E-GROCERY

Developed By

Syed Sameed Hasan

3980-FBAS/BSCS4/F18

M. Sarosh Bin Imran

3981-FBAS/BSCS4/F18

BS Software Engineering

Supervisor

Mr. Muhammad Imran Saeed

Department of Computer Science & Software Engineering
Faculty of Basic and Applied Sciences
International Islamic University,
Islamabad

2021

Department of Computer Science and Software Engineering

International Islamic University, Islamabad.

Final Approval

Dated:

It is certified that we have read the project report submitted by [Student name, Registration no] and it is our judgment that this project is of sufficient standard to warrant its acceptance by the International Islamic University, Islamabad for the BS Degree in Computer Science/Software Engineering.

Committee:

External Examiner

[External Examiner Name]
[Designation] [Signature, Date]
[Department, Organization]

Internal Examiner

[Internal Examiner Name]
[Designation] [Signature, Date]
[Department]

Supervisor

[Supervisor Name]
[Designation] [Signature, Date]
[Department]

Dissertation

A dissertation submitted to the Department of Computer Science & Software Engineering, International Islamic University, Islamabad, as a partial fulfillment of the requirements for the award of the degree of Bachelor's in Computer Science

Dedication

I dedicate this project to my beloved parents, respected teachers and all those who prayed for our success.

Declaration

I hereby declare that this Software, neither as a whole nor as a part thereof has been copied out from any source. It is further declared that I have developed this Software entirely on the basis of my personal efforts made under the sincere guidance of my teachers and supervisor.

No portion of the work presented in this report has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

Syed Sameed Hasan
(3980-FBAS/BSCS4/F18)

M. Sarosh Bin Imran
(3981-FBAS/BSCS4/F18)

Acknowledgement

All praise to Almighty Allah, who gave me the understanding, courage and patience to complete this project.

Thanks to my parents and all well wishers, who helped me during my most difficult times and it is due to their untiring efforts that I am at this position today.

I express my gratitude to my kind Teacher **Mr. Muhammad Imran Saeed** for providing me the opportunity to learn and enhance my knowledge. She had been ready to help and guide me throughout the project.

Project in Brief

Project Title:	E-Grocery System
Objective:	The Objective of this project is to make grocery shopping easy and effortless.
Undertaken By:	Syed Sameed Hasan M. Sarosh Bin Imran
Supervised By:	Mr. Muhammad Imran Saeed Assistant Professor
Date Started:	*September, 2021*
Date Completed:	*August, 2022*
Tools Used:	*Microsoft Office 365 Microsoft SQL Server 2005 Microsoft Visual Studio 2008 Adobe Photo Shop 7.0 DHTML Menu 9.0 Macromedia Flash Player*
System Used:	*Core I5 3rd Generation, Processor 2.56 GHz,8GB RAM, Windows 10 Pro*

Abstract

Due to the expanding need of online shopping and considering the progressions in our lives after this pandemic , we have chosen to create a web-based application for online shopping for food. In our application we will be taking grocery stores of a specific region. Our web-based application will have all the staple things present in the genuine stores. The reason for this application is to mechanize and work with the entire course of shopping for food. Our framework will benefit both the retailer and customers in an interesting manner.

For retailers our framework will be a stage which can fill in as a virtual store where they can sell every one of their items on the web. Selling on our framework will reduce off their all-out expense cost by limiting the use of assets and duplicate their income.

Our project limits physical as well as mental exertion of customers and save their valuable time.

Table of Contents

Chapter 1: Introduction........1
1.1 Introduction........1
1.2 Project Motivation........2
1.3 Project Scope........2
1.4 Existing System........2
1.5 Objectives........3
Chapter 2: Basic Concepts/Existing System........4
2.1. Assumptions & Constraints........4
2.1.1. Assumptions........4
2.1.2. Constraints........4
2.2. Existing System........4
2.3. Drawbacks in Existing System........4
2.4. Proposed Solution........5
2.5. Existing Systems........5
2.5.1. Deliver It........5
2.5.2. Sabzi Mandi Online........5
2.5.3. Fowrry........6
Chapter 3: Problem/System Analysis........7
3.1 Problem Statement........7
3.2 Features........7
3.3 Requirements........8
3.3.1 Functional........8
3.3.2 Non-functional........9
3.4 Stakeholders........9
3.5 Use cases........10
3.5.1 Use case (Admin)........10
3.5.2 Use case (Mart Manager)........11
3.5.3 Use case (Rider)........12
3.5.4 Use case (Customer)........13
3.6 Fully Dressed Use cases........14
3.6.1 Use case: Sign up........14
3.6.2 Use case: Manage User’s Sign ups........15
3.6.3 Use case: Log In/Out........16
3.6.4 Use case: Manage Rider........17
3.6.5 Use case: Manage Mart Manager........18
3.6.6 Use case: Manage Customer........19
3.6.7 Use case: Manage Profile........20
3.6.8 Use case: Manage Product........21
3.6.9 Use case: Manage Order........22
3.6.10 Use case: Manage Mart........23
3.6.11 Use case: Manage Cart........24
3.6.12 Use case: Search Product........25

3.6.13 Use case: Place Order 26
3.6.14 Use case: Track Order 27
3.6.15 Use case: Picks Order 28
3.6.16 Use case: Track Path 29
3.6.17 Use case: Deliver Order 30
3.7. Domain Model 31
3.7.1. Use case Diagram 31
3.8. System Sequence Diagrams 32
3.8.1. Sign Up 32
3.8.2. Log In 32
3.8.3. Log Out 33
3.8.4. Add Rider 33
3.8.5. Update Rider 34
3.8.6. Delete Rider 34
3.8.7. Manage Profile 35
3.8.8. Add Manager 35
3.8.9. Update Mart Manager 36
3.8.10. Delete Mart Manager 36
3.8.11. Add Product 37
3.8.12. Update Product 37
3.8.13. Delete Product 38
3.8.14. Manage Order 38
3.8.15. Add Mart 39
3.8.16. Update Mart 39
3.8.17. Add to Cart 40
3.8.18. Remove from Cart 40
3.8.19. Search Product 41
3.8.20. Place Order 41
3.8.21. Track Path 42
3.8.22. Deliver Order 42
3.8.23. Pick Order 43
3.9. Domain (Conceptual) Model 44
Chapter 4: System Design 45
4.1 Design Model 45
4.1.1 Interaction Diagram 1
4.1.2 Class Diagram 1
4.2 Activity Diagram 1
4.2.1 Admin activity diagram ....
4.2.2 Mart manager activity diagram ....
4.2.3 Rider activity diagram ....
4.2.4 Customer activity diagram ....
4.3 Data Model 1
4.3.1 ERD 1
Chapter 5: Implementation 4
Chapter 6: Testing 4

Chapter 1
Introduction

1.1. Introduction

Web programming, also known as web development, is the creation of dynamic web applications. Web development is the building and maintenance of websites; it's the work that happens behind the scenes to make a website look great, work fast and perform well with a seamless user experience. In our modern times Web has become an important aspect of our lives and day by day there is advancement in technology and things have moved to web and is easily accessible to the users.

With the increasing popularity of online shopping and considering the changes in our lives after this pandemic situation, we have decided to develop a web-based application for online grocery shopping. In our application we will be taking onboard major marts of a particular area. Our system will have all the grocery items present in the actual marts. The purpose of this application is to automate and facilitate the whole process of grocery shopping. Our system will benefit both the retailer and consumers in a unique way.
For retailers our system will be a platform which can serve as a virtual mart where they can sell all their products online. Selling on our system will cut off their total cost price by minimizing the usage of resources and multiply their earnings.

Our system minimizes physical as well as mental effort of shoppers and save their precious time. It will facilitate the working mothers since they can do all of their shopping easily at any place. The main attribute of online grocery system is that customers can purchase the grocery with just one click of mouse button and the items ordered get delivered at home. This system will provide convenience for the customers as they will not have to physically visit a grocery store and search for the grocery items and their availability. An online shopping aims at making shopping comfortable and enjoyable process.

1.2. Project Motivation

Online shopping has gained significant importance during the outbreak of Covid-19 to limit physical interactions. So, our aim is to facilitate the customers to limit their physical interactions in this pandemic situation and to digitalize the process of grocery shopping. The main purpose of this project is to make grocery shopping easy in this pandemic situation and to save the time and effort required for grocery shopping. This project will provide a platform for customers to buy their desired grocery items without going to markets and searching for them in multiple shops. This project will also help marts to increase their sale by selling their products online without launching their own website.

1.3. Project Scope

- To provide an online platform for grocery shopping.
- To make the whole grocery shopping process less time consuming and easy.
- To minimize the resources of buyers they spend to visit a market.
- To provide platform for renders to sell online.
- To facilitate renders to maximize their sales while using minimum resources.
- To provide part-time job opportunities for students

1.4. Existing System

There are few similar applications already present in the market. Some of which includes Deliver It, Sabzi Mandi Online, and Fowrry etc. Even though these applications are similar to our application, but they all lack some of the features which our application will provide. i.e., Tracking, add/edit profile etc. None of these applications directly involve marts to sell their products online. So, our application will provide the platform for marts to directly sell their products online. These applications do not provide all the grocery items on one platform.

1.5. Objectives

The main objective of this web-based application is to bring all the useful household grocery items online at one platform.

- To make a grocery shopping process easy and less time-consuming.
- To provide convenience for the people by bringing marts in their mobile phones.
- To provide a platform for renders to multiply their sales while using fewer resources than an actual mart or a shop.
- To provide part-time job opportunities for students, as a rider.

Chapter 2
Basic Concepts/Existing System

2.1 Assumptions & Constraints

2.1.1 Assumptions

- Defined project scope will be available.
- Required technology, tools and equipment will be available.
- Project members will be available.
- Performance of project members will be accurate.
- Skills of project members will enhance during the development of this project.

2.1.2 Constraints

- Project advancement must finish in given time periods and cutoff times.
- Project advancement should totally follow characterized venture scope.
- Domain necessities must be tended to totally.

2.2 Existing System

There are few similar applications already present in the market. Some of which includes Deliver It, Sabzi Mandi Online, and Fowrry etc. Even though these applications are similar to our application, but they all lack some of the features which our application will provide. i.e., Tracking, add/edit profile etc. None of these applications directly involve marts to sell their products online. So, our application will provide the platform for marts to directly sell their products online. These applications do not provide all the grocery items on one platform and no proper tracking feature.

2.3 Drawbacks in Existing Systems

Even though these applications are similar to our application but there are few drawbacks in these applications like no tracking of orders, slow delivery process, no stock information, limited delivery areas, and no guaranteed availability of the desired product(s).The major drawback of these application is that they do not provide all the grocery items on single platform. Quality of the products are not assured because they are not brought from good marts. So, their users are not satisfied with the provided services. Another major drawback of these applications is no direct involvement of mart(s).So, there is no guarantee if the ordered product(s) is available or not and will be delivered or not. Some of the applications even postpones the delivery to next day.

2.4 Proposed Solution

To over the problems/drawbacks mentioned above, we have come up with a solution to facilitate the customers as well as retailers. We have proposed an "E-Grocery" System which will overcome all the drawbacks that we have mentioned above and will provide a user-friendly interface and many other features. The E-Grocery system will provide the order tracking feature, fast delivery, available stock information, delivery in all areas and guaranteed availability of the desired product(s). The E-Grocery System provides a platform where customers can find all the desired grocery items at one place. There are categories of products to make it easy for the customers to find their desired items. The E-Grocery System also provides the facility to check the quality of the products first and then pay for them if they are satisfied with the quality otherwise, they can return the product(s) if they are not satisfied with the quality of the products without paying for them. The E-Grocery system directly involves the Marts to assure the quality of the products and to provide a platform for Marts to sell their products online.

2.5 Existing Systems

Current Existing System are as follow:

2.5.1 Deliver It

Deliver It is an online delivery application which provides home delivery of food & grocery items including fruits and vegetables at the doorstep of customers in Islamabad Only. It provides all the grocery items at one place but there are still few problems in this application like no order tracking, order can take hours to be delivered, no order confirmation and does not provide quality products according to their customer’s reviews. There is also no direct involvement of Marts to provide them a platform where they can sell their products. There is no option to add/edit information in Deliver It Application.

2.5.2 Sabzi Mandi Online

Sabzi Mandi Online is also an online application which provides free delivery of vegetables and fruits in Islamabad, Rawalpindi and Abbottabad at Government approved rates. The advantage of this application is that is provides free delivery and prices are DC approved. The major drawback of this application is that it only provides fruits and vegetables and there is no order tracking feature. Even it postpones the delivery to next day if the order is placed after evening. It does not notify the customers about the availability of the items. Quality of the product(s) is not assured and there is no confirmed delivery time. Another drawback is that the minimum order amount is Rs.1000.

2.5.3 Fowrry

Fowrry is also an online grocery delivery application which provides all the grocery items at one place and provides the category of products as well. It also provides the facility to purchase the desired product from the store of your own choice. This is the best application as compared to Sabzi Mandi Online and Deliver It. The drawback is that even this application does not provide order tracking feature and it only delivers in Islamabad. Another drawback according to customer’s review is that the product prices are not updated daily thus the issue of difference in bills is reported.

Chapter 3
Problem/System Analysis

3.1. Problem Statement

The problem of	This type of website does not exist in our country in which all the features are combine to a single platform and if something exist it does not combine under an umbrella.
Affects	Buyers have to visit grocery stores physically even if they have to buy a single product, which surely consumes time and effort. Meanwhile renders also have to use maximum resources which minimize their actual profits
The Impact of which is	Buyers have to take out free time from their schedules to visit markets and find their desired product(s). On the other hands renders also have to pay for the excessive resources which can be minimized by providing them with a platform like this.
A successful solution will be	To provide a platform for both renders and buyers. Where multiple marts can enroll and do their sales online. By opting this approach both renders, and buyers will be benefited.

