



Addis Ababa Science and Technology University
College of Electrical & Mechanical Engineering

Department of Software Engineering
Graduation Project Documentation
Title: OnDemand logistics System

June 29, 2021

ADDIS ABABA, ETHIOPIA

Group Members

Name

ID

1. Asier Nebiyou ETS 0164/10
2. Asdenaki Tamrat.....ETS 0159/10
3. Abenet WoldeETS 0040/10
4. Abebaw Gebre.....ETS 0027/10

Advisor Name: Mr Muleta

Signature _____

Examiner Name

_____.

_____.

_____.

_____.

Signature

_____.

_____.

_____.

_____.

Acknowledgment

First and foremost, our greet thanks go to Almighty God, for his help us in every aspect of our life. We would like to acknowledge the support of all our class mates and those who helped us directly and indirectly motivating us and with their encouragement, support, and understanding which had a great value in every effort of our study. Our heartfelt appreciation goes to our adviser **Mr Muleta** for his guidance, valuable suggestion, moral support, and constant encouragement in this work and also for constructive opinion and willingness to participate in each part of our documentation and his effective direction. We would also like to thank EMS employees for their cooperation and for their willingness to be interviewed and their patience in answering our numerous questions.

TABLE OF CONTENTS

Chapter one: Introduction	1
1.1 Background of the project	1
1.2 Statement of the problem	3
1.2.1 Existing System	3
1.2.2 Major problems of an existing system	4
1.2.3 Proposed system.....	6
1.2.4 Advantages of the proposed system	7
1.3 Motivation	8
1.4 Scope and Limitation of the Project	9
1.4.1 The scope of the project	9
1.4.2 Limitations of the project.....	9
1.5 project goal and objective.....	10
1.5.1 Goal.....	10
1.5.2 General objective	10
1.5.3 Specific Objective	10
1.6 Methodology	11
1.6.1 Data collection methodology.....	11
1.6.2 System design analysis tools.....	11
Chapter 2: System Requirement Specification	12
2.1. Background Overview	12
2.2. Functional Requirements	12
2.3. Non-Functional Requirements.....	15
2.4. Feasibility Study	17
Chapter 3: System Analysis and modeling	19

3.1 Overview	19
3.2 Scenario-Based Modeling	19
3.2.1 Use Case Identification.....	20
3.2.2 Actor Identification	22
3.2.3 Use case description and use case diagram	23
3.2.4 Activity Diagram.....	41
3.3 Behavioral/ Dynamic Modeling.....	47
3.3.1 Sequence Diagram	47
3.3.2 State Diagrams	52
3.4 Class-Based Modeling	56
3.4.1 Identifying Classes	57
3.4.2 Class Diagram	60
Chapter 4: System Design	61
4.1 Overview	61
4.2 System Design	61
4.3 Architecture of the System	61
4.4 System decomposition	62
4.5 Component Diagram.....	64
4.6 Deployment Diagram	65
4.7 Database Design.....	65
4.7.1 Entity relationship Diagram	66
4.5 User interface design.....	67

List of Figures

Figure 2. 1: Gantt Figure for project schedule.....	19
Figure 3. 1: Use case diagram for OnDemand logistics system	23
Figure 3. 2: Activity diagram for admin update and delete.....	41
Figure 3. 3: Activity diagram for signup	42
Figure 3. 4: Activity diagram for login	43
Figure 3. 5: Activity diagram for place order	44
Figure 3. 6: Activity diagram for update status	45
Figure 3. 7: Activity diagram for register driver	46
Figure 3. 8: Sequence Diagram of login	48
Figure 3. 9: Sequence Diagram of Sign Up.....	49
Figure 3. 10: Sequence Diagram of Account Register	50
Figure 3. 11: Sequence diagram of Place Order	51
Figure 3. 12: State diagram for signup	52
Figure 3. 13: State diagram for login	53
Figure 3. 14: state diagram for place order	54
Figure 3. 15: state diagram for update status.....	55
Figure 3. 16: state diagram for delete user	56
Figure 3. 17: Class diagram for OnDemand logistics system	60
Figure 4. 1 : System architecture.....	62
Figure 4.2: Component Diagram.....	64
Figure 4. 3: Deployment diagram.....	65
Figure 4. 4: Er diagram	66
Figure 4. 6: UI design.....	67

List of Tables

Table 2. 2: schedule of the project	18
Table 3. 1: Use case lists	21
Table 3. 2: Use case description for signup.....	24
Table 3. 3: Use case description for Login	25
Table 3. 4: Use case description for place order	25
Table 3. 5: Use case description for Track order	26
Table 3. 6: Use case description for Report driver	27
Table 3. 7: Use case description for View history	27
Table 3. 8: Use case description for cancel order.....	28
Table 3. 9: Use case description for Update profile	29
Table 3. 10: Use case description for Accept/decline order	29
Table 3. 11: Use case description for View customer info.....	30
Table 3. 12: Use case description for Update status	31
Table 3. 13: Use case description for Enter receiver info.....	32
Table 3. 14: Use case description for Choose vehicle.....	32
Table 3. 15: Use case description for Specify receiver info	33
Table 3. 16: Use case description for Enter pickup and delivery point	34
Table 3. 17: Use case description for Logout.....	34
Table 3. 18: Use case description for Register driver	35
Table 3. 19: Use case description for Update driver account	36
Table 3. 20: Use case description for Ban User	36
Table 3. 21: Use case description for Generate report	37
Table 3. 22: Use case description for Supervise conflict.....	38
Table 3. 23: Use case description for Set terms.....	38
Table 3. 24: Use case description for delete account user	39
Table 3. 25: Use case description for update admin account.....	40
Table 3. 26: Table for Admin class.....	57
Table 3. 27: Table for Customer class.....	57

Table 3. 28: Table for System User class 58

Table 3. 29: Table for Driver class 58

Table 3. 30: Table for Order class 58

Table 3. 31: Table for Payment class 59

Table 3. 32: Table for Location class 59

Table 3. 33: Table for History class..... 59

Table 3. 34: Table for feedback class 59

Abstract

People have always been searching for a better way of getting a proper courier services, where they can entrust their items for delivery without them wondering about moving the item themselves. Now a days, people can have any type of delivery services in their reach easily, whether it is food, item or documents, delivery services have made a wide approach to satisfy the need of the customers whether with traditional or modern ways of deliveries.

The time essential behavior the item delivery has is what made the issue of delivery more challenging. The Traditional courier service is time-consuming as the result of the cumbersome processes involved. Moreover, people may not know the exact time when they will need some item delivered or by the time, they want it delivered, they can't communicate with the service providers easily or they can't find on-demand delivery services which means they will have to take some steps towards finding the service instead of finding the service on their door steps. This is the main concern why this study attempts to address the aforementioned problems with Automated OnDemand logistics system, a courier service which is simple, helps the users and service provider to have direct communication via the application.

One of the modern ways delivery made is, using a smartphone in hand and order the desired service over the internet without bothering about the worries of how to make it happen. On the OnDemand logistics system, the customer would get the courier service desired with a few clicks of a button. The customer just has to specify the location where to get the item from and where to get the item to, then the system has registered drivers all around the place to assign the orders to.

The proposed OnDemand logistics system is entirely chain of process that serve as an interface between the courier service providers and customers. The main aim of the study is to digitalize the courier services, where the customer orders an item and receive corresponding the item in timely fashion. This saves money and time and brings customer satisfaction with the advantages it brings.

CHAPTER ONE: INTRODUCTION

1.1 BACKGROUND OF THE PROJECT

As long as this world operates there will always be movement. whether it is people or goods, the hustle never ends. There will always be some point to get from and others to get to. There is no doubt that time is a decisive entity that needs to be taken into account, people need not only goods and products to be moved from place to place in a short period but also curious about the mode of delivery as well. Courier services have evolved magnificently from the days of their starting and below are basic steps the courier service took over time from then till now.

Early Courier Services:

A Trusted Delivery Person:

Before the middle of the 19th century, people simply pay a courier to take their packages to their destination [1]. The courier would drop it off in a central location for the recipient to pick up and then be on his way. People had to hope they would come across someone who was making the journey to wherever their mail needed to go.

Pony Express:

One of the earliest delivery methods was the pony express, which was built to connect the populated East Coast to the wild frontier of the West Coast in the United States [1]. It was the first and earliest form of today's more modern mail service. The pony express consisted of riders who were positioned across the 2,000 miles from sea to shining sea. The way it worked was that riders would switch horses every 15-20 miles to accomplish the delivery as quickly as they could [1]. The chain of riders operated across the entire country.

Meanwhile in Ethiopia, before the establishment of the postal service on March 9, 1894, following an imperial edict, correspondence was conducted through messengers known as “melektegnas” or “postegnas” [2]. These tough individuals traveled great distances, often on foot, overcoming rough landscapes and weathering hostile climates. They endured the pangs of hunger and thirst and

carried their letters over their heads, on cleft sticks (which later became the symbol of the post office still today) until they reached their destination.

Courier Services in modern days:

Today, significant advancements in information technology enable customers to track a package's movement from origin to destination. In some areas, consumers can order a product online and have it in hand by the end of the day.

From stagecoach package delivery to same-day walk-in delivery solutions, the industry has evolved drastically to meet consumers' growing expectations. The goal remains the same, however: meet delivery needs with speed, ease, and great customer service.

After some time, the construction of the Djibouti - Addis Ababa train made it possible for letters, parcels, and merchandise, which were previously transported on camelback [2]. This was a crucial factor that greatly improved the pace and efficiency of the postal service while it laid the basis for the international exchange of mail. Then Ethiopia became a member of the Universal Postal Union in 1908 [2], which in turn led the Ethiopian postal service to join the global community of EMS (Express mail service) which professionalized the enterprise in parcel and merchandise delivery.

Delivery evolution in a nutshell:

- The first stagecoaches, the Pony Express, the Railroad Express Agency were historical precursors to today's delivery options [8].
- In 1975, UPS began providing service to all of the 48 contiguous states. What followed was a significant shift to truck and air transport, creating a surge in the modern package-delivery industry [8].
- The 1990s brought new services and enhancements, including multiple delivery options, same-day or next-day service, and a variety of time-specific delivery choices. Guarantees for next-day delivery soon became commonplace, an option today's consumer expects [8].

1.2 STATEMENT OF THE PROBLEM

1.2.1 EXISTING SYSTEM

For many businesses, delivery schedules are sporadic and unpredictable. It is never known when people need that critical part or document delivered. This is one of the biggest courier challenges for businesses. people need the delivery made when they need it. But it can be impossible to predict when the situation will arise.

If we see around, there are diverse kinds of Courier services around, some of which are standardized like EMS and some just out there for business just like the random truck owners who deliver any type of merchandise for customers.

The mini pickups around Merkato: The mini pickup owners are best known for their courier service providing around merkato, Addis Ababa. Their customers mostly are businessmen around there, who want their merchandise delivered to the market as soon as possible. When people around there want to get their service, they have to call them and check in whether they get an immediate response or not. As the service providers are limited to some numbers and may always not be around the customers sometimes have to be lucky to get an on-time response.

EMS (Express Mail Service): EMS is an international postal Express Mail Service, for documents and merchandise. The introduction of EMS has made the Ethiopian Postal Service involve more in the express delivery market. Using vehicles owned by the enterprise which are mostly motor bicycles, items will be transported to the areas intended within a considerable time. Customers must bring their item to the local branch of the enterprise and it will be weighed and registered for delivery.

Some other courier service providers such as DHL and FedEx, which are not indigenous to Ethiopia have the same kind of service as EMS. Even more concerned on international delivery than the domestic one.

Lately, on-demand delivery services are starting to sprout in Ethiopia. Services like Ride and Feres are on-demand transport for people, which in the process people get picked up from wherever they

are and dropped off to the destination they intended. These on-demand services use smartphones to make the order and use registered drivers to execute it. However, On-demand courier services for logistics are not easy to come by. The concept of on-demand delivery is poorly adapted in terms of logistics.

OnDemand logistics system aims to integrate the concepts taken from EMS and RIDE and narrow the service gap observed in them. Since the courier service of EMS is not on-demand, and the on-demand service of RIDE does not work for logistics. OnDemand logistics system integrates the courier service of EMS and the on-demand service of RIDE to realize on-demand courier service for any kind of logistics no matter the size and type, which is ordered using smartphones and executed by registered drivers.

1.2.2 MAJOR PROBLEMS OF AN EXISTING SYSTEM

During contact we made with the information desk of EMS, there are some concerns raised about the service. Some of which are:

- ❖ EMS doesn't pick up items, instead, they have to be delivered to the nearest branch.
- ❖ There is a weight restriction of parcels, which sets the maximum weight allowed to 30 KGs
- ❖ All the vehicles used for delivery are owned by EMS, which sometimes during busy delivery orders gets difficult to get free vehicles because of their limited number.
- ❖ Delayed delivery is one of the problems, when all vehicles are occupied, the item fails to reach the intended destination on time.

EMS aside, what if someone wants a heavy-duty delivery? Besides, what if that customer wants real-time tracking of that heavy-duty delivery? Of course, EMS enables tracking but heavy-duty courier services are not given by EMS. Even with the weight acceptable by EMS, the customer has to go to a local branch to hand the item to the delivery services.

So, when heavy-duty delivery is required, the person has to go to local drivers who deliver the type of service wanted. Mostly the local freight service providers do not have a reasonable price for their delivery, which is sometimes very expensive to even hear it.

Traditional courier services can be unreliable and unpredictable in situations like this. Traditional courier services are often low-tech operations, so scheduling deliveries is a slow, manual process, and usually don't have a great deal of transparency into the location of the driver, the status of the delivery, or the estimated costs. Because of the limited number of drivers and delivery vehicles.

Some of the generalized shortcomings of the courier service providers are:

- **Cumbersome Processes:** In the traditional logistics and courier business model, there is too much procedure required for ordering the service.
- **Changing Customer Expectations:** With the rise of digital adoption in all walks of life, users are becoming more demanding. They want the services to be delivered to their door-steps.
- **Delayed Delivery:** Many times, drivers fail to reach the destination or deliver couriers and other goods on time because of the limited resource the service providers have.
- **Higher Cost:** Be it reimburses for the delayed delivery, coordinate with third parties, or buy new tools to make a customer strategy, everything adds to the cost associated with running a business; making it yet another challenge hindering the path to a successful venture.
- **Poor Feedback System:** Last but not least, the tiresome processes and huge amount of work have left workers with less time to check and respond to the feedback shared by the customers. In fact, not every online logistics company offers this service. Something that accounts for a poor feedback system to be another challenge faced by the logistics and parcel delivery service providers.

1.2.3 PROPOSED SYSTEM

In the proposed system, OnDemand logistics system is borrowing ideas developed for the delivery of market items to customers and applying them to industrial and business deliveries, where items and logistics get delivered from the initial to the destination point of the customer's choosing. This technology-based solution would allow to quickly and easily request a courier from a smartphone. It would be possible to set pickup and drop-off locations, vehicle requirements, and more. And because this OnDemand logistic service has a target to employ a pool of reliable and efficient, independent drivers who are already on the road and in the area, customers can be sure that their delivery will be made when they need it.

Technology-based OnDemand delivery service offers significant benefits when the user wants to track the driver and the status of the deliveries. This software solution sends the delivery request to a nearby driver the second it is made and utilizes GPS to track the driver's route in real-time. With this tracking software, the customer knows exactly where the driver is and when to expect them to pick up and drop off the package. It is possible to share this information with coworkers, clients, or customers who are expecting the delivery, so they know when the package is on the way and don't need to waste time waiting for the delivery. This is especially important in business locations where drivers might get caught in traffic.

With all this technological advancement mentioned above, the proposed system would find a way to avoid too much process by creating a direct line between the drivers and the customer. The system would eliminate external parties who bring nothing but complication for the customer. The system also aims to satisfy customers' expectations by the easy access to the courier service using the technology in hand and fast response offered to the customer. Since being on the road and in the area means fast response; the possibility of service delay comes to a minimum as the service providers would always be geared up and close by. With no parties in the middle asking for their cut, less time and effort it takes to accomplish deliveries and, because of the convenient business environment this project planned to create, the service cost minimizes by a considerable amount. The system would enable customers to report or give feedback about the service they had which is very helpful to improve the service from time to time.

1.2.4 Advantages of the proposed system

With smartphones in pockets and apps like Uber, BiteSquad, and Amazon Prime, people have come to expect that anything they need or want can be delivered at a moment's notice. Packages, Documents, and some household essentials, the list below are some of the advantages that could be experienced with the proposed system.

- ❖ **Ordering delivery from the comfort bed:** The OnDemand logistics system set a target to make it possible for consumers to come across the service providers, book for delivery, choose the pickup location and delivery, and much more using smartphones.
- ❖ **Process automation and Efficiency:** By investing in OnDemand logistics service app development, business leaders have a chance to be provided with high-end services to their target customers without involving multiple third parties into the equation, asking for endless questions, as well as organize the inhouse processes and assign different assignments to their drivers in real-time.
- ❖ **Cost-Effective courier service:** OnDemand logistics system would bring the customers directly in touch with the courier service providers easily. This project aims to eradicate the need for third-party agents and even streamlines the process, which will eventually aid in gathering more outcomes with minimum investment. Also, we plan to make it possible for the entrepreneurs to run their business without hiring new drivers or vehicles, which again adds to the profitability of the business.
- ❖ **Real-time Tracking of the courier:** The biggest advantage of considering the OnDemand logistics system would be real-time tracking. As the name suggests, this functionality enables consumers as well as business leaders to track the vehicles (or fleets) in real-time. While this assures consumers about courier delivery, it aids companies to see how to optimize the process and grab better opportunities in the marketplace.
- ❖ **Feeling Secured about the delivery:** The real-time tracking, better optimization of processes, and other such factors may help customers to feel secured about the item. This aids in building trust between customers and service vendors as well as bring more profits into the environment.
- ❖ **Effective Fleet Management:** Traditionally, it is quite tough for the logistics and parcel delivery parties to keep a track of every single vehicle they have, the routes they have

traveled, the number of deliveries made, and much more. However, with the advent of on-demand mobility solution, this project plans to make this much easier, quicker, and efficient.

