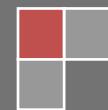


# F-COMMERCE MANAGEMENT SYSTEM

Software Requirement Specification



# **f-Commerce**

## **Management System**

**Submitted to**

Nadia Nahar

Lecturer

Institute of Information Technology

University of Dhaka

**Submitted by**

MD Rakibul Islam	BSSE0903
MD Aminul Islam	BSSE0906
Md Atiquer Rahman	BSSE0908
Amran Hossain	BSSE0917
Pritom Kumar Das	BSSE0919
Ishrat Jahan Emu	BSSE0927

Submission Date: 18 November 2018

## **Letter of Transmittal**

Nadia Nahar

Lecturer

Institute of Information Technology

University of Dhaka

**Subject: Submission of term report on “f Commerce Management System”.**

Madam,

With due respect, we are submitting the report on the above topic you assigned us. In this report, we have given our best effort albeit some shortcomings. We earnestly hope that you would excuse our errors and oblige thereby.

Yours sincerely,

MD Rakibul Islam	BSSE0903
MD Aminul Islam	BSSE0906
Md Atiquer Rahman	BSSE0908
Amran Hossain	BSSE0917
Pritom Kumar Das	BSSE0919
Ishrat Jahan Emu	BSSE0927

2<sup>nd</sup> year, 4<sup>th</sup> semester, 9<sup>th</sup> batch

Institute of Information Technology

University of Dhaka

Session: 2016-17

## Acknowledgement

We are highly indebted for getting such a tremendous opportunity to prepare the report on f-Commerce Management System. We would like to thank whole-heartedly our course teacher, Nadia Nahar, Lecturer, Institute of Information Technology, University of Dhaka, for giving us guidelines about how we can prepare this report. In completing this report, we have collected various important data and information from various facebook based online shops. We are thankful to all of the works cited.

## **Abstract**

This study is made for f Commerce Management System. The scope of the study is to analyze the facebook based shop management system and to know the functions and drawbacks and design the SRS (software requirements and specification) of the system. The object of the study is to develop an SRS of a facebook based online shop management system. This study also describes the current system of the online shop.

# Table of Contents

<b>Chapter1.....</b>	<b>1</b>
<b>Chapter2.....</b>	<b>3</b>
<b>Chapter3.....</b>	<b>8</b>
<b>Chapter4.....</b>	<b>21</b>
<b>Chapter5.....</b>	<b>86</b>
<b>Chapter6.....</b>	<b>100</b>
<b>Chapter7.....</b>	<b>130</b>
<b>Chapter8.....</b>	<b>161</b>
<b>Chapter9.....</b>	<b>162</b>

# Chapter1

## Introduction

This chapter is a part of the software requirement specification for the project “f Commerce Management System”. In this chapter intended audience for the project are focused on.

### 1.1 Purpose

This document briefly describes the Software Requirement Analysis of f commerce management system. It contains functional, non-functional, and supporting requirements and establishes a requirement baseline for the development of the system. The requirements specified in the SRS are independent, uniquely numbered and organized by topic. The SRS serves as an official mean of communicating user requirements to the developer and provides a common reference point for both the developer team and the stakeholder community. The SRS will evolve over time as users and developers work together to validate, clarify and expand its contents.

### 1.2 Intended Audience

This SRS is intended for several audiences including the customers as well as the project managers, designers, developers and testers.

- The customers will use this SRS to verify that the developer team has created a product that is acceptable to the customers.
- This project managers of the developer team will use this SRS to plan milestones and a delivery date and ensure that the developing team is on track during development of the system.

- The designers will use this SRS as a basis for creating the system's design. The designers will fulfill the customer's needs.
- The developers will use this SRS as a basis for developing the system's functionality. The developers will link the requirements defined in this SRS to the software they create to ensure that they have created a software that will fulfill all the customer's documented requirements.
- The testers will use this SRS to derive test plans and test cases for each documented requirement. When portions of the software are complete, the testers will run their tests on that software to ensure that the software fulfills the requirements documented in this SRS. The testers will again run their tests on the entire system when it is complete and ensure that all requirements documented in this SRS have been fulfilled.

### **1.3 Conclusion**

This analysis of the audience helped us to focus on the users who will be using our analysis. This overall document will help each and every person related to this project to have a better idea about the project.

# Chapter 2

## Inception

In this chapter, the inception part of the SRS will be discussed briefly.

### 2.1 Introduction

Inception is the beginning phase of requirements engineering. It defines how a software project gets started and what the scope and nature of the problem to be solved is. The goal of the inception phase is to identify concurrent needs and conflicting requirements among the stakeholders of a software project. At project inception, we established a basic understanding of the effectiveness of preliminary communication and collaboration between the other stakeholders and the software team.

To establish the ground work, the following factors have been worked onto the inception phase:

- List of stakeholders
- Recognizing multiple viewpoints
- Working towards collaboration
- Requirements questionnaire

#### 2.1.1 List of Stakeholders

Stakeholders refers to any person or group who will be affected by the system directly or indirectly. Stakeholders include end-users who interact with the system and every one else in an organization that might be affected by its installation. At inception, a list of people who will contribute input as requirements are elicited. The initial list will grow as stake holders are contacted because every stakeholder will be asked: "whom else do you

think I should talk to?"

The following stakeholders were identified for the grocery shop management system.

**Owner:** Owner owns the shop, operates the business in facebook and recruits the employees. Some shops are owned in shares. If he works alone then, he does not demand day to day transaction. He also profits from the net gain of the business.

**Customer:** A customer is an individual or business that purchases or consumes the goods or services produced by a business. Attracting customers is the primary goal of most businesses as it is the customer who creates demand for goods or services. Businesses often compete through advertisements or lowered prices to attract customers.

**Supplier:** A party of the respective company that supplies goods or services. A supplier may be distinguished from a contractor or subcontractor, who commonly adds specialized input to deliverables.

**Software Developer :** A software developer is concerned with facets of the software development process, including the research, design, programming, maintenance and testing of computer software. He will be responsible for the outcomes of the software.

### **2.1.1 Recognizing Multiple Viewpoints**

Different stakeholders demand different features from the software. To satisfy the stakeholders, most of these features should be included in the software.

#### **Owner's viewpoint**

- Error free system
- Getting all the information about profit on a specific date
- Managing employees' salary
- Keeping all information regarding shop maintenance
- Net profit after paying the due
- Keeping profit records on monthly basis
- Limited budget
- Strong authentication system

#### **Employee's viewpoint**

- User friendly
- Error free system
- Keeping a track on remaining products
- Getting accurate information of total received cash and profit of a day
- Identifying the profitable products
- Identifying the unpopular products among customers
- Being notified about the products near to expire
- Strong authentication system
- Keeping all information regarding the shop maintenance
- Informing the suppliers about the needed products
- Keeping a backup of the data

#### **Supplier's viewpoint**

- Getting information about the needed products
- Total products and transport expenses

#### **Customers' viewpoint**

- Getting a receipt of the purchase
- Privileges of online shopping

#### **Developer's viewpoint**

- Easy to built
- Error free effective software
- No ambiguous requirement
- Getting a decent amount of money for project budget

### **2.1.2 Working towards collaboration**

While working with different stakeholders, some conflicting and common viewpoints can be noticed. For this reason, final requirements can be gotten by collaborating the viewpoints.

#### **Common viewpoints**

- Error free effective system
- User friendly
- Easy to maintain the software
- Strong authentication system

#### **Conflicting viewpoints**

- Keeping information regarding day to day profit
- Developing the project in minimum budget

- Getting a receipt for clarification
- Online shopping system
- Managing employees' salary
- Send information to the supplier about the needed products and expenses
- Identifying the profitable products
- Keeping all information regarding the shop maintenance

### **Final Requirements**

- Error free effective system
- User friendly
- Easy to maintain the software
- Strong authentication system
- Net profit after paying the due
- Keeping a track of the remaining products
- Being notified about the products near to expire
- Keeping a data backup

#### **2.1.3 Requirements Questionnaire**

At first some context free questions were asked for identifying the stakeholders. Context free questions are helpful to identifying some stakeholders who cannot be identified by structural questions. Then questions regarding the software were regarding to know their demands. The questionnaires are included in the appendix section.

#### **2.2 Conclusion**

In this inception phase, a basic understanding of the problem was developed and a preliminary nature of the solution was obtained. The

requirements which are identified in this phase, will be used later for further steps of requirement engineering.

# Chapter 3

## Elicitation

This chapter specifies the Elicitation phase.

### 3.1 Introduction

Requirements Elicitation is a part of requirements engineering that is the practice of gathering requirements from the users, customers and other stakeholders. Many difficulties were faced, like understanding the problems, making questions for the stakeholders, limited communication with the stakeholders due to a short amount of time and volatility. Though it is not easy to gather requirements within a very short time, these problems have been surpassed in an organized and systematic manner.

### 3.2 Eliciting Requirements

The main task of this phase is to combine the elements of problem solving, elaboration, negotiation and specification. The collaborative working approach of the stakeholders is required to elicit the requirements. The following tasks were done for eliciting requirements-

- Collaborative Requirements Gathering
- Quality Function Deployment
- Usage Scenarios
- Elicitation work products

#### 3.2.1 Collaborative Requirements Gathering

The meetings with the stakeholders created an indecisive state to elicit the

requirements. To solve this problem, more than one meeting was held with the stakeholders. A slightly different scenario from these approaches has been found following activities have been completed to accomplish this task.

- The meetings were conducted with the owner, employees and suppliers. They were questioned about their requirements and expectations from the Grocery Management System.
- They were asked about the problems they were facing with the current manual system.
- Lastly final requirement list was selected from the meetings.

### **3.2.2 Quality Function Deployment**

Quality Function Deployment (QFD) is a technique that translates the needs of the customer into technical requirements for software. Ultimately the goal of QFD is to translate subjective quality criteria into objective ones that can be quantified and measured and which can then be used to design and manufacture the product. It is a methodology that concentrates on maximizing customer satisfaction from the software engineering process. The requirements, which are given below, are identified successfully by the QFD.

#### **3.2.2.1 Normal Requirements**

The normal requirements are generally the objectives and goals that are stated for a product or system during meetings with the customer. The presence of these requirements fulfills customers' satisfaction. These are the normal requirements for the project.

- Sign up
- Product Search
- Order & Payment System
- Getting information about daily, weekly, monthly and yearly profit.

- Managing employees' salaryproperly
- Transaction History
- Net profit after paying thedue
- Storing profitrecords
- Using limited budget for making thesoftware
- Keeping track of remaining products
- Storing accurate records oftransactions
- Identifying profitable and unpopularproducts
- Get notified about products nearexpiration
- Contacting suppliers forgoods
- Calculating transport expenses for carryinggoods

### **3.2.2.2 Expected Requirements**

These requirements are intrinsic to the product or system and may be so elementarythatthecustomerdoesnotexplicitlystatethem.Theirabsence will be a cause for significant dissatisfaction. Below the expected requirements for our project are brieflydescribed.

- Error freesoftware
- Strong authenticationsystem
- Secured Transaction
- Userfriendly
- Effectivesystem
- No ambiguousfeature
- Data back up

### **3.2.2.3 Exciting Requirements**

These requirements are for features that go beyond the customer's expectations and prove to be very satisfying when present. Following are some exciting requirements of thisproject.

- Giving privileges of onlineshopping
- Direct Facebook LogIN
- Messenger BOT
- Live Chat Box
- Search History Analysing of user then Suggestion to user
- Datagraph
- Pointing System
- Suppliers can contact owner through text messages and viceversa

# Usage scenario

## Authentication

**Sign Up:** In the FC(F-Commerce), there is an authentication part, where it allows the user to access the system. Admins, employees, suppliers and customers are the users of this system. To access the system, a user requires an account and for that he/she must fill up a form. Users have to give first name, lastname,facebook id, emailid, address, mobile number, password and account type (types control the access to the system). User can fill either email id or mobile number or both. No user name can contain any number, punctuation mark or any special character and the length of the name should be between 2 characters and 30 characters. There will be duplicity and validity (syntax) checking for email and mobile number. If all the information is correct, user will be sent a verification code to his/her email address(if he/she provides only email address) or mobile number(if he/she provides only mobile number) or both(if he/she provides both email id and mobile number). If the account type is "Employee", "Supplier" or "Admin", he/she will have to wait for any of the current admin's approval. There will be one preinstalled account as "Admin" type.

**Sign In:** If any user has an account, he/she can log into the system. To log in, user has to give his/her email id or mobile number and password. The email id or mobile number and password will be verified. If the verification is successful, the user can log in to the system successfully and his/her login information (date and time of login) will be stored. If the password is wrong,

there is a retry option. If the retry count is 5, the user will be blocked for a while. After block time has finished user can attempt to log into the system again. User can update their profile information anytime except their email ID and phone number. The updated information will be stored in database. User has an option to log out from the system. User's will automatically be logged out of his/her account if the account remains idle for more than an hour. User's log out information (date and time of logout) will be stored.

**Recover Account:** User can recover their account with "Forgotten Password" option . To recover account , he/she has to give his/her email id or mobile number that he/she has used to sign up. If the provided email id or mobile number is valid, an email will be sent to their email address or a message will be sent to their mobile or both (If both options are chosen.) containing the link to change their password. The user can not use their old password and have to use a new password for security. After confirming their new password they can log in to their account with their new password.

## Administrative Functionalities

**Product Management :** Admin can include and exclude products. If a new product is added in the product list, the description of the product, price, quantity, product name, discount amount are the fields needed to be filled out by the Admin. The system will generate a product id for each product. In case of supplement of a product, Admin can just edit the quantity of the product or add the increasing quantity of the product. There will be an option for editing the product details. The system cannot identify the missing products by its own.

The employee stores the missing product information which are product id, product name, supply date and missing quantity. The product quantity will be updated. If Admin modifies, adds or removes any product, then those activities will be stored in activity log. If any employee adds missing products, those activities will be stored in activity log.

**Customer Management :**Admin can block customer and delete account that's harmful for the company . Blocked customers can not access website or place order for some time . They will be monitored in the future . The employee can contact customer for opinion or to confirm order placement . There is a helpline for customers , too. They can ask any query about product over the helpline .Customer comfort is the main inspiration of the company .So , all customer queries are taken seriously.

## Profit Strategy Management

**Product Data Analysis:** Identifying the needed products is one of the major tasks of the Admin. The system can also show weekly and monthly popular products. It can also show popular products based on a given interval. To identify the popular products, the system checks the transaction data and sorts out those product IDs which have been sold over 55% of total products in week and month. The system can sort out the profitable products (earning 20% of total profit) over the current month and week or in any period by using the transaction data. The system checks the profit earned from each product and popularity of that product. Then it shows the result in a tabulated way. The product list shows the quantity for each product. A minimum amount of each product will be stored by the Admin. If any product quantity is less than

minimum amount, a notification is sent to the Admin's account that those products are near the limit. Then the Admin checks the product quantity in the list.

**Customer Data Analysis:** For better understanding our customers and their needs the system keeps cookies for all customers . It helps the admin to have better understanding of customer psychology and market demands . If a customer spend 50% more time on a product page he/she will see more advertisement of related products in future visit . It helps the customer to look for things that they are interested in . Admin can also analyse customer history to get valued customer list . When a customer spent over 5000 taka in a week or over 20000 taka in a month they become our valued customer for the next week or month . We keep these customers in high esteem. Their preferences and choice will be taken more seriously when restocking products . The valued customer list will be stored in database.

**Advertisement:** Advertising products is one of the key income source . To sell a product it has to be send to a appropriate audience . Customer cookie analysis helps to find potential customer . Then advertising containing exciting feature of a product is send to the customer . If advertisement is clicked then the customer is taken to product page . The customer can not stop advertisement from website .

## **Order and Delivery Management**

**Customer Order :**The customers having accounts can only view or buy products online. At first, they will have to log in to their account (described in the authentication section). They can sort products according to their popularity, prices, names, special discount or by category. If they want to buy any particular product, they will have to click the product icon and choose the quantity that they want to buy. The products will be stored in a list (their purchased products will be shown in a “cart” icon). When the customer is finished buying the products, information of all the products (product name, quantity, price) including total price of those will be shown in another pop-up interface (the products information will be shown like a list). Customer can remove a particular product from the list or keep buying the products. After finishing buying the products, the customer will see all the products he/she bought along with the quantity of that product, total price of each product (price of a unit multiplied by the quantity of the product) and total cost of all the things he/she bought. After that customer can choose to keep shopping or place order . Customer will choose a payment method they prefer and a transaction id will be generated for that order .His/Her purchase information will be stored and a notification will be sent to the Admins so that they send an employee to deliver the products that had been bought. Each product quantity will decrease according to the amount that has been sold and it will be updated. If any product quantity is insufficient (that amount was enlisted by the Admin or employee for each product), the product will automatically be removed from online and a notification will be sent to the Admins so that they can take necessary steps.

**Delivery :**An employee will be sent to the customer address after one day of transaction . 24 hours will be given to customer in case they want to cancel their purchase . The employee will deliver the product to customer address and also get customer confirmation . It will be seen as transaction being complete and update database.

## Transaction

**Payment:**Customers can pay for the products with three methods . They are “Cash on Delivery” , “Debit Card” or “bCash” . Customer can choose any one of the options . After selecting payment method a order will be placed with a order number . Customers can cancel order with in time limit of 24 hours.

**Refund Policy :**Customer can click on refund policy icon to issue a refund . He/She has to give transaction ID of given product . The refund window is 72 hours after transaction completion .After that time the product can not be refunded . Customer has to state their reason for refund .A employee will be send to customer address to collect the product. A expert will than check the product thoroughly to verify the defect . If the problem matches customer description customer will get full refund as well as an apology . If not the product will be returned to the customer .

## Search

**Customer search:**To search for a customer , their email ID or phone number has to be given . Then the admin can view all the information of the customer , as well as their past activity and transaction . If user is an Admin he/she can search Admin information, employee information and customer information. If user is a customer, he/she cannot search another customer information but

only Admin or employee basic information. An employee can search for customer information .

**Product search:** Purpose of this part is to show a product is exist or not in the shop. All kind of user can search product. To search product user has to give product name (can be substring). If product name exists, all stored information of entered product will show. After that, product information can be searched by using product id. If productid is valid, all stored information of entered product id information will show.