3.2. Features

Our E-grocery web applications provide many features which helps people in terms of ease and reliability, each feature is discussed below:

- Our web application connects people with different marts and renders online.
- User can visit store online and cross check product prices in different stores and buy the reliable one.
- User can add different items from mart just by a click.
- User can place order and report their order in case of any mishap.
- Mart owners or renders can register their mart online, list their products and make sales online.
- So, mart managers will be having a virtual mart on our web application.

3.3. Requirements

Following are the Functional and Non-Functional Requirements of E-Grocery System.

3.3.1. Functional

Functional requirements are user requirements. These requirements work with functions that are visible and vital to users that a system must satisfy the business objectives to fulfil the design. When the user is using our app, given below are the functions that can be performed by the user.

- **FR-1**

  A user can create an account that can login through their respective account.
- **FR-2**

  User can visit marts and see their listed products.
- **FR-3**

  User can also add products to their cart and place order accordingly.
- **FR-4**

  User can rate their order and report in case of any mishap.
- **FR-5**

  Admin can add, update and delete mart managers and riders to keep system concurrent.
- **FR-6**

  Mart owners can register their store, list their products and make their sales.
- **FR-7**

  Mart manager can add, update and delete products in the list.
- **FR-8**

  Mart manager will also accept the order for their respective store and can see all the orders entertained by that store.
- **FR-9**

  Rider can accept orders and can keep a check on their profiles (total order delivered, earnings)
- **FR-10**

  Users can manage their profile.

3.3.2. Non-functional

Non-functional requirements are the requirements that shows rule to judge the operation of a system, rather than the behavior of the system. Non-functional requirements of our system are given below.

- Capacity
- Reliability
- Security
- Accuracy

3.4. Stakeholders:

Admin:

Admin will be assigned to keep a system wide check. Admin can add, update and delete mart managers and riders. Admin will have a check and control over their data and their actions.

User:

User can sign up through their accounts so they can login through their respective accounts later. User can visit different marts online, can see their listed products and can also search for the desired product. They can also add desired product into their product and place their order accordingly. They can also keep a track of their rider and order details. In case of any mishap, they can also report their order and rate the store too.

Mart Manager:

Mart mangers will be maintaining their product list by adding, updating and deleting products according to their availability. Mart manager will also be accepting orders for their respective stores and can keep a check on all of their orders. Mart manager can also update their mart data.

Rider:

Riders can register their selves. Accept nearby orders and deliver them. Rider has to update the order status when picked and delivered. Rider can track the path to mart and customer addresses. Rider can also keep a check of their profile (delivered orders and account) and can also manage their accounts.

3.5. Use cases

Following are the Use Cases of E-Grocery System.

Sign-Up:

User will sign up to get registered and to be able to perform further actions. User will click sign up button on the main screen and will get signup screen where he can enter the required data and get registered after validation.

Manage User Sign-ups:

Mart manager will validate signups of users (rider, mart manager), so they can perform their actions. Mart manager will be notified when any user will sign-up and he will validate that sign up and accept that signup.

Log In:

User will login by the account they have already signed up with, to perform actions. User will go to the login page and enters the required data(email/password) so he can login with his/her account.

Manage Riders:

Admin will manage riders by adding or updating their data or deleting them. Admin will go the manage rider activity and search for rider he wants to manage data of. By this, the data in the system will be concurrent and consistent.

Manage Mart manager:

Admin will manage mart managers by adding or updating their data or deleting them. Admin will go the manage rider activity and search for rider he wants to manage data of. By this, the data in the system will be concurrent and consistent.

Manage Customer:

Admin can manage customer and will be having a limited access to user’s data. Admin can manage customer by going to manage customer activity search customer by his ID and he will be given the customers data.

Manage Profile:

Users can manage their profile by updating their data like (address, cell-no). User can go to manage profile activity and can update their data there, so their data will be up to date and consistent.

Manage Product:

Mart manager can manage products by adding deleting items or updating item data from the product list. Mart manager will go to the manage product activity, there he can select products he wants to perform action on from list. By this the product list will be up to dated and concurrent.

Manage Order:

Mart manager will be managing orders by accepting or rejecting orders on the basis of their final availability. Mart manager will be notified on receiving an order for his mart and he will accept or reject that order.

Manage Mart:

Mart manager can manage their marts where they can view and update mart data. Mart manager will go to the manage mart activity where he can update their mart's data and view orders and total earnings for their mart.

Manage Cart:

User can add and delete products to their cart so they can place their final order after they are done. User can add products in the cart while viewing items and pressing add to cart button for the item. User can click cart button and can drop the product they want.

Search Product:

User can search for their desired product. User will click search button and enters name of the product they are looking for and the product will be displayed if it is available in any store.

Place Order:

User(customer) will place order when they are finally done adding items in cart. User will go their cart and clicks place order button, after confirmation customer’s order will be placed and he will get his order and rider details.

Track path:

Rider will track the path of mart and customer’s address. After accepting an order rider will click track path button on which he will be showed with map to mart and then customers home.

Track Order:

Customer can track his order after rider has picked his order. When a rider will pick the order customer will get the rider detail where he can click track order button and can track his delivery.

Picks Order:

Rider will update order status to picked. After picking order from the mart rider will update order status to picked so customer can be notified and order status gets picked in the system.

Delivers Order:

Rider will update order status to delivered. After delivering order and receiving payment rider will update order status to delivered so every user related to that order will get notified and order status can be finalized as delivered.

3.5.1. Use case (Admin)

The Figure 3.5.1 shows the Use Case Diagram of Admin.

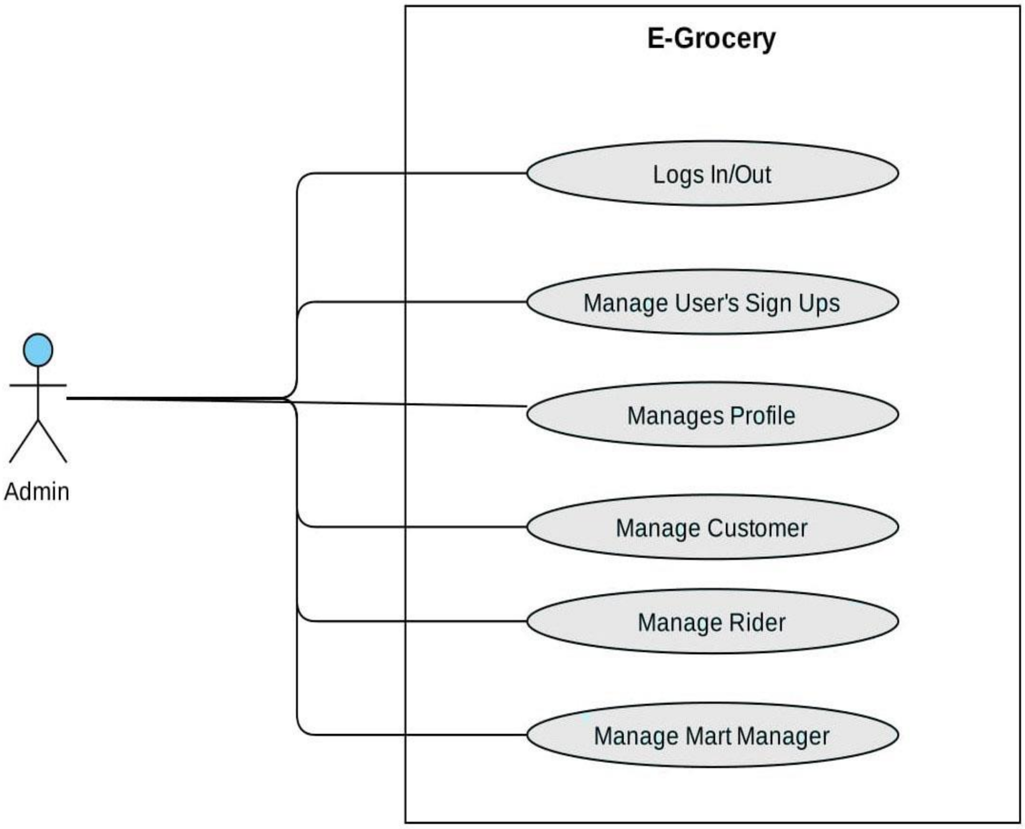

Figure 3.5.1

3.5.2. Use case (Mart Manager)

The Figure 3.5.2 shows the Use Case Diagram of Mart Manager.

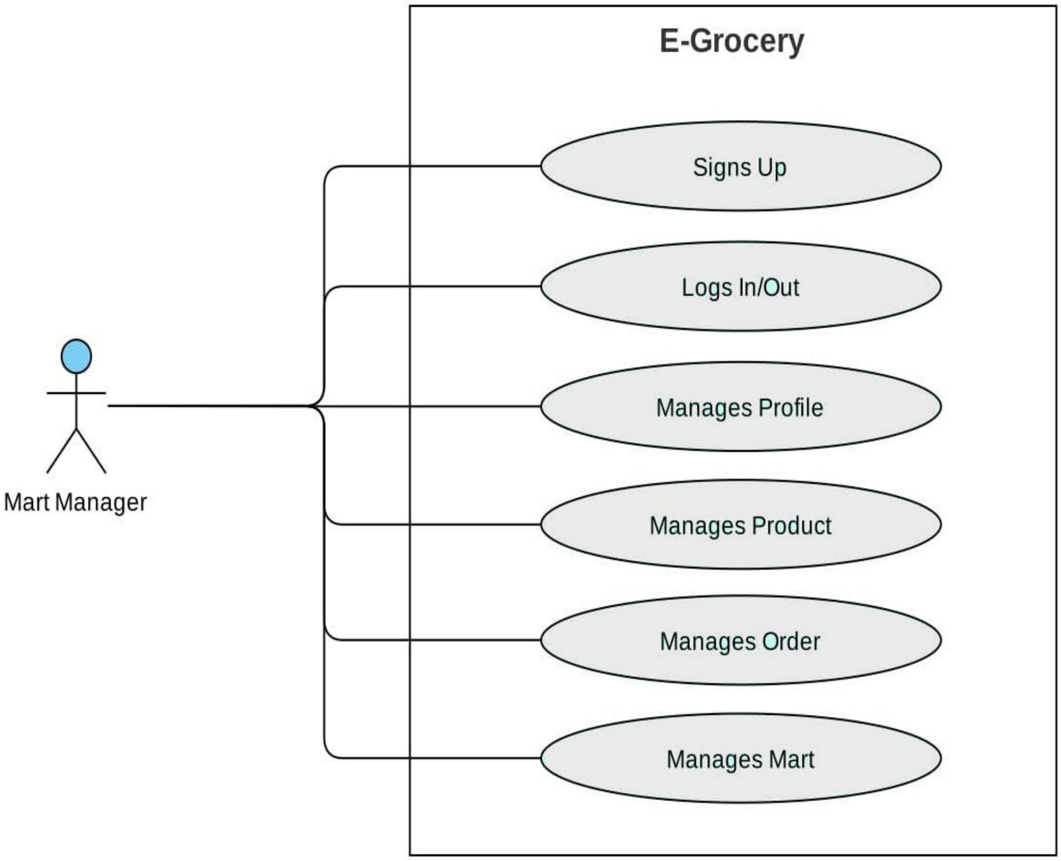

Figure 3.5.2

3.5.3. Use case (Rider)

The Figure 3.5.3 shows the Use Case Diagram of Rider.

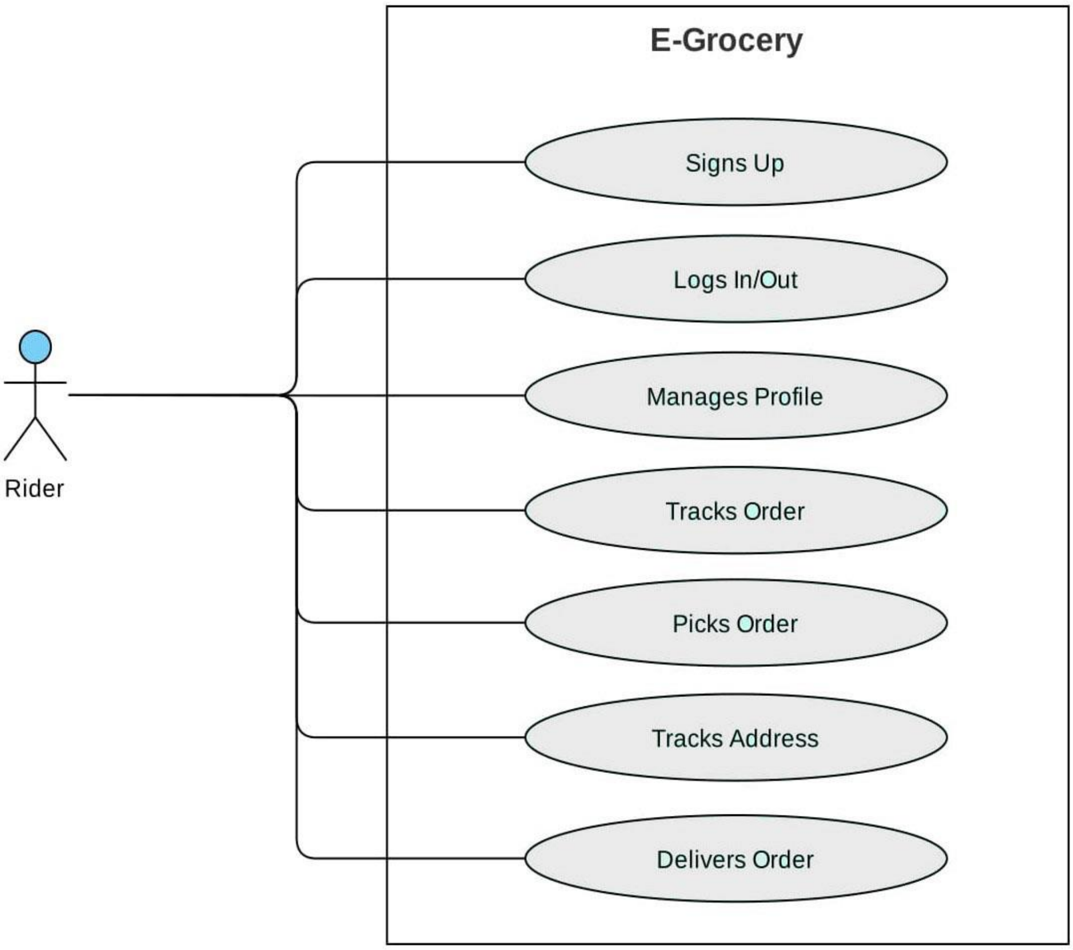

Figure 3.5.3

3.5.4. Use case (Customer)

The Figure 3.5.4 shows the Use Case Diagram of Customer.

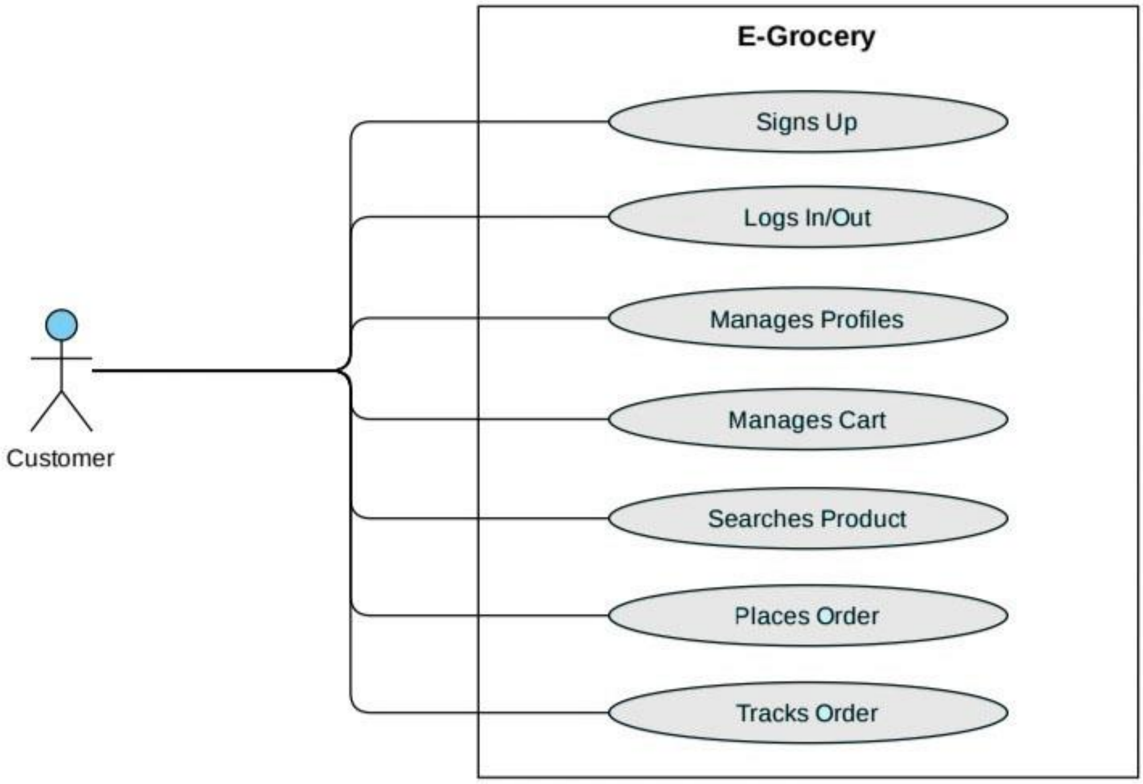

Figure 3.5.4

3.6. Fully Dressed Use cases

Following are the Fully Dressed Use Cases of E-Grocery System.

3.6.1. Use-case 0-1: Sign Up

Following are the details of Use Case Sign Up

USECASE ID	**UC-01**
Name:	Sign up
Actor:	Customer, Mart manager, Rider.
Description:	To create an account to proceed further.
Goal:	To create an account.
Pre-Condition:	User should have internet connection.
Post-condition:	Admin will approve the request for signup and user account will be created.
Main Flow:	User will open the website and goes to signup page, enter required data and proceed to signup.
Alternate-flow:	If user couldn't create account, then they cannot access other pages.
Error:	……….

3.6.2. Use-case 0-2: Manage User Sign Ups

Following are the details of Use Case Manage User Sign Ups

USECASE ID	**UC-02**
Name:	Manage User Signups
Actor:	Admin.
Description:	To manage Mart managers and Rider’s signups.
Goal:	To validate user signups,
Pre-Condition:	User must access signup page and enter data needed for signup.
Post-condition:	User’s account will be created and they can perform further actions.
Main Flow:	Admin will get the request for the signup by user and he will accept it after validating
Alternate-flow:	If admin doesn’t accept user signups, then user account will not be created and user cannot perform any action
Error:	……….

3.6.3 Use-case 0-3:Log In/Out

Following are the details of Use Case Log In/Out

USECASE ID	**UC-03**
Name:	Log in/out
Actor:	Admin, Customer, Mart manager, Rider.
Description:	To log in or out their respective account.
Goal:	To enter or exit their account.
Pre-Condition:	User must have registered account.
Post-condition:	User can perform further in bound actions with their accounts.
Main Flow:	User will go the login page if he has an account, enter the required data and proceed to login.
Alternate-flow:	If user couldn't login they must try again later or choose forget password option if they have forgot their password.
Error:	……….

3.6.4 Use-case 0-4: Manage Rider

Following are the details of Use Case Manage Rider

USECASE ID	**UC-04**
Name:	Manage rider
Actor:	Admin
Description:	Add, delete, update riders.
Goal:	To add, remove riders from the system or update their record.
Pre-Condition:	Rider must have registered before.
Post-condition:	Rider’s data will be consistent.
Main Flow:	Admin will have a panel to manage riders where he will be adding, deleting and keeping a check on riders.
Alternate-flow:	If the action couldn’t be performed for any reason, Admin have to return to the panel and try again.
Error:	………

3.6.5 Use-case 0-5: Manage Mart Manager

Following are the details of Use Case Manage Mart Manager

USECASE ID	**UC-05**
Name:	Manage mart manager
Actor:	Admin
Description:	Add, delete, update mart managers.
Goal:	To add, remove managers from the system or update their record.
Pre-Condition:	Mart manager must have registered before.
Post-condition:	Mart managers data will be consistent.
Main Flow:	Admin will have a panel to manage mart managers where he will be adding, deleting and keeping a check on riders.
Alternate-flow:	If the action couldn’t be performed for any reason, Admin have to return to the panel and try again.
Error:	……….

3.6.6 Use-case 0-6: Manage Customer

Following are the details of Use Case Manage Customer

USECASE ID	**UC-06**
Name:	Manage customer
Actor:	Admin
Description:	Admin can keep a check on customer data, so in case of false use of system, admin will take an appropriate action.
Goal:	To keep a check on customer activity.
Pre-Condition:	Admin must have registered account and has logged on to the website.
Post-condition:	Customer can shop on their behalf and incase of false use of system they will be blocked.
Main Flow:	Admin will have a panel to manage customers where he will be adding, deleting and keeping a check on customer details.
Alternate-flow:	If the action couldn’t be performed for any reason, Admin have to return to the panel and try again.
Error:	……….

3.6.7 Use-case 0-7: Manage Profile

Following are the details of Use Case Manage Profile

USECASE ID	**UC-07**
Name:	Manage Profile
Actor:	Admin, Customer, Mart manager, Rider.
Description:	User can update their data like (name, email, address) or log in or out their account.
Goal:	To manage account or profile.
Pre-Condition:	User must have registered account and has logged on to the website.
Post-condition:	User data will be updated in profile if changed.
Main Flow:	User will go to manage profile page and update their profile there.
Alternate-flow:	User must retry again.
Error:	………

3.6.8 Use-case 0-8: Manage Product

Following are the details of Use Case Manage Product

USECASE ID	**UC-08**
Name:	Manage Product
Actor:	Mart manager
Description:	Mart manager can add, delete and update product.
Goal:	To manage their product list.
Pre-Condition:	Mart manager and their mart must be registered before.
Post-condition:	Customer can see the available products and make purchasing.
Main Flow:	Manager will go to his panel and there he can add, delete or update product/product details there.
Alternate-flow:	If failed to perform required action, Manager must reload page and try again.
Error:	……….

3.6.9 Use-case 0-9: Manage Order

Following are the details of Use Case Manage Order

USECASE ID	**UC-9**
Name:	Manage Order
Actor:	Mart manager
Description:	Mart manager will accept or reject order on the basis of availability.
Goal:	To receive, accept or decline order.
Pre-Condition:	Their must be a received order.
Post-condition:	Order will be accepted or rejected and customer will be notified.
Main Flow:	Manager will go to his panel and there he can add, delete or update product/product details there.
Alternate-flow:	If failed, Mart manager must reload page and try again.
Error:	……….

3.6.10 Use-case 10: Manage Mart

Following are the details of Use Case Sign Up

USECASE ID	**UC-10**
Name:	Manage Mart
Actor:	Mart manager
Description:	Add, remove or update mart data like mart address, opening closing time.
Goal:	To keep marts data and status up to date.
Pre-Condition:	Mart manager must have registered account and has logged on to the website.
Post-condition:	Mart’s data and status will be up to date.
Main Flow:	Manager will go to his panel and there he can add, delete or update mart details there.
Alternate-flow:	If failed, Mart manager must reload page and try again.
Error:	………

3.6.11 Use-case 11: Manage Cart

Following are the details of Use Case Manage Cart

USECASE ID	**UC-11**
Name:	Manage Cart
Actor:	Customer
Description:	To add and delete products to the cart.
Goal:	To add or delete products before placing final order.
Pre-Condition:	Customer must have registered account and has logged on to the website.
Post-condition:	Items will be added to user’s cart.
Main Flow:	Customer logs in to the website, selects wanted items and clicks add to cart button. User will be brought to the cart page and displayed the items he has chosen.
Alternate-flow:	If failed, customer must reload and try again.
Error:	………

3.6.12 Use-case 12: Search Product

Following are the details of Use Case Search Product

USECASE ID	**UC-12**
Name:	Search Product
Actor:	Customer
Description:	Customer can search their wanted items so they can purchase it.
Goal:	To search a product
Pre-Condition:	Customer must have registered account and has logged on to the website.
Post-condition:	Customer will find the desired product and he can now add that to cart.
Main Flow:	Customer logs in to the website, on the main page by clicking search product they can search the product they wish for.
Alternate-flow:	If failed, Customer must reload page and try again.
Error:	………

3.6.13 Use-case 13: Place Order

Following are the details of Use Case Place Order

USECASE ID	**UC-13**
Name:	Place Order
Actor:	Customer
Description:	Customer can place order after they have added their desired products to the cart.
Goal:	To place final order.
Pre-Condition:	Their must e a product in a cart.
Post-condition:	Customer’s order will be placed. He will get order details and can track his order.
Main Flow:	After adding items to the cart, customer will click place order button to place his order.
Alternate-flow:	If failed, Customer must reload page and try again.
Error:	………

3.6.14 Use-case 14: Track Path

Following are the details of Use Case Track Path

USECASE ID	**UC-14**
Name:	Track path
Actor:	Rider
Description:	Rider can track path to the mart and to the customer’s address.
Goal:	To track the accurate location of mart and customer
Pre-Condition:	Rider must have registered account and has logged on to the website.
Post-condition:	Rider will be shown the map with exact address of mart and customer’s address.
Main Flow:	When the order arrives and rider gets assigned to the order, He can track path by clicking track path and he will be shown the address details map for both mart and customer.
Alternate-flow:	If failed, Rider must reload page and try again.
Error:	………

3.6.15 Use-case 16: Picks Order

Following are the details of Use Case Picks Order

USECASE ID	**UC-16**
Name:	Picks Order
Actor:	Rider
Description:	To collect the order(items) from mart, update order status to picked and deliver it.
Goal:	To pick order(items) from the mart and update order status to picked.
Pre-Condition:	Rider must have picked the order.
Post-condition:	Order status will be updated to picked.
Main Flow:	After receiving order rider will click order picked button to update order status to picked and then deliver it.
Alternate-flow:	If failed, Rider must reload page and try again.
Error:	………

3.6.16 Use-case 15: Deliver Order

Following are the details of Use Case Deliver Order

USECASE ID	**UC-15**
Name:	Deliver Order
Actor:	Rider
Description:	To update the order status to delivered after delivery.
Goal:	To deliver order and update order status to delivered.
Pre-Condition:	Rider must have delivered the order at right location and payment must be received.
Post-condition:	Order status will be updated to delivered.
Main Flow:	When the rider delivers the order and receives payment. Rider goes to the order page and updates order status to delivered.
Alternate-flow:	If failed, Rider must reload page and try again.
Error:	………

USECASE ID	**UC-16**
Name:	Track order
Actor:	Customer
Description:	Customer can track his order when his order is picked by the rider
Goal:	To track rider and be up to date about the delivery.
Pre-Condition:	Order must be accepted and rider must have picked the order,
Post-condition:	Customer can track his rider or delivery.
Main Flow:	When rider will pick the order customer will be notified by his rider’s details. Customer can click track order button and track his delivery
Alternate-flow:	If failed, Customer must reload page and try again.
Error:	………

3.7. Domain Model

A domain model is a conceptual model of the domain that includes both behavior and data in Software Engineering. A domain model is a formal representation of a knowledge domain including concepts, roles, datatypes, individuals, and rules, usually anchored in description logic, in ontology engineering.