- ❖ **Improved User Experience:** Unlike traditional when users supposed to meet multiple agents and stand in long queues to get a perfect deal for them, this project aims to make it possible for them to come across the service providers and make a booking with a few taps on their smartphones. Also, the app would offer them the opportunity to track their orders in real-time, communicate with the associated driver. Something that has indicated a better user experience.
- ❖ **Business Expansion:** Lastly, the inception of the On-demand concept plans to open a new set of opportunities for the businesses. It would encourage them to look beyond their premises and see what and how introducing new technologies into their business process can enhance their productivity and profitability.

1.3 MOTIVATION

The state-owned Ethiopian Postal Service Enterprise is the only indigenous courier service in Ethiopia. The Ethiopian Postal Service Enterprise provides both international as well as domestic express delivery services under its program called Express Mail Service (EMS). Other international courier service providers such as UPS, FedEx, and DHL have offices in Ethiopia and provide express delivery services. The services provided by these companies have too many limitations concerning the type of item, amount, weight, and mode of delivery. Even with other local delivery service providers, most of their work is manual and difficult to acquire.

There is no doubt that Ethiopia's business environment has improved over the years and many investors have set up shop in wide and different places. This has further created fresh demand for the speedy delivery of goods and packages. A need that has not been properly met by the national postal service. The few big courier companies that rule the industry on the other hand have not been able to match the efficiency that most clients are looking for. What if clients are looking for courier companies that are small enough to offer personalized services and big enough to provide same-day deliveries even at odd hours. The gap we saw between the demand and supply of the

service took us high to the inspiration to think of a system which is fast and interactive in a way that can accommodate the need and expectation of the business and logistics environment.

1.4 Scope and Limitation of the Project

1.4.1 THE SCOPE OF THE PROJECT

The scope of this project is to develop a web-based application which is accessible by smartphones to offer customers with on-demand courier service which enables real time item delivery of different types from pickup to drop locations of customers' choosing. The proposed system is bounded only in delivering items and not the provision of legal documents required for item transport. i.e., if required documents are provided by the customer.

1.4.2 LIMITATIONS OF THE PROJECT

- ❖ The system only allows Same-Day deliveries, which means anyone who uses the system should be aware the delivery will be made as soon as the customer books it.
- ❖ The items should be handed to the driver assuming there will be no purchase and delivery.
- ❖ Whether it is natural or man-made the system does not notify any inconveniences, this has its effect on the delivery time or condition.
- ❖ Due to the low network infrastructure of our country, the system only operates in cities where there is a good internet connection, this means the deliveries will only be made within the city in our case Addis Ababa.
- ❖ The other limitation of this project is the payment method, because of the pre requisite the available payment methods require, we will not be able to implement an online payment service.
- ❖ For there is no digitalized data stored in the transport sector, Online verification of drivers' data such as license and plate number are impossible.
- ❖ The system cannot limit the number of accounts a customer can have, unless it is the same phone number the customer registered with.
- ❖ Customers cannot cancel an order while the order is in motion.

1.5 PROJECT GOAL AND OBJECTIVE

1.5.1 GOAL

The attempted goal is providing both heavy-duty & lightweight transport delivery of parcels and packages securely, reliably & visibly around the clock without the hassle of having to deal with many drivers/deliverers with different prices, schedules, and sometimes even extra fees after delivery. In general, the goal of this project is to give the service of delivery to the extent of from special to everyday commodities for the customer in need.

1.5.2 GENERAL OBJECTIVE

The general objective of this project is to design and implement Web-based On-Demand logistic Application system.

1.5.3 SPECIFIC OBJECTIVE

To meet the general objective stated above some considerable objectives must be taken into account which in this case are:

- ❖ Identifying the existing problems in getting satisfying courier services.
- ❖ Relating users' problems with the solution in hand.
- ❖ Making the system user-friendly.
- ❖ Developing an interface for courier service providers and customers, Provide satisfying means of communication between the driver and the user.
- ❖ Introducing businesses with digitalized and convenient courier service providing environment.
- ❖ Making logistics delivery cost-effective.
- ❖ Building trust between drivers and customers.
- ❖ Helping businesses to sprout with a little bit of investment.
- ❖ Minimize the time it takes for customers to contact drivers and agree on terms.
- ❖ Making customers comfortable while using the application.
- ❖ Design, implement, test, and deploy the system
- ❖ Train users of the system.

1.6 METHODOLOGY

1.6.1 DATA COLLECTION METHODOLOGY

The first step the project going to take towards collecting the data needed is observing the existing system. It is important to take points in the gaps between the demand and supply, difficulties in transportation of products. Conducting interviews also helps to get the view of the problem from the customers' angle and researches towards the existing service providers in Ethiopia like EMS will help to get what essence they missed from the actual problem on the ground. The final analysis of the compiled document whether from the interview or the researches will be more helpful to relate the needs of the interviewee to the services offered by providers and use that data to draw more existing requirements fulfill them.

1.6.2 SYSTEM DESIGN ANALYSIS TOOLS

Design and analysis tools are the basic components of system development in the action of system design setting a foundation. In our system we used:

Microsoft Office: is an application used to prepare requirement specification document, we have used Microsoft word 2019.

Draw.io: draw.io is completely free online (can also be offline) diagram editor that enables the user to create flowcharts, UML, entity relation, network diagrams and also helps to design and develop database schemas.

1.6.3 System development tools

As the design and analysis phase has laid some basic foundations and set a skeleton of the project the system needs to be transited to the implementation and development phase and we have chosen what we thought is the best for the intended project whether on the database or front or backend, which are:

Front-end tool: these tools help to develop the user side of the application.

- **Flutter:** is an open-source mobile UI framework that allows to create native mobile applications with only one codebase. It is used to develop applications for Android, iOS, Linux, and Windows.

Back-end tool: these tools mostly emphasize the operational endeavors of the system which is hidden from the customer and it is more detailed part of the system.

- **Dart:** is a simple, clean, and object-oriented language that helps to develop web and mobile applications.
- **Firebase:** is a multifunctional tool that has functionalities to develop database and back-end functions in mobile apps.

Databases: database development tool we used is:

- **Firebase:** is a NoSQL database that has a real-time document store

Testing tools and Editors: system testing and editing are done using these tools:

- **Visual Studio, IntelliJ:** These are code editors for developing many types of programs or software with a proper extension.

CHAPTER 2: SYSTEM REQUIREMENT SPECIFICATION

2.1. BACKGROUND OVERVIEW

On demand courier logistics application is dedicated to satisfying both drivers and customers by making the interaction & communication between them easy and efficient.

2.2. FUNCTIONAL REQUIREMENTS

Functional requirements are requirements that are used to build the main functionality of a system in which it interacts directly with both customers and system owners. Our project includes many functional requirements. These requirements are collected, organized, and ordered through different requirement gathering methods and finally, we have used the elicitation process to select the best one, more appropriate, and these functionalities are used as a backbone for the system to apply it without any doubt.

A) Customer registration

- The system allows customers to be registered online through the mobile application.
- The information needed for registration is first name, last name, phone number and password.

B) Login

- The system lets the admin to login with his/her username and password
- The system gives the admin the ability to control the drivers by registering or deleting, and modifying information about them.
- The system doesn't allow drivers to change information about themselves but can do so only through contacting an admin.
- The system lets admins to login with their username and password given by the system.
- It allows admins to access any system information.

C) Modify services

- The system allows the admin to change service information of the system, including modifying price/km and price/Kg.
- The system allows the admin to add vehicle categories.

D) Get and respond to feedback

- The system allows the branch Admin to enforce rules that govern how the interaction of driver and customer.

E) Delete Information

- The system allows the admin to delete the driver account; if necessary. Delete customers' accounts too.

F) View Information

View history

- The system lets the customers view their order history.
- The system lets the drivers view their fulfilled orders history.
- The system lets admin view all of customer and driver's history

View profile information

- The system lets customers and drivers view their own profile information.
- The system lets admin view all of customer and driver's information.
- The system lets customers and drivers view their respective information during an order that is being fulfilled.

Feedback or comments

- The system allows Admins to see the comments and feedback.

View notification from the system

- The system gives notification about some content that is to be managed.

View and track services

- The system lets the customers view estimated tariffs and offers
- The system lets the customer's view and track their goods.
- The system lets the admin's view and track their goods

G) Sorting information

- The system can sort the list of information such as transactions

H) Generate Reports

Generate a report about the active customers and drivers

- The system generates reports about active customers and gives a summary.

Generate a report about customer feedback

- The system generates reports about the customers' feedback about the service.

Generate a report about a customer and driver transaction

- The system generates organized reports about the customers' transactions.
- The system generates organized reports about the driver's transactions.

- The system generates reports about the total transaction spent in the system with their respective time.

Generate a report of customer distribution like activity.

- The system generates reports about customers' gender distribution to know in detail about the type of customers.

I) Alert Notification

- The system gives alert notification when customer sign up for the first time like welcome
- The system gives alert notification after customer make transaction
- The system gives alert notification when order status changes (like picked up, on route and delivered)
- The system gives an alert notification when the user leaves or wants to delete an account

J) User logout and login

- The system allows any user to log out and login when necessary.

Forgotten Account

- If any user will lose his or her password, then, the system allows resetting their passwords.

2.3. NON-FUNCTIONAL REQUIREMENTS

Nonfunctional requirements are the properties that your product must-have. Think of these properties as the characteristics or qualities that make the product attractive, usable, fast, or reliable, and they generally support all users in that they describe the business standards and the business environment, as well as the overall user's experience (user attributes). They do not alter the product's functionality. That is, the functional requirements remain the same no matter what properties you attach to them. Our project includes some of the quality attributes.

- **Usability** – It allows customers and system owners to use this system easily and efficiently since it supports user-friendly interface including using the Amharic language. The app also supports real-time data receiving and providing to the stakeholders without any redundancy of information. It will be accessible on any smartphone, or tablet, and others.
- **Reliability** – The system will have crash handling mechanisms to recover from any failure during the transaction process in a few seconds. This mechanism is supported by the database itself.
- **Performance** – Our system has a short response time for given access, accuracy, efficiency, and high throughput (high rate of work), since the back end will be built with firebase that is a non-blocking and asynchronous, no one is waiting for others process to complete when accessing in the same time that is many users access the system in an instant.
- **Availability** - the system is accessible anytime, anywhere via mobile devices, and tablets with an internet connection.
- **Security**– We use a secure and non-relational database to protect customer information and customer points from any harm forced by unauthorized access. Any user of this system will have a username and password stored in the database. Both the username and password are encrypted data, no one can use them without authentication.

a. Issue related to Access Control: It provides a form of authenticated user accesses only what they are authorized for. This issue suggests that accesses to resources and services should be permission based. If the authenticated user is permitted/allowed to access resources and services then these permitted users should not be denied access to that services that they are legitimately expect to received. To produce secure software, it is mandatory to implement access control at requirement perspective.

b. Issue related to Accountability: Accountability is a functional requirement that plays a vital role for maintaining number of logs for any task/ activities or action performed to accomplish a work in order to maintain security of prevention. It involves the application tracking activities of processes and maintaining log details. Security can be maintained through audit transactions which help to determine the attacker or source of attack if occurred in software.

c. Issue related to Accuracy: For any secured and reliable software system, it is necessary to produce accurate requirement for the system. The issues relate that the software development team should perform various activities such as scenarios, interviews, etc. in order to produce correct and accurate requirement for the software. Accuracy can also be considered in terms quality factors such as speed, time.

- **Sustainability** – The capacity of the software to endure. In other words, it means that the software will continue to be available in the future, on new platforms, meeting new needs which will require continual maintenance and updates.

2.4. FEASIBILITY STUDY

The feasibility study is the procedure to predict the outcome of an investigation examination or assessment of a planned scheme along with possible gain. We have tried to check whether our new proposed project is feasible or not feasible in different conditions and situations. Some of the evaluated visibility conditions are listed below.

- **Product feasibility** – The final product of our new project is a mobile application used to, give them some offers, communicate with customers directly to increase benefits and satisfaction, and store customer information's to determine who is loyal.

- **Economic Feasibility** – This project is economically feasible since the cost is lowered for our customers in their day-to-day endeavors, but the objective of this project is to lessen the cost of manually finding and hiring transport services Generally, this new project is cost-effective and efficient.

- **Technical Feasibility** -- Technical feasibility is mainly associated with the technologically evaluates the project. The main technologies and tools included with the system are:

- **Microsoft office** – used to write and edit the project's documentation.
- **Draw Io** – used to draw all the design models needed for the system.
- **Flutter** – cross platform tool used to develop a mobile application for both iOS and Android.
- **Dart** – the language in which flutter is built on.

- **Firestore** – the database for our system.
- **Legal Feasibility** – The on-demand courier service is legally feasible because this system does not conflict with national or international legal requirements. Any personal information is stored securely. This system is focused on the transport services only, not on providing the legal Document for the goods.
- **Operational Feasibility** – We have studied the behavioral feasibility of our project. It will fulfill all the functional and business requirements to satisfy user needs. These actions forecast all possible schemes to recognize and resolve troubles. This system also supports user-friendly interfaces with minimum input forms and any help to avoid confusion.
- **Schedule Feasibility** – We have planned to complete this system in 4 months with different phases and with iterative requirement analysis. If it is necessary through the development cycle. Currently, we are going with the right schedule to finish all phases with the specified time interval. So, we are sure our proposed project is schedule feasible. The schedule of our project is shown in table 2.1

TABLE 2. 1: SCHEDULE OF THE PROJECT

Task	Start date	Duration
Analysis		
Requirement analysis	May, 05, 2021	6
Requirement's document	May, 11, 2021	5
Design		
Database design	May, 16, 2021	11
Ui design	May, 16, 2021	12
Software Design	May,28, 2021	20
Development		
Develop system modules	Jun,18, 2021	35
Integrate system modules	Jul,23, 2021	15
Initial Testing	Aug,8, 2021	5
Testing		
Unit testing	Aug,13, 2021	7
Integration	Aug,20, 2021	5
Complete testing	Aug,25, 2021	3
Deployment		
Deployment	Aug,28, 2021	5
Maintaining	-	-

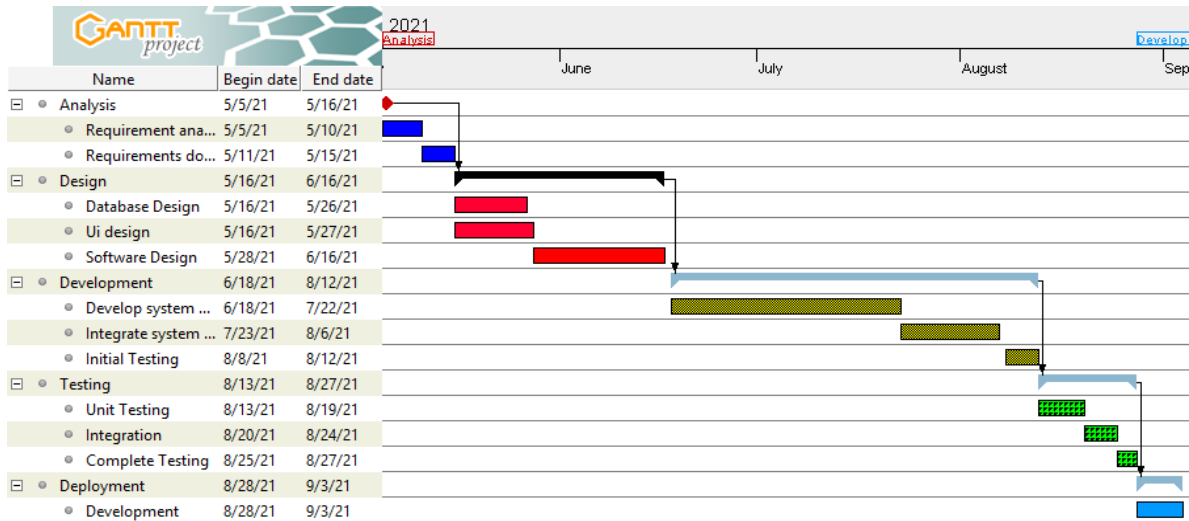


FIGURE 2. 1: GANTT FIGURE FOR PROJECT SCHEDULE

CHAPTER 3: SYSTEM ANALYSIS AND MODELING

3.1 OVERVIEW

In the previous chapters, we have seen the basic theme of the project we have intended to develop. Based on the requirements we are aiming to satisfy the goals we set, in this chapter, we are going to see and analyze the basic processes and models we are going to implement in the overall development of the project and the chronological operations and activities the system going to perform afterward. In modeling the proposed system, we will display the basic functions, activities, system structure, and actors involved using UML diagrams.

3.2 SCENARIO-BASED MODELING

Sets of scenarios can be used to stress-test or analyze the implications of any critical decision facing the user. And a set can be used to gauge the full plausible variability of any quantity of interest, as well as the interactions between various quantities [4]. In this accord, we are going to see the future dimensions of the system towards the actors, use cases, and the activities which are triggered and performed by the actors which are made possible by the system.

3.2.1 USE CASE IDENTIFICATION

Function Contextualization models the interactions between a function and its actors. We use diagrams to show the whole system operations based on the actor and the interactions between the actor and the function. The use cases imply the activities/functions which are triggered by the user/actor of the system. In other words, it shows a system function as a whole, with its inputs and outputs events to and from other external entities called actors.