**Transactional search:** Only owner and employee can perform transactional search. For every transaction type, there will be a search option. To search any transaction, user has to give transactionID. If the transaction id is valid, all stored information of the entered transaction id will be shown.

# Chapter 4

## Scenario Based Modeling

This chapter describes the scenario based model for the Grocery Management System.

### 4.1 Introduction

Although the success of a computer-based system or product is measured in many ways, user satisfaction resides at the top of the list. If the software developer team understands how end users (and other actors) want to interact with a system, they will be better able to properly characterize requirements and build meaningful analysis and design models. Hence, requirements modeling begins with the creation of scenarios in the form of Use Cases, activity diagrams and swim lane diagrams.

### 4.2 Definition of UseCase

A Use Case captures a contract that describes the system behavior under various conditions as the system responds to a request from one of its stakeholders. In essence, a Use Case tells a stylized story about how an end user interacts with the system under a specific set of circumstances. A Use Case diagram simply describe satisfactory using corresponding actors who perform important roles in the story and makes the story understandable for the users.

The first step in writing a Use Case is to define that set of "actors" that will be involved in the story. Actors are the different people that use the system or product within the context of the function and behavior that is to be described. Actors represent the roles that people play as the system

operators. Every user has one or more goals when using system.

### **Primary Actor**

Primary actors interact directly to achieve required system function and derive the intended benefit from the system. They work directly and frequently with the software.

### **Secondary Actor**

Secondary actors support the system so that primary actors can do their work. They either produce or consume information.

#### 1. Scenario Based Model

##### Use Case Scenario

Level 0	Level 1	Level 2	Level 3	Level 4	Actors
FCMS	Account Creation	Open account Verification Sign in Recovery Sign out			Primary Actor: Customer, Admin Secondary Actor : Admin

			Delete Account		
			Communicate	Customer to employee	
				Employee to customer	
Profit Strategy Management	Customer Data Analysis	Customer cookie analysis		Primary Actor : Admin	
		Customer History analysis			
		Product Data Analysis			
Delivery order and management	Customer Order			Primary Actor : Customer, Admin , Employee	
	Delivery				
Search	Customer Search			Primary Actor : Customer, Admin , Employee	
	Product Search				
	Transactional Search				
Transaction	Payment System			Primary Actor : Customer, Admin , Employee	
	Refund Policy				

2. Use Case Description:

### **1.1. Account Creation**

Account Creation is divided into five subsystems

#### **1.1.1. Open Account**

**Use Case:** Open Account

**Primary Actor:** Customer , Admin , Employee

**Secondary Actor:** Admin

**Goals in Context:** To open an account

**Preconditions:**

1. System has been designed for opening an account
2. System has interface for opening account

**Triggers:** Each customer , employee and admin has to open an account

**Scenario:**

1. Visit Page and initiate open account
2. Provide name, email address, facebook id, mobile number, date of birth , user name and password.
3. Information verified by an delegated administrator
4. Confirmation message sent via email or SMS

**Exception:**

1. Ambiguous input
2. Same registration number or employee number used more than once

**Priority:** Essential, Must be implemented

**When Available:** First Increment

#### **1.1.2. Verification**

**Use Case:** Verification

**Primary Actor:** Admin

**Goals in Context:** To verify given information provided by customers and employees

**Preconditions:**

1. System has been designed for verification
2. A delegated administrator should be set for verifying information

3. User has provided information

**Triggers:** Administrator needs to verify the information

**Scenario:**

1. Customer or Employee has provided information
2. The information is verified by the admin
3. An auto generated email or SMS has been sent after successful verification

**Exception:**

1. Wrong or false information provided
2. SMS or email service temporarily unavailable

**Priority:** Essential, Must be implemented

**When Available:** First increment

#### **1.1.3. Sign in**

**Use Case:** Sign in

**Primary Actor:** Customer , Admin , Employee

**Secondary Actor:** Admin

**Goals in Context:** To sign in to the system

**Preconditions:**

3. System has been designed for signing in an account
4. System has interface for opening account

**Triggers:** Each customer , employee and admin has to sign or log in an account

**Scenario:**

5. Visit Page and sign in to access into the account
6. Provide user name and password.

**Exception:**

1. Ambiguous input
2. Same registration number or employee number used more than once

**Priority:** Essential, Must be implemented

**When Available:** First Increment

#### **1.1.4. Recovery**

**Use Case:** Recovery

**Primary Actor:** Customer , Admin , Employee

**Secondary Actor:** Admin

**Goals in Context:** To recover an account

**Preconditions:**

1. System has been designed for recovering an account
5. System has recovered for opening account

**Triggers:** Each customer , employee and admin can recover their account

**Scenario:**

1. Recover account
2. Provide email address/ mobile number
3. A code will be sent via email or SMS

**Exception:**

4. Ambiguous input
5. Same registration number or employee number used more than once

**Priority:** Essential, Must be implemented

**When Available:** First Increment

### 1.1.5. Sign out

**Use Case:** Sign out

**Primary Actor:** Customer , Admin , Employee

**Secondary Actor:** Admin

**Goals in Context:** To sign out

**Preconditions:**

1. System has been designed for signing out
2. System has interface for signing out

**Triggers:** Each customer , employee and admin can sign out after their work

**Scenario:**

1. Visit Page
2. After finishing their task, account owner can sign out

**Exception:**

3. Ambiguous input

- After 30 minutes of signing in , the system will be signed out the account automatically

**Priority:** Essential, Must be implemented

**When Available:** First Increment

## **1.2. Administrative Functionalities**

Administrative Functionalities has two subsystems

### **1.2.1. Product Management**

Product management is divided into five subsystems

#### **1.2.1.1. Add item**

**Use Case:** Add item

**Primary Actor:** Admin , Employee

**Goals in Context:** Admin or employee can item

**Precondition:**

- System has been designed for adding documents
- Admin/Employee is logged in
- Admin/Employee has some document to upload

**Trigger:** Admin/Employee has something to add item

**Scenario:**

- Admin finishes login
- Admin or employee provides information of her document (Title of the work, Type of the work, Synopsis, Keywords, Date)
- Document must be in jpeg or png format and the file size will be checked
- If an employee wants to upload document, it must be verified by admin
- An admin does not have to verify her document

**Exception:**

- File is not in jpeg or png format
- File size exceeds
- Authentication failure
- Provided information is not correct

**Priority:** Essential, Must be implemented

**When Available:** First Increment

### **1.2.1.2. Edit item**

**Use Case:** Edit item

**Primary Actor:** Admin , Employee

**Goals in Context:** To edit products

**Preconditions:**

1. System has been designed for editing product details
2. System has been designed for editing
3. Has a prepared document

**Triggers:** Admin or employee can edit

**Scenario:**

1. Admin or employee has to log in
2. Admin can edit
3. Admin or employee can edit documents

**Priority:** Essential, Must be implemented

**When Available:** First increment

### **1.2.1.3. Add promotion**

**Use Case:** Add promotion

**Primary Actor:** Admin , Employee

**Goals in Context:** To add promotion

**Preconditions:**

4. System has been designed for adding promotion
5. System has been designed for adding promotion
6. Has a prepared document

**Triggers:** User has something to add promotion

**Scenario:**

4. Admin can add promotion
5. Employee has something to add
6. Admin or employee can add promotion

**Priority:** Essential for business, Must be implemented

**When Available:** First increment

#### **1.2.1.4. Stock Update**

**Use Case:** Stock Update

**Primary Actor:** Admin , Employee

**Goals in Context:** To stock update

**Preconditions:**

7. System has been designed to stock update
8. System can access to user information
9. Has a prepared document

**Triggers:** Admin has something to update the stock

**Scenario:**

7. Is a user
8. System can update the stock
9. Admin or employee can update the stock

**Priority:** Essential, Must be implemented

**When Available:** First increment

#### **1.2.2. Customer Management**

Customer management has three subsystems

##### **1.2.2.1. Block Customer**

**Use Case:** Block Customer

**Primary Actor:** Admin , Employee

**Goals in Context:** To block customer

**Preconditions:**

10. System has been designed for blocking customer
11. Admin or employee can block customer

**Triggers:** Admin or Employee can block customer if they want

**Scenario:**

10. Is a user
11. User can be blocked by admin

**Priority:** Essential, Must be implemented

**When Available:** First increment

### **1.2.2.2. Delete Account**

**Use Case:** Delete Account

**Primary Actor:** Admin , Employee , Customer

**Goals in Context:** To delete account

**Preconditions:**

12. System has been designed for deleting account
13. For deleting an account, admin permission must be needed

**Triggers:** Anyone can delete their own account

**Scenario:**

12. Can be customer, admin or employee
13. Anyone can delete their own account

**Priority:** Essential, Must be implemented

**When Available:** First increment

### **1.2.2.3. Communicate**

Communicate has two subsystems

#### **1.2.2.3.1 Customer to Employee**

**Use Case:** Customer to Employee

**Primary Actor:** Customer , Employee

**Goals in Context:** To communicate with employee from customer

**Preconditions:**

14. System has been designed for communicating with employee
15. System has been designed for communication

**Triggers:** Customer has to communicate

**Scenario:**

14. Is a customer
15. Customer has to communicate
16. Customer can communicate with employee

**Priority:** Essential, Must be implemented

**When Available:** First increment

### **1.2.2.3.2 Employee to Customer**

**Use Case:** Employee to Customer

**Primary Actor:** Customer , Employee

**Goals in Context:** To communicate with customer from employee

**Preconditions:**

16. System has been designed for communicating with customer
17. System has been designed for communication

**Triggers:** Employee has to communicate

**Scenario:**

17. Is a employee
18. Employee has to communicate
19. Employee can communicate with customer

**Priority:** Essential, Must be implemented

**When Available:** First increment

## **1.3. Profit Strategy Management**

Profit Strategy Management has three subsystems

### **1.3.1. Customer Data Analysis**

Customer Data Analysis has two subsystems

#### **1.3.1.1. Customer cookie analysis**

**Use Case:** Customer cookie analysis

**Primary Actor:** Customer , Employee

**Goals in Context:** To analysis customer cookie

**Preconditions:**

18. System has been designed for analyzing customer cookie

**Triggers:** System can analyze customer cookie

**Scenario:**

1. Is a customer
2. System can show the products that customer can choose

**Priority:** It can be a wow factor, should be implemented

**When Available:** First increment

### **1.3.1.2. Customer history analysis**

**Use Case:** Customer history analysis

**Primary Actor:** Customer , System

**Goals in Context:** To analyze customer history

**Preconditions:**

3. System has been designed for analysis

**Triggers:** System can analyze customer cookie

**Scenario:**

1. Is a system
2. It can be helped for showing useful products

**Priority:** Essential, Must be implemented

**When Available:** First increment

### **1.3.2. Product Data analysis**

**Use Case:** Product Data analysis

**Primary Actor:** System , Admin , Employee

**Goals in Context:** To analyze product data

**Preconditions:**

3. System has been designed for analyzing product data
4. It can be helpful for finding out productive products

**Triggers:** System can analyze product data

**Scenario:**

5. Is a system
6. System can analyze the product data

**Priority:** Essential, Must be implemented

**When Available:** First increment

## **1.4. Delivery order and management**

Delivery order and management has two subsystems

#### **1.4.1. Customer Order**

**Use Case:** Customer Order

**Primary Actor:** Customer ,Employee

**Goals in Context:** To order from customer

**Preconditions:**

7. System has been designed for taking order from customer

**Triggers:** Customer has something to order

**Scenario:**

8. Is a customer
9. Customer has something to order
10. Any type of customer can order

**Priority:** Essential, Must be implemented

**When Available:** First increment

#### **1.4.2. Delivery**

**Use Case:** Delivery

**Primary Actor:** Employee

**Goals in Context:** To delivery orders

**Preconditions:**

11. System has been designed for delivery
12. Employee will delivery products according to order

**Triggers:** Employee will delivery the ordered products

**Scenario:**

13. Is an employee
14. Employee has to delivery

**Priority:** Essential, Must be implemented

**When Available:** First increment

#### **1.5. Search**

Search has three subsystems

### **1.5.1. Customer Search**

**Use Case:** Customer Search

**Primary Actor:** Customer , Admin , Employee

**Goals in Context:** To search for customers

**Precondition:**

1. System has been designed for searching
2. Admin uses search box to search

**Trigger:** Admin has customer to search

**Scenario:**

1. Admin or employee search for some products by giving a keyword
2. Searched customers are showed chronologically
3. Searched customers can be found in a range of time

**Exception:**

1. System is not ready
2. Keyword does not match with any of the stored document

**Priority:** Essential, Must be implemented

**When Available:** First Increment

### **1.5.2. Product Search**

**Use Case:** Product Search

**Primary Actor:** Customer , System

**Goals in Context:** To search for a product

**Precondition:**

3. System has been designed for searching
4. Customers uses search box to search

**Trigger:** Customers has something to search

**Scenario:**

4. Customers search for some document by giving a keyword
5. Searched products are showed chronologically
6. Searched product can be found in a range of time

**Exception:**

3. System is not ready
4. Keyword does not match with any of the stored document

**Priority:** Essential, Must be implemented

**When Available:** First Increment

**1.5.3. Transactional Search**

**Use Case:** Transactional Search

**Primary Actor:** Admin , Employee

**Goals in Context:** To search for transaction

**Precondition:**

5. System has been designed for searching
6. Admin uses search box to search

**Trigger:** Admin can search customer's transaction

**Scenario:**

7. Admin search for some transaction by giving a keyword
8. Searched transactions are showed chronologically
9. Searched transaction can be found in a range of time

**Exception:**

5. System is not ready
6. Keyword does not match with any of the stored document

**Priority:** Essential, Must be implemented

**When Available:** First Increment

**1.6. Transaction**

Transaction has two subsystems

**1.6.1. Payment System**

**Use Case:** Payment System

**Primary Actor:** Customer , System

**Goals in Context:** To pay

**Preconditions:**

15. System has been designed for various payment system

**Triggers:** Customer has various ways to pay

**Scenario:**

16. Is a customer

17. :Customer has various ways to pay

**Priority:** Essential, Must be implemented

**When Available:** First increment

### **1.6.2. Refund Policy**

**Use Case:** Refund Policy

**Primary Actor:** Customer, Employee

**Goals in Context:** To refund

**Preconditions:**

18. System has been designed for refunding

**Triggers:** Customer has something to refund

**Scenario:**

19. Is a customer

20. Customer has something to refund

**Priority:** Essential, Must be implemented

**When Available:** First increment

4. Use Case Diagram

## Level 0

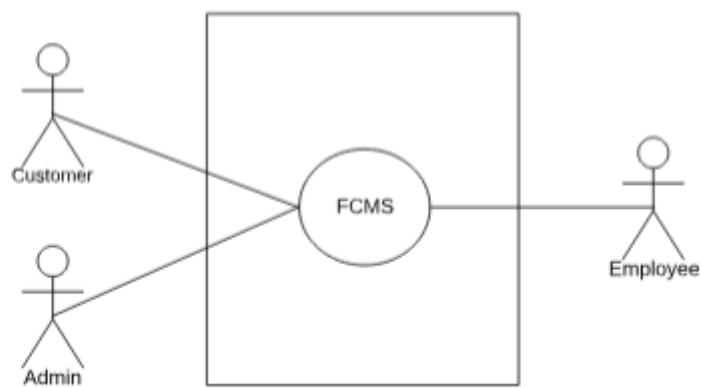
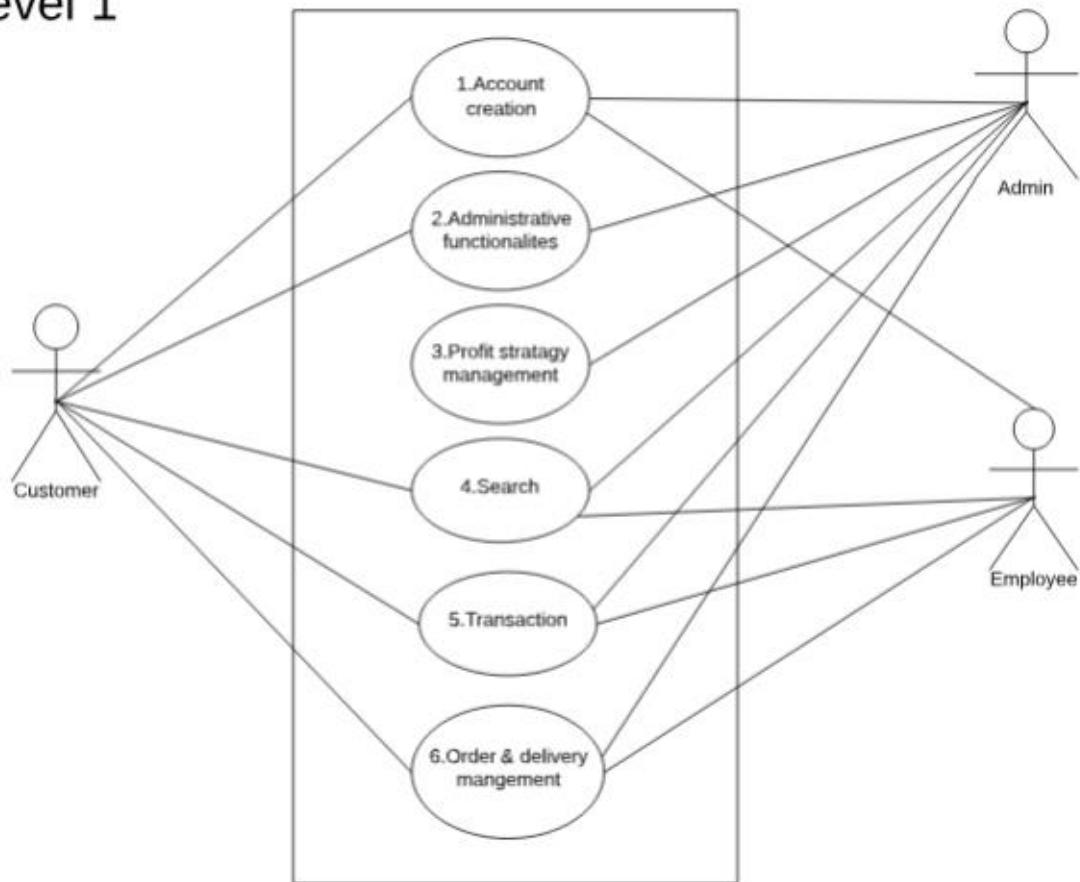