3.7.1. Use Case Diagram

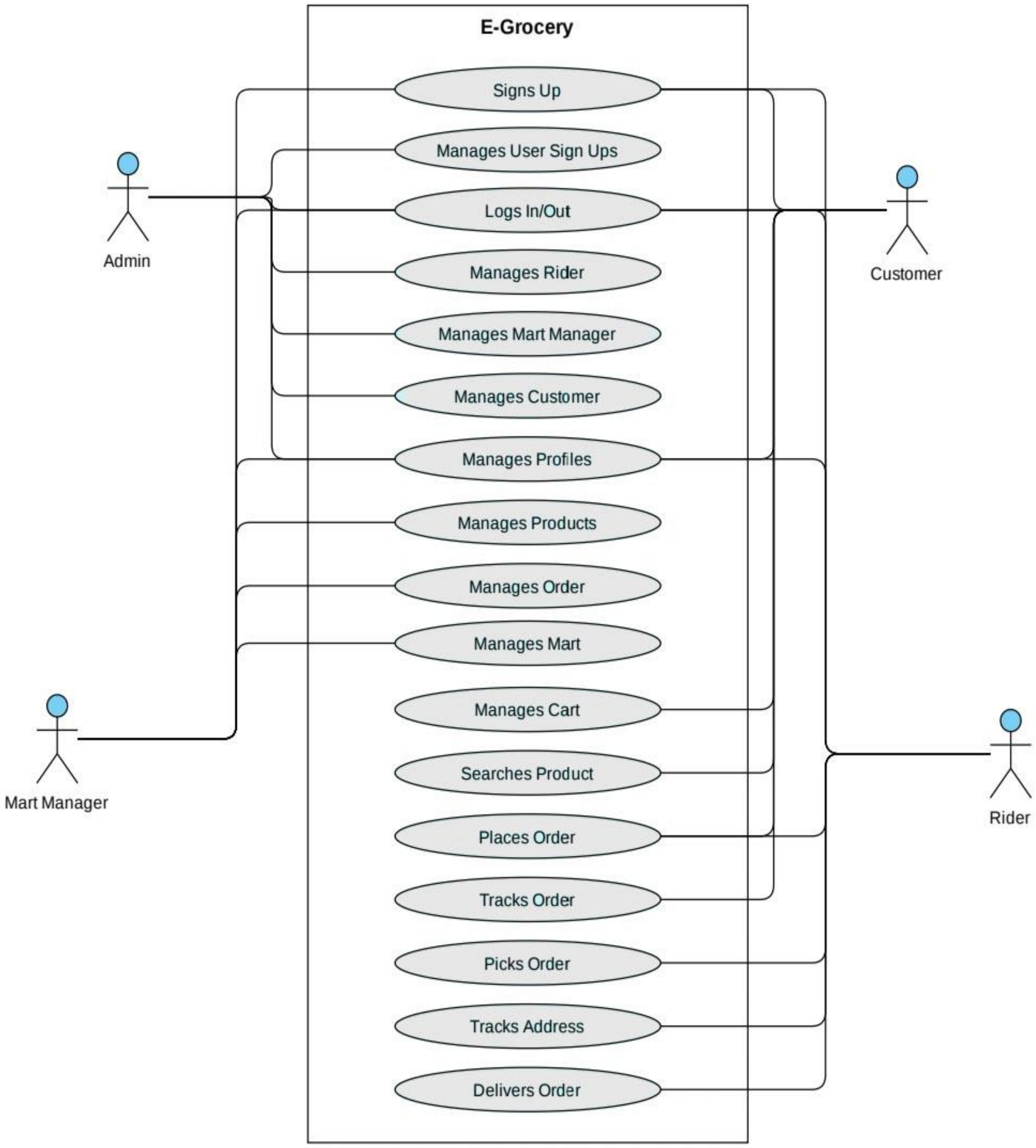

Figure 3.7.1

3.8. System Sequence Diagram

Following are the System Sequence Diagrams of E-Grocery System

3.8.1. Sign Up

The figure 3.8.1shows the sequence diagram for sign up.

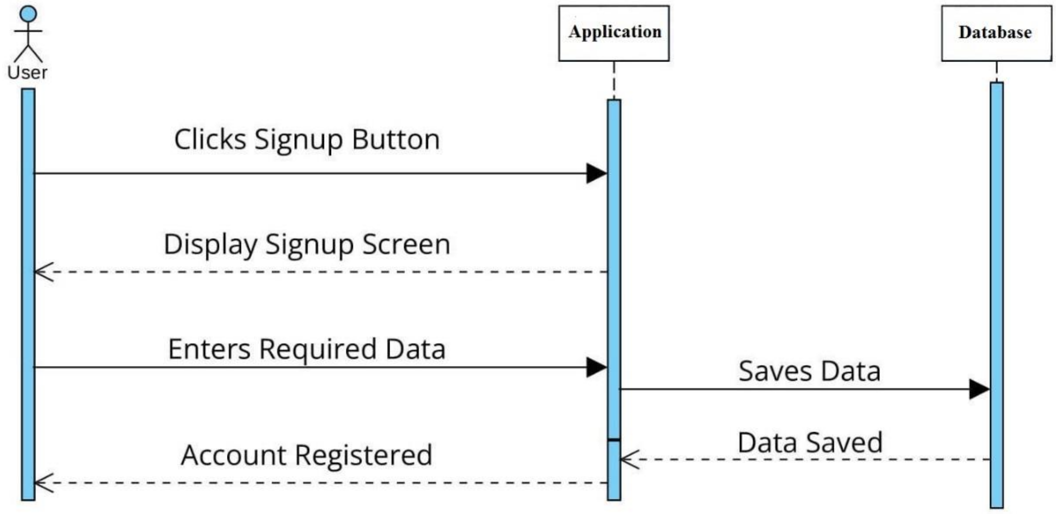

Figure 3.8.1

3.8.2. Log In

The figure 3.8.2 shows the sequence diagram for Log In

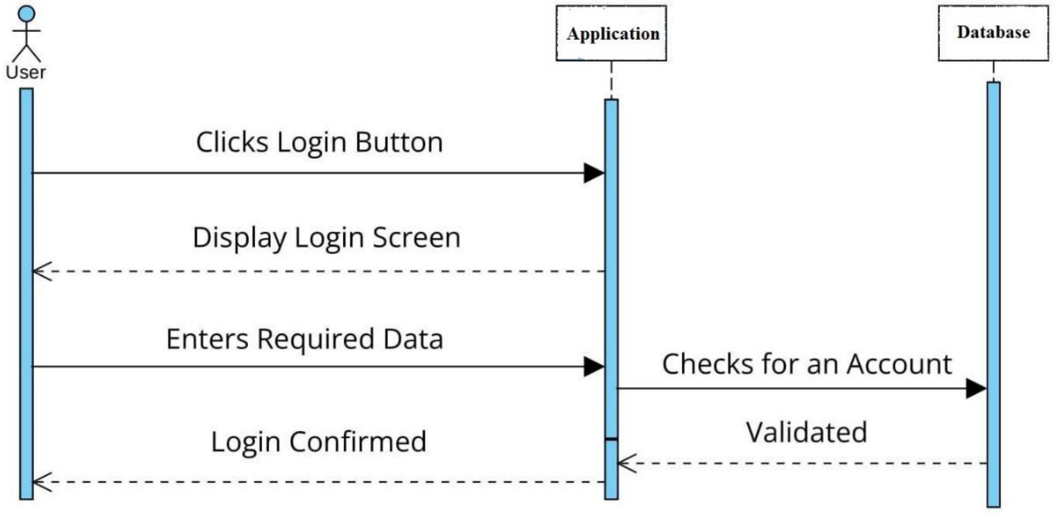

Figure 3.8.2

3.8.3. Log Out

The figure 3.8.3 shows the sequence diagram for Log Out

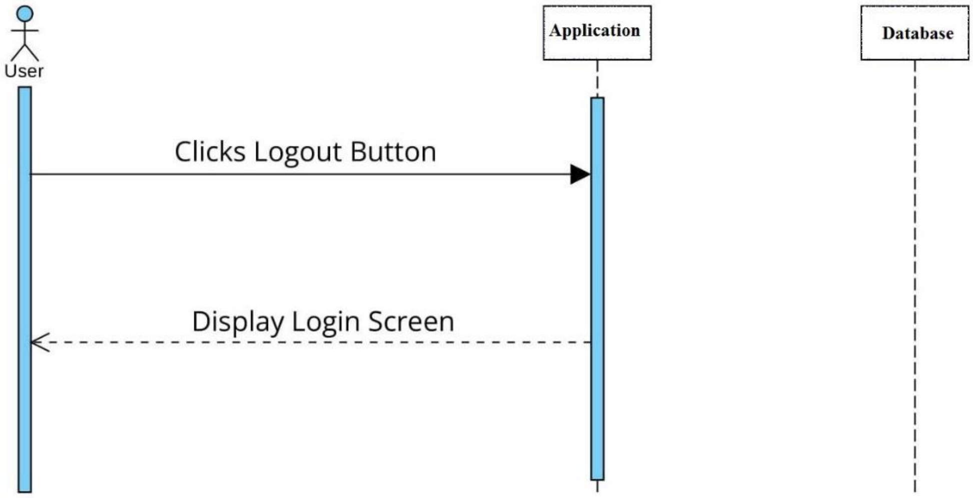

Figure 3.8.3

3.8.4. Add Rider

The figure 3.8.4 shows the sequence diagram for Add Rider

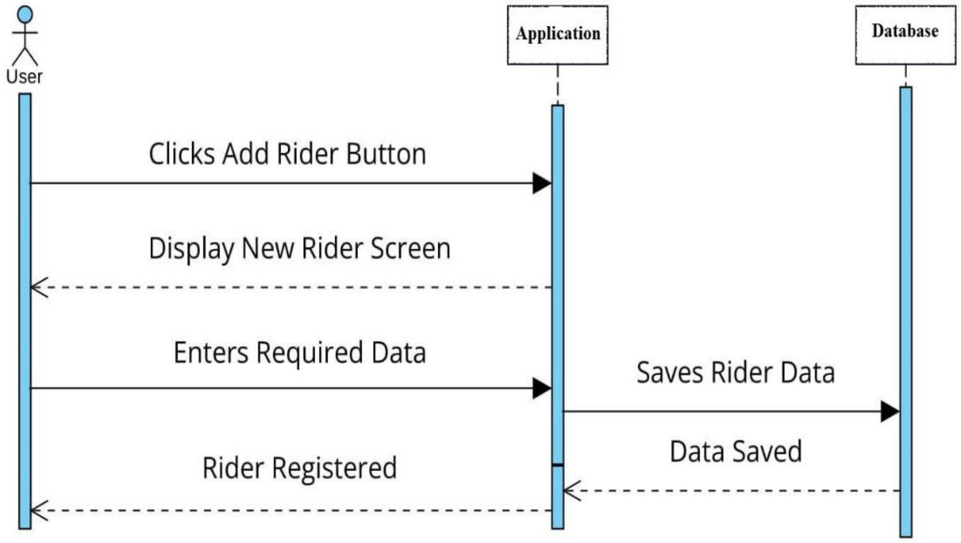

Figure 3.8.4

3.8.5. Update Rider

The figure 3.8.5 shows the sequence diagram for Update Rider

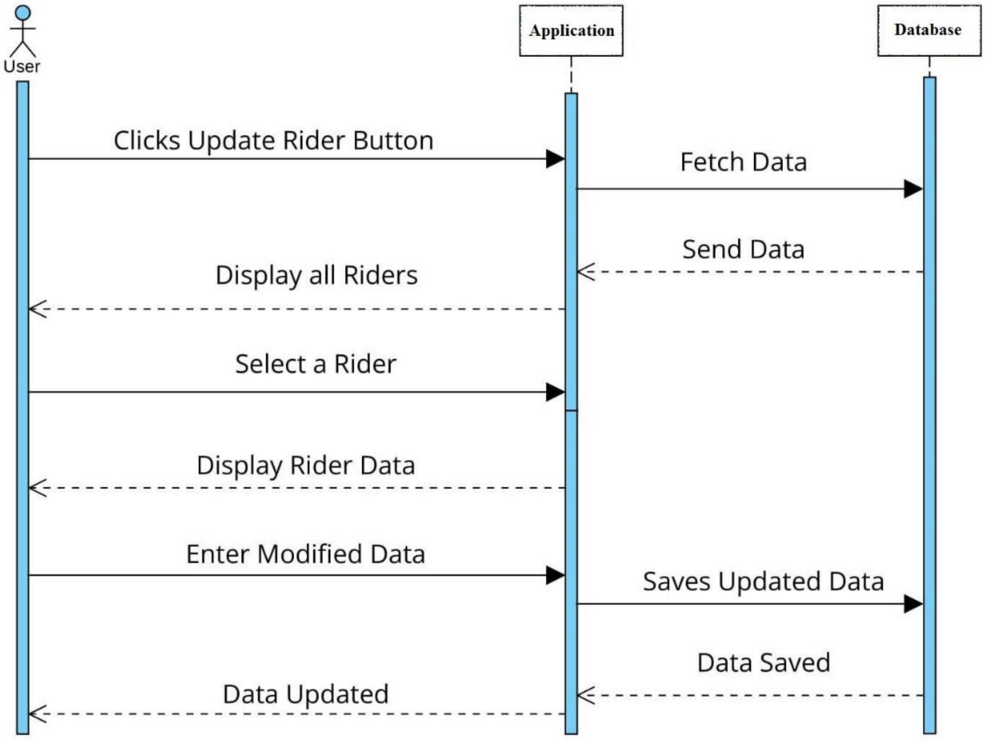

Figure 3.8.5

3.8.6. Delete Rider

The figure 3.8.6 shows the sequence Diagram of Delete Rider

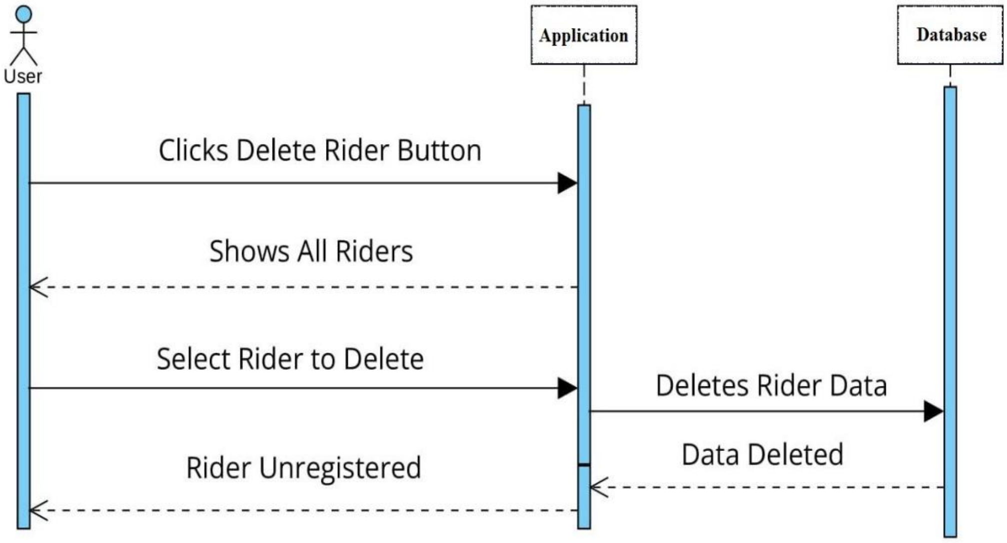

Figure 3.8.6

3.8.7. Manage Profile

The figure 3.8.7 shows the sequence diagram for Manager Profile

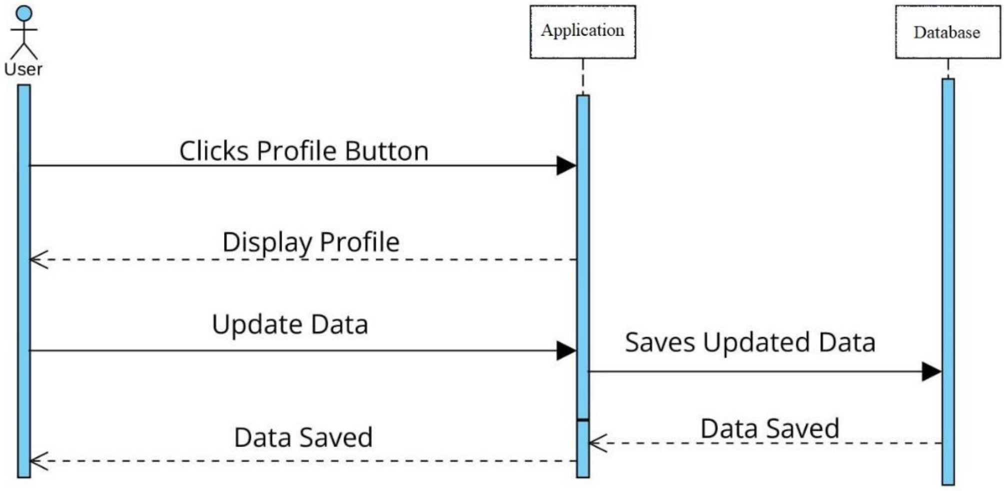

Figure 3.8.7

3.8.8. Add Mart Manager

The figure 3.8.8 shows the sequence diagram for Add Mart Manager

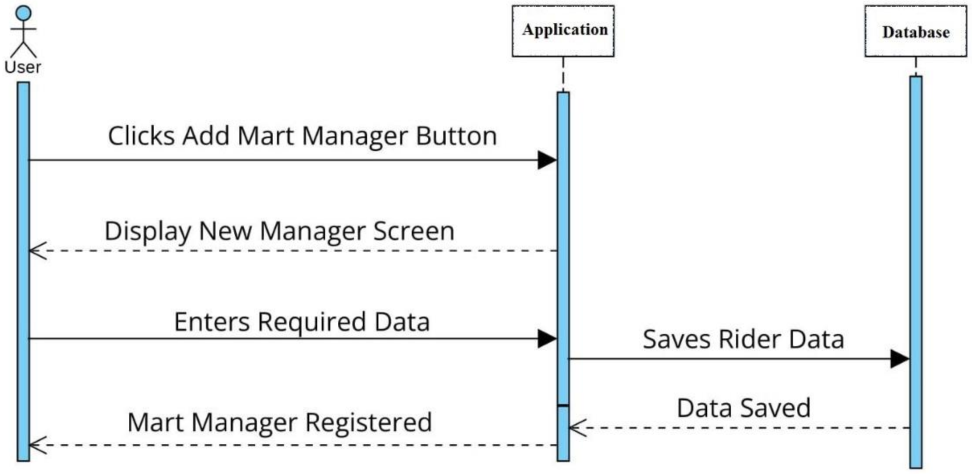

Figure 3.8.8

3.8.9. Update Mart Manager

The figure 3.8.9 shows the sequence diagram for Update Mart Manager

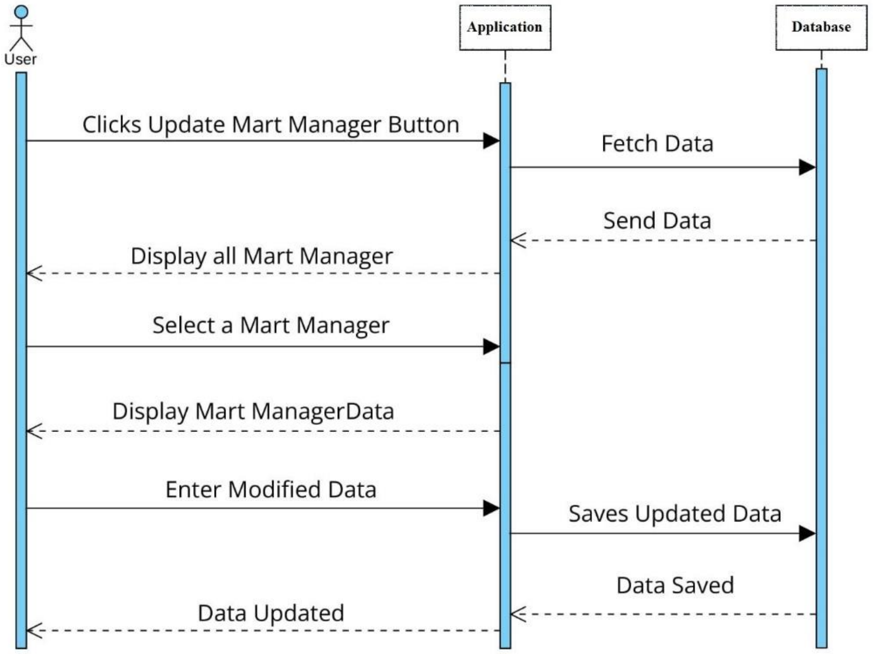

Figure 3.8.9

3.8.10. Delete Mart Manager

The figure 3.8.10 shows the sequence diagram for Delete Mart Manager

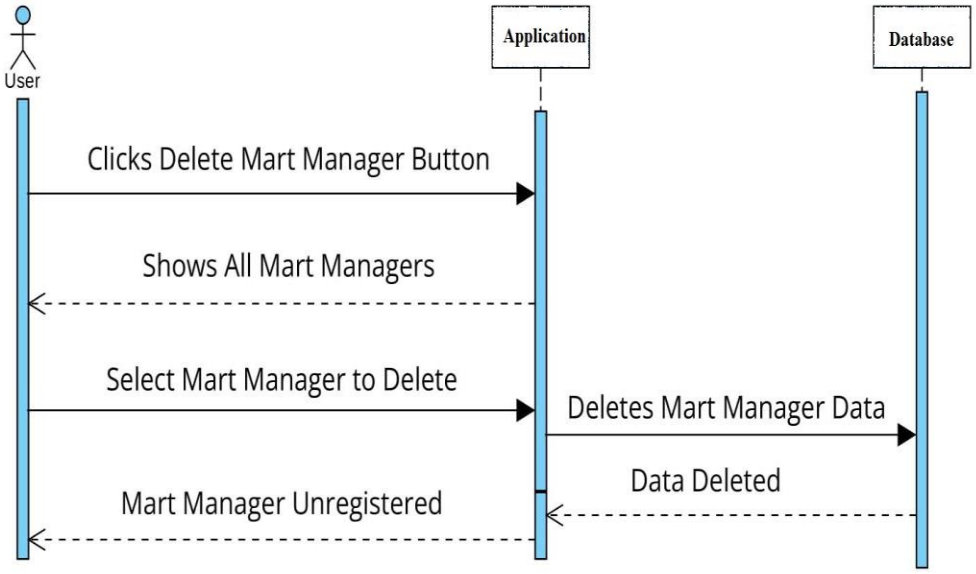
User
Application
Database
Clicks Delete Mart Manager Button
Shows All Mart Managers
Select Mart Manager to Delete
Deletes Mart Manager Data
Data Deleted
Mart Manager Unregistered