Considering the proposed system, we have tried to extract the activities the system does and the actors involved as shown below.

TABLE 3. 1: USE CASE LISTS

UC_ID	Use Case
UC_1	Signup
UC_2	Login
UC_3	Place order
UC_4	Track order
UC_5	Report driver
UC_6	View history
UC_7	Cancel order
UC_8	Update profile
UC_9	Accept/decline order
UC_10	View customer info
UC_11	Update status
UC_12	Enter receiver info
UC_13	Choose vehicle
UC_14	Specify receiver info
UC_15	Enter pickup and delivery point
UC_16	Logout
UC_17	Register drivers
UC_18	Update driver account
UC_19	Ban/unban user
UC_20	Generate report
UC_21	Supervise conflict
UC_22	Set terms
UC_23	Delete user account
UC_24	Update admin account

3.2.2 ACTOR IDENTIFICATION

Identifying actors is one of the first steps in use case analysis. Each type of external entity with which the system must interact is represented by an actor. For instance, the operating environment of a software system consists of the users that the system interacts with. We used the term actor simply to imply the future users of the system after completion of the project. In this way, we will have a better way of analyzing the system at the operational level.

The proposed system has the following actors:

- ❖ **Administrator:** Performs the necessary tasks which are mandatory in managing the system as a whole, it could be driver's account, customer's account, or the system settings generally. In addition, the administrator will do disciplinary works which may be banning user or driver and supervising conflicts which is maybe from observation or the reports submitted by the customer. The administrator also generates reports of orders, users, drivers and reports submitted and also has the authority to set and modify basic system entities such as cost and terms.
- ❖ **Driver:** Performs tasks related to picking up, transportation, and delivery of goods as per customer request. In the process, the driver will be able to accept or decline a request and if the order is accepted the driver will update the status from accepted to picked up, then delivered. In the course of delivery, the pickup and drop locations will be displayed on the driver's phone.
- ❖ **Customer:** is the actor that requires services from the driver to transport their merchandise. The customer is the one who places the order with necessary information including the pickup and drop locations and also the receiver. After placing the order, the customer will be able to track and cancel the order using the phone in hand. If the customer ever gets inconvenience, he/she will be able to report the situation or the driver.

3.2.3 USE CASE DESCRIPTION AND USE CASE DIAGRAM

3.2.3.1 USE CASE DIAGRAM

The use case diagram is a way of modeling the functionality of a system using actors and use cases. A system is an entity that is being developed or operated. Use cases are the activities or functions which are triggered by the actor and performed by the system. Actors are entities operating the system with a certain role.

Use case diagram relates these basic factors of a system in a systematic way

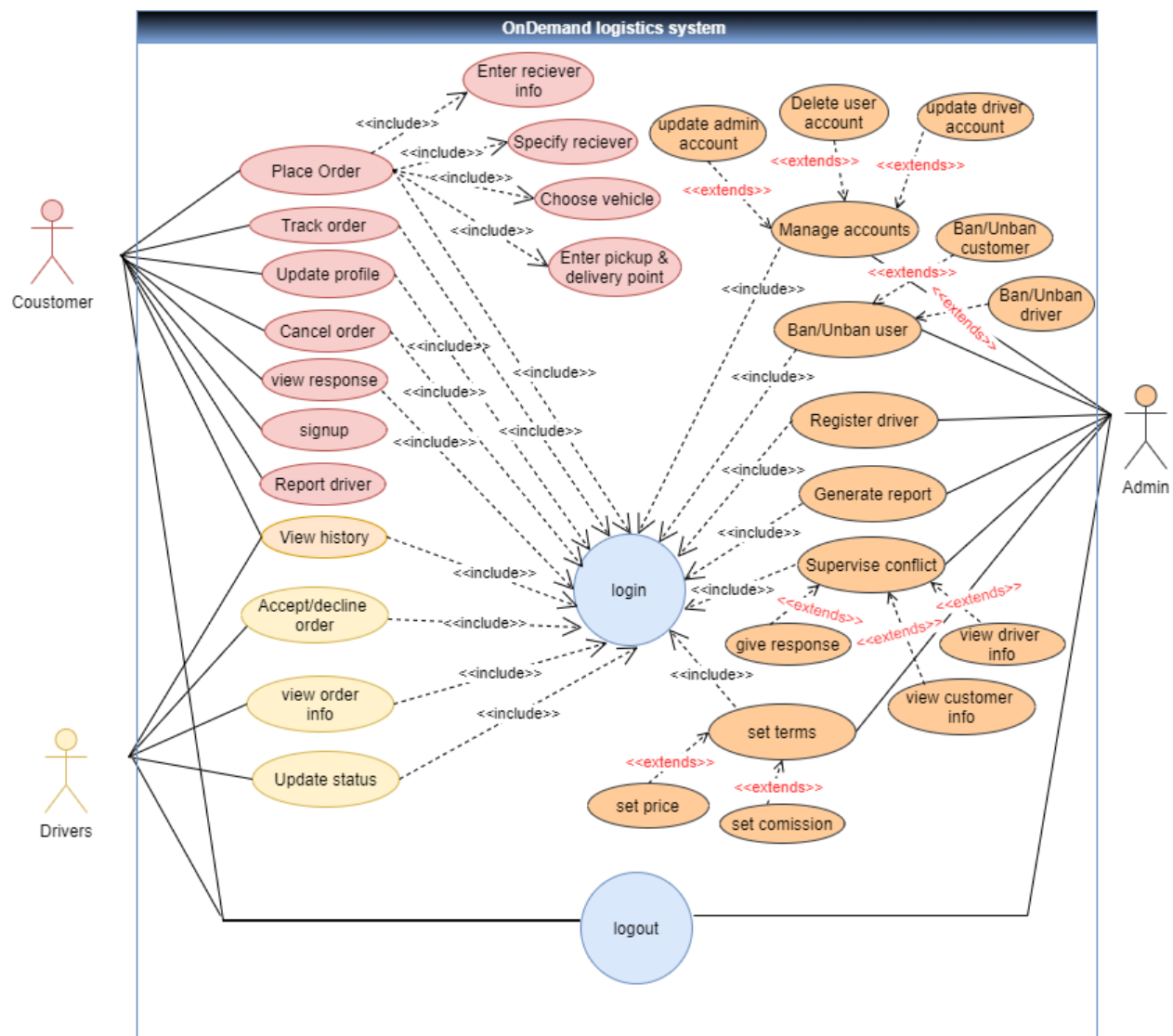


FIGURE 3. 1: USE CASE DIAGRAM FOR ONDEMAND LOGISTICS SYSTEM

3.2.3.2 USE CASE DESCRIPTION

Use case description another way of describing a use case, which is sometimes called use case narrative a use case description explains the interaction between the system and the user which is demonstrated graphically in the use case diagram.

The tables below describe the use cases and correspond to the use cases of the proposed system we seen in the use case list or the use case diagram.

TABLE 3. 2: USE CASE DESCRIPTION FOR SIGNUP

Use case name	Signup
Use case ID	UC_1
Actor	Customer
Description	In this use case, anyone who wants to use the system fills necessary information and opens an account
Pre-condition	A user just needs to install the application
Basic course of action	Action: <ol style="list-style-type: none"> 1. Open program 2. User clicks signup 3. The user enters any necessary information required 4. User clicks signup 5. User's account created and opened 6. End of the use case
The alternate flow of action	A1 if the account already exists the system displays "this account already exists"
	A2 if the customer didn't fill in all required information system displays "fill in all required information"
Post-condition	The user will know have an account with which he/she can access any service available

TABLE 3. 3: USE CASE DESCRIPTION FOR LOGIN

Use case name	Login
Use case ID	UC_2
Actor	Customer, Driver, Admin
Description	This use case describes how users can be able to login into their accounts.
Pre-condition	A user must register first to acquire an account to login to
Basic course of action	Action: <ol style="list-style-type: none"> 1. Open program 2. User clicks “login” 3. The user fills in any provided information 4. System authentication follows 5. User account page opens 6. End of the use case
The alternate flow of action	A1 if the filled information is not correct the system will display “please fill your correct username and password”
	A2 if the customer forget password the customer will go by the choice “forgot password?”
Post-condition	Since the user page is open user will access any service available in reach

TABLE 3. 4: USE CASE DESCRIPTION FOR PLACE ORDER

Use case name	Place order
Use case ID	UC_3
Actor	Customer
Description	This use case describes how users place an order/delivery of the type they want

Pre-condition	A user has to specify pickup and delivery points and must provide contact information
Basic course of action	Action: <ol style="list-style-type: none"> 1. User clicks “place order” 2. Specify pickup and delivery location 3. Specify vehicle type as they see fit for their item 4. Choose receiver 5. Confirm order 6. End of the use case
The alternate flow of action	A1 if all the required information was not filled by the customer before placing order system displays “fill all the required information”
Post-condition	The user will now get the service he/she requires and can monitor the status using the phone in hand

TABLE 3. 5: USE CASE DESCRIPTION FOR TRACK ORDER

Use case name	Track order
Use case ID	UC_4
Actor	Customer
Description	This use case describes how users track the item while it is moving till delivered
Pre-condition	A user has to set initial and delivery location and place order properly
Basic course of action	Action: <ol style="list-style-type: none"> 1. Place order 2. Follow the movement whether it is on route or not on the screen 3. End of the use case

The alternate flow of action	There is no alternate action
Post-condition	The user will follow if the item is on the right track or if it has reached the intended destination

TABLE 3. 6: USE CASE DESCRIPTION FOR REPORT DRIVER

Use case name	Report driver
Use case ID	UC_5
Actor	Customer
Description	This use case describes how users report the if there is any issue
Pre-condition	Specify the driver which is about to be reported and enter reason
Basic course of action	Action: <ol style="list-style-type: none"> 1. User opens account 2. User clicks “report” 3. User fill in report 4. Submit report 5. End of the use case
The alternate flow of action	No alternate action here
Post-condition	The report will be sent to admin page so that it could be responded

TABLE 3. 7: USE CASE DESCRIPTION FOR VIEW HISTORY

Use case name	View history
Use case ID	UC_6
Actor	Customer, driver

Description	This use case describes how the user or the driver reviews the history based on the time frame
Pre-condition	The user should login to the system
Basic course of action	Action: <ol style="list-style-type: none"> 1. User click on “history” 2. History page opens 3. User views history 4. End of the use case
The alternate flow of action	No alternate flow of action
Post-condition	The user will be able to view and monitor the information in the history

TABLE 3. 8: USE CASE DESCRIPTION FOR CANCEL ORDER

Use case name	Cancel order
Use case ID	UC_7
Actor	Customer
Description	This use case shows the way how the user cancels placed order
Pre-condition	There should be placed order first
Basic course of action	Action: <ol style="list-style-type: none"> 1. User places order 2. If there is a change in mind, the user clicks on “cancel order” 3. The system cancels the order 4. End of the use case
The alternate flow of action	No alternate flow of action
Post-condition	The placed order will be canceled and no further action will be taken

TABLE 3. 9: USE CASE DESCRIPTION FOR UPDATE PROFILE

Use case name	Update profile
Use case ID	UC_8
Actor	Customer
Description	This use case shows the way how the user updates the user profile information
Pre-condition	User must have an account to make a change to and login to it
Basic course of action	Action: <ol style="list-style-type: none"> 1. User clicks on “update profile” 2. User makes changes on preferred information 3. User click on “confirm change” 4. The system applies the update 5. End of the use case
Post-condition	The changes made by the user will be saved and the system starts managing the account with the new change

TABLE 3. 10: USE CASE DESCRIPTION FOR ACCEPT/DECLINE ORDER

Use case name	Accept/decline order
Use case ID	UC_9
Actor	Driver
Description	This use case shows how the driver accept/decline an order placed by a nearby customer
Pre-condition	First, the system should notify the driver of the request made by the customer
Basic course of action	Action:

	<ol style="list-style-type: none"> 1. The system notifies the driver about the requested order 2. Driver clicks “accept order”/” decline order” 3. End of the use case
The alternate flow of action	No alternate flow of action
Post-condition	If the driver accepts, he/she proceeds with the order, otherwise, the system searches for another driver around

TABLE 3. 11: USE CASE DESCRIPTION FOR VIEW CUSTOMER INFO

Use case name	View order info
Use case ID	UC_10
Actor	Driver
Description	This use case shows how the driver is going to see the customer’s info which he/she accepted the order for
Pre-condition	The driver must accept the order to see the information of the customer who placed the order
Basic course of action	<p>Action:</p> <ol style="list-style-type: none"> 1. Driver clicks on “accept order” 2. The system displays necessary order information 3. Driver observes the information 4. End of the use case

The alternate flow of action	A1 if the driver declines the order the system hides the customer information from the driver
Post-condition	After acquiring the necessary information, the driver will get in touch with the customer using it

TABLE 3. 12: USE CASE DESCRIPTION FOR UPDATE STATUS

Use case name	Update status
Use case ID	UC_11
Actor	Driver
Description	This use case shows how the driver updates the status of the item whether if it is picked up or delivered or if he/she is available
Pre-condition	There should be a request in the reach of the driver
Basic course of action	<p>Action:</p> <ol style="list-style-type: none"> 1. User request reaches the driver 2. If the driver is free clicks “accept order” 3. As soon as the driver picks up the item driver clicks “item picked up” 4. The driver clicks “delivered” after the delivery is made 5. End of the use case
The alternate flow of action	A1 if the driver declines the order the system will quit the notification and looks for another driver around
Post-condition	The customer will be notified of each step and tracks the status of the item

TABLE 3. 13: USE CASE DESCRIPTION FOR ENTER RECEIVER INFO

Use case name	Enter receiver info
Use case ID	UC_12
Actor	Customer
Description	This use case describes the way the customer enters the necessary contact info of the customer
Pre-condition	The customer has to be in the process of placing an order and the receiver has to be another person
Basic course of action	Action: <ol style="list-style-type: none"> 1. User clicks “another person”, in the option receiver 2. User fills in preferred information about the receiver 3. User clicks “submit” 4. End of the use case
The alternate flow of action	A1 if the person who placed the order is the receiver the system fetch contact info from the account
	A2 if the required information is not filled the system displays “please fill customer information”
Post-condition	The system sends this information to the driver so that the driver could make contact with the receiver

TABLE 3. 14: USE CASE DESCRIPTION FOR CHOOSE VEHICLE

Use case name	Choose vehicle
Use case ID	UC_13
Actor	Customer
Description	This use case shows how the customer chooses the type of vehicle which is convenient for the item

Pre-condition	The customer has to be in the process of placing an order and the customer has to know the exact/approximate fit of the item to order a vehicle
Basic course of action	Action: <ol style="list-style-type: none"> 1. User clicks the item's category of weight 2. The System displays types of vehicles under that category 3. User clicks the preferred type of vehicle 4. End of the use case
The alternate flow of action	No alternate flow of action here
Post-condition	The system notifies drivers with that certain type of vehicle about the placed order

TABLE 3. 15: USE CASE DESCRIPTION FOR SPECIFY RECEIVER INFO

Use case name	Specify receiver info
Use case ID	UC_14
Actor	Customer
Description	This use case describes the way how the customer specifies which person is to receive the item
Pre-condition	The customer has to be in the process of placing an order and the customer has to know the person who is to receive the item
Basic course of action	Action: <ol style="list-style-type: none"> 1. The system displays two choices whether the item is to be received by "self" or "other person" 2. User clicks on the preferred choice 3. End of the use case
The alternate flow of action	No alternate flow of action

Post-condition	If the receiver is another person the system requires the contact information of the receiver to send it to the driver or if it is the customer itself the system sends the contact from the existing account information
-----------------------	---

TABLE 3. 16: USE CASE DESCRIPTION FOR ENTER PICKUP AND DELIVERY POINT

Use case name	Enter pickup and delivery point
Use case ID	UC_15
Actor	Customer
Description	This use case describes the way how the customer enters pickup and drop locations
Pre-condition	The customer has to be in the process of placing an order and should know the exact pickup and delivery locations
Basic course of action	Action: <ol style="list-style-type: none"> 1. The Customer enters the pickup and delivery point 2. System processes and displays the route on the map 3. End of the use case
The alternate flow of action	No alternate action here
Post-condition	The system then takes the distance, calculates and displays the price for the customer for the next step

TABLE 3. 17: USE CASE DESCRIPTION FOR LOGOUT

Use case name	Logout
Use case ID	UC_16
Actor	Customer
Description	This use case shows how users sign-out the system after use

Pre-condition	The customer has to has an account and must login into it
Basic course of action	Action: <ol style="list-style-type: none"> 1. The Customer clicks “logout” 2. System sign-out the account 3. End of the use case
The alternate flow of action	No alternate flow of action
Post-condition	The user account will be logged out and authentication will be required for future login

TABLE 3. 18: USE CASE DESCRIPTION FOR REGISTER DRIVER

Use case name	Register driver
Use case ID	UC_17
Actor	Admin
Description	This use case shows how the admin add drivers to the system
Pre-condition	The admin must have credentials and login to the system
Basic course of action	Action: <ol style="list-style-type: none"> 1. The admin clicks “Add driver” 2. The system displays the information to be filled 3. Admin enters the information 4. Admin clicks “Add” 5. End of the use case
The alternate flow of action	A1 if all the required information no filled in the system displays “Required information not filled”
	A2 if the account already exists system displays “account already exists”
Post-condition	The driver will have an account and will be notified if there is any order around

TABLE 3. 19: USE CASE DESCRIPTION FOR UPDATE DRIVER ACCOUNT

Use case name	Update driver account
Use case ID	UC_18
Actor	Admin
Description	This use case shows how the admin updates driver's account
Pre-condition	The admin should login to the admin account
Basic course of action	<p>Action:</p> <ol style="list-style-type: none"> 1. The admin logs into account 2. The admin clicks driver's account 3. System displays the selected account 4. The admin clicks update profile 5. Admin fills in account update 6. Admin clicks Done 7. System updates the account 8. End of the use case
The alternate flow of action	A1 if the update input is not valid system notifies the input is not valid
	A2 if admin didn't fill in correct input system notifies to put proper input
Post-condition	System saves and applies the updates and the data will be stored to database

TABLE 3. 20: USE CASE DESCRIPTION FOR BAN USER

Use case name	Ban User
Use case ID	UC_19
Actor	Admin

Description	This use case shows how the admin ban drivers and customers in any deserving issues.
Pre-condition	The admin must know the account to be banned
Basic course of action	Action: <ol style="list-style-type: none"> 1. The admin clicks the account which is to be banned 2. The system displays the account 3. Admin clicks “Ban” 4. System bans the user/driver account 5. End of the use case
The alternate flow of action	No alternate flow of action
Post-condition	The banned driver/user will not be able to use the account until the issue is resolved

TABLE 3. 21: USE CASE DESCRIPTION FOR GENERATE REPORT

Use case name	Generate report
Use case ID	UC_20
Actor	Admin
Description	This use case shows how the admin generates and reviews the report and operation
Pre-condition	The admin must log in to the system
Basic course of action	Action: <ol style="list-style-type: none"> 1. Admin open program 2. The admin clicks “get report” 3. The system displays report 4. End of the use case
The alternate flow of action	No alternate flow of action
Post-condition	The admin watches and reviews the orders, history, and actors

TABLE 3. 22: USE CASE DESCRIPTION FOR SUPERVISE CONFLICT

Use case name	Supervise conflict
Use case ID	UC_21
Actor	Admin
Description	This use case shows how the admin manages complains and reports submitted
Pre-condition	There should be conflicts in hand
Basic course of action	Action: <ol style="list-style-type: none"> 1. The system notifies the admin about the conflict 2. The admin views the complaint information 3. Admin gives response 4. End of the use case
The alternate flow of action	No alternate flow of action
Post-condition	The admin will take the necessary measures to resolve the conflict

TABLE 3. 23: USE CASE DESCRIPTION FOR SET TERMS

Use case name	Set terms
Use case ID	UC_22
Actor	Admin
Description	This use case shows how the admin sets terms for order cost and commission paid for drivers.
Pre-condition	The admin must log in to the system and has to fill reasonable cost
Basic course of action	Action: <ol style="list-style-type: none"> 1. The admin clicks terms 2. The system displays the terms 3. The admin clicks change terms

	<ol style="list-style-type: none"> 4. Admin enters the change 5. Admin clicks Done 6. End of the use case
The alternate flow of action	No alternate flow of action
Post-condition	The system will calculate the cost based on the new modification for any orders to be placed onward

TABLE 3. 24: USE CASE DESCRIPTION FOR DELETE ACCOUNT USER

Use case name	Delete user
Use case ID	UC_23
Actor	Admin
Description	This use case shows how the admin deletes users' (customer/driver) account.
Pre-condition	Admin should log into admin account
Basic course of action	<p>Action:</p> <ol style="list-style-type: none"> 1. The admin clicks user's account 2. The system displays options 3. The admin clicks delete 4. System warns the admin about the action 5. Admin agrees by clicking ok 6. System deletes the account 7. End of the use case
The alternate flow of action	A1 if admin clicks NO on warning, the system cancels the process.

Post-condition	The account will be deleted and the user will no longer be able to use the account
-----------------------	--

TABLE 3. 25: USE CASE DESCRIPTION FOR UPDATE ADMIN ACCOUNT

Use case name	Update admin account
Use case ID	UC_24
Actor	Admin
Description	This use case shows how the admin updates the admin account
Pre-condition	The admin must log into the system
Basic course of action	Action: <ol style="list-style-type: none"> 1. The admin logs into account 2. The admin clicks update profile 3. The admin fills in the update 4. The admin clicks Done 5. System updates the account 6. End of the use case
The alternate flow of action	A1 if the update input is not valid system notifies the input is not valid
	A2 if admin didn't fill in correct input system notifies to put proper input
Post-condition	Admin credentials will be updated and the new updated admin-info will be stored to database

3.2.4 ACTIVITY DIAGRAM

Activity is a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. It is also called object-oriented flowchart. The UML activity diagrams below focus on the execution and flow of activities of the proposed system instead of implementation. In another word, A UML activity diagram helps to visualize a certain use case at a more detailed level which helps to see complicated use cases in step by step and easy fashion.

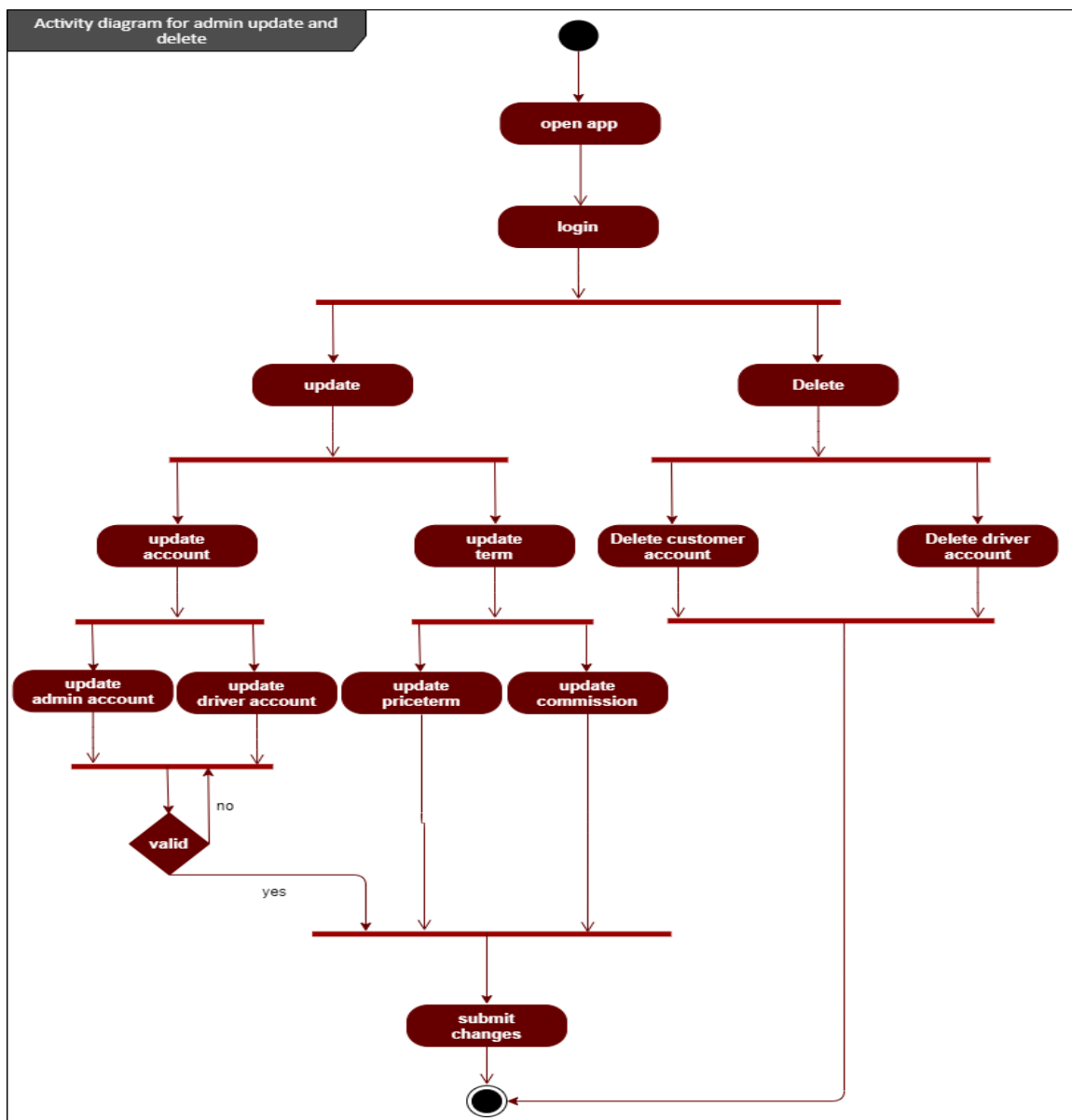
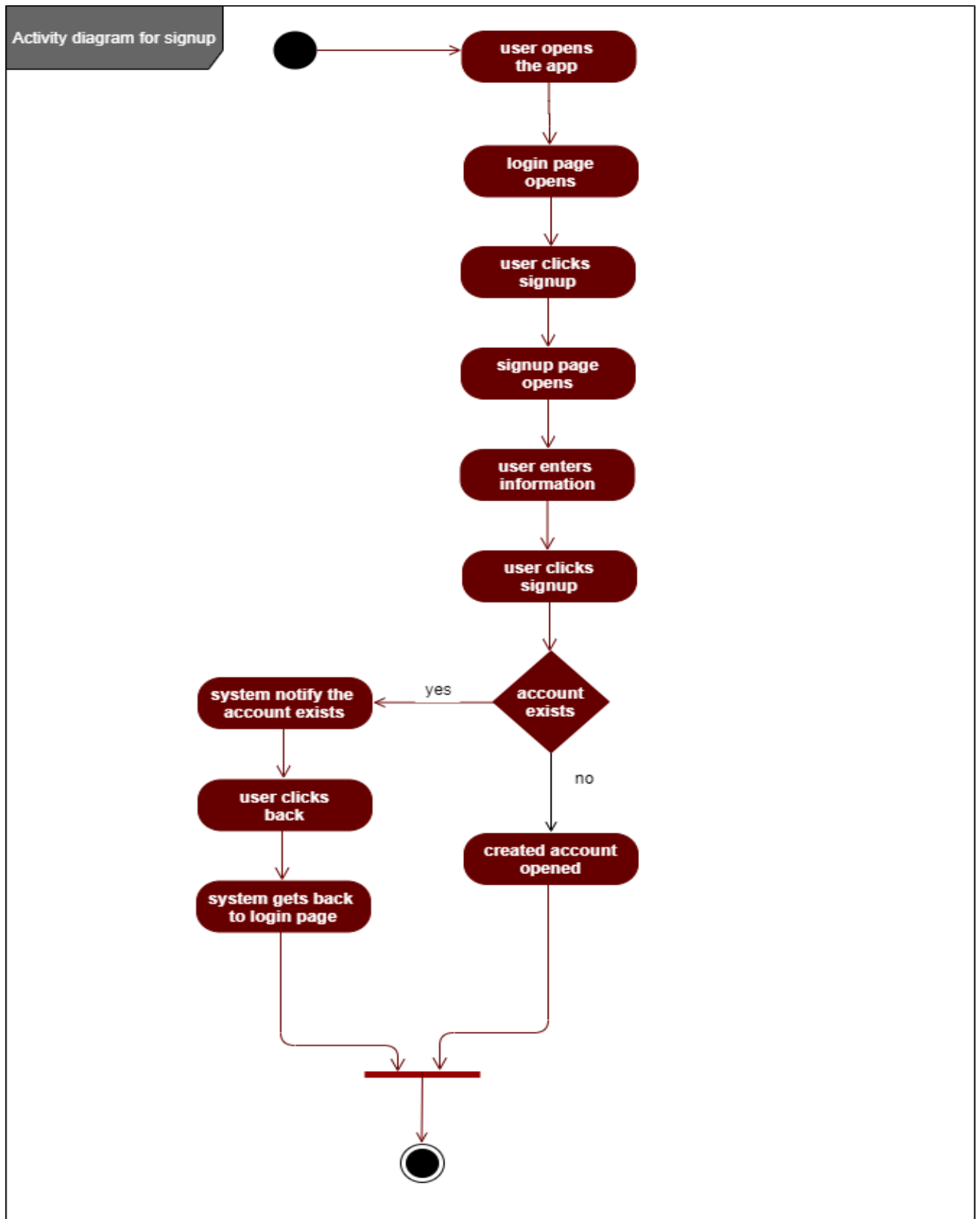
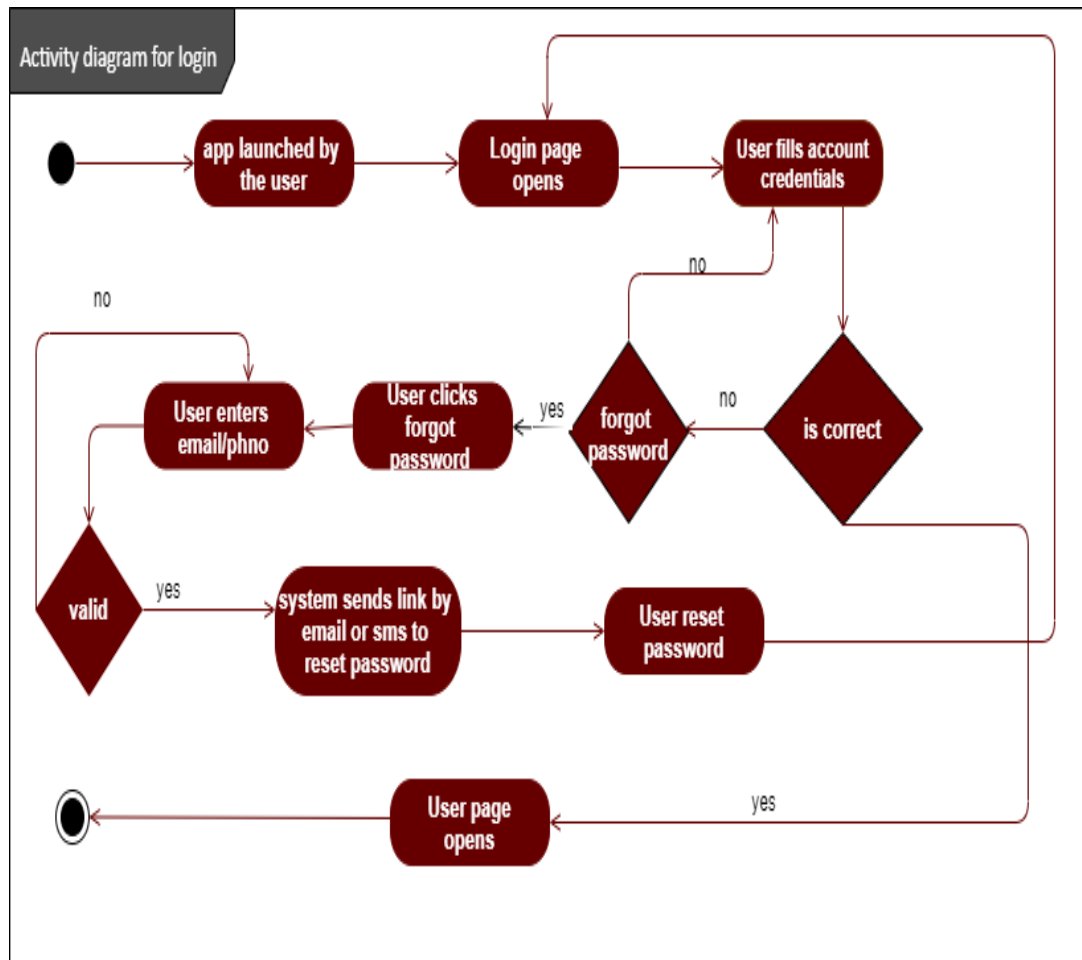
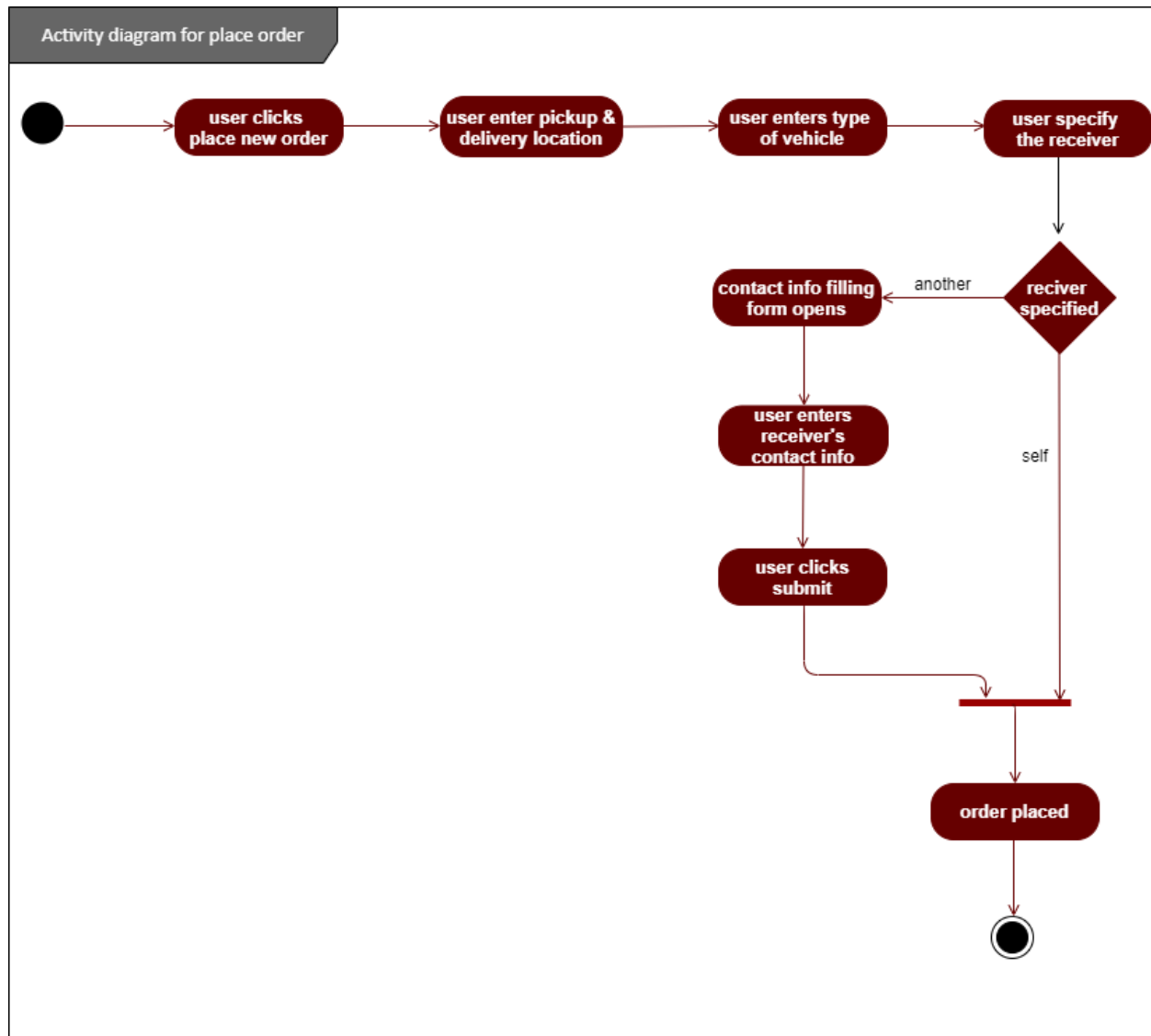
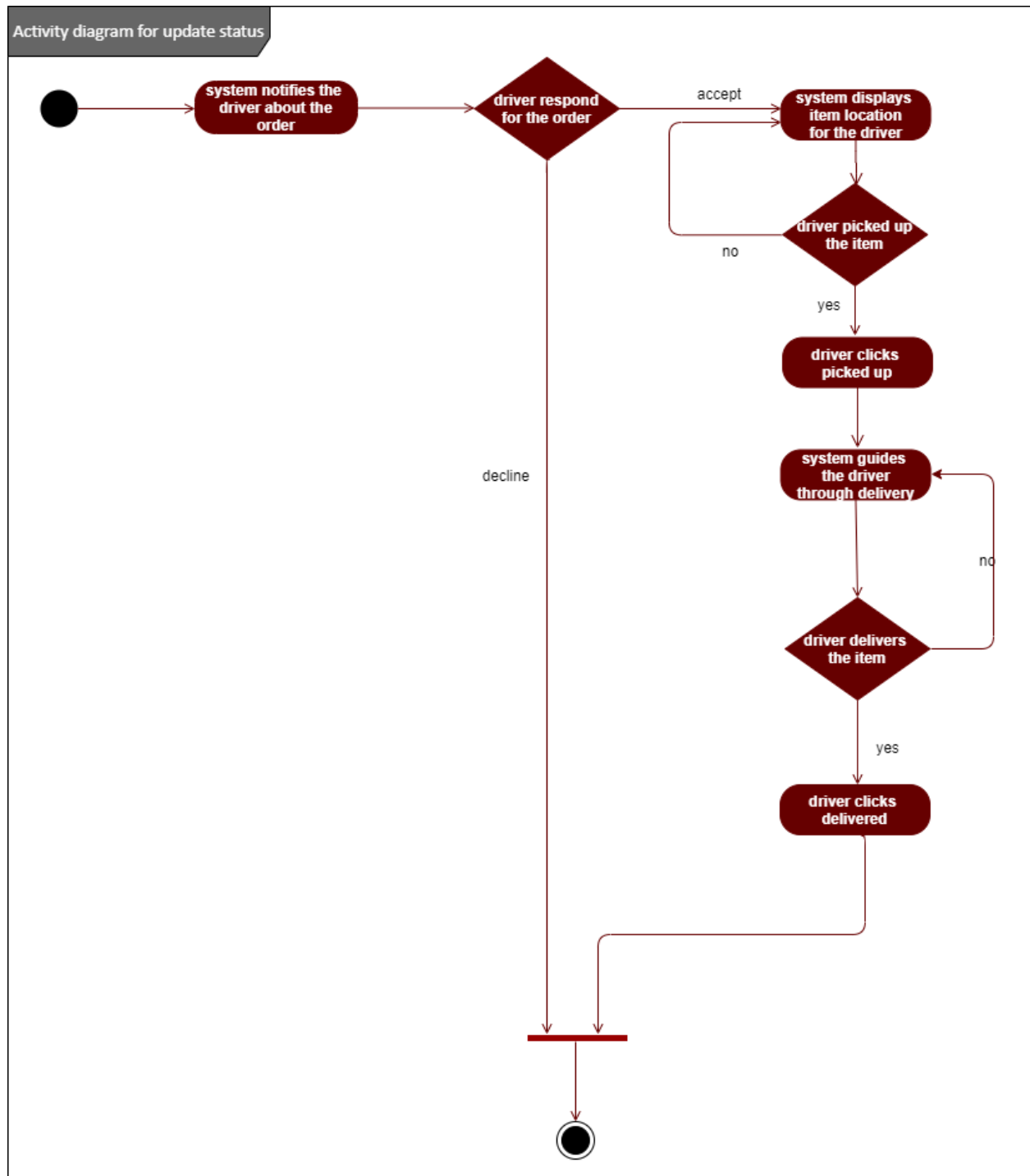