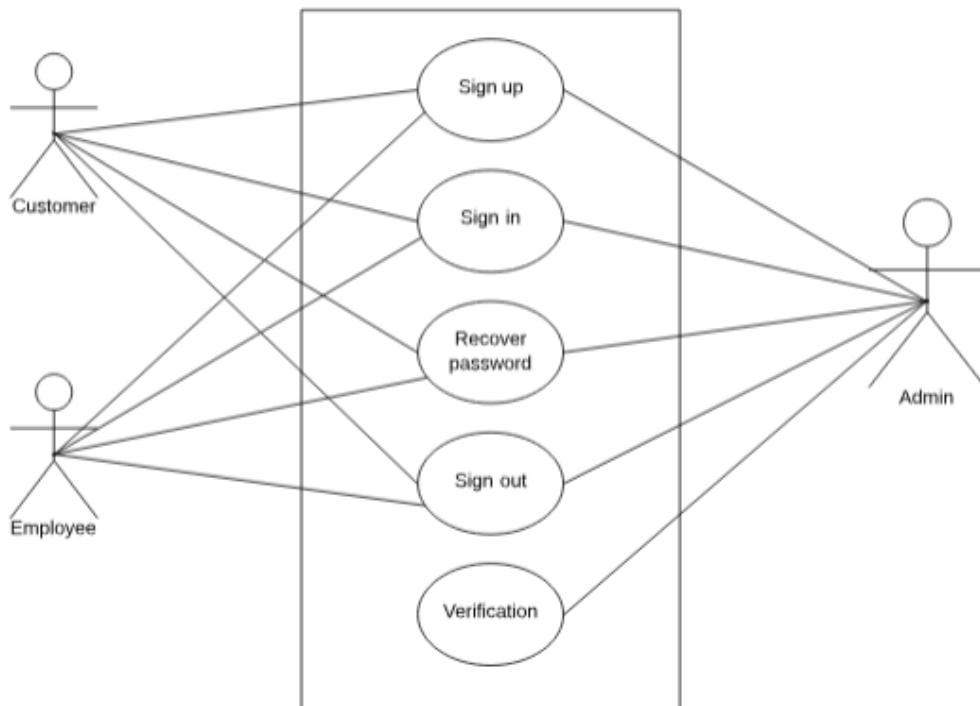
Fig 1: level 0 for FCMS

## Level 1



---

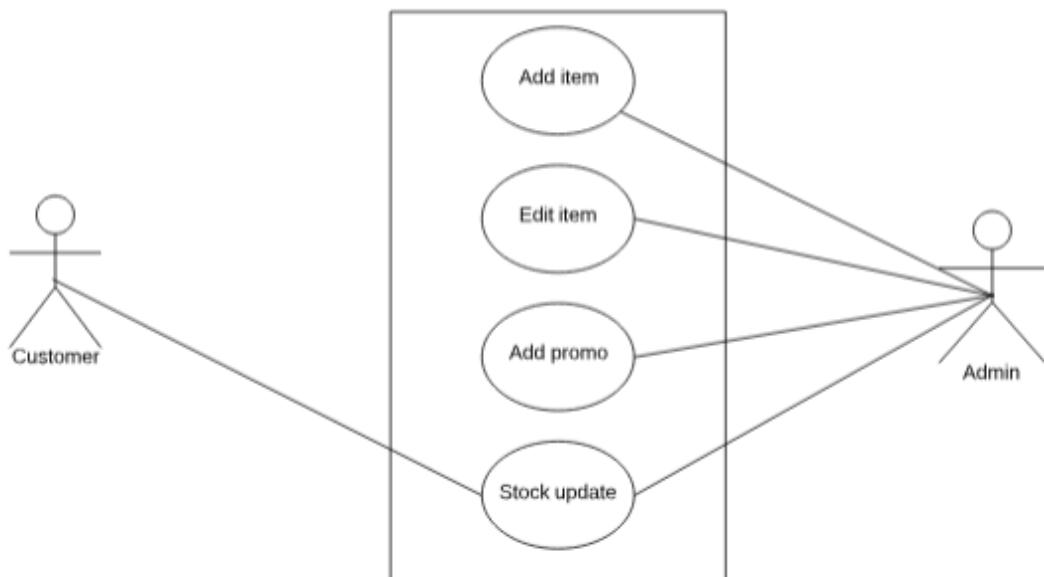
## Level 2.1



---

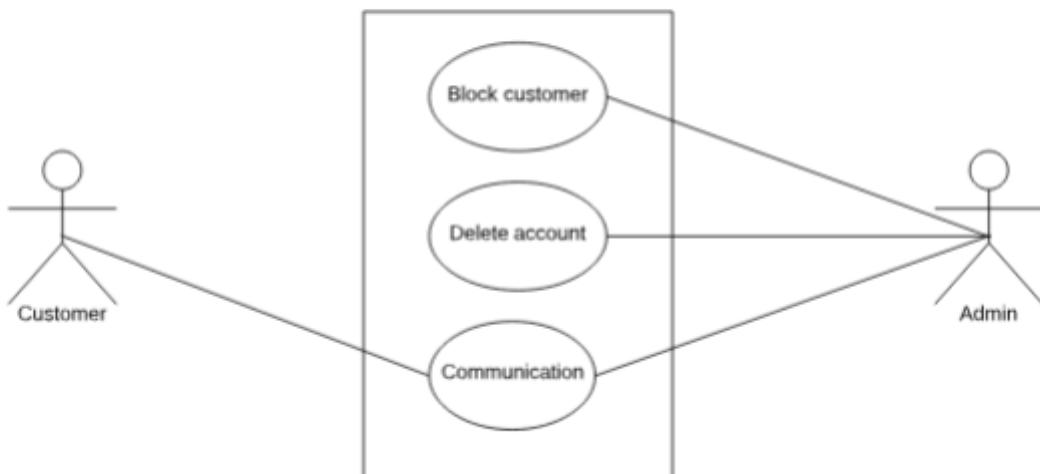
## Level 2.2.1

(Product management)



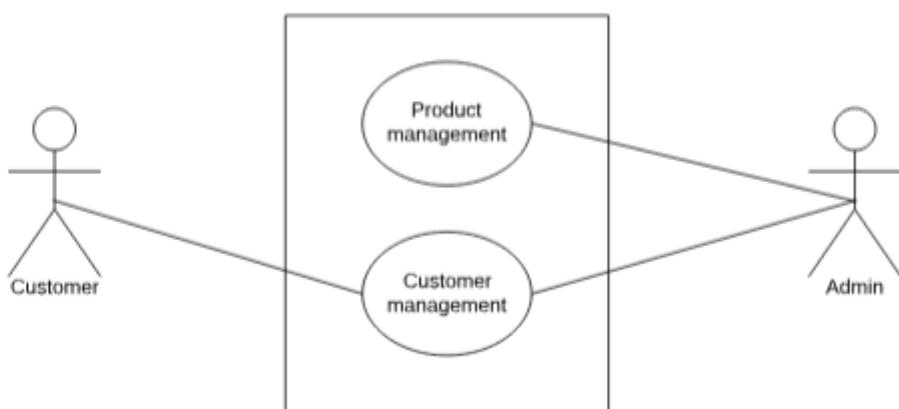
## Level 2.2.2

(Customer management)



## Level 2.2

(Administrative functionality)



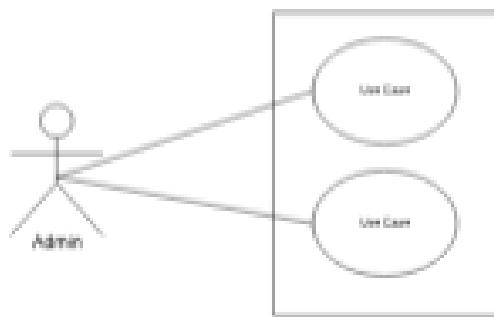
## Level 2.3.1

(Customer Data Analysis)



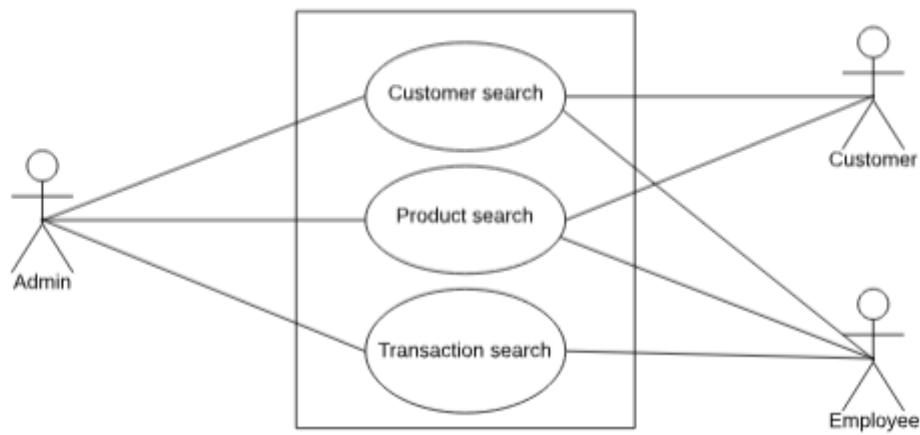
## Level 2.3

(Priority management)



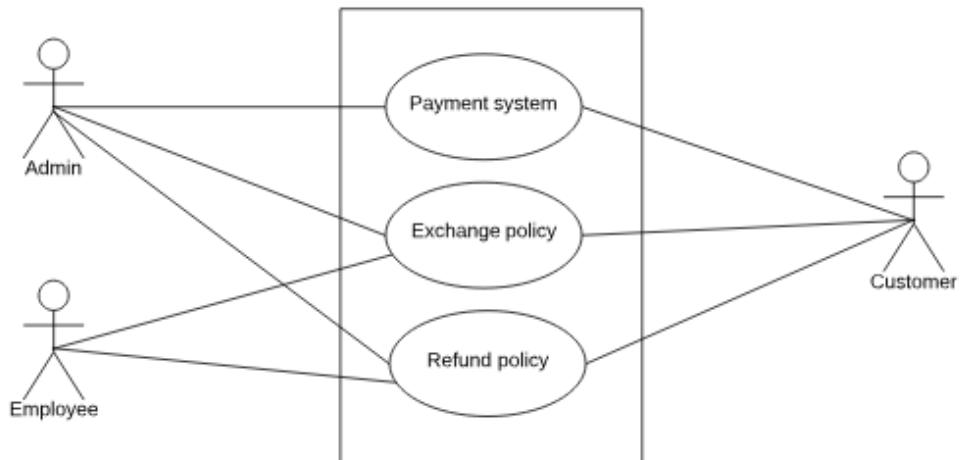
## Level 2.4

(Search)



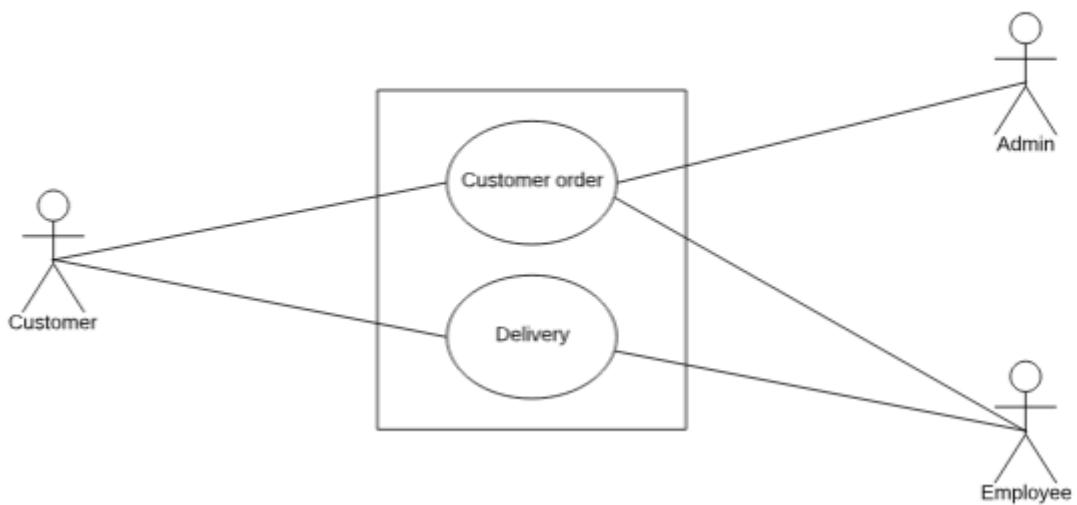
## Level 2.5

(Transaction)



## Level 2.6

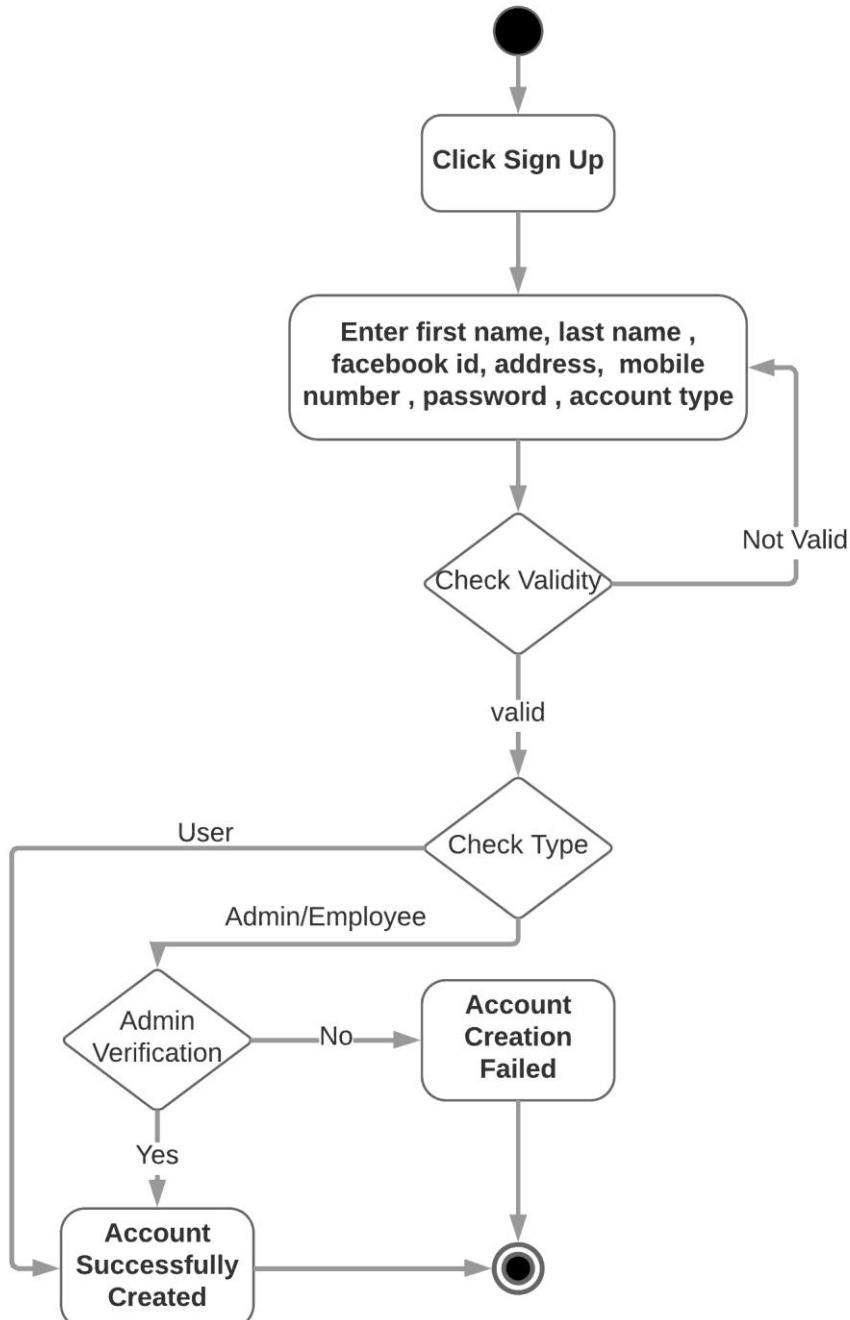
(Order and delivery)



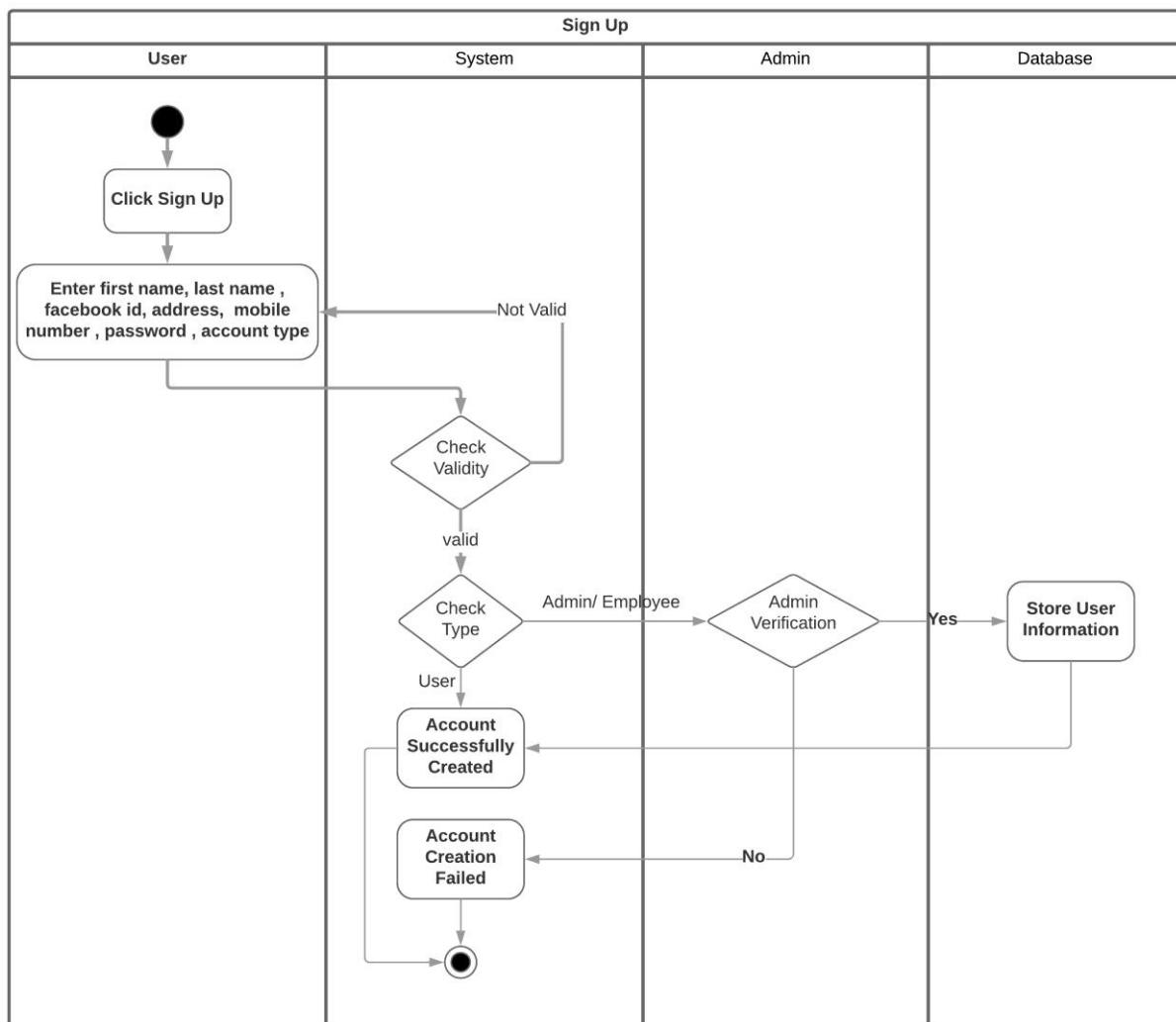
#### 4. Activity Diagram and Swimlane Diagram

##### Use Case 1: Open Account

Activity Diagram:

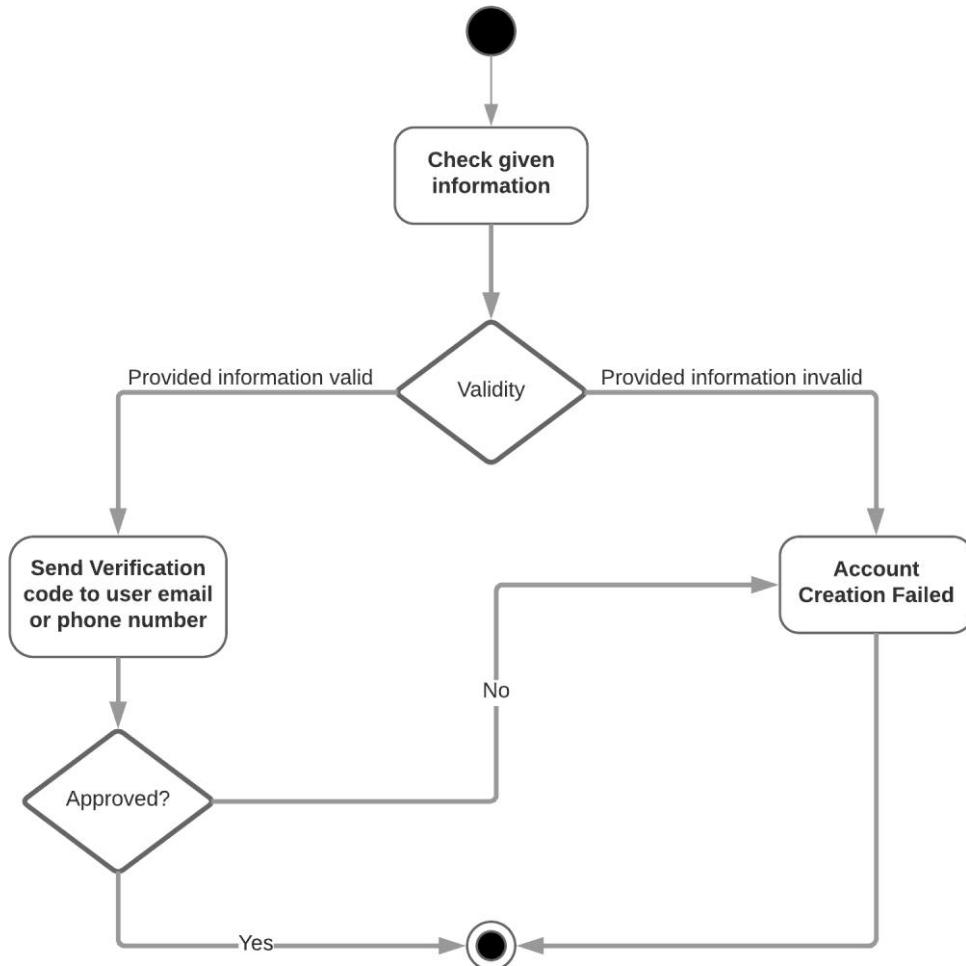


## Swimlane Diagram:

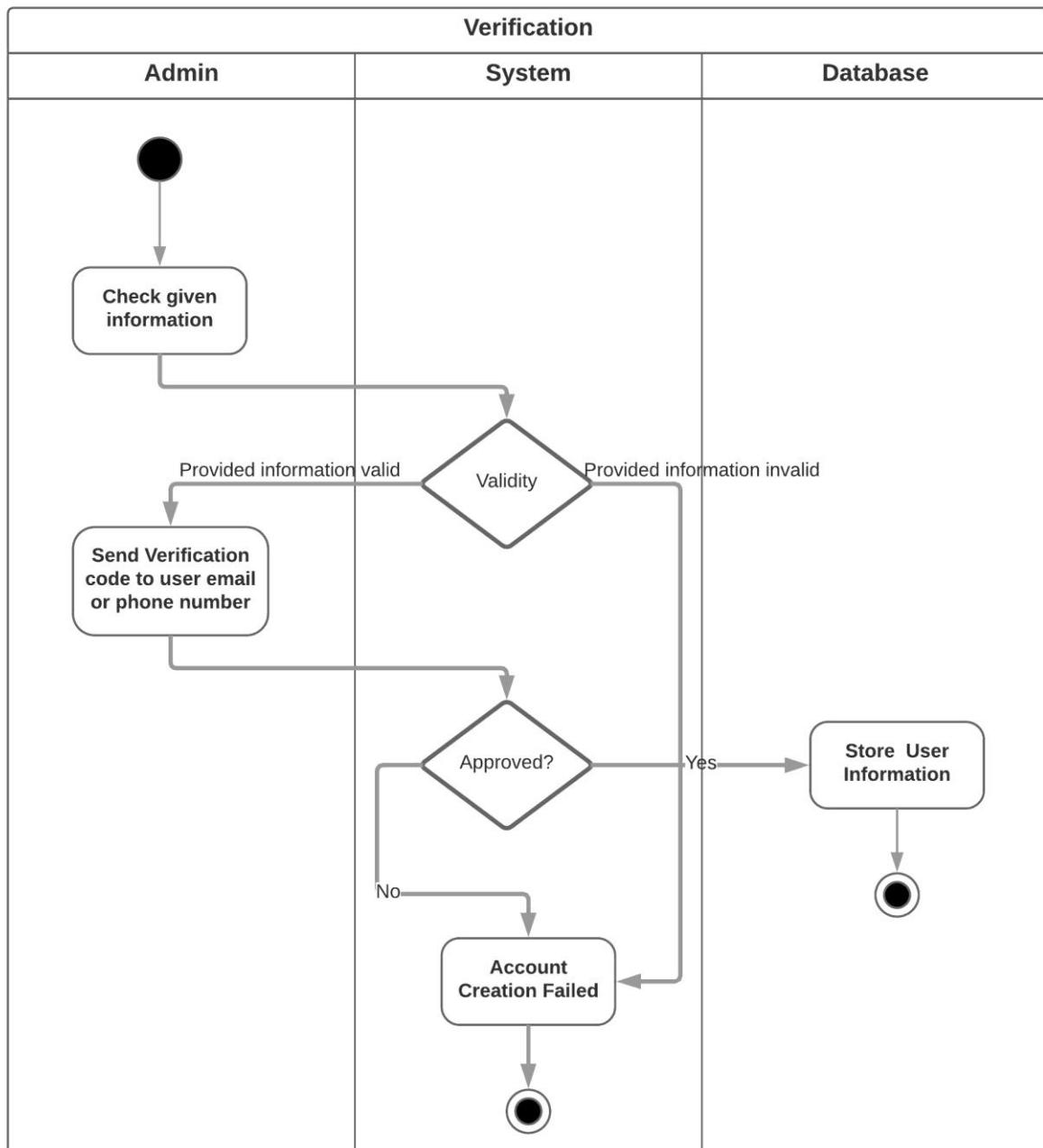


## Use Case 2: Verification

Activity Diagram:

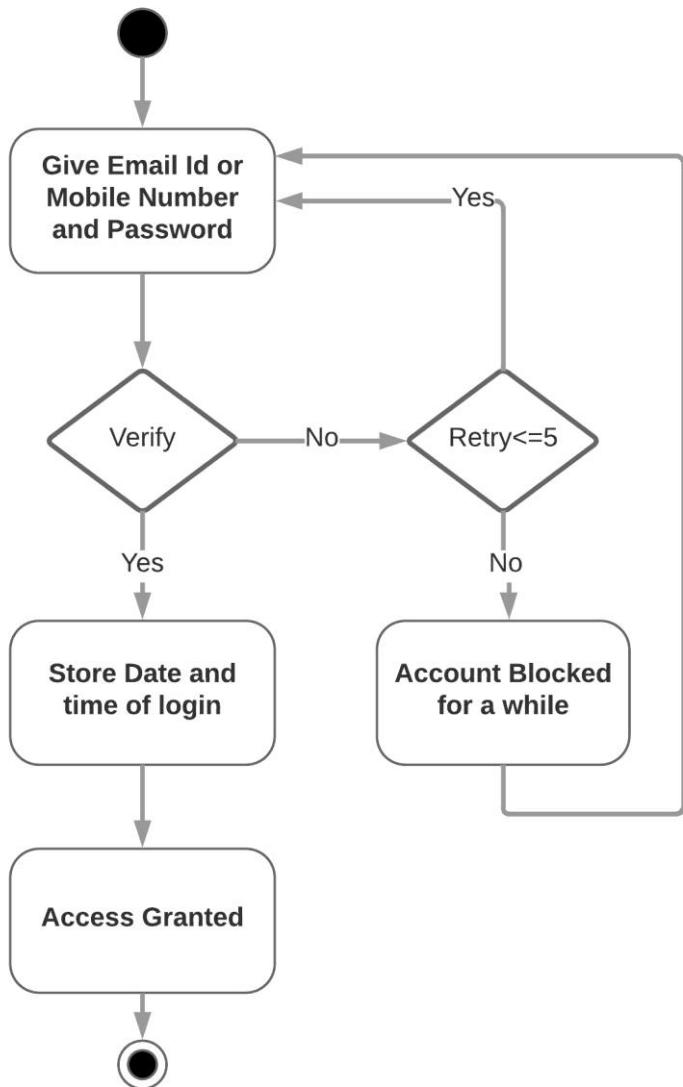


Swimlane Diagram:

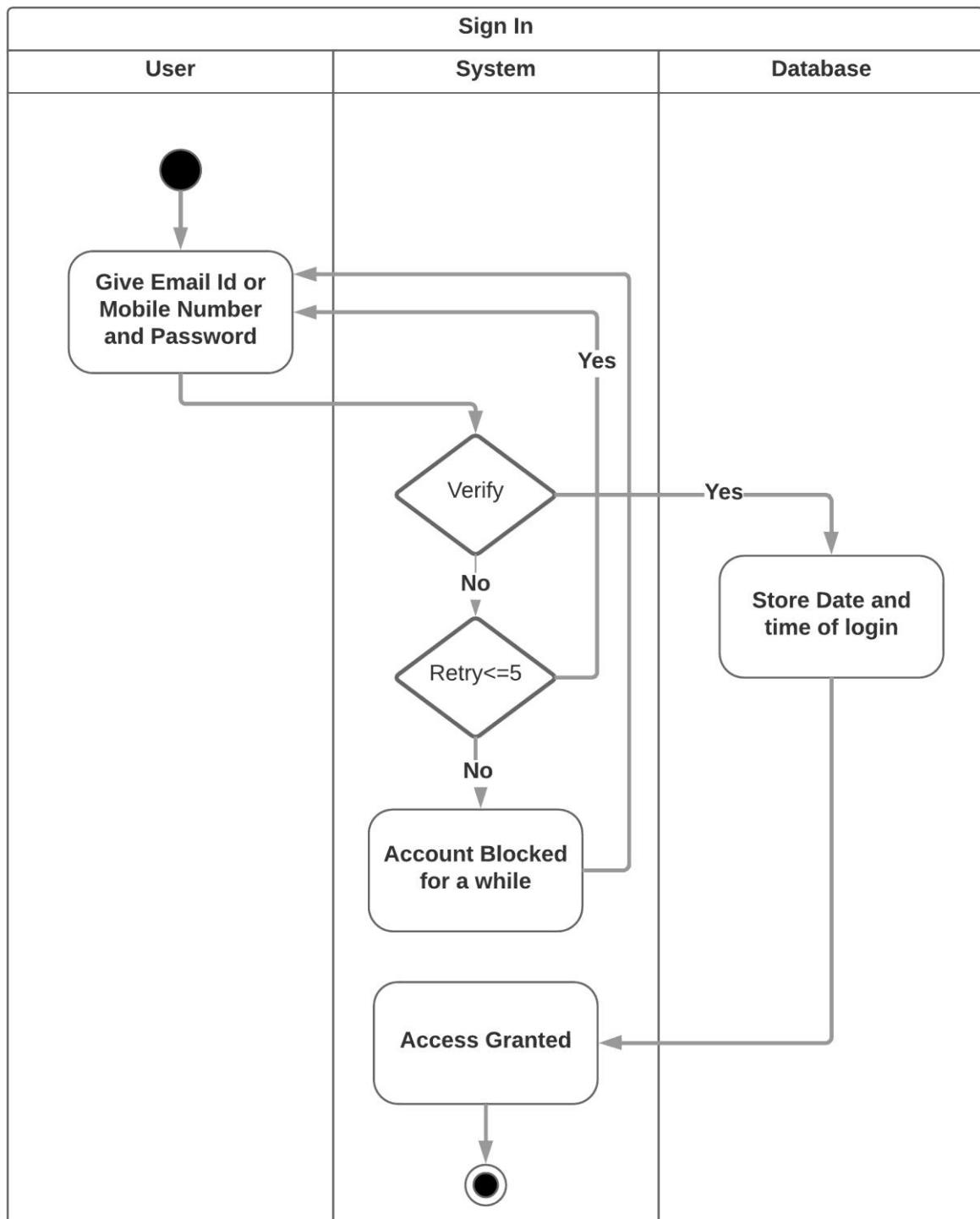


### Use Case 3: Sign in

Activity Diagram:

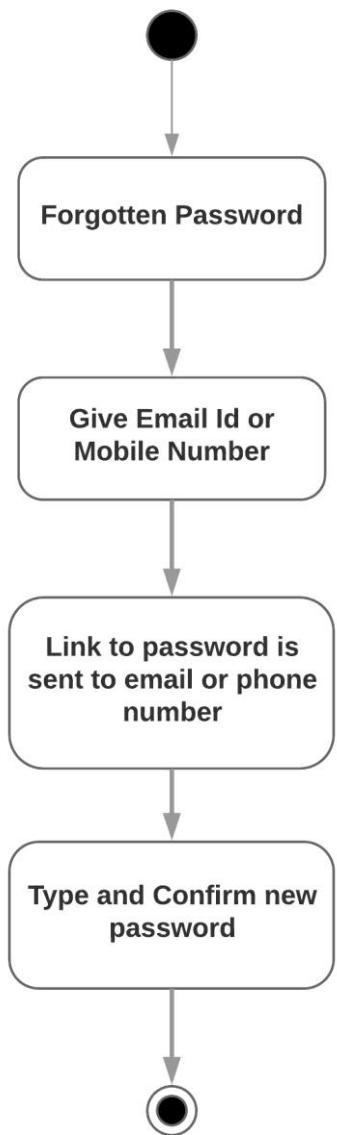


Swimlane Diagram:

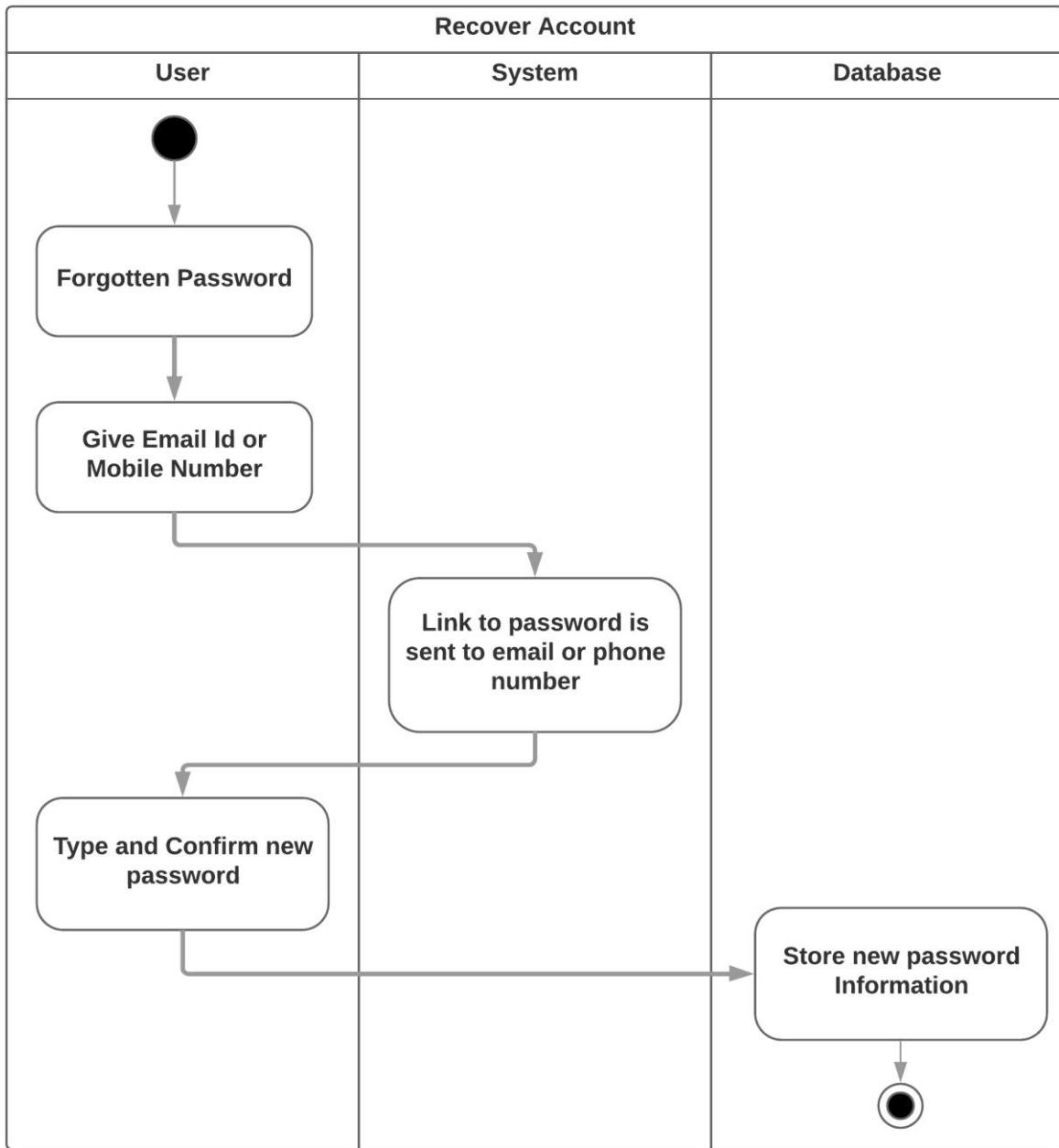


## Use Case 4: Recovery

Activity Diagram:

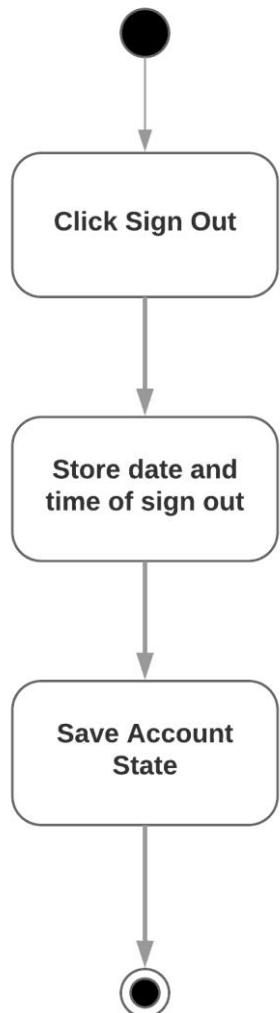


Swimlane Diagram:

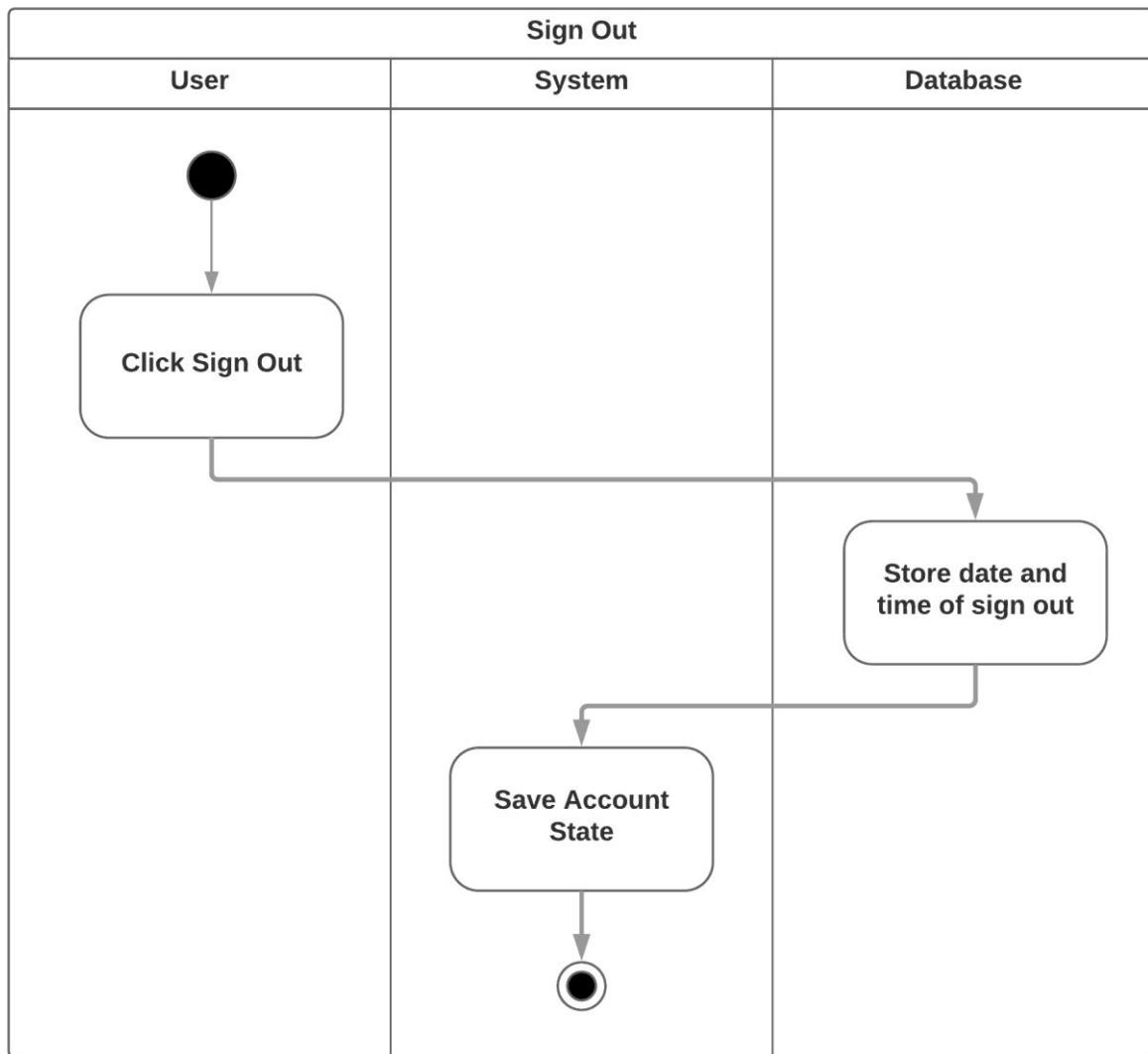


**Use Case 5: Sign out**

Activity Diagram:



Swimlane Diagram:

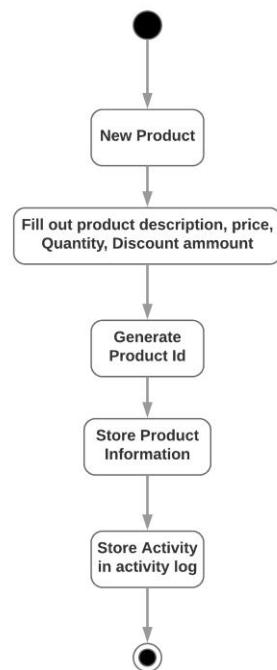


## Use Case 6: Add item

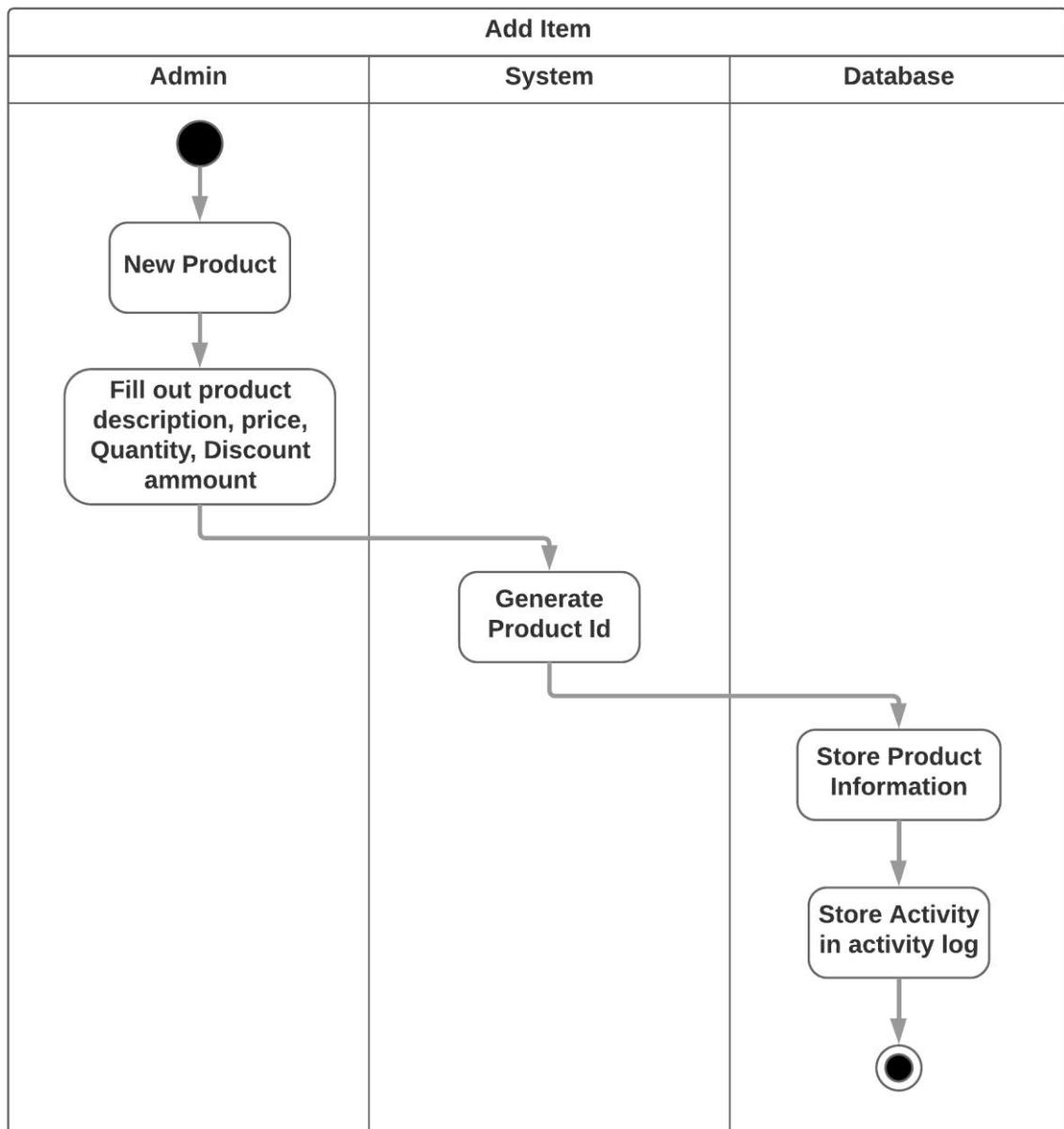
Activity Diagram:

ADD ITEM

Pritom | November 9, 2018

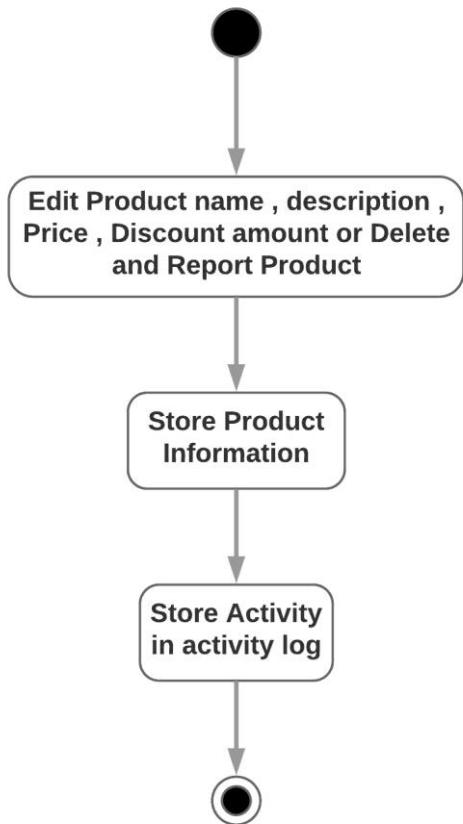


Swimlane Diagram:

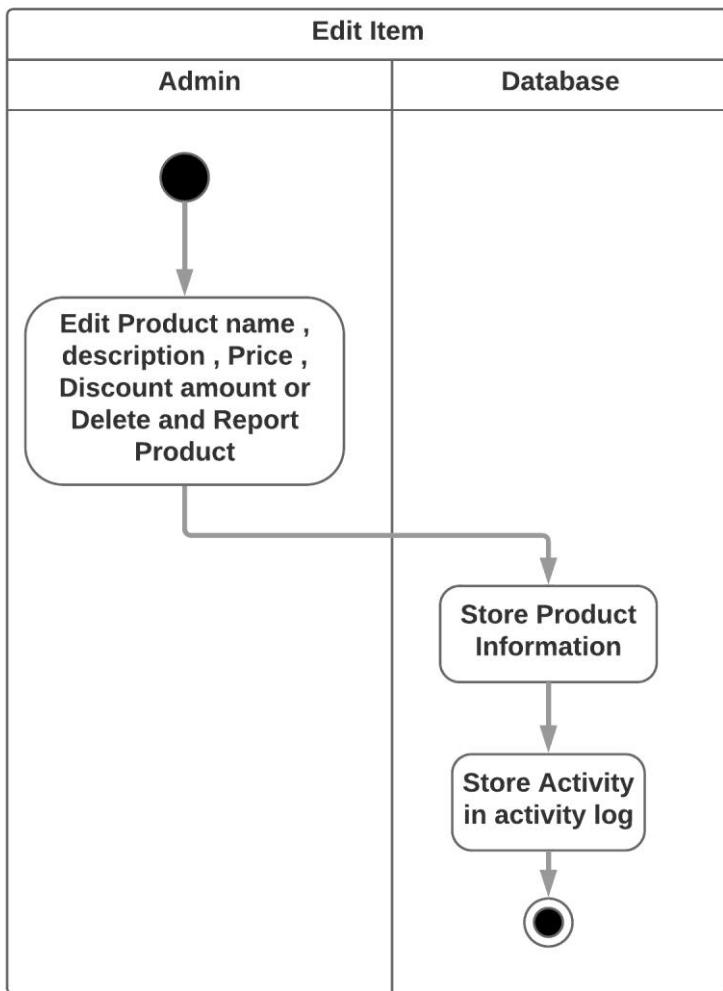


### **Use Case 7: Edit item**

Activity Diagram:

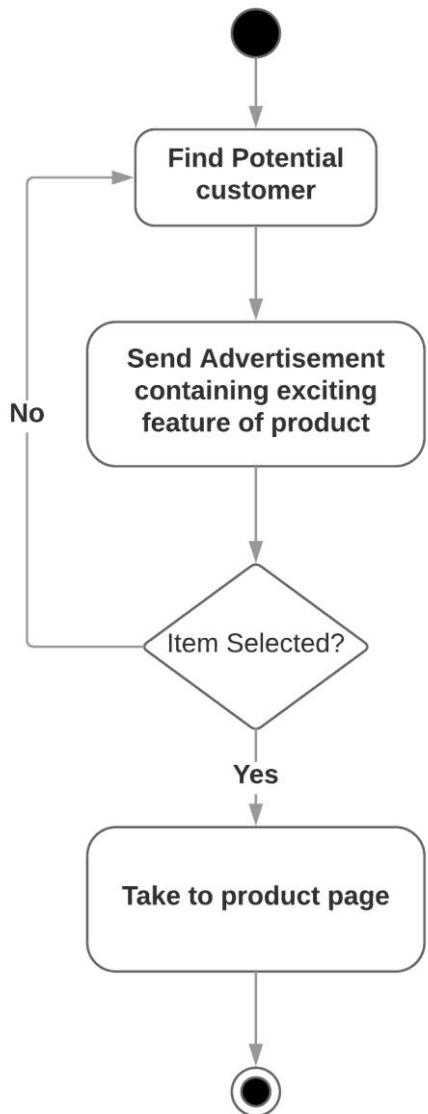


Swimlane Diagram:

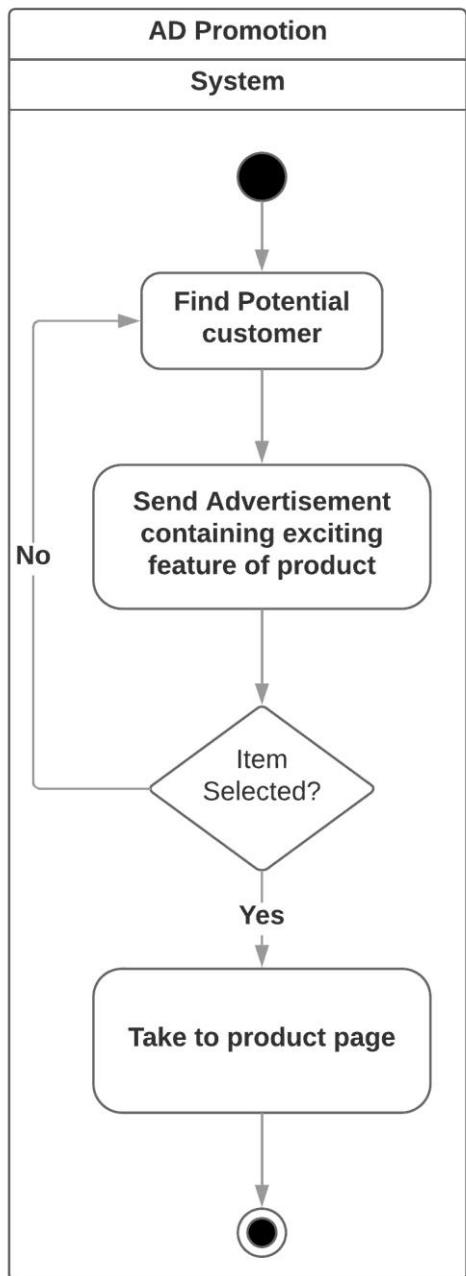


## Use Case 8: Add promotion

Activity Diagram:

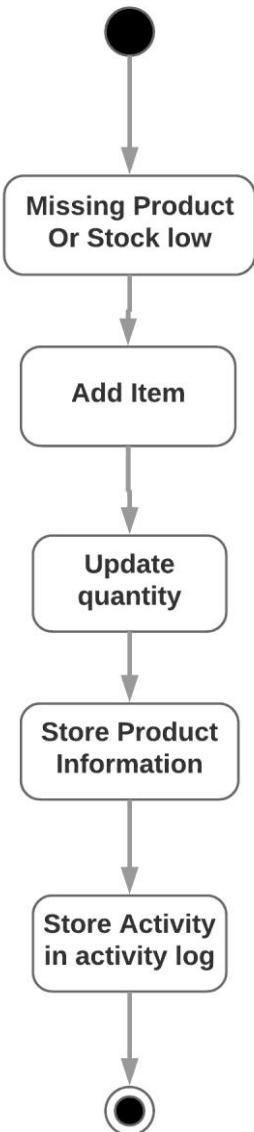


Swimlane Diagram:

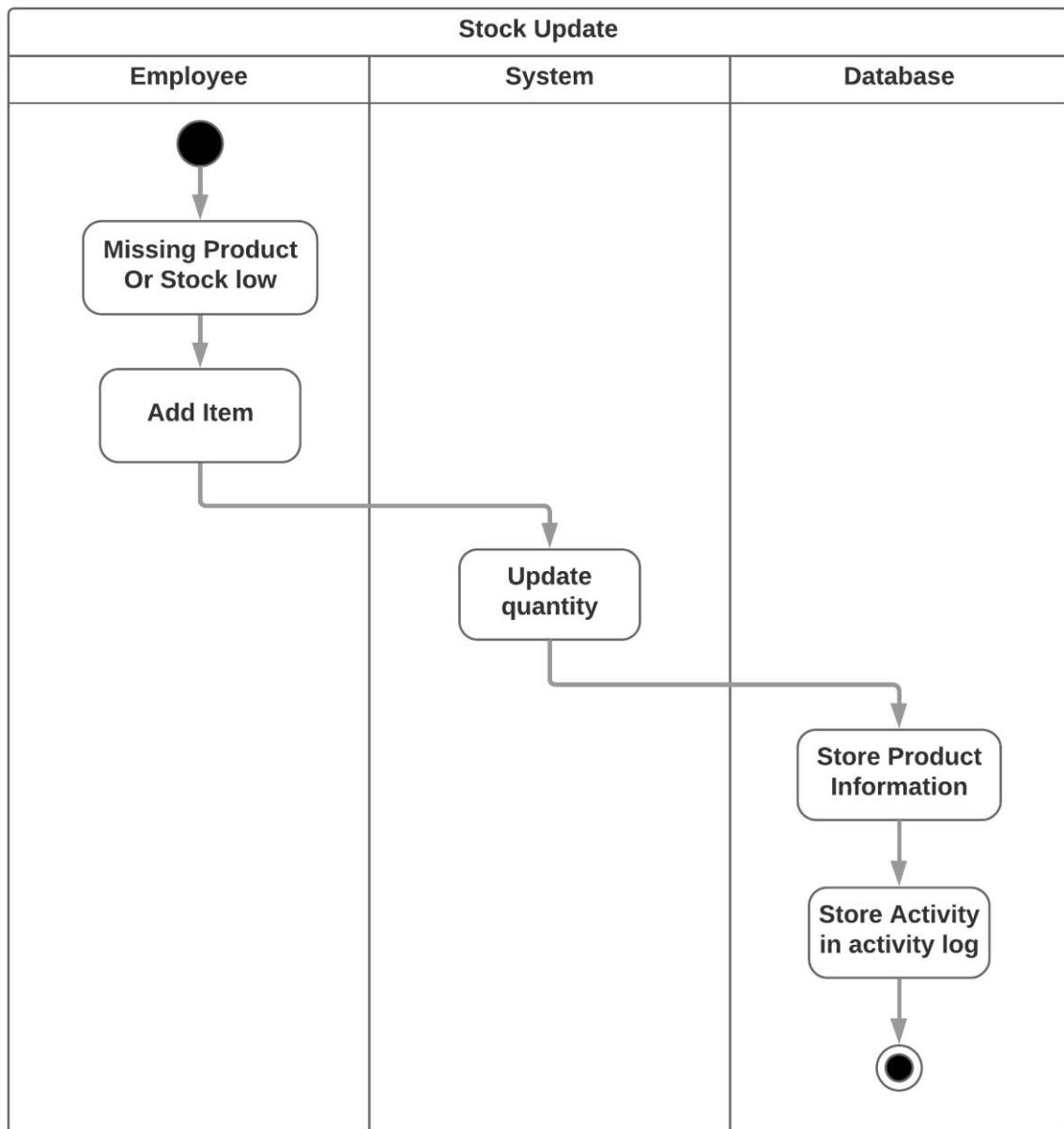


## Use Case 9: Stock update

Activity Diagram:

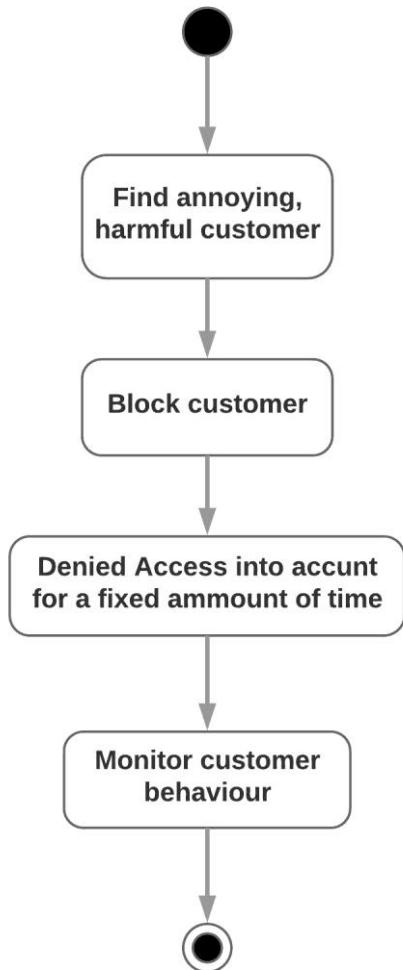


Swimlane Diagram:

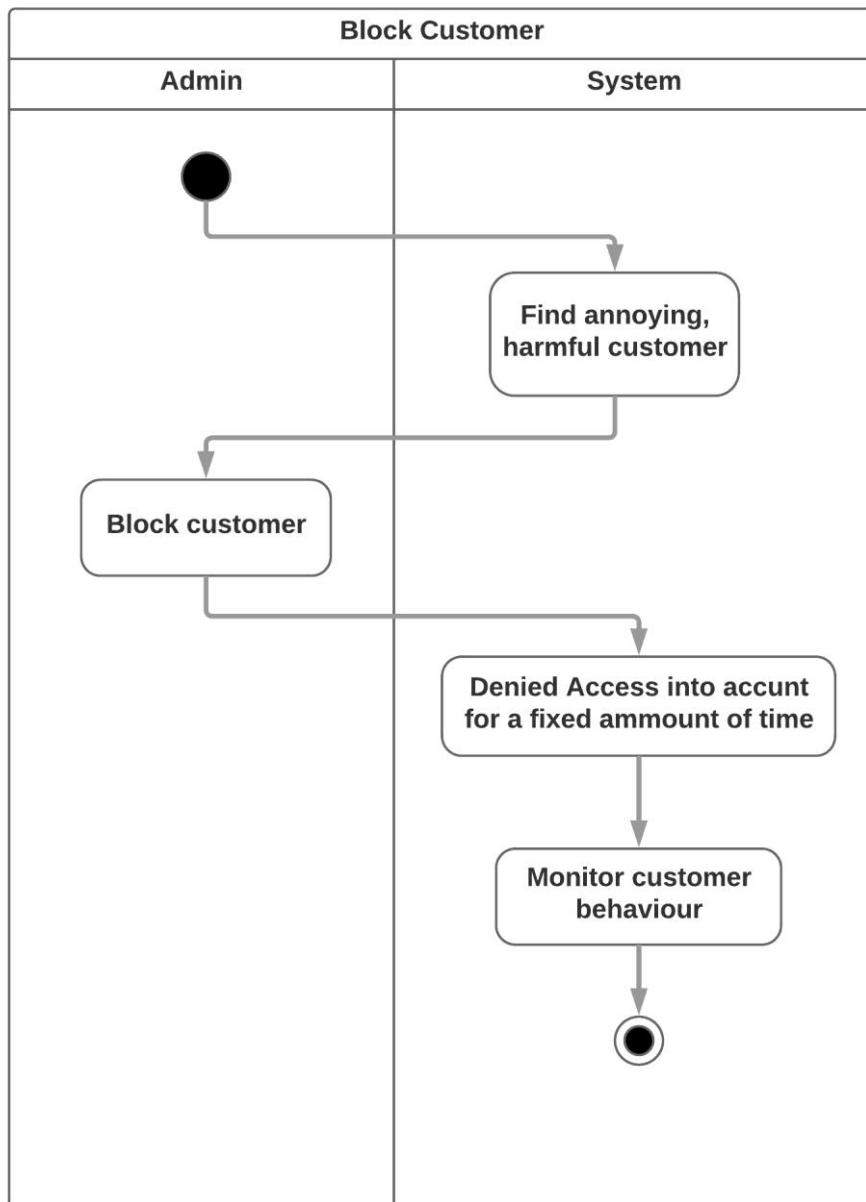


## Use Case 10: Block customer

Activity Diagram:

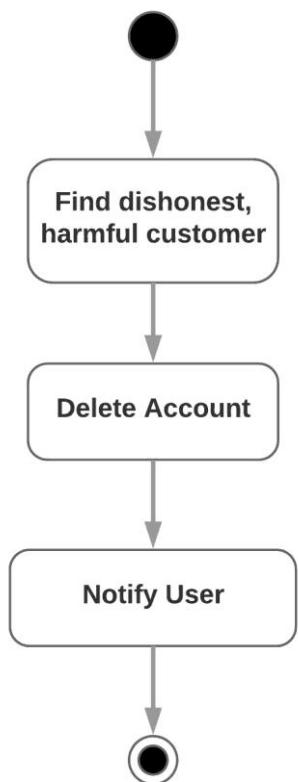


Swimlane Diagram:

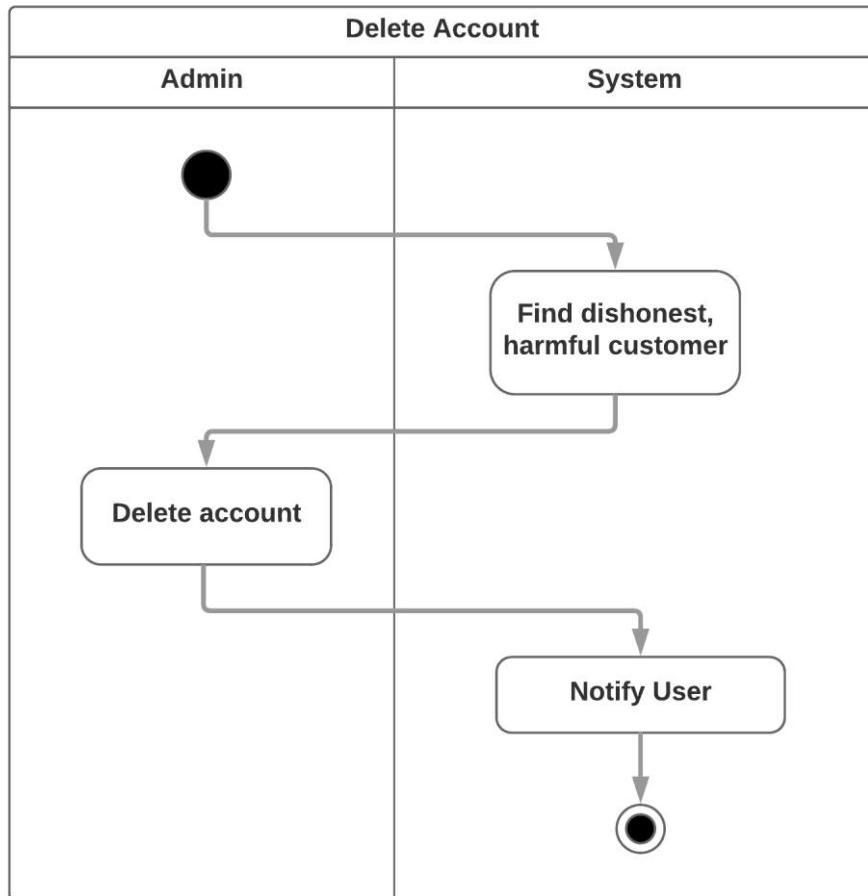


## Use Case 11: Delete Account

Activity Diagram:

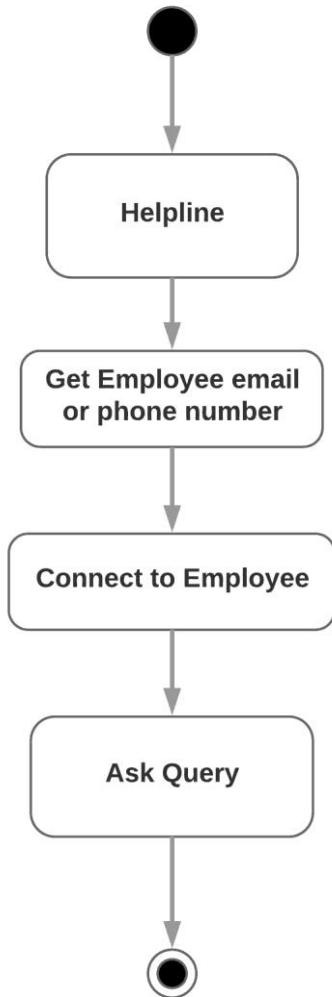


Swimlane Diagram:

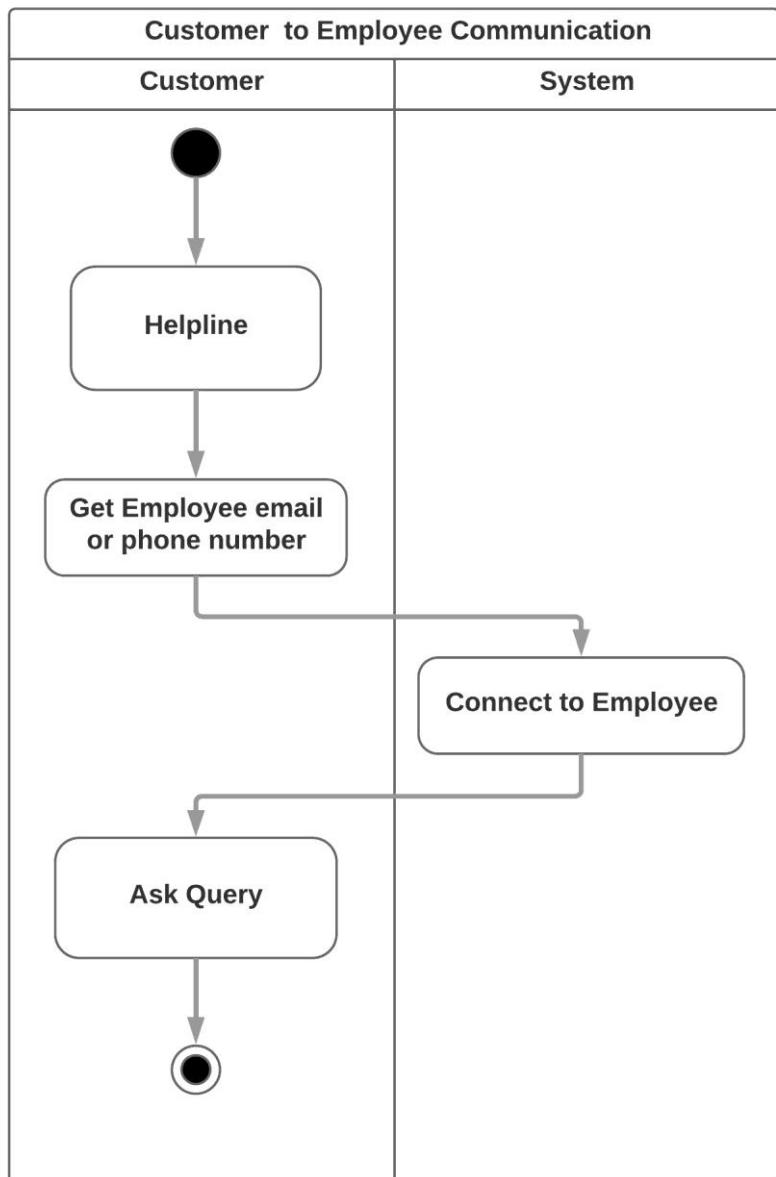


## Use Case 12: Customer to Employee

Activity Diagram:

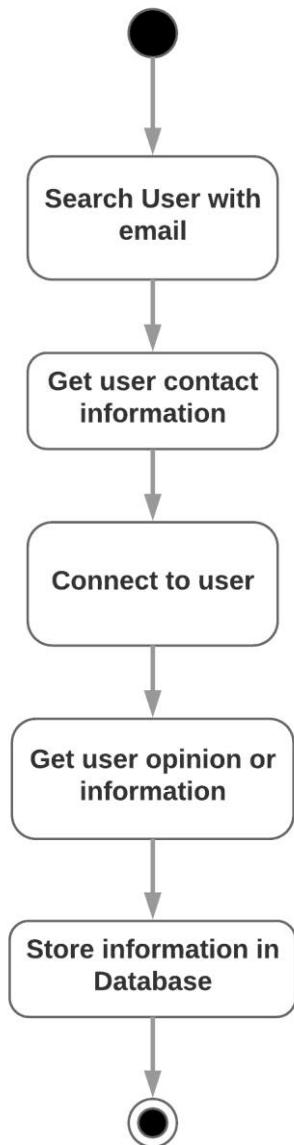


Swimlane Diagram:

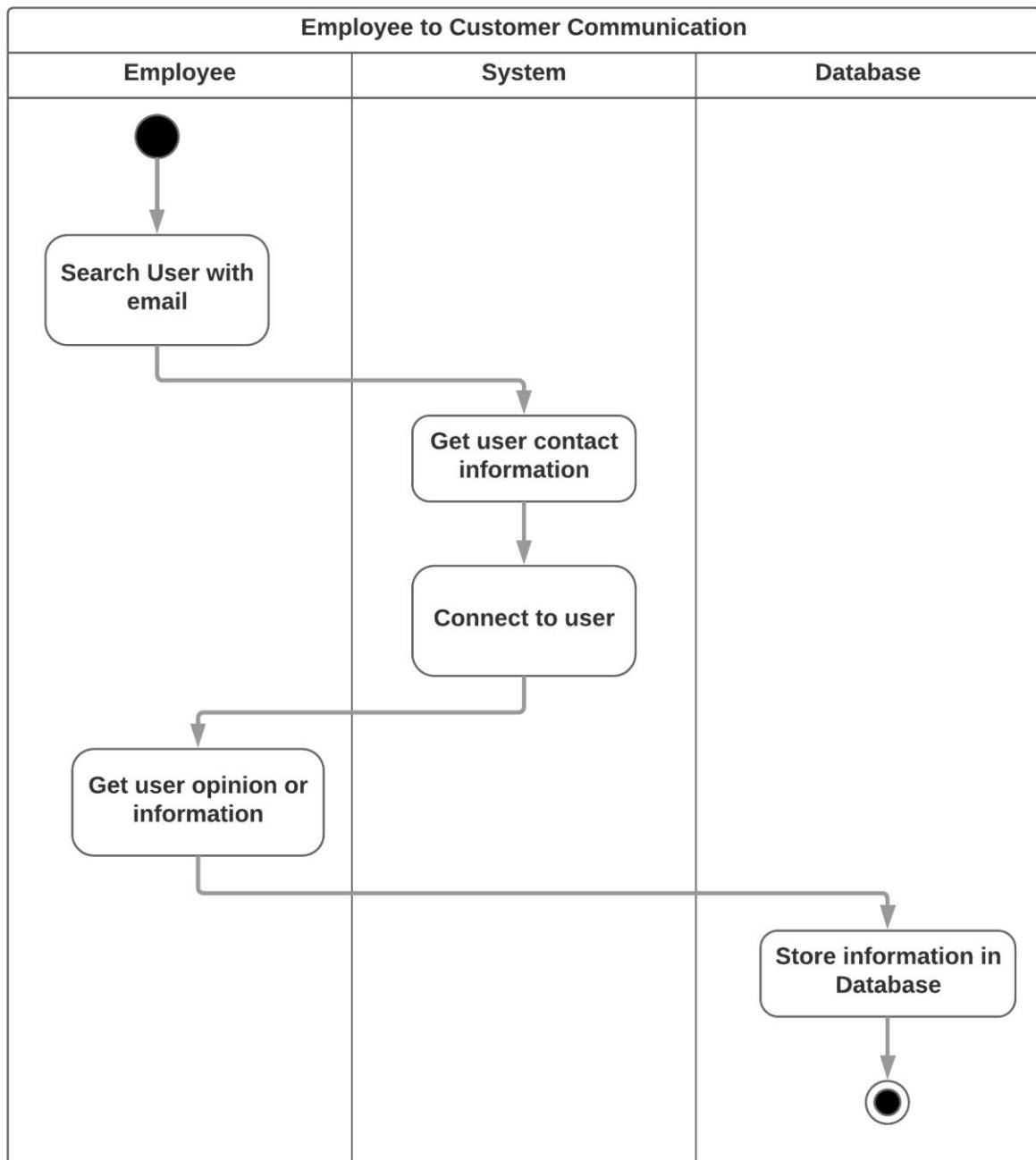


### **Use Case 13 : Employee to Customer**

Activity Diagram:

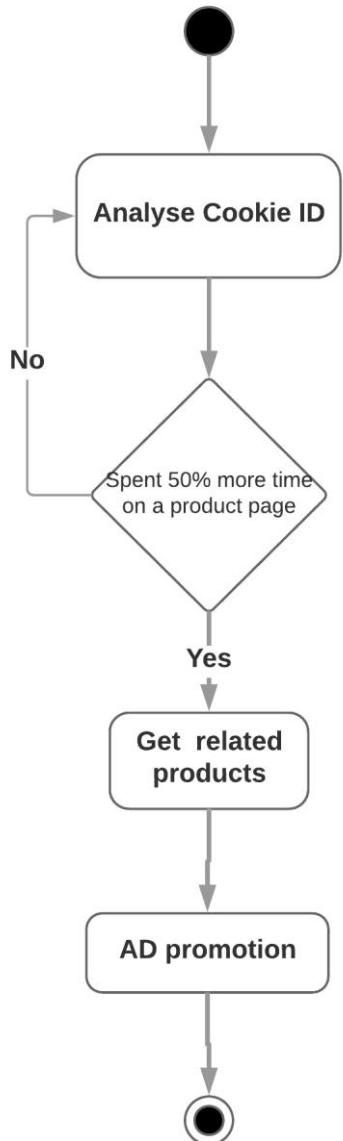


Swimlane Diagram:

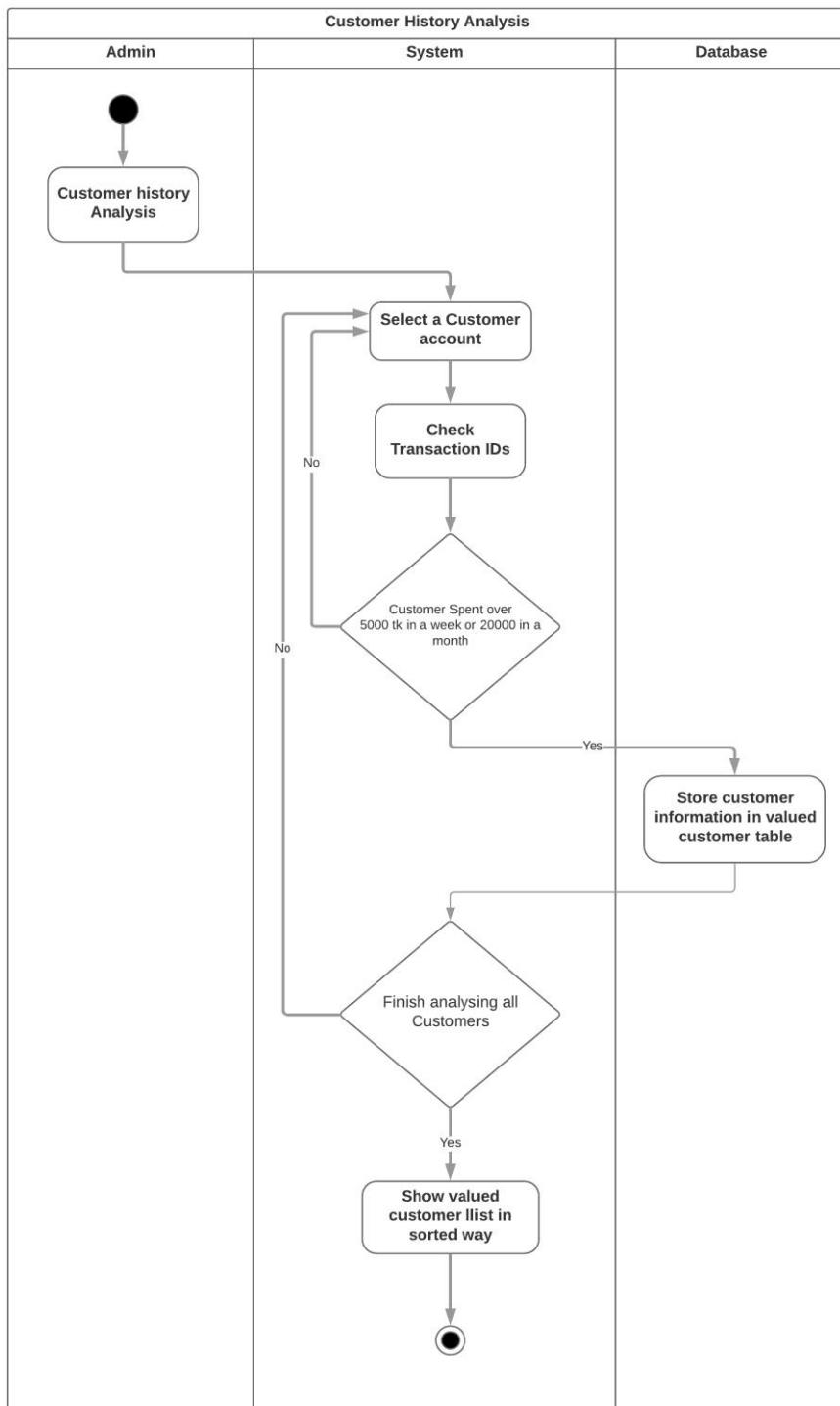


### Use Case 13: Customer cookie analysis

Activity Diagram:

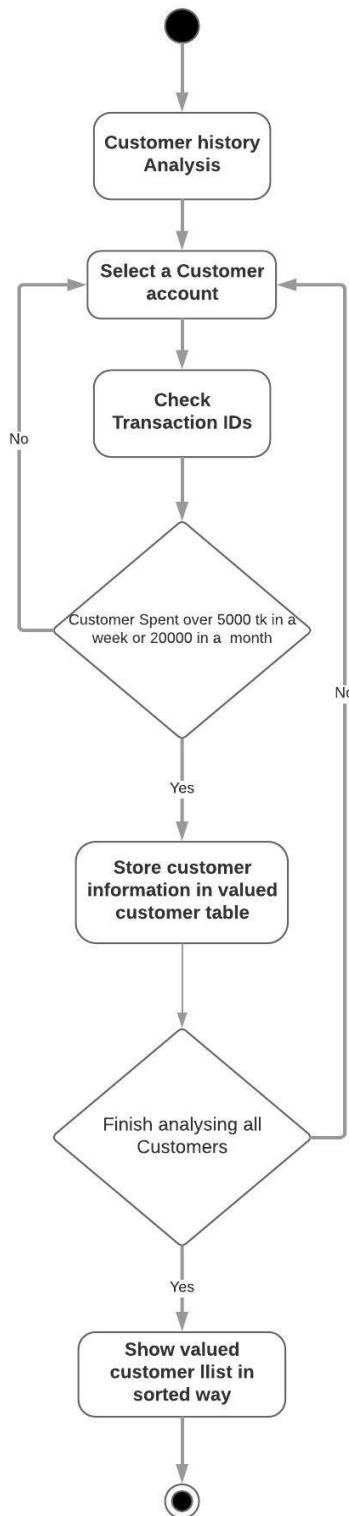


## Swimlane Diagram:

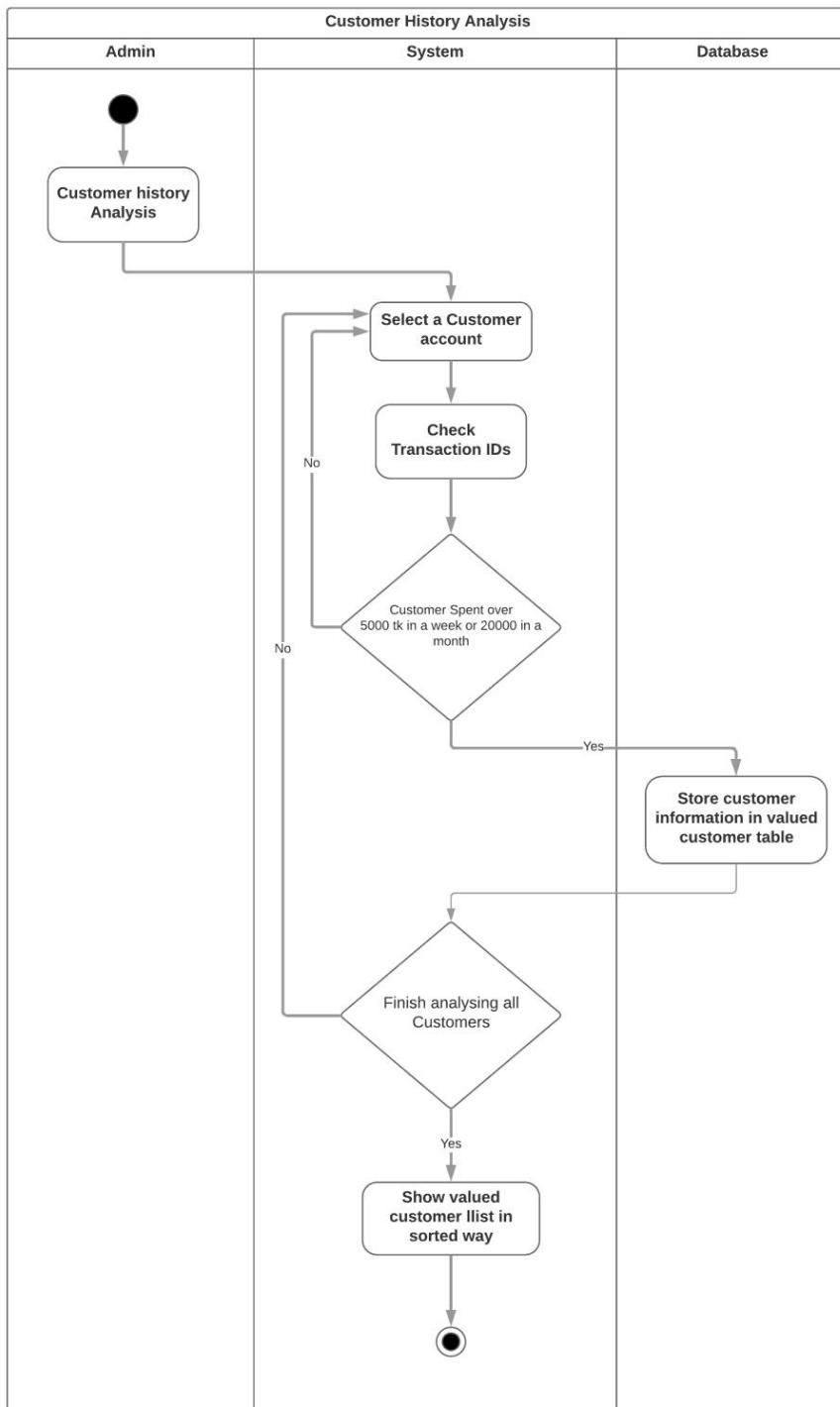


## Use Case 14: Customer history analysis

Activity Diagram:

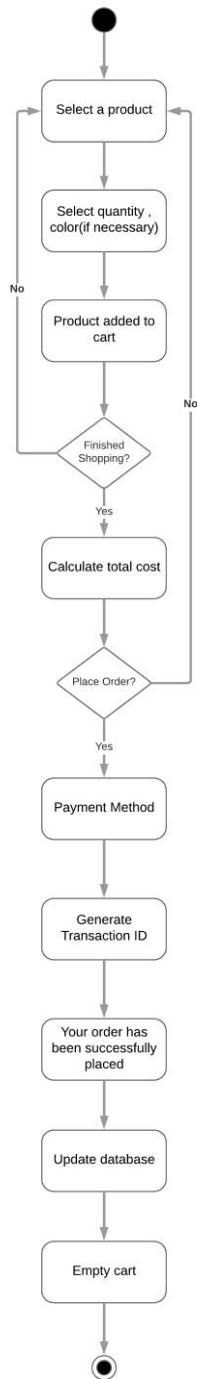


## Swimlane Diagram:

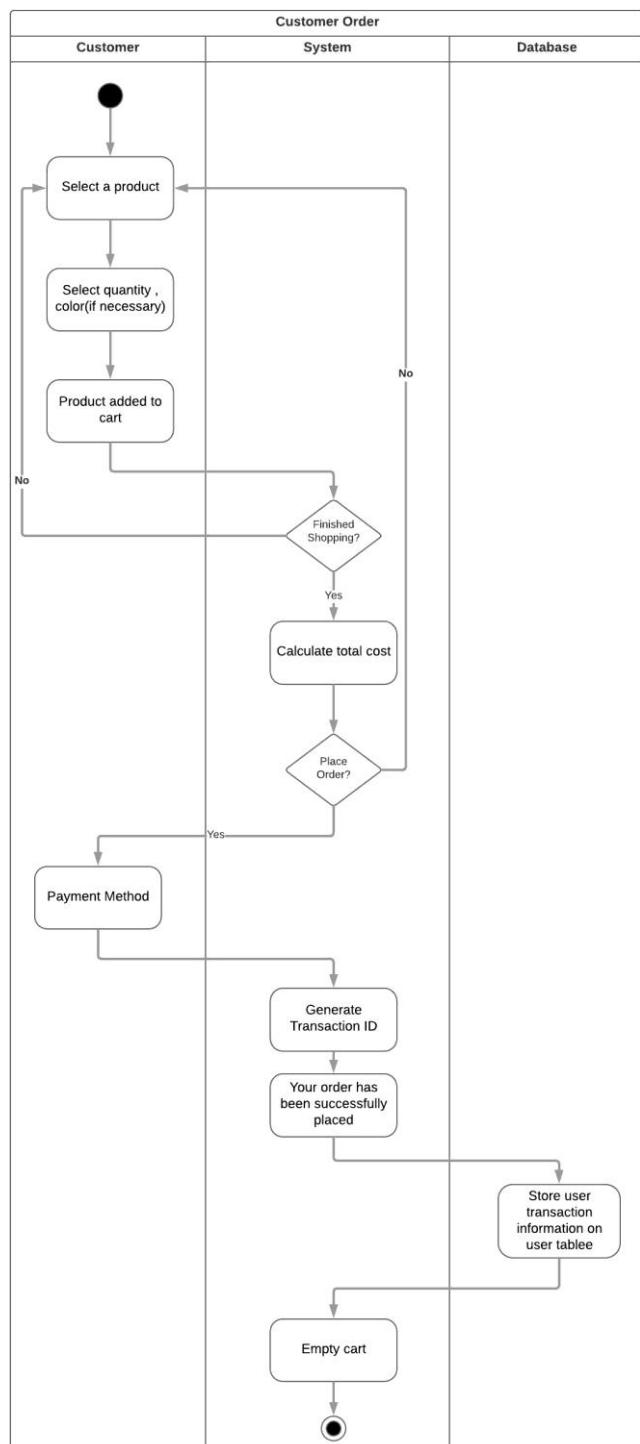


## Use Case 15: Customer order

Activity Diagram:

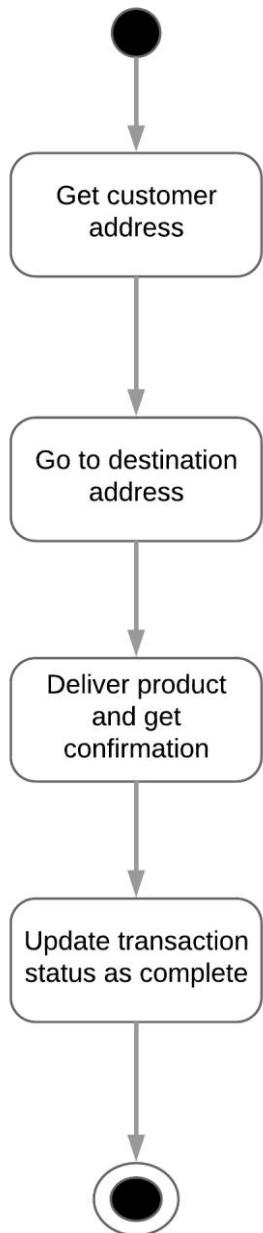


## Swimlane Diagram:

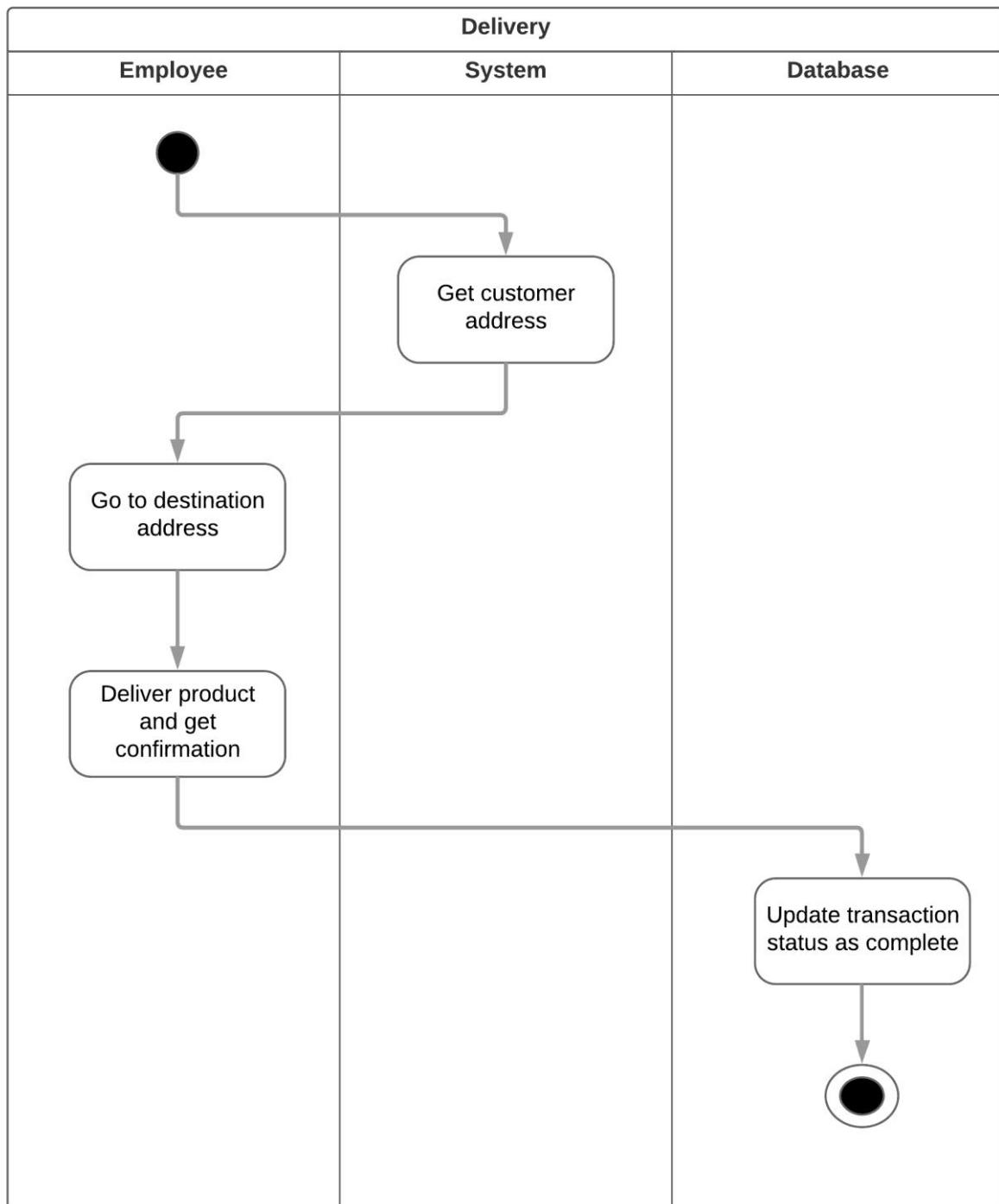


## Use Case 16: Delivery

Activity Diagram:

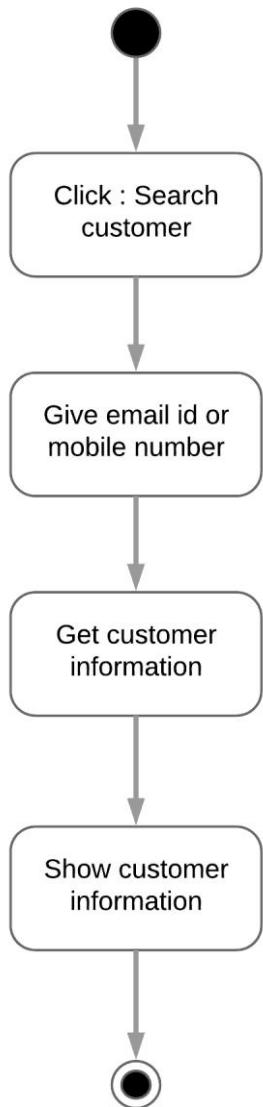


Swimlane Diagram:

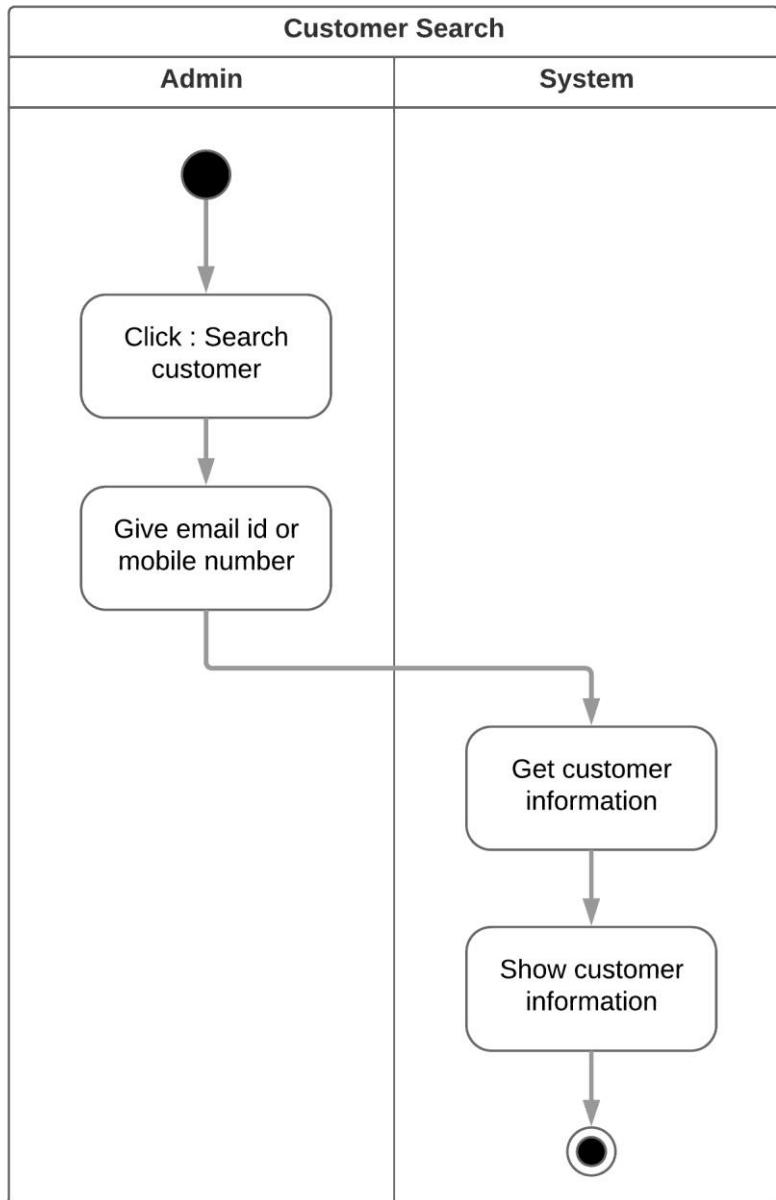


### **Use Case 17: Customer search**

Activity Diagram:

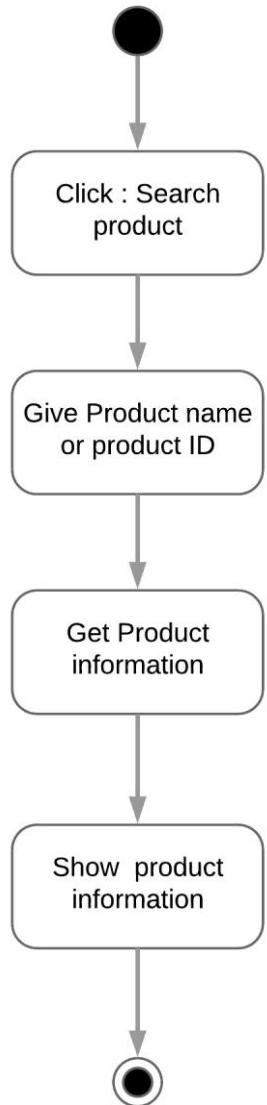


Swimlane Diagram:

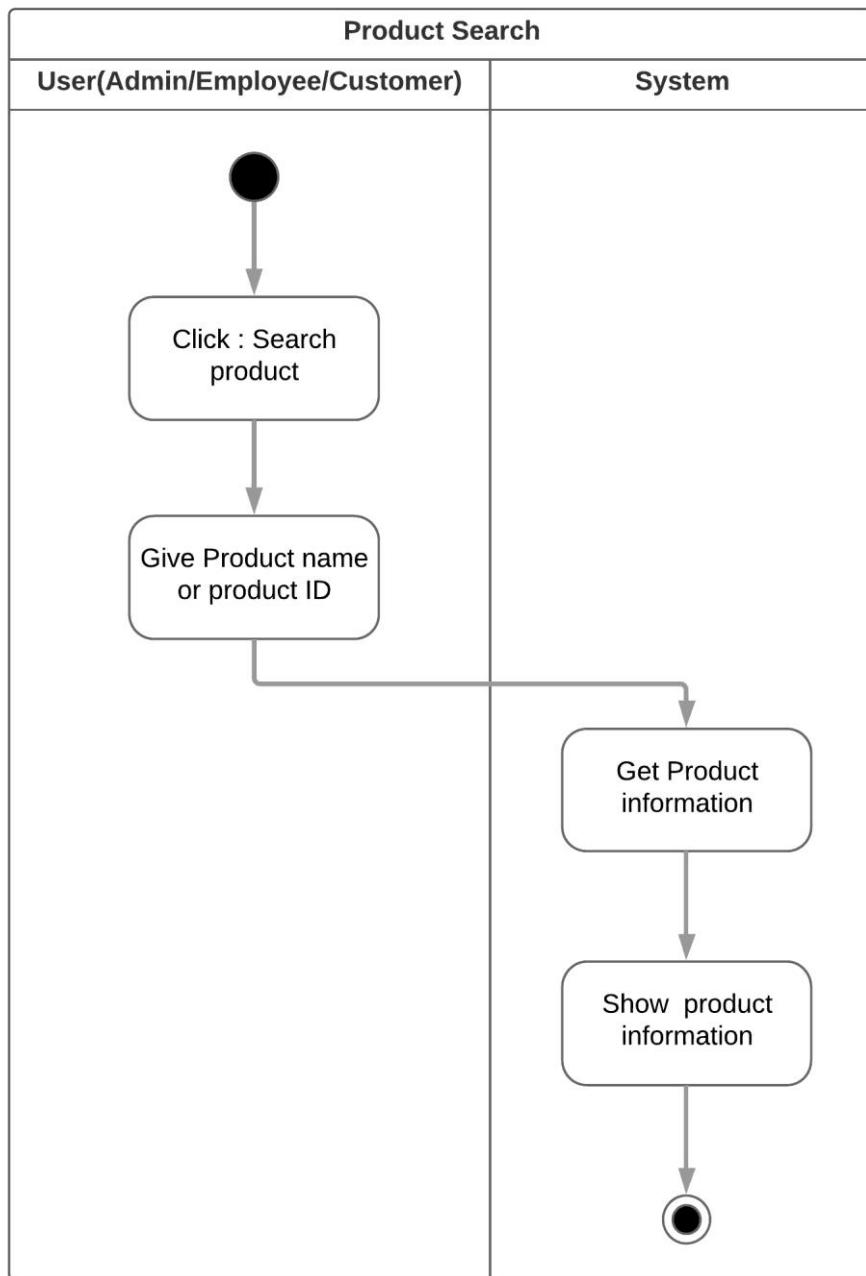


## Use Case 18: Product Search

Activity Diagram:

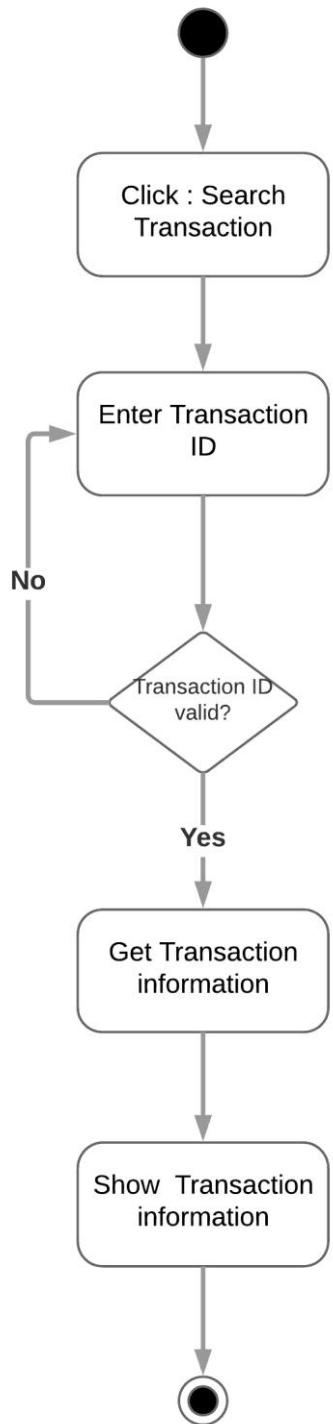


Swimlane Diagram:

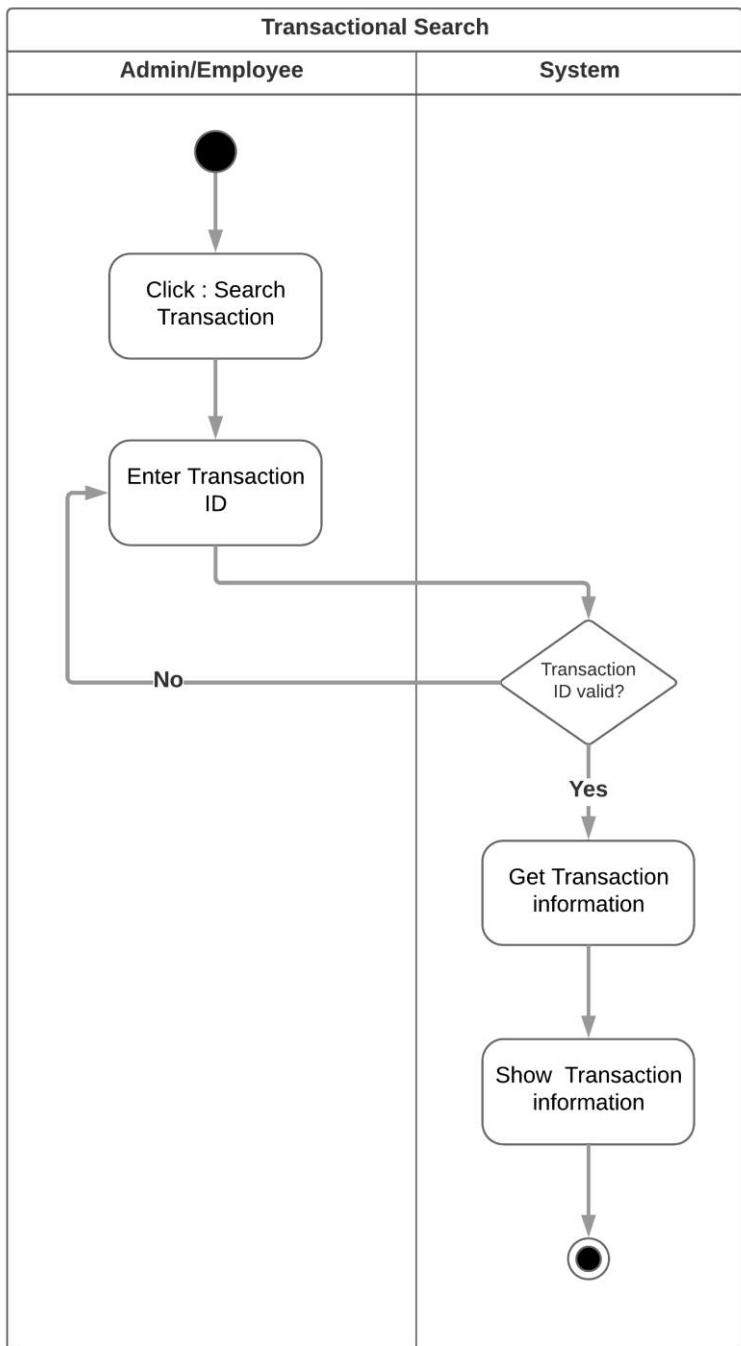


## Use Case 19: Transactional Search

Activity Diagram:

