bank is notified of second transaction
Pre-Conditions	Customer must have funds in bank account, Driver must have bank account ready to receive, pre defined rates for differing distances
Post-Conditions	Customer is no longer charged, Service must approve as valid transaction
Special Requirments	Bank information must be valid to make transactions

5.5 User Requirements Table

Table 5.3: User Requirements Table

R#	User Requirements	Priority	Use Case
R1	Pay with Card	Must	2
R2	Pay with Cash	Should	2
R3	Tip	Could	2
R4	Set Destination	Must	1
R5	Set Pickup Spot	Must	1
R6	Driver’s Quickest route	Should	1
R7	Canceling Ride	Must	1
R8	Refunds	Should	2
R9	Set Number of people to pick up	Would	1
R10	Rerouting due to delays and change of driver request	Could	1

5.6 Matching Use Cases to Requirements

Table 5.4: Matching Use Cases to Requirements

Use Case 1	R4, R5, R6, R7, R9, R10
Use Case 2	R1, R2, R3, R8

5.7 Activity Diagram 1: Logistics

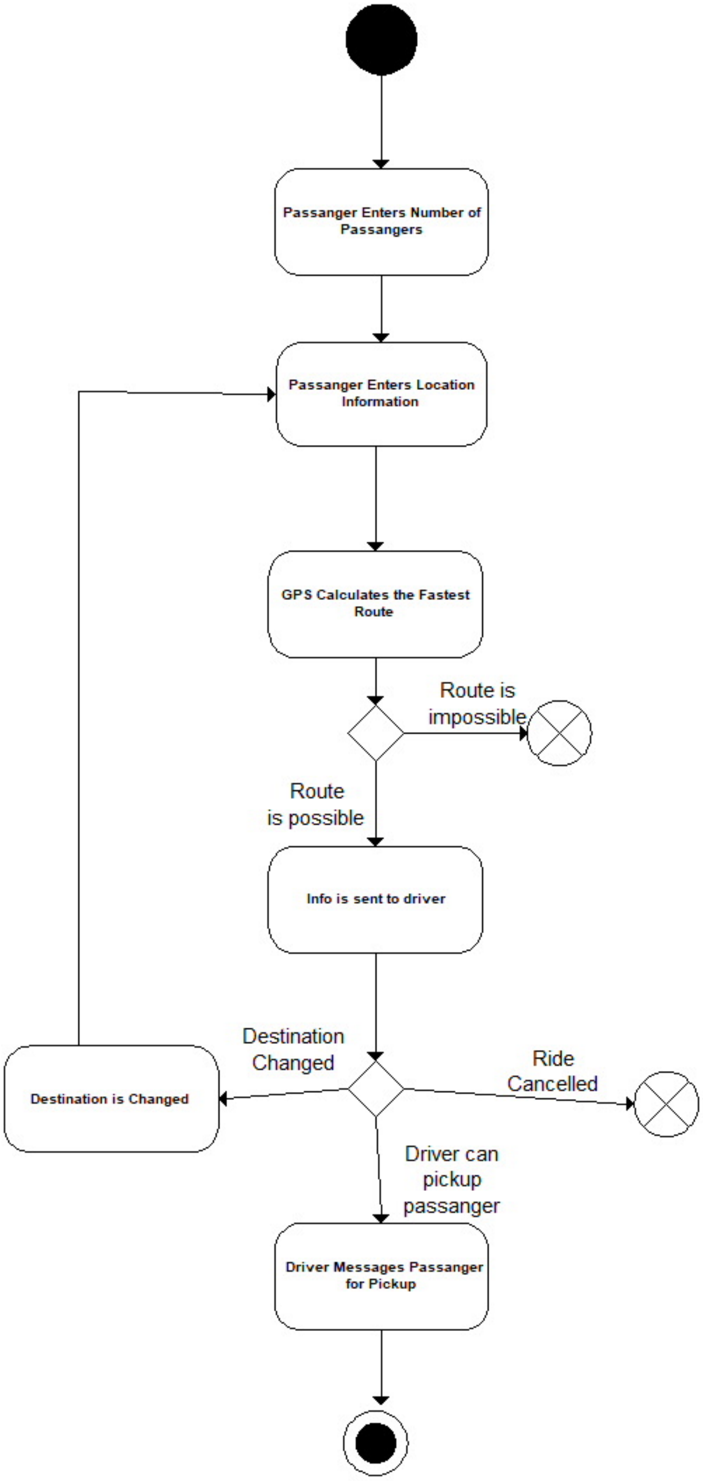
Passanger Enters Number of Passangers
Passanger Enters Location Information
GPS Calculates the Fastest Route
Route is impossible
Route is possible
Info is sent to driver
Destination Changed
Ride Cancelled
Destination is Changed
Driver can pickup passanger
Driver Messages Passanger for Pickup