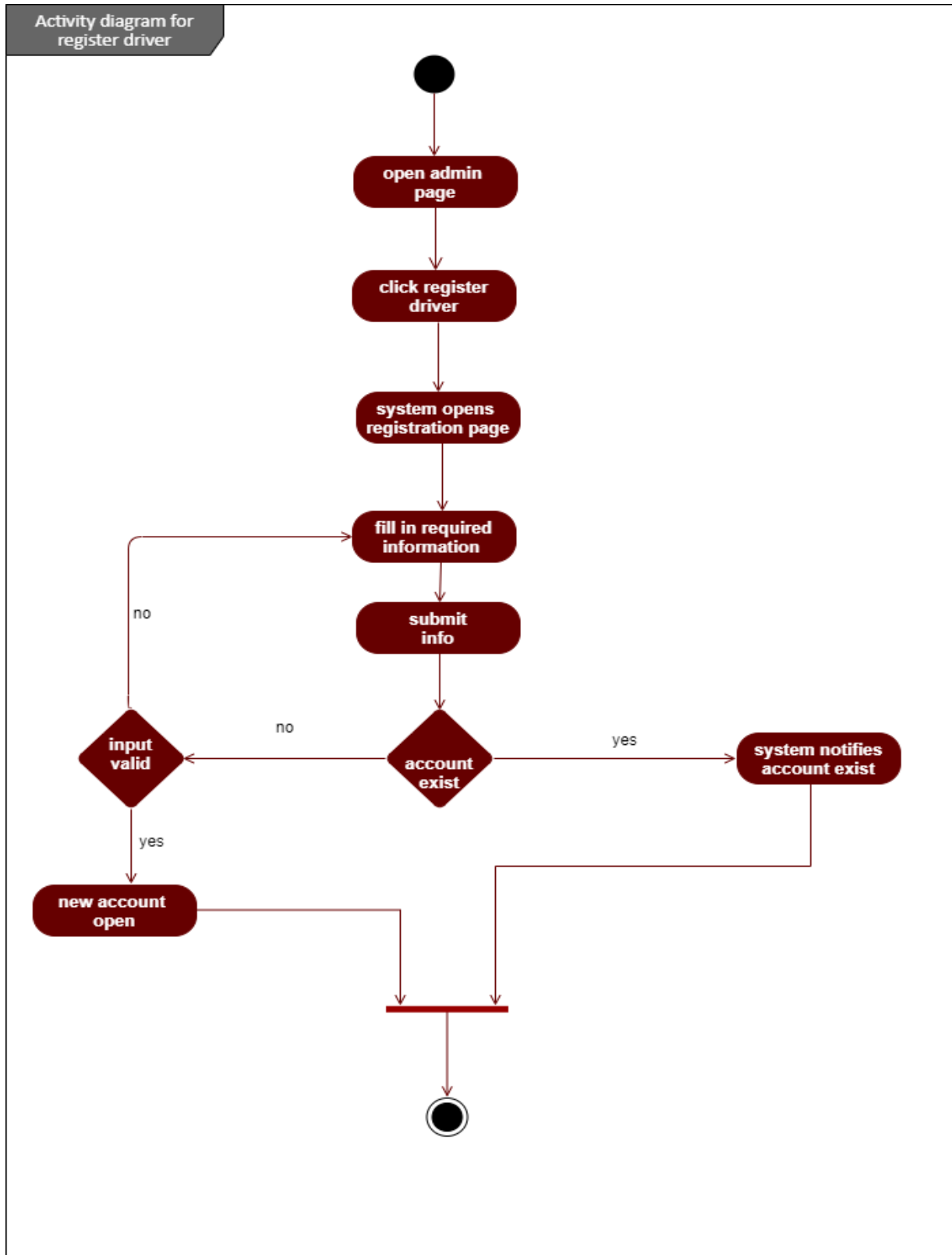
FIGURE 3. 2: ACTIVITY DIAGRAM FOR ADMIN UPDATE AND DELETE

**FIGURE 3. 3: ACTIVITY DIAGRAM FOR SIGNUP**

**FIGURE 3. 4: ACTIVITY DIAGRAM FOR LOGIN**

**FIGURE 3. 5: ACTIVITY DIAGRAM FOR PLACE ORDER**

**FIGURE 3. 6: ACTIVITY DIAGRAM FOR UPDATE STATUS**

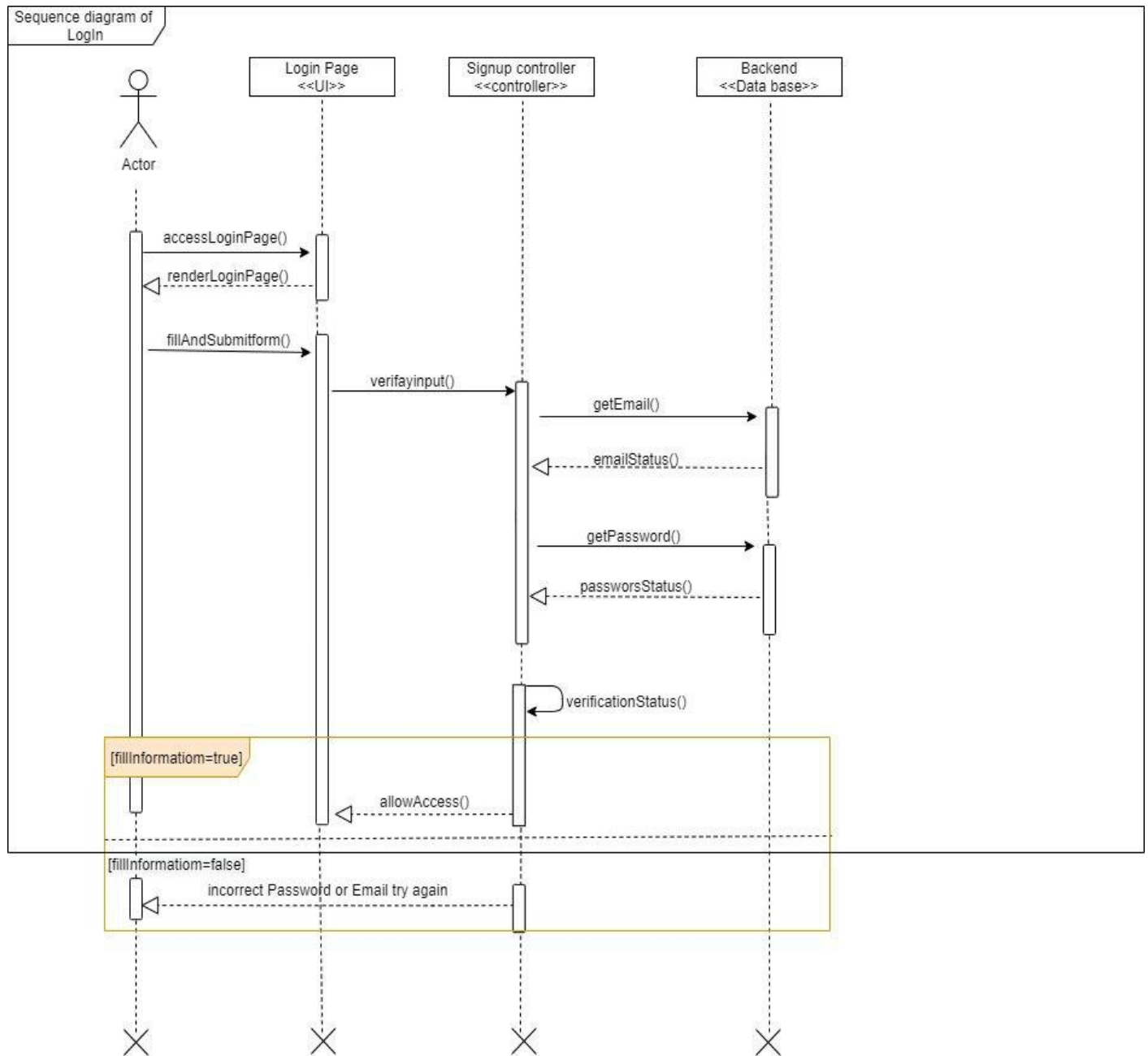
**FIGURE 3. 7: ACTIVITY DIAGRAM FOR REGISTER DRIVER**

3.3 Behavioral/ Dynamic Modeling

Behavioral models describe the internal behavior of a system. Representation of the details of a system process identified by use-cases show how objects collaborate to provide the functionality defined in the use cases [5]. Representations of changes in the data behavioral state focus is on the dynamic view of the system, not on how it is implemented.

3.3.1 Sequence Diagram

UML Sequence Diagrams which are also called event diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when [5]. These can help to predict how a system will behave and to discover responsibilities a class may need to have in the process of modeling a new system. In this part of the document, we aim to show how the different elements of the system interact with each other and exchange messages to fulfill the objectives. We have discussed the main sequences of the system as shown below.