Figure 3.8.10

3.8.11. Add Product

The figure 3.8.11 shows the sequence diagram for Add Product

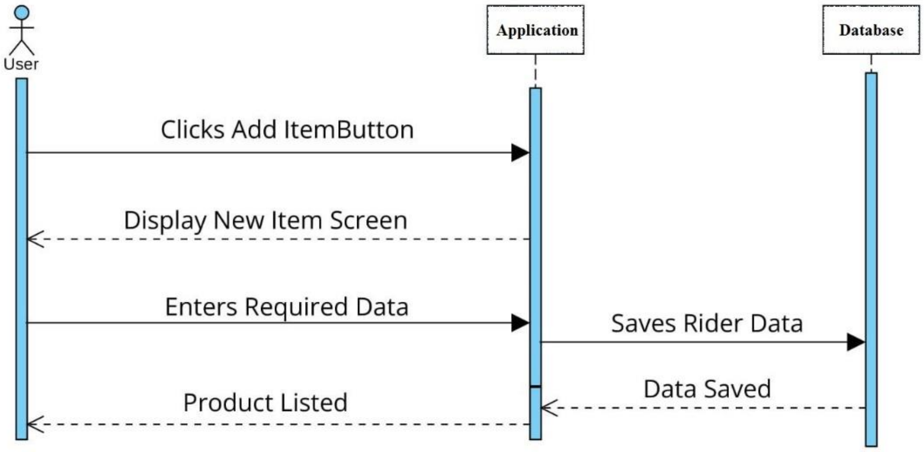

Figure 3.8.11

3.8.12. Update Product

The figure 3.8.12 shows the sequence diagram for Update Product

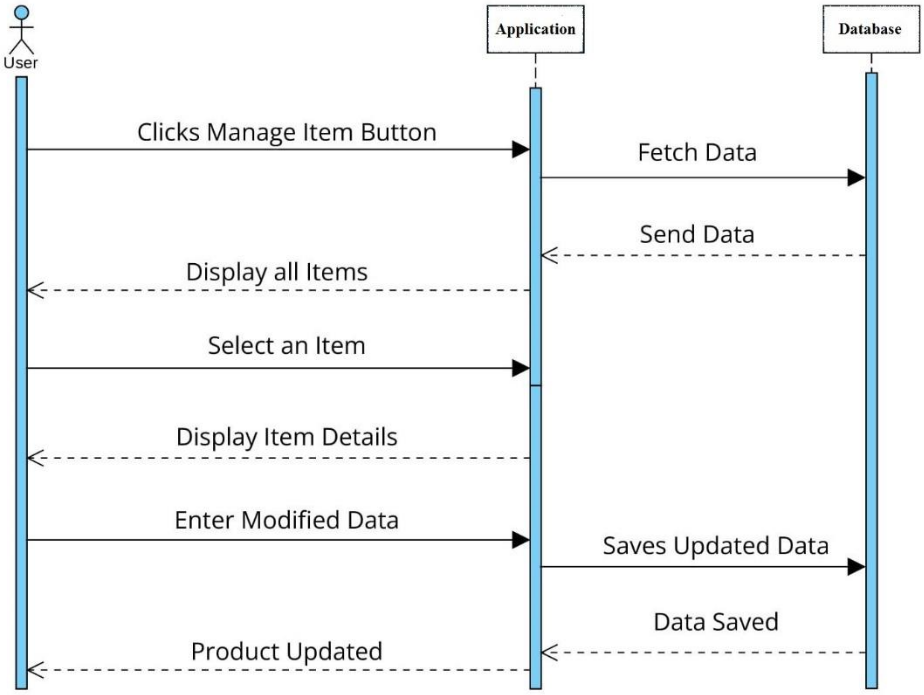

Figure 3.8.12

3.8.13. Delete Product

The figure 3.8.13 shows the sequence diagram for Delete Product

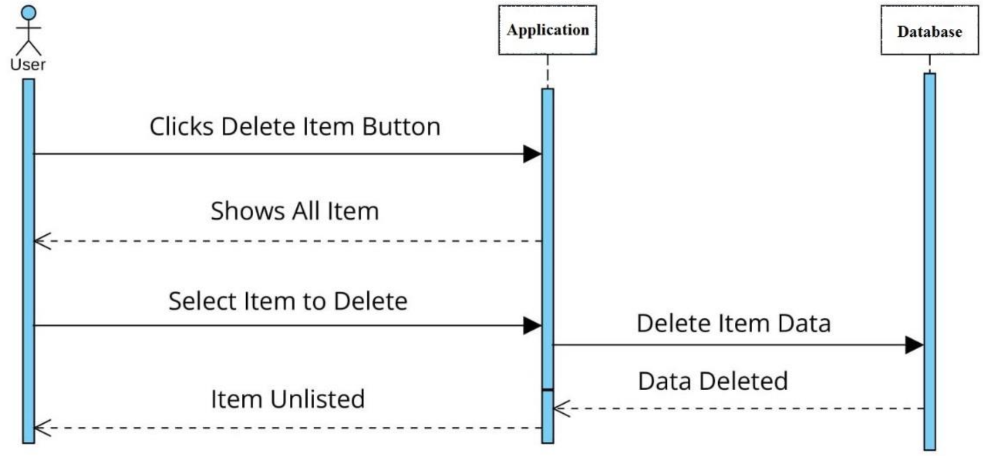

Figure 3.8.13

3.8.14. Manage Order

The figure 3.8.14 shows the sequence diagram for Manage Order

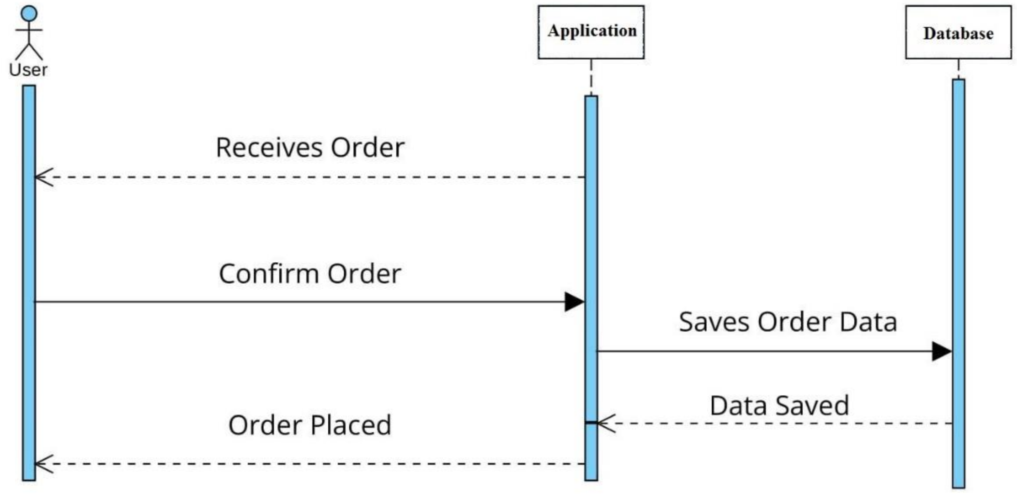

Figure 3.8.14

3.8.15. Add Mart

The figure 3.8.15 shows the sequence diagram for Add Mart

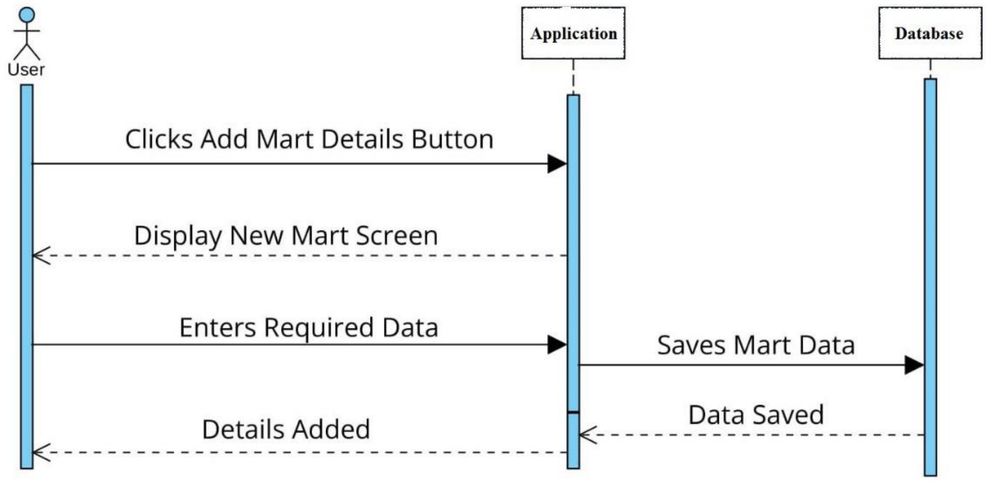

Figure 3.8.15

3.8.16. Update Mart

The figure 3.8.16 shows the sequence diagram for Update Mart

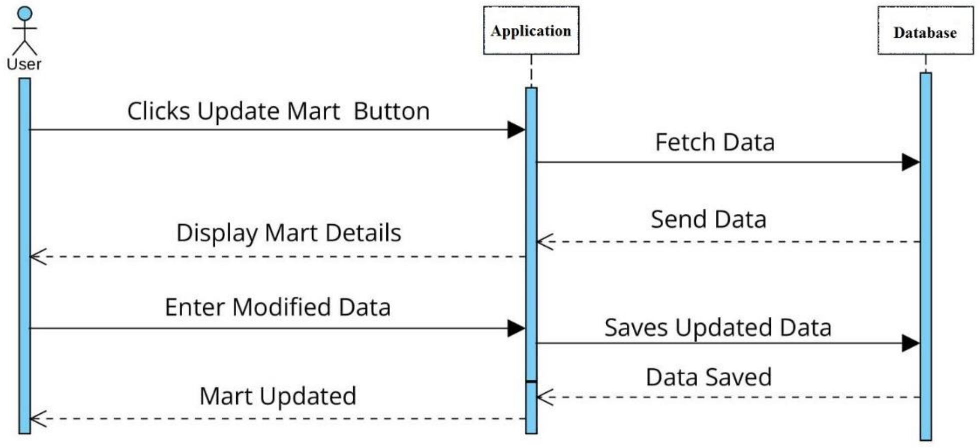

Figure 3.8.16

3.8.17. Add to Cart

The figure 3.8.17 shows the sequence diagram for Add to Cart

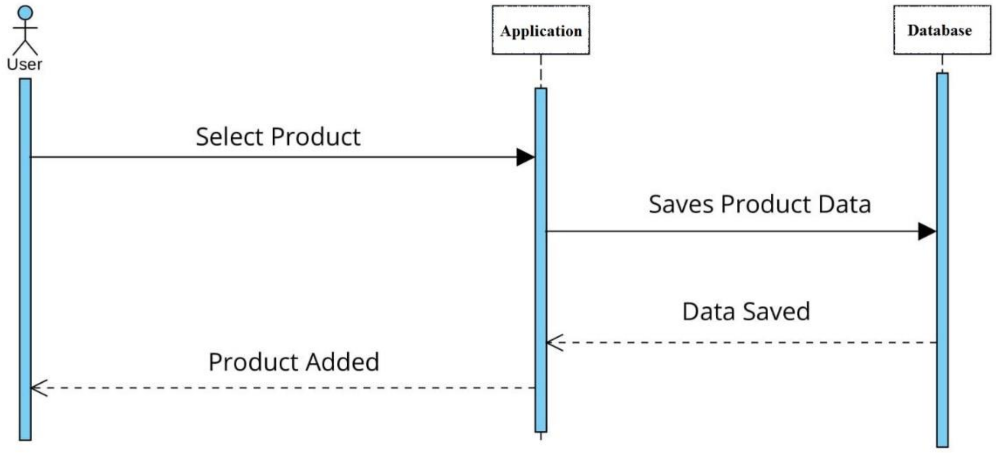

Figure 3.8.17

3.8.18. Remove from Cart

The figure 3.8.18 shows the sequence diagram for Remove from Cart

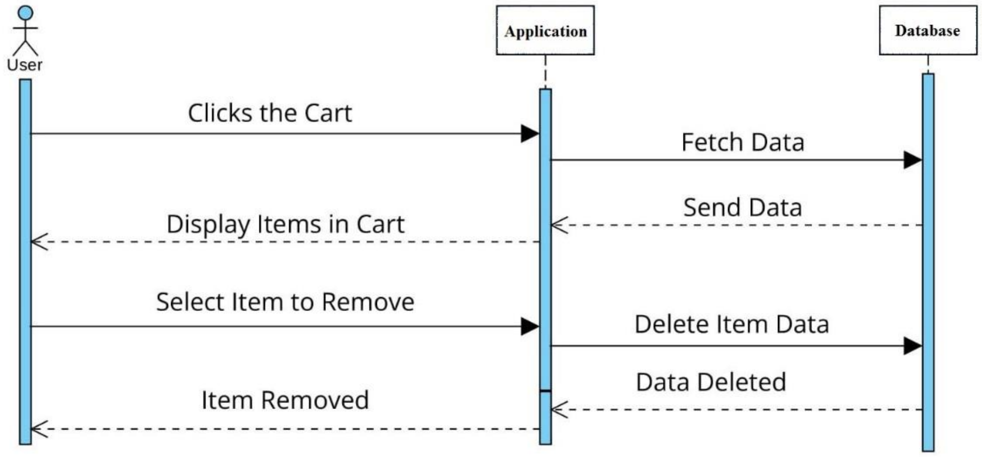

Figure 3.8.18

3.8.19. Search Product