Figure 5.3: Use Case 2 Activity Diagram: Logistics

5.8 Activity Diagram: Payment

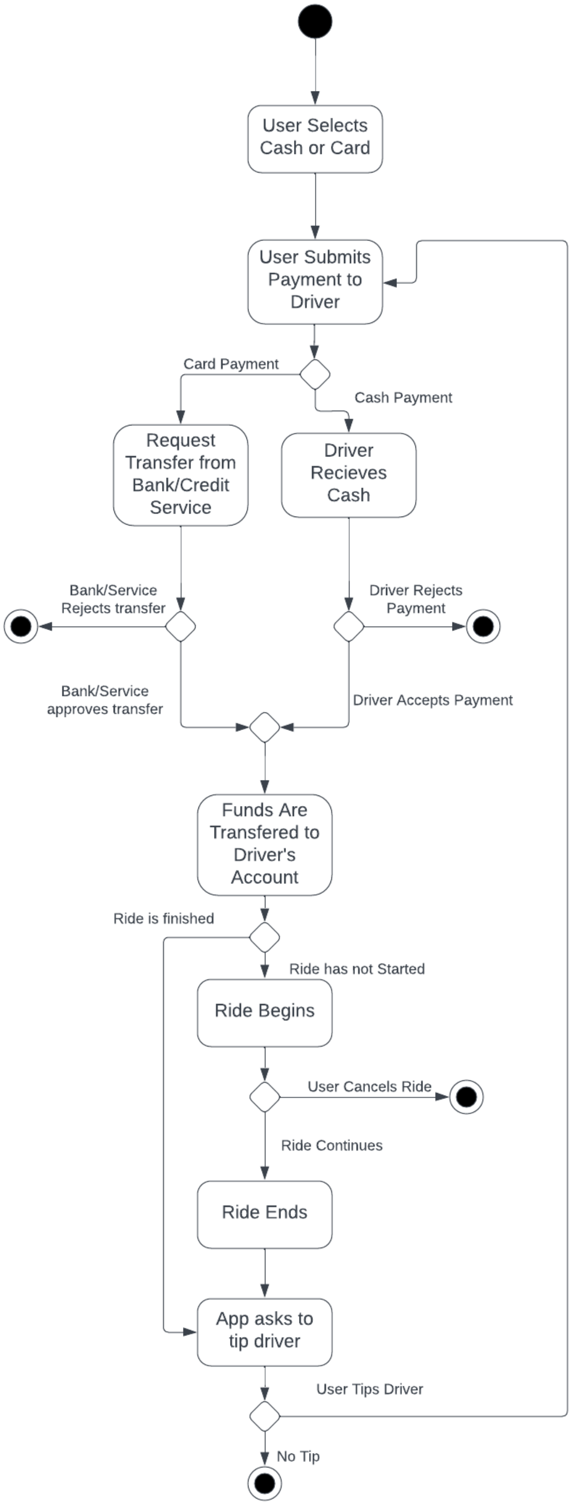

Figure 5.4: Use Case 2 Activity Diagram: Payment to Driver

Bibliography

Index