**FIGURE 3. 8: SEQUENCE DIAGRAM OF LOGIN**

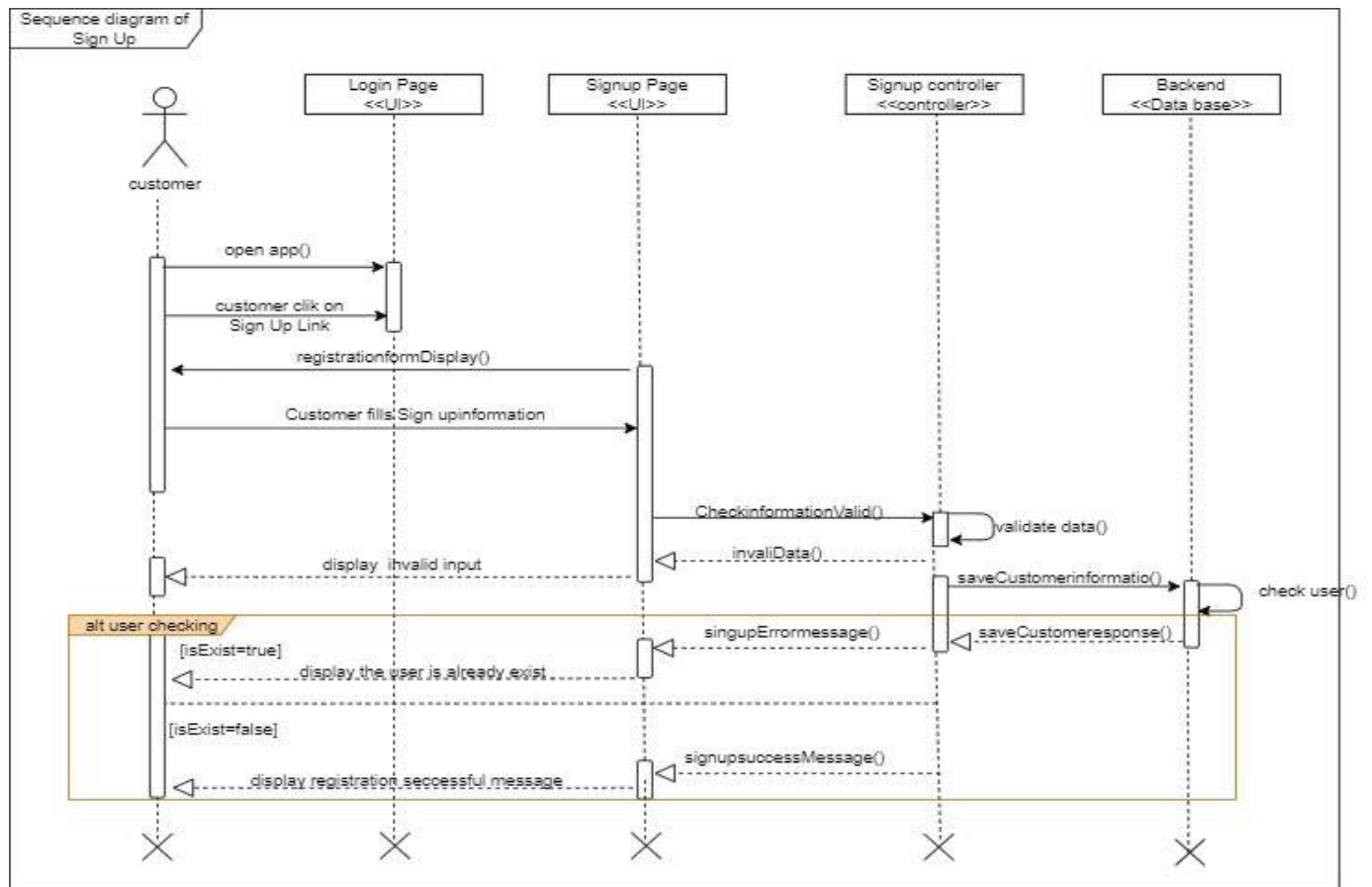


FIGURE 3. 9: SEQUENCE DIAGRAM OF SIGN UP

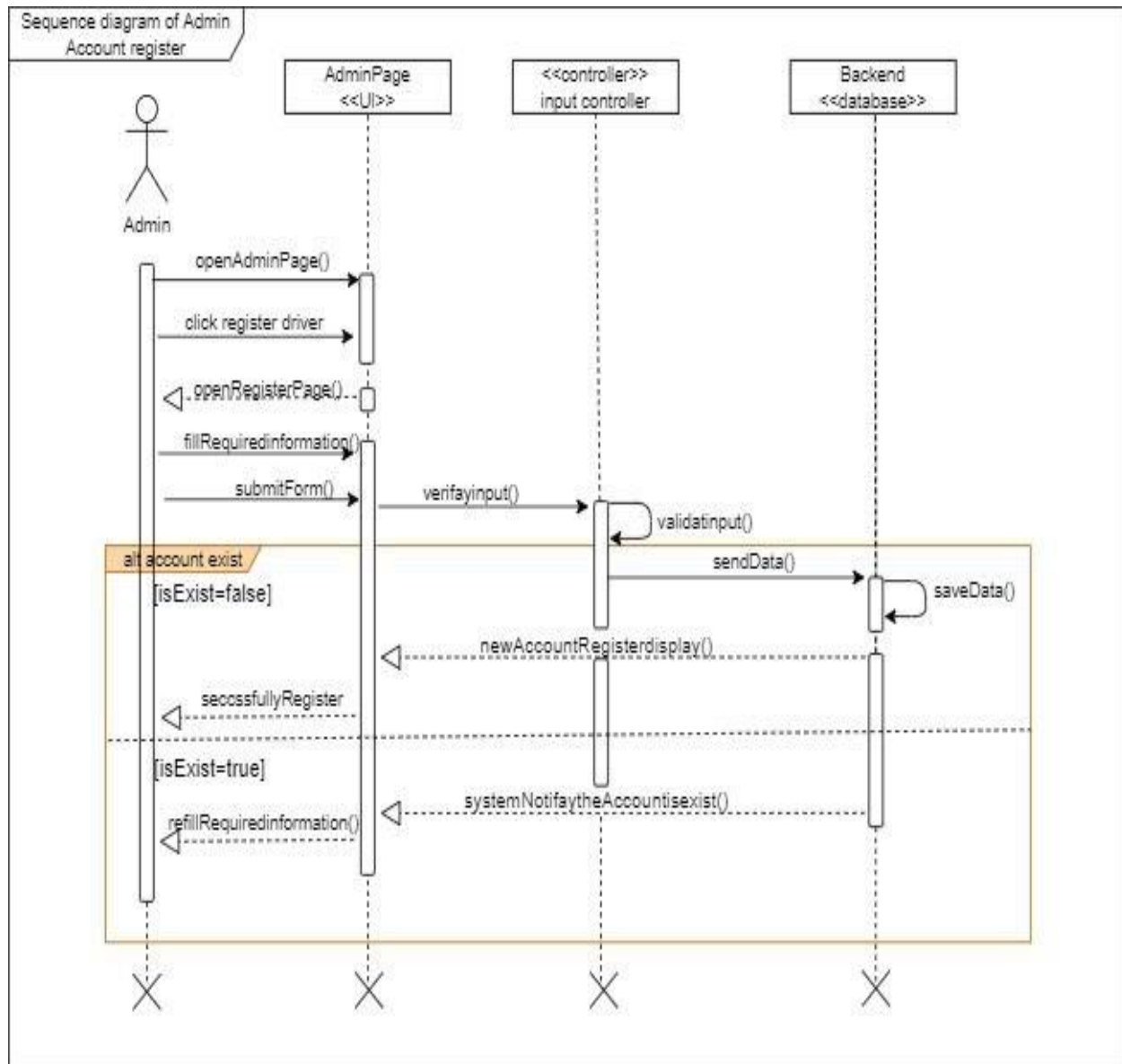


FIGURE 3. 10: SEQUENCE DIAGRAM OF ACCOUNT REGISTER

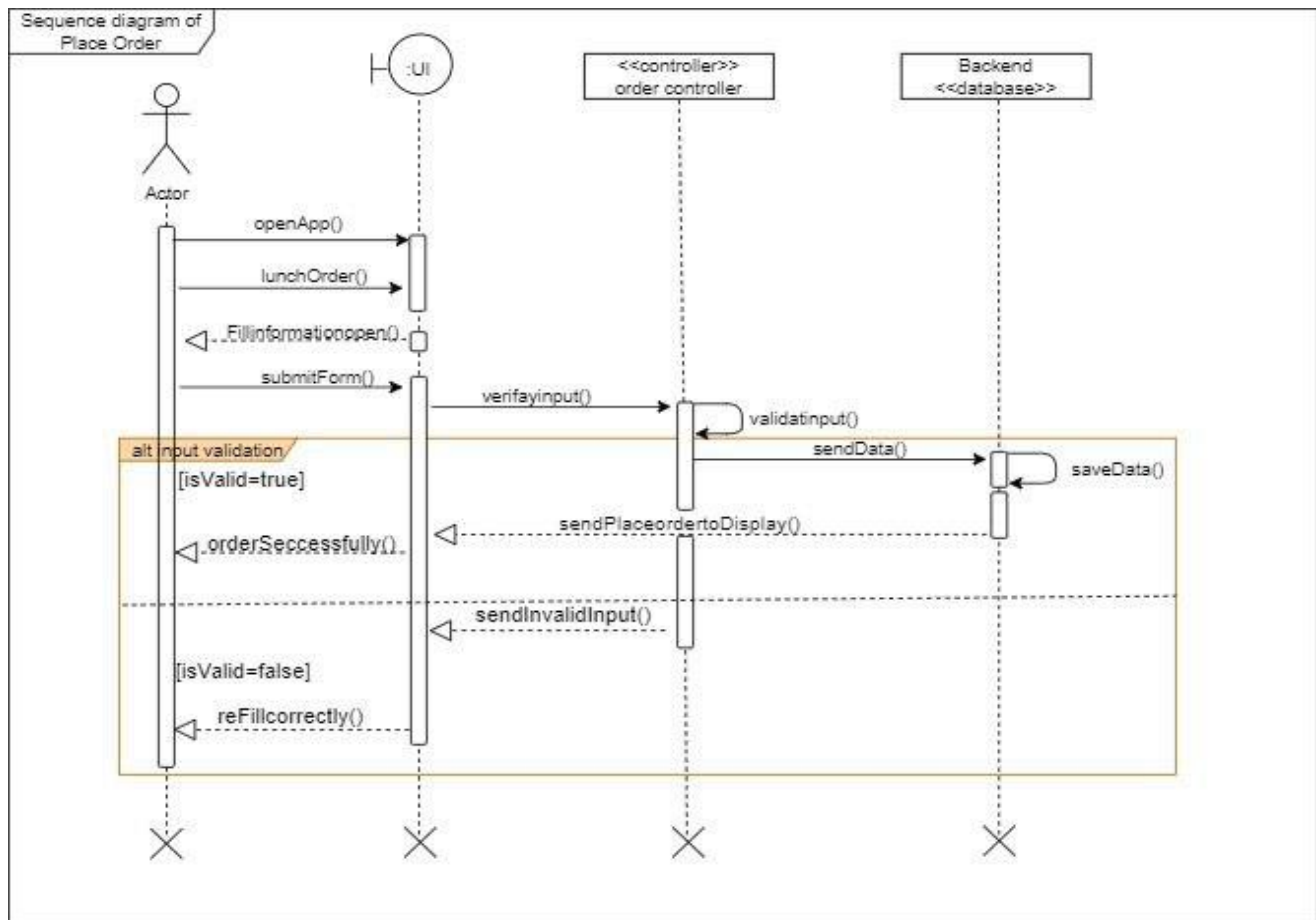


FIGURE 3. 11: SEQUENCE DIAGRAM OF PLACE ORDER

3.3.2 State Diagrams

A state diagram is used to represent the condition of the system or part of the system at finite instances of time [4]. it represents the behavior using finite state transitions. State diagrams are also referred to as State machines and State-chart Diagrams. These terms are often used interchangeably. So simply, in this part of the document we aim to model the dynamic behavior of a class in response to time and changing external stimuli.

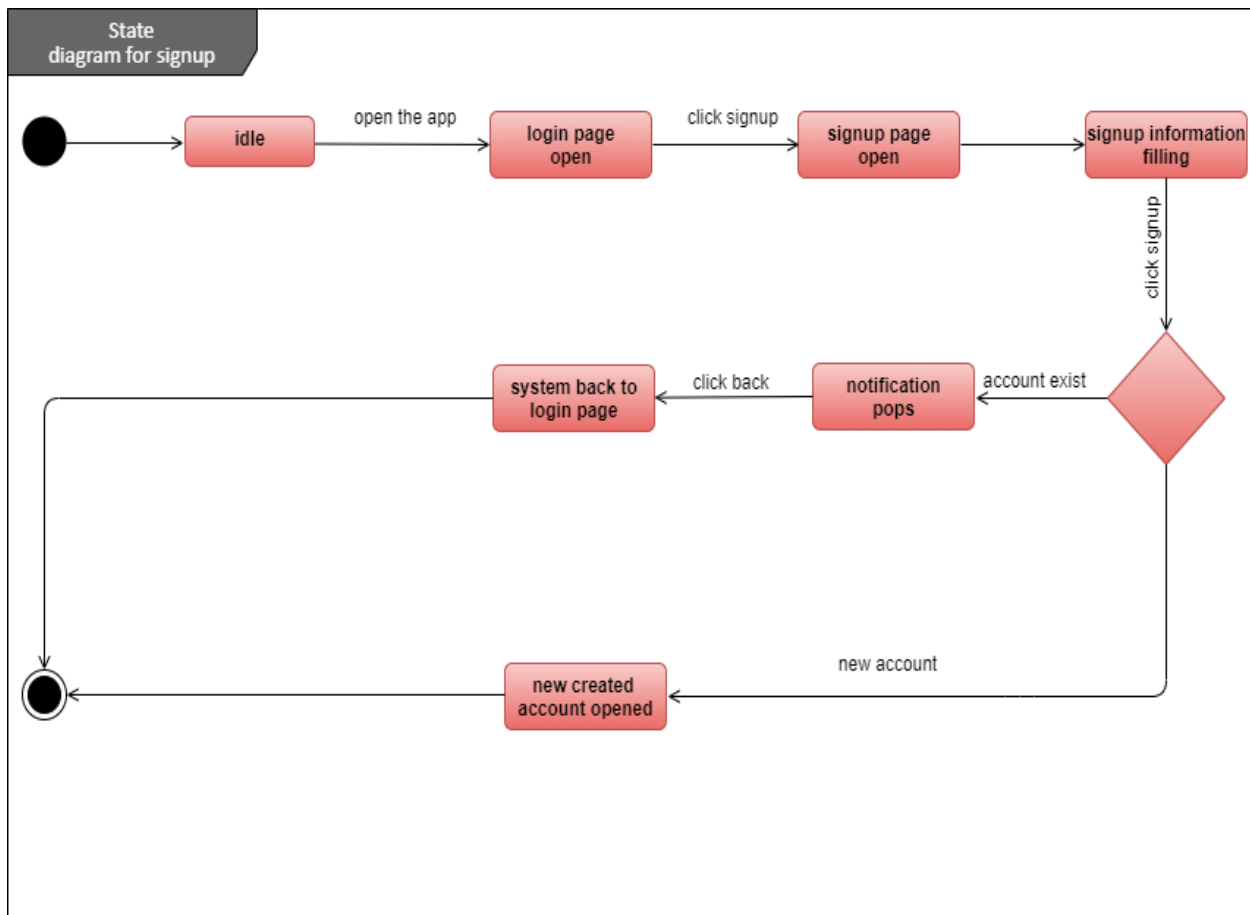


FIGURE 3. 12: STATE DIAGRAM FOR SIGNUP

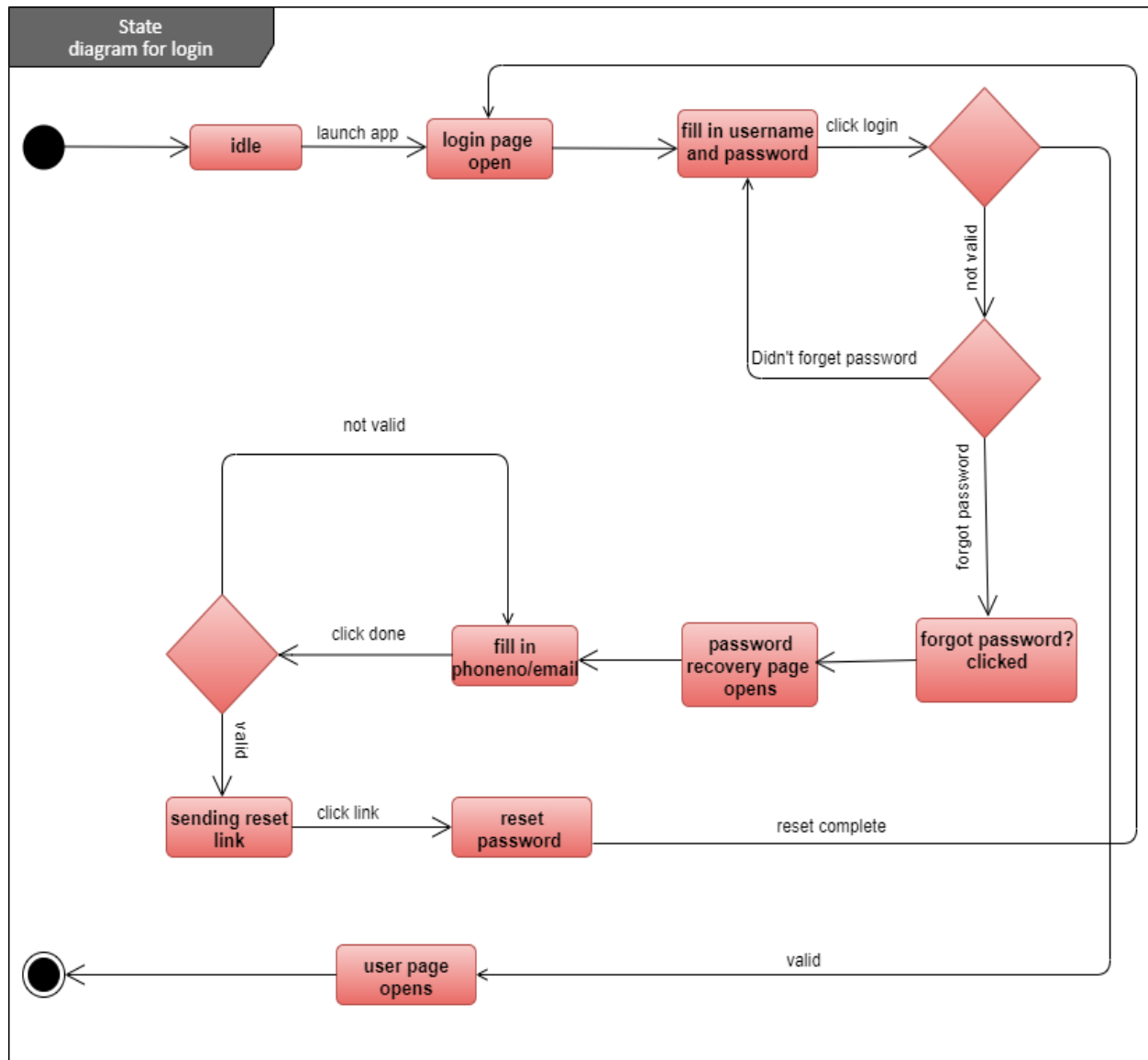
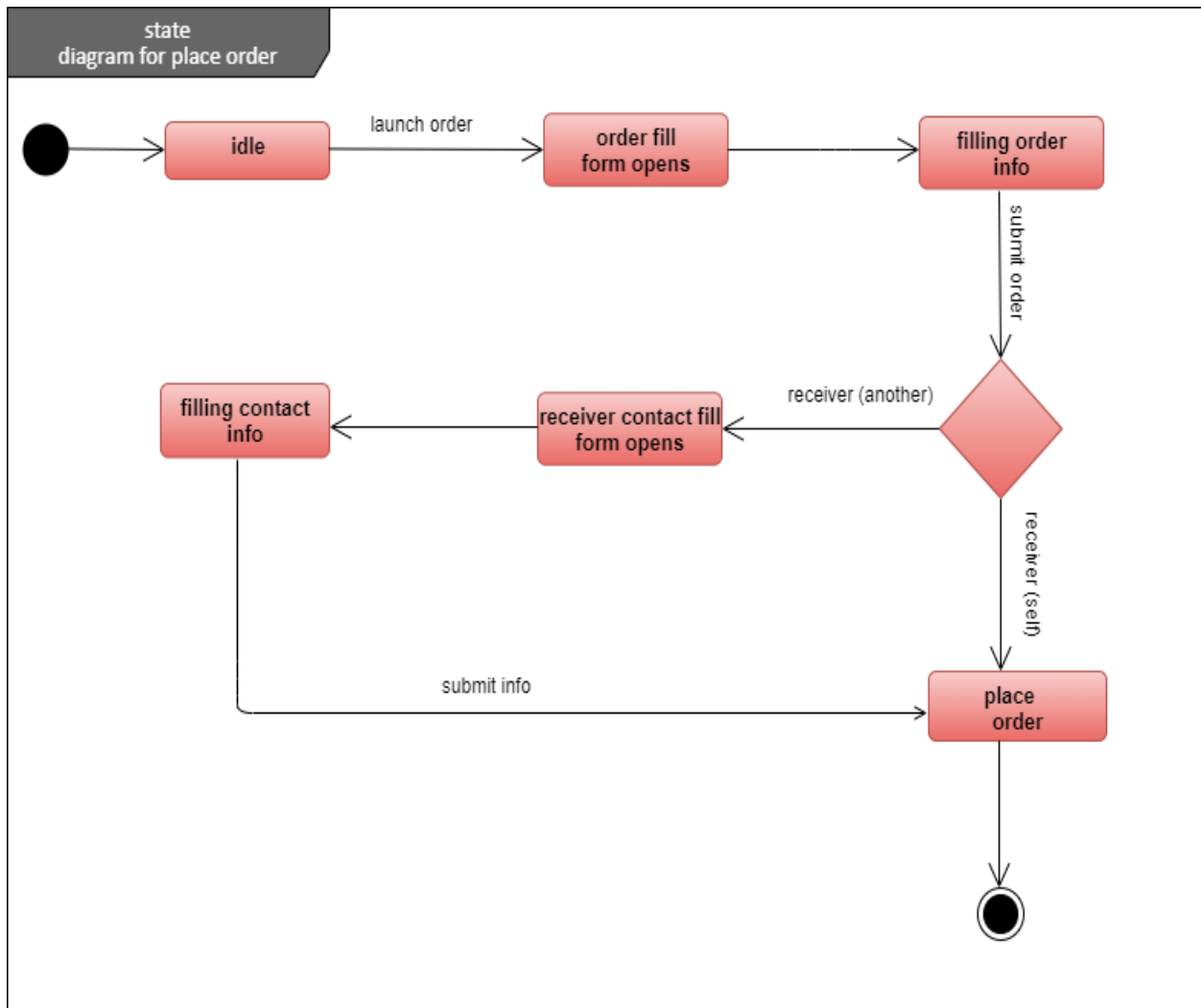
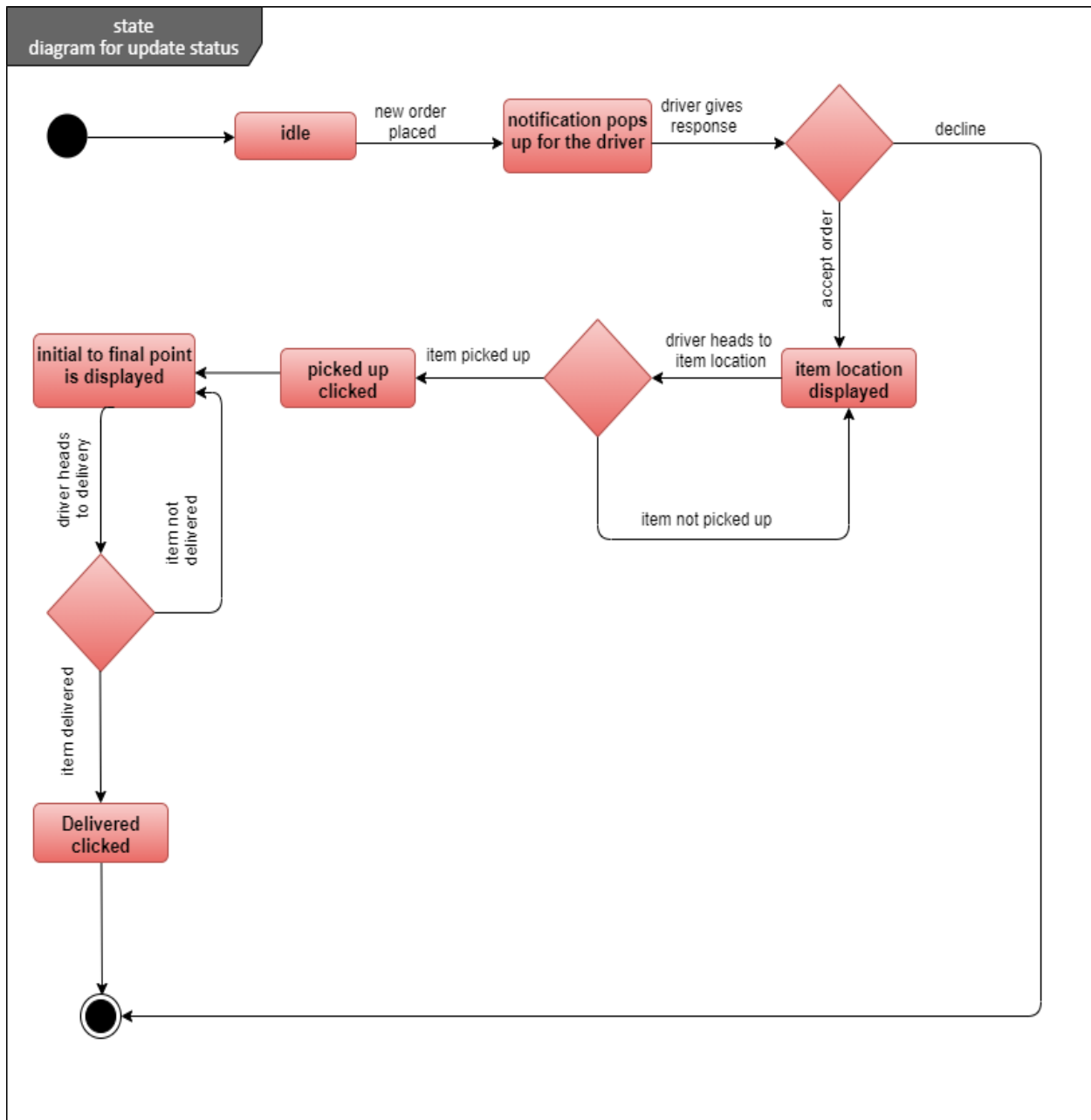


FIGURE 3. 13: STATE DIAGRAM FOR LOGIN

**FIGURE 3. 14: STATE DIAGRAM FOR PLACE ORDER**

**FIGURE 3. 15: STATE DIAGRAM FOR UPDATE STATUS**

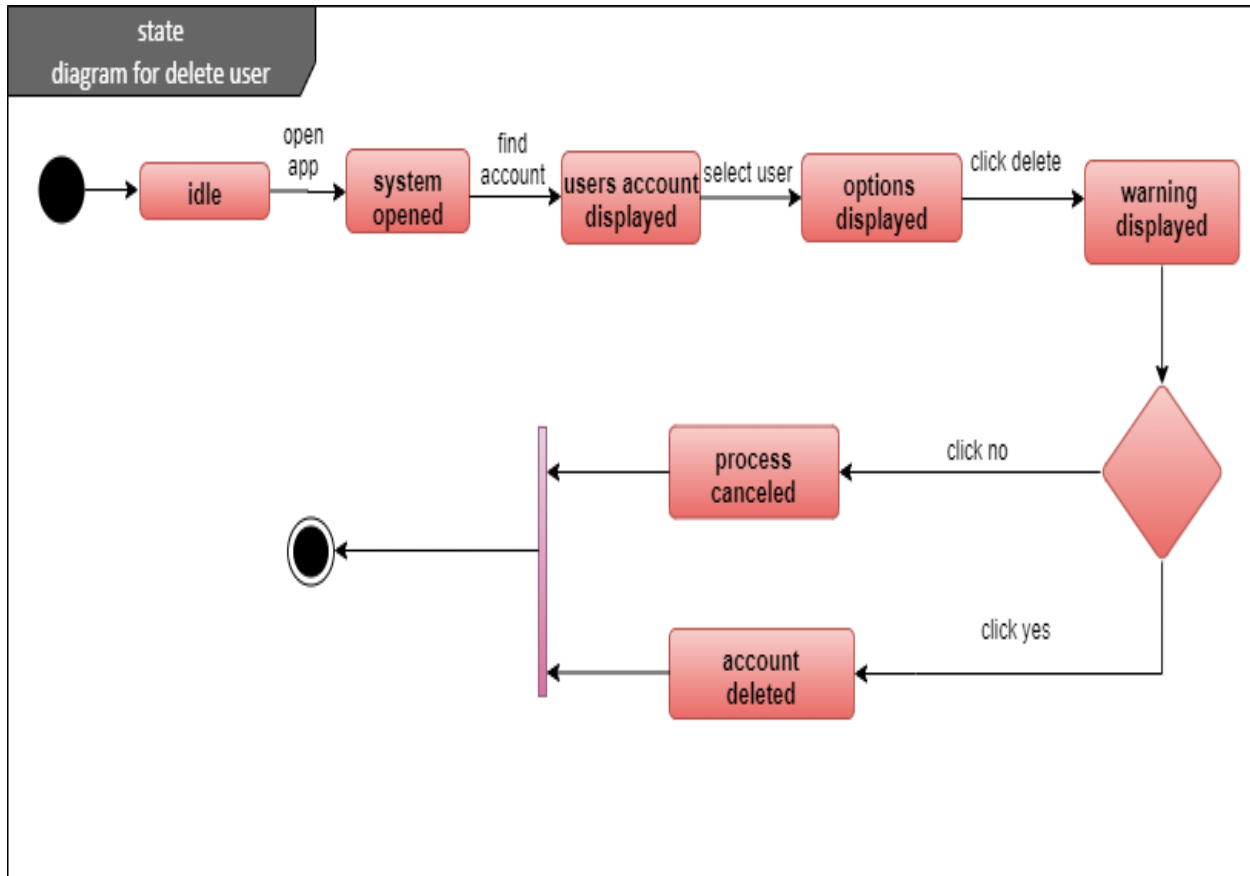


FIGURE 3. 16: STATE DIAGRAM FOR DELETE USER

3.4 Class-Based Modeling

Class-based modeling is a way of making a view of a system structure using classes, attributes, operations/functions, and the interaction between them [4]. In this manner, this section of the document shows system development resources by helping identify the basic attributes and methods in the system.

3.4.1 Identifying Classes

In this section, we set a target to show classes that are basic structural entities of the system. In the proposed system there will be certain classes as well which are implied as shown below:

1. Admin class
2. Customer class
3. Driver class
4. System User class
5. Payment class
6. Location class
7. Order class
8. History class
9. Feedback class

The tables below clarify more about the classes in the proposed system by showing the attributes in the corresponding classes. Attributes are the basic characters of classes on which the operations are applied.

TABLE 3. 26: TABLE FOR ADMIN CLASS

Attribute	Data Type	Required
Address	String	True
Gender	String	True

TABLE 3. 27: TABLE FOR CUSTOMER CLASS

Attribute	Data Type	Required
Receiver name	String	True
Receiver Last Name	String	True
Receiver Phone	String	True

TABLE 3. 28: TABLE FOR SYSTEM USER CLASS

Attribute	Data Type	Required
User id	String	True
User Name	String	True
User last name	String	True
Password	String	True
Phone	String	True
e-mail	String	Optional
Reset code	String	True

TABLE 3. 29: TABLE FOR DRIVER CLASS

Attribute	Data Type	Required
Vehicle type	String	True
Plate no	String	True
License renewed date	String	True
Address	String	True
Photo	String	True
Gender	Char	True

TABLE 3. 30: TABLE FOR ORDER CLASS

Attribute	Data Type	Required
ID	String	True
Distance	String	True
Category	String	True

Cost	String	True
Date	Date	True
Status	String	True
Time	Datetime	True

TABLE 3. 31: TABLE FOR PAYMENT CLASS

Attribute	Data Type	Required
Cost per KM	Float	True
Cost per category	Float	True
Cut percent	Float	True

TABLE 3. 32: TABLE FOR LOCATION CLASS

Attribute	Data Type	Required
Pickup location	String	True
Drop location	String	True

TABLE 3. 33: TABLE FOR HISTORY CLASS

Attribute	Data Type	Required
Date	Date	True
Time	Datetime	True

TABLE 3. 34: TABLE FOR FEEDBACK CLASS

Attribute	Data Type	Required
Response	String	True

Report	String	True
--------	--------	------

3.4.2 Class Diagram

As Class diagrams are the basic structural showcase of a system, the class diagram shown below gives the overall glimpse of the proposed system's structure.

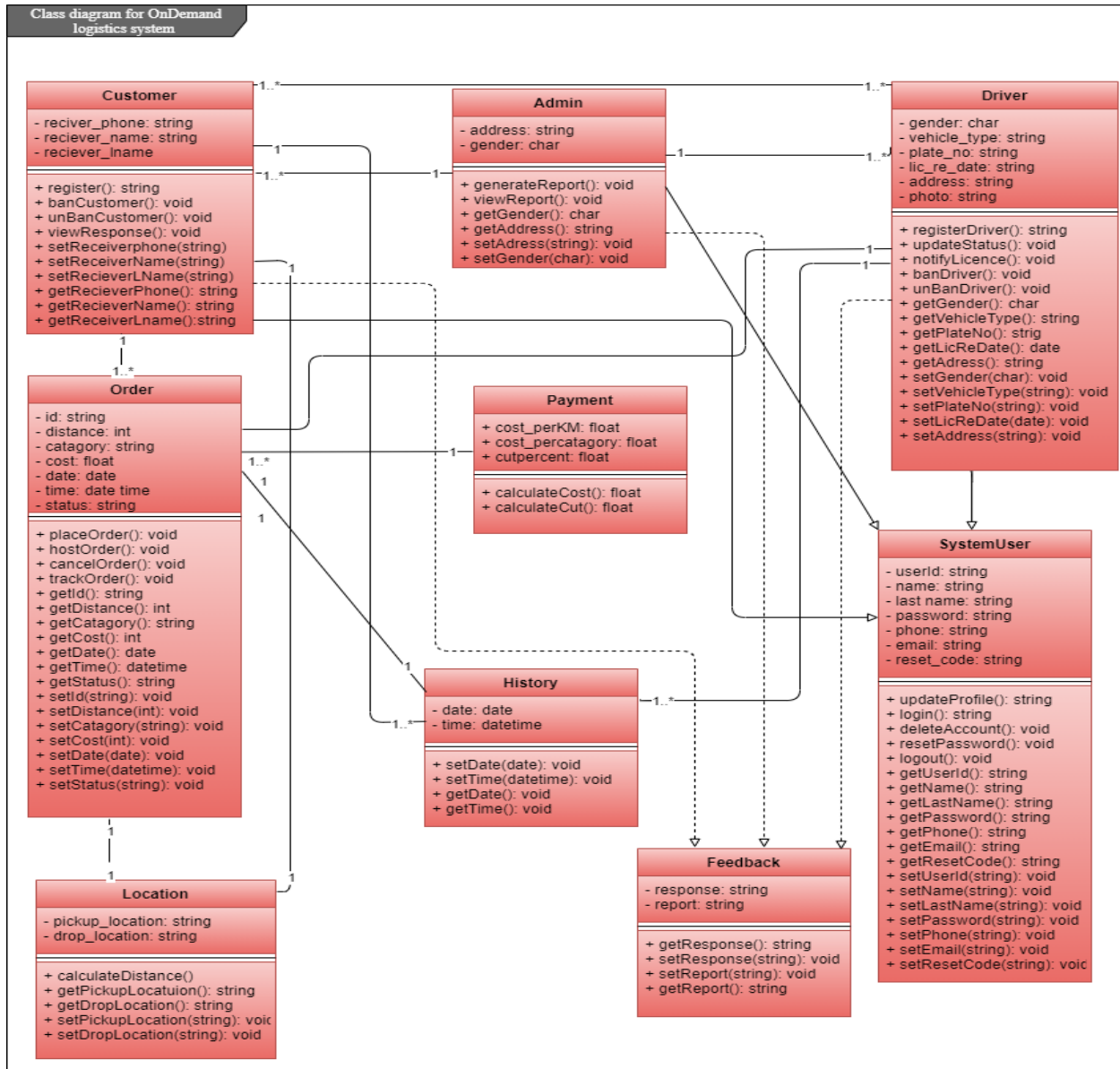


FIGURE 3. 17: CLASS DIAGRAM FOR ONDEMAND LOGISTICS SYSTEM

CHAPTER 4: SYSTEM DESIGN

4.1 OVERVIEW

This chapter mainly focuses on the design part of the on-demand logistics system. Since system design is the first part to get into the solution domain in software development, this part will focus on transforming the analysis model into the design model that takes into account the non-functional requirements and constraints described in the problem statement and requirement specification sections discussed earlier this design model and shows the direction of how our system is built and to provide sufficient detailed data and information about the system and its system elements to enable the implementation consistent with architectural entities.

4.2 SYSTEM DESIGN

System design is the process of designing the elements of a system such as the architecture, modules, and components, the different interfaces of those components, and the data that goes through that system. Thus, the system design helps us to understand some of the design goals. The design goals of a system are related with the nonfunctional requirements of the system listed in chapter 2.

4.3 ARCHITECTURE OF THE SYSTEM

The software architecture of a system depicts the system's overall structure of the components the system, and provides an explanation of how they behave. The proposed system implemented based on layered based architecture which separates the system in to three; these are

- 1. Presentation Layer** This is the top most level of the application. It displays information such as services the user interacts with. It communicates with the process layer by

which it outputs the results to be displayed through User interfaces. In simple terms, it is a layer that users can directly access.

2. Logic Layer Concerned with application-specific functionality by performing detailed processing.

3. Data Layer The data layer includes the data persistence mechanisms like database servers and the data access layer that encapsulates the persistence mechanisms and exposes the data through application program interface that helps to manage the stored data. The proposed system uses Firebase as its database.

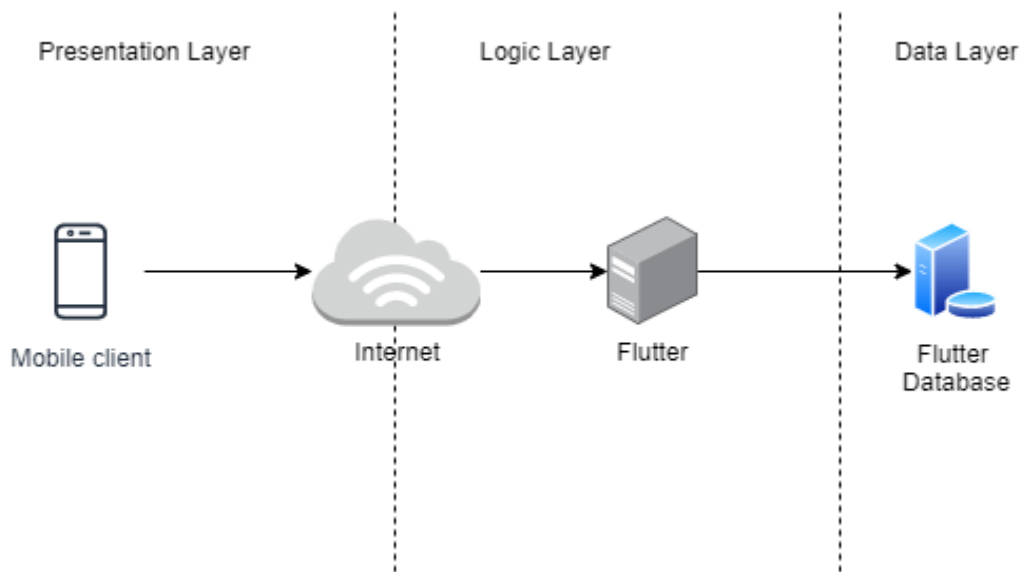


FIGURE 4. 1 : SYSTEM ARCHITECTURE

4.4 SYSTEM DECOMPOSITION

In this section, the system will be decomposed into cohesive, well-defined subsystems that are. set of related operations that share common purpose and also reduce the complexities of the system.

The following are the major subsystems (system decompositions) identified in our proposed subsystem:

1. Account management subsystem

This subsystem is concerned with allowing the system users to create, update, view, and delete different accounts based on their access privilege and access levels.

2. Request management subsystems

This subsystem includes basic functions of the system like requests sent by Users, accept/reject request received by the driver This sub system generally includes

- Request
- Response for request

3. History management subsystem

This subsystem contains records/logs of the performed activities in the system like user's history and driver's history which is only accessible by themselves and system admin. The general history about the activity of the system includes login trial records, new user registration records, request records, etc.

4. Report management subsystem

This subsystem is concerned with the creation and management of report related activities like generation of report, printing of report, posting/sharing reports and view those reports by using different tables and charts for visualization. The reports of the system use other subsystems to get data for the different reports.

5. Database connection subsystem

This subsystem contains ways to store and retrieve data that is generated by the system by different activities. The other sub systems are connected with the database to coordinate and operate to perform activities that are required of the system.

4.5 COMPONENT DIAGRAM

Component diagrams are used to visualize the organization of system components and the dependency relationships between them [4]. They provide a high-level view of the components within a system. In our system, we have used different components to understand the system and how its components are connected as shown in

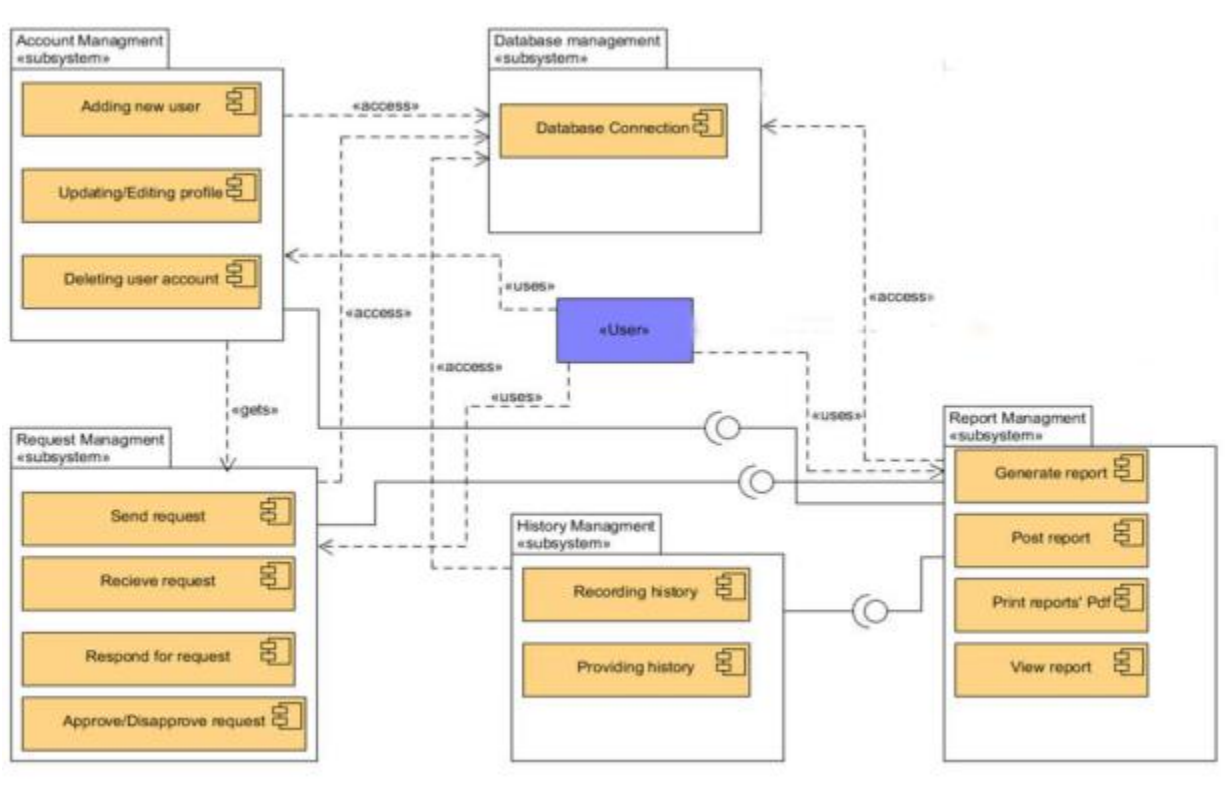


FIGURE 4.2: COMPONENT DIAGRAM

4.6 DEPLOYMENT DIAGRAM

The hardware Software mapping (deployment diagram) is described to show the various hardware devices and equipment's used in the system and its interaction with the software components' [4] it is one of the UML diagrams types that shows the execution architecture of a system, including nodes, such as, hardware or software execution environments, and the middleware connecting them. We plan on deploying our system as shown in the below diagram

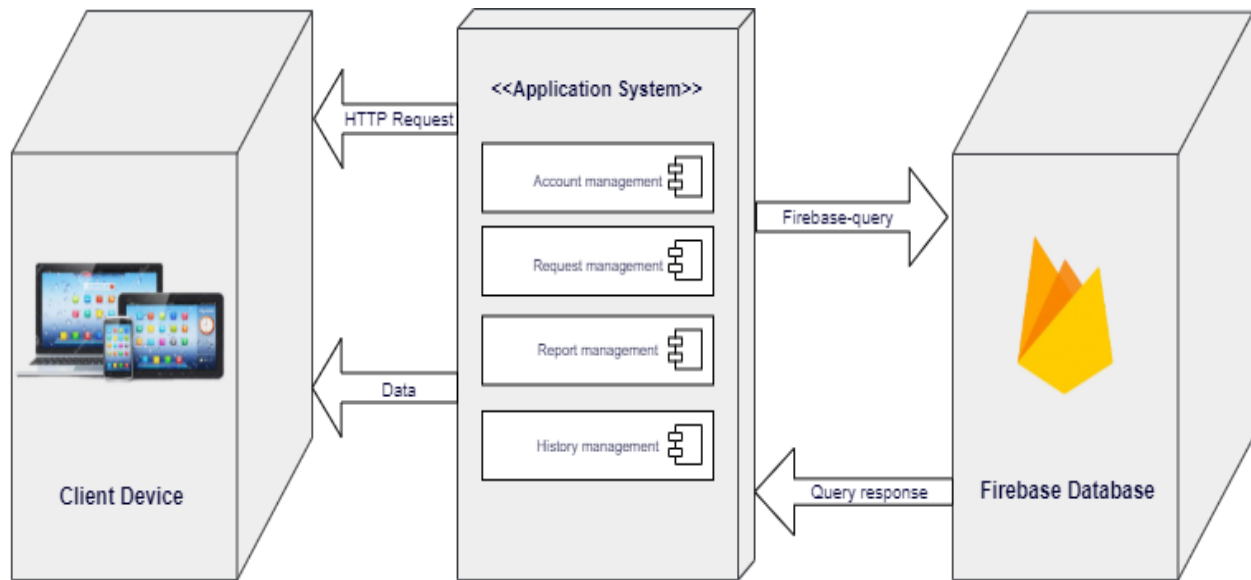


FIGURE 4. 3: DEPLOYMENT DIAGRAM

4.7 DATABASE DESIGN

Database design is the organization of data according to a database model collection of processes that facilitate the designing, development, implementation and maintenance of data management systems in order to produce database systems that meet the requirements of users. The main objective of database designing is to produce logical and physical design models of the proposed database system.

4.7.1 ENTITY RELATIONSHIP DIAGRAM

ER diagram displays the entity sets stored in a database and their relationships. The ER Diagram for the proposed system is used to indicate the relationship between the database collections in the system.

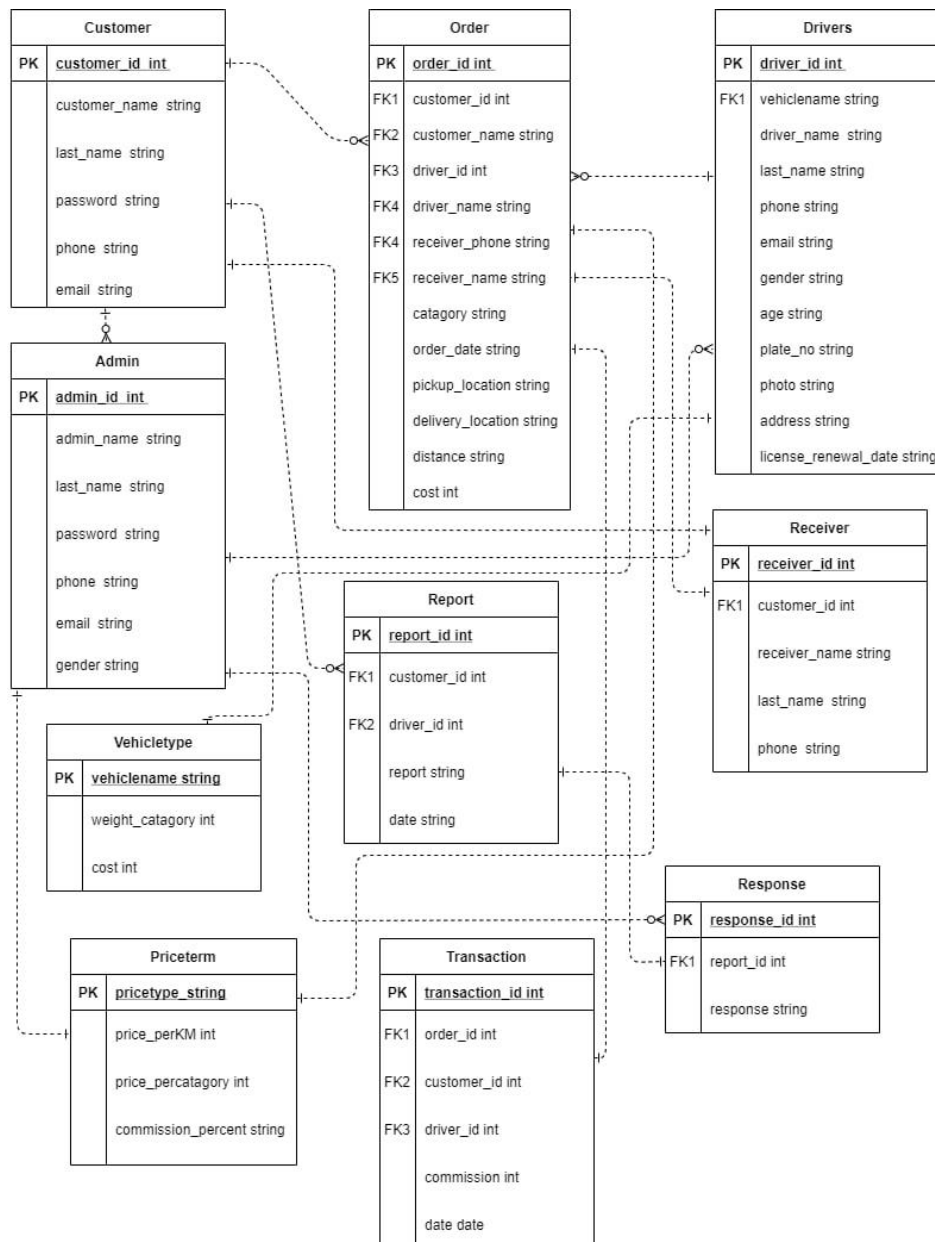


FIGURE 4. 4: ER DIAGRAM

4.5 USER INTERFACE DESIGN

User interface design refers to the visual layout of the elements that a user might interact with and this section of the document describes the user interface of the proposed system as shown below:

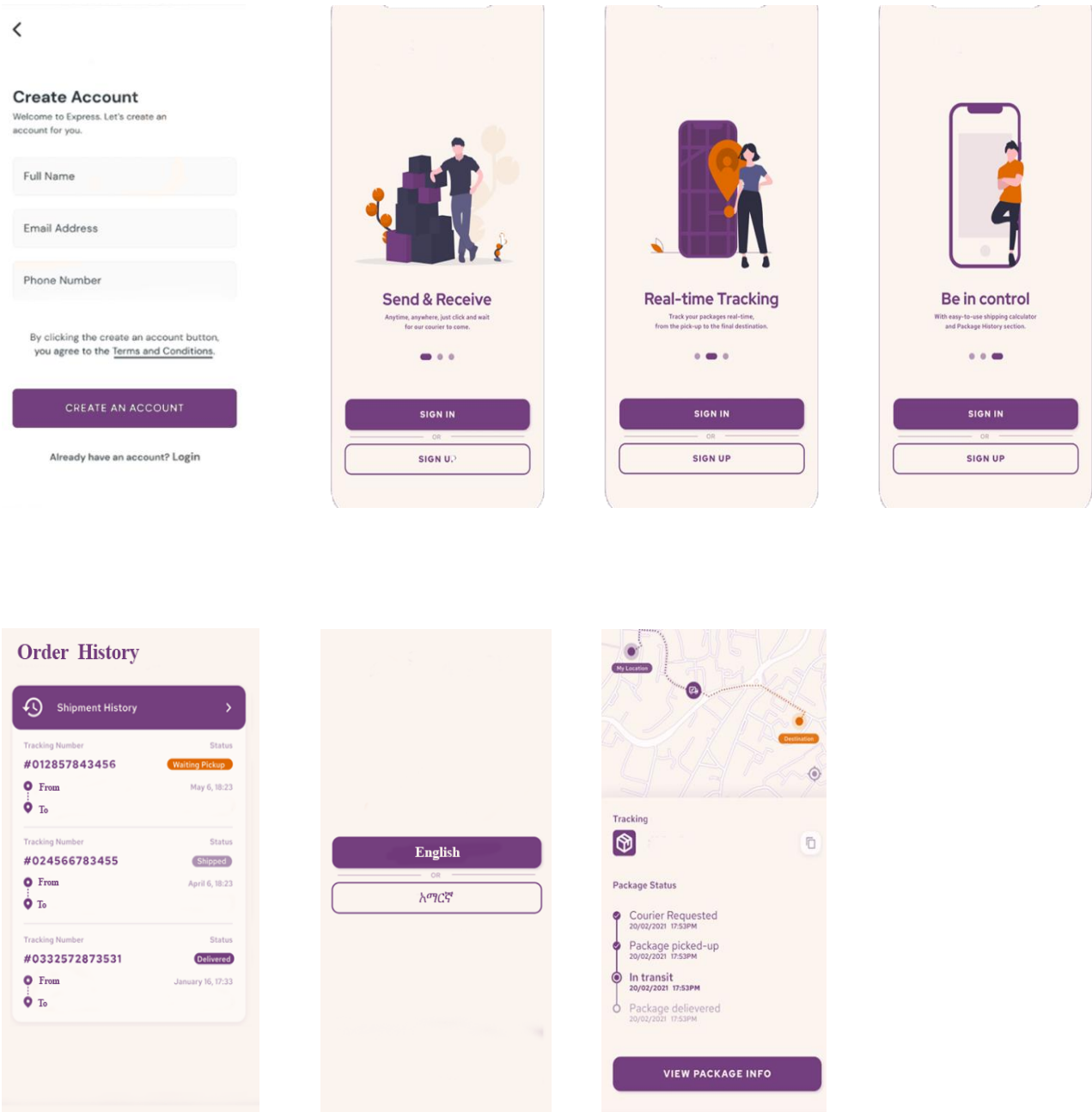


FIGURE 4. 5: UI DESIGN

References

- [1] <https://expresswaycourier.com/history-courier-services/>
- [2] <https://www.ethiopostal.com/index.php/history>
- [3] <https://en.wikipedia.org/wiki/Courier>
- [4] <https://www.visual-paradigm.com/guide/>
- [5] <https://creatly.com/blog/diagrams/uml-diagram-types-examples/>
- [6] https://www.tutorialspoint.com/software_engineering/software_requirements.htm
- [7] <https://blog.prototypr.io/software-documentation-types-and-best-practices-1726ca595c7f>
- [8] <https://www.curagroup.com/blog/the-evolution-of-the-courier-service-industry>