The figure 3.8.19 shows the sequence diagram for Search Product

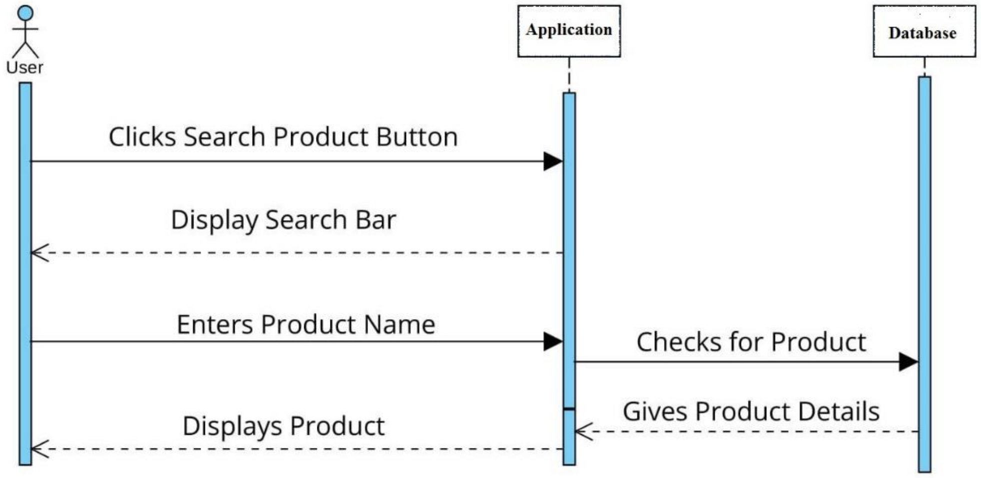

Figure 3.8.19

3.8.20. Place Order

The figure 3.8.20 shows the sequence diagram for Place Order

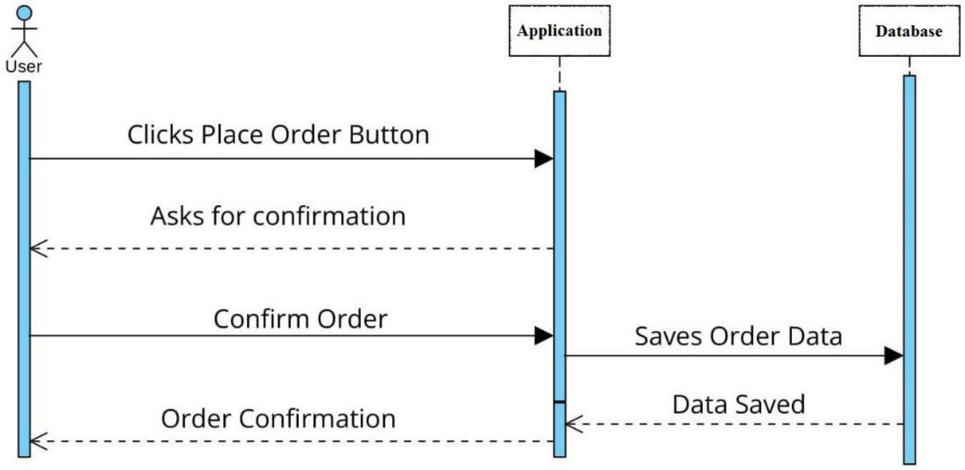

Figure 3.8.20

3.8.21. Track Path

The figure 3.8.21 shows the sequence diagram for Track Path

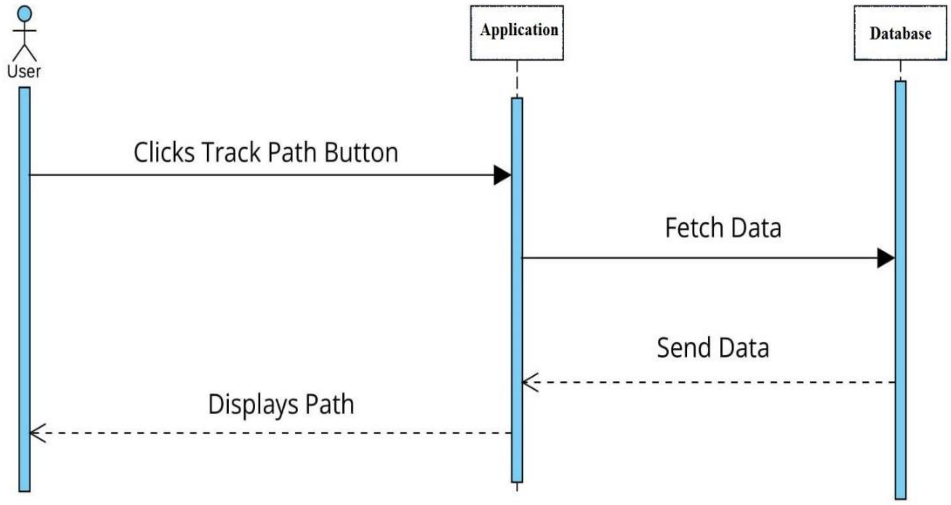

Figure 3.8.21

3.8.22. Deliver Order

The figure 3.8.22 shows the sequence diagram for Deliver Order

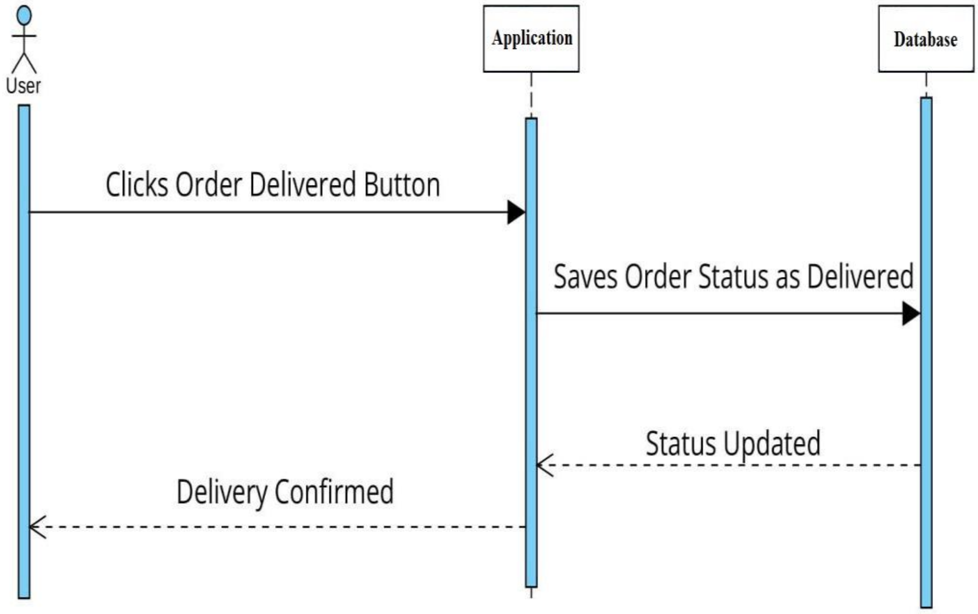
User
Application
Database
Clicks Order Delivered Button
Saves Order Status as Delivered
Status Updated
Delivery Confirmed

Figure 3.8.22

3.8.23. Pick Order

The figure 3.8.23 shows the sequence diagram for Pick Order

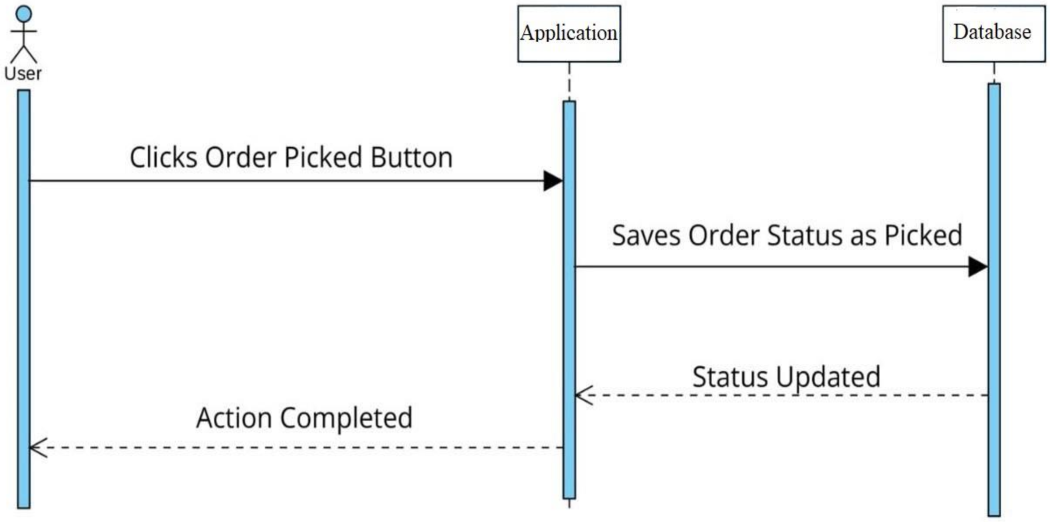

Figure 3.8.23

Chapter 4
System Design

4.1 Design Model

4.1.1 Interaction Diagram

4.1.2 Class Diagram:

Figure 4.1.2 shows the class diagram of the system.

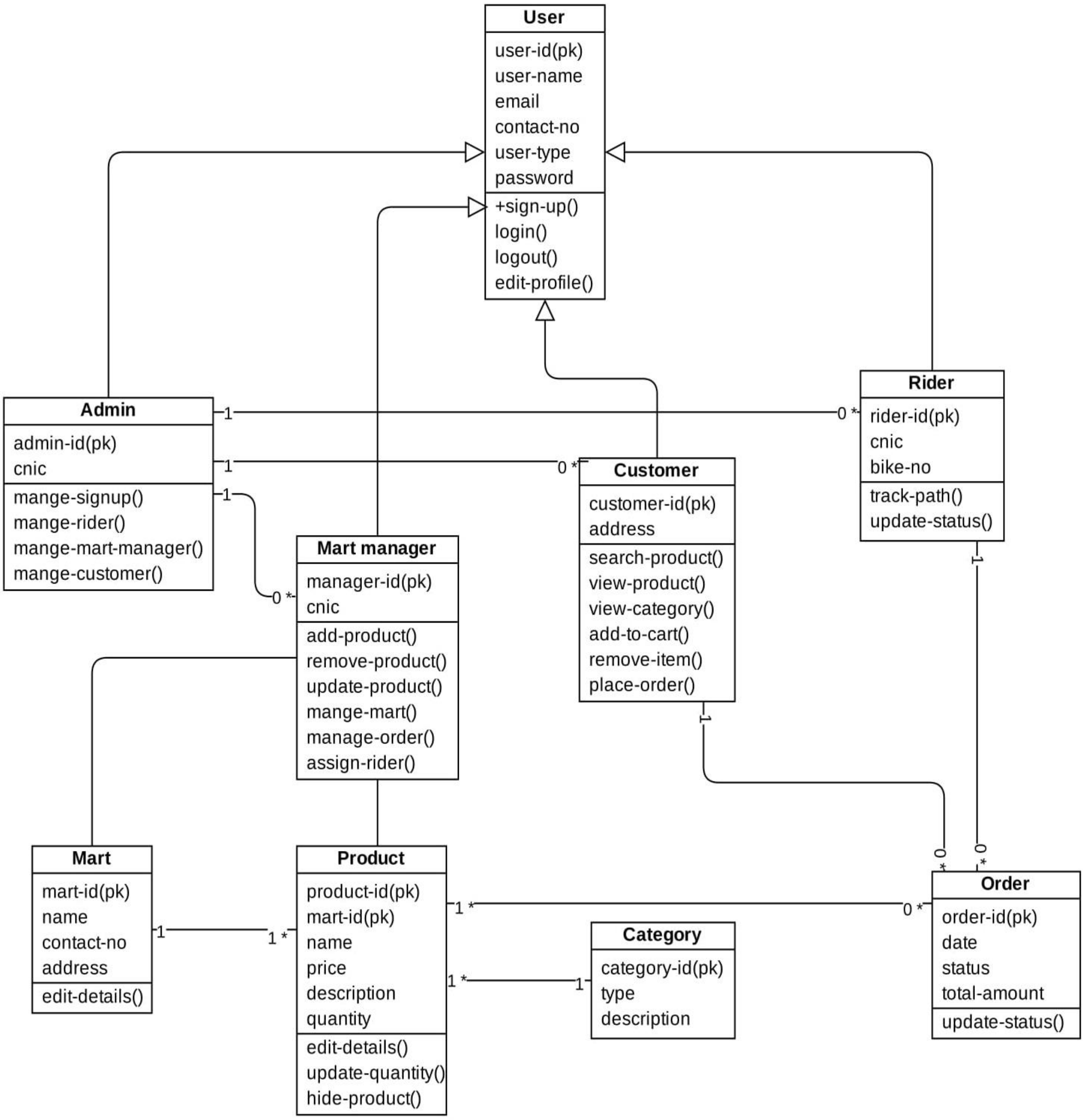

Figure 4.1.2

4.2 Activity Diagram

4.2.1 Admin:

Figure 4.2.1 shows the activity diagram for admin.

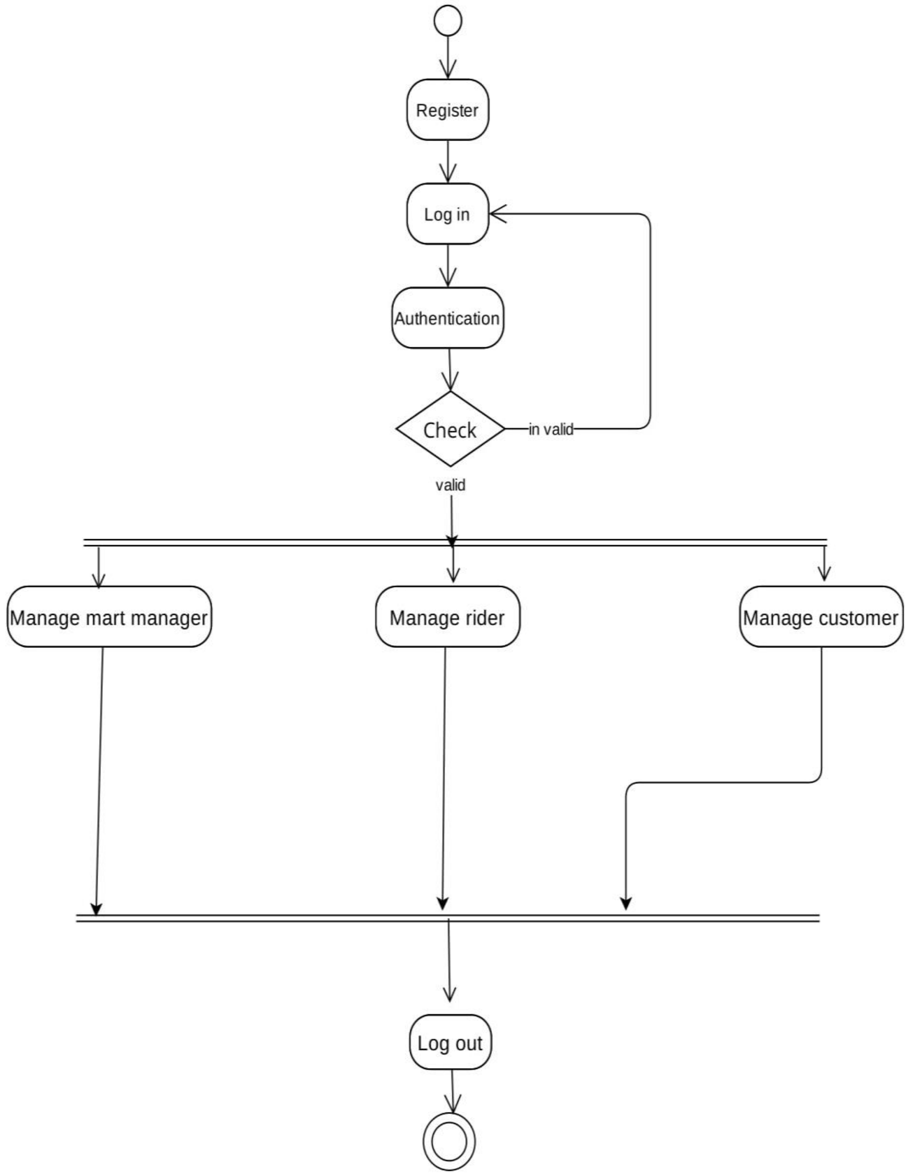

Figure 4.2.1

4.2.2 Mart manager:

Figure 4.2.2 shows the activity diagram for admin.

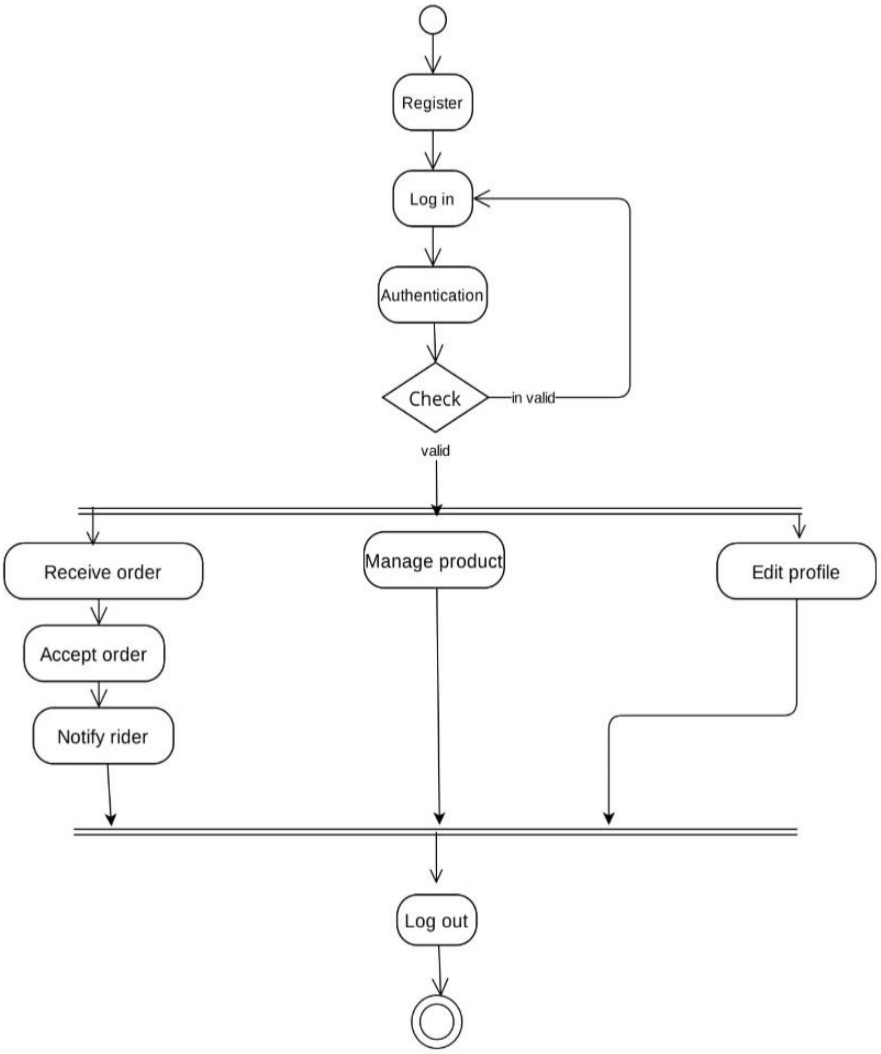

Figure 4.2.2

4.2.3 Rider:

Figure 4.2.3 shows the activity diagram for admin.

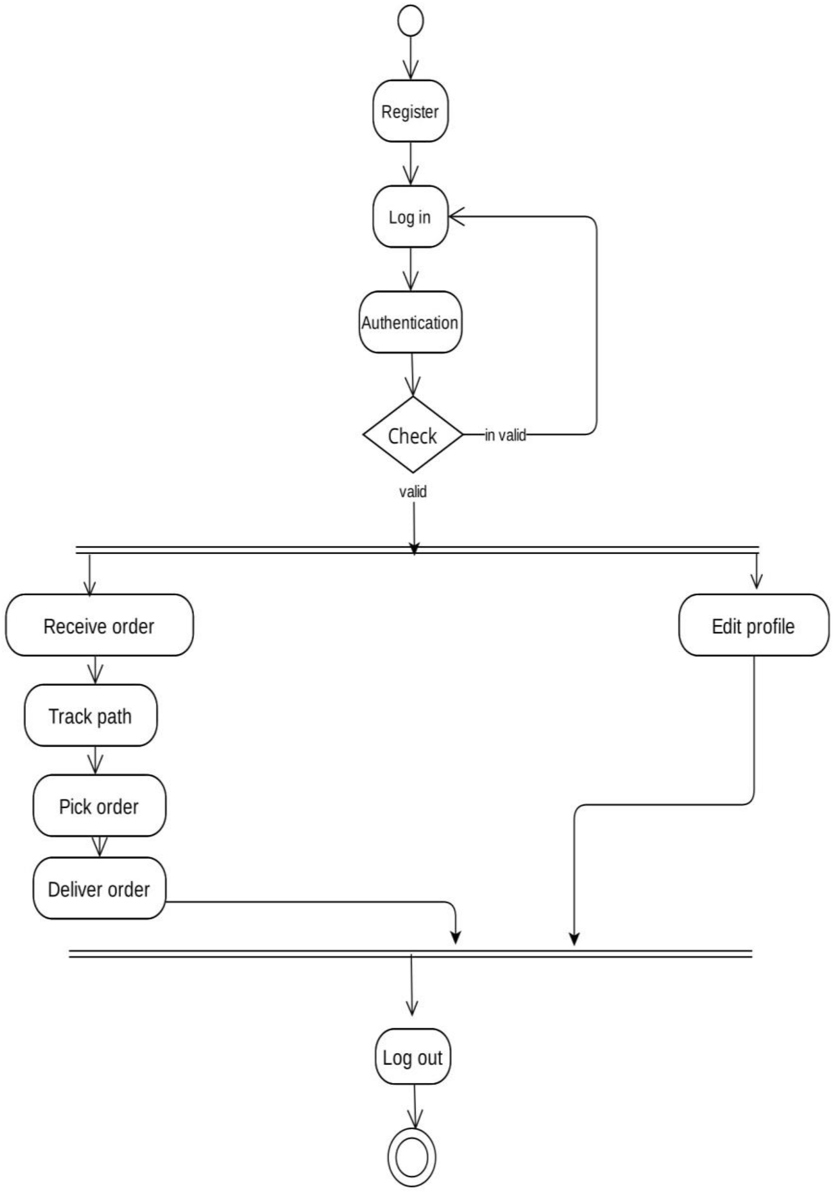

Figure 4.2.3

4.2.4 Customer:

Figure 4.2.4 shows the activity diagram for admin.

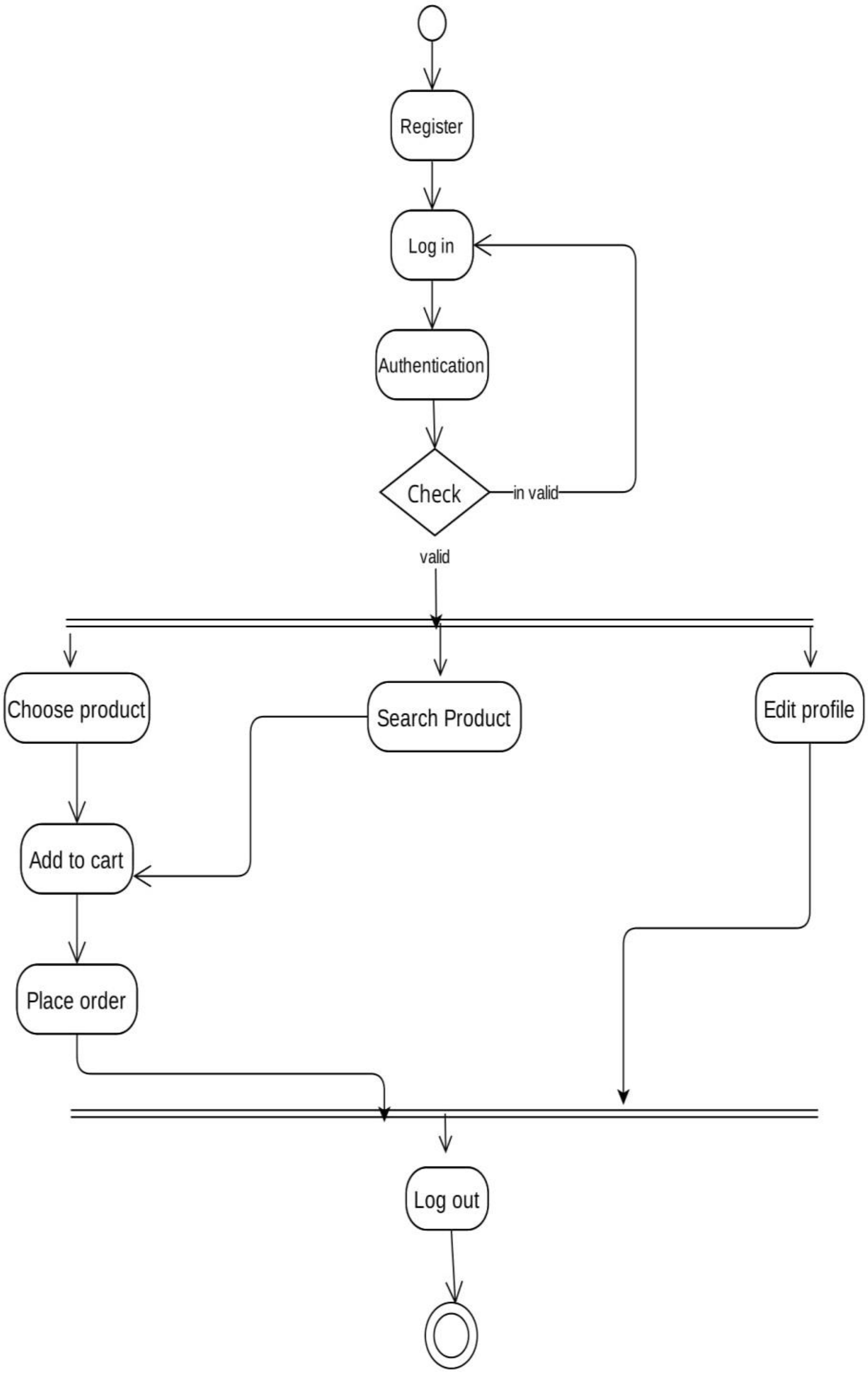

Figure 4.2.4

4.3 Data Model

4.3.4 ERD

The ERD of E-Grocery System is given below

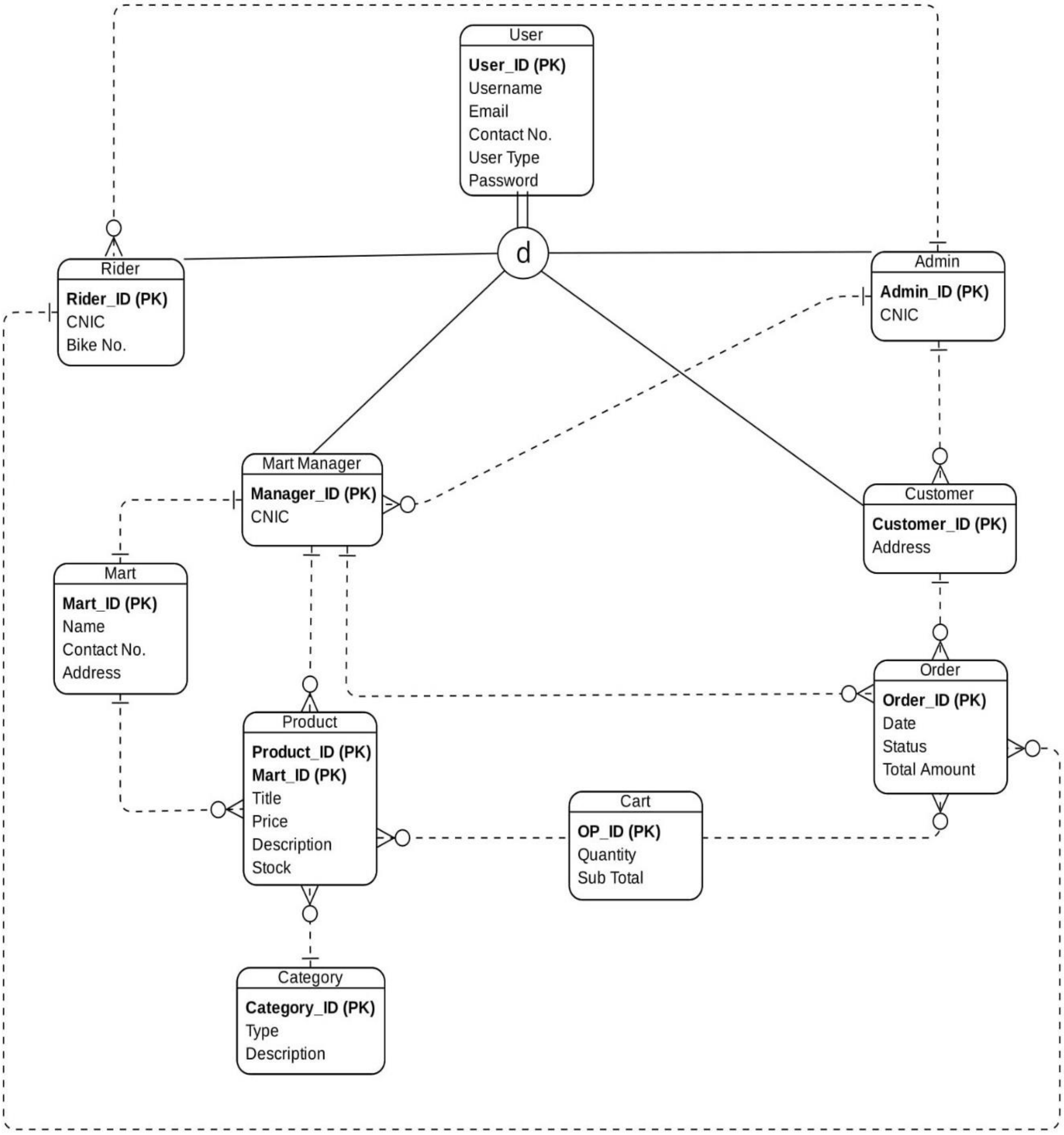

Figure 4.3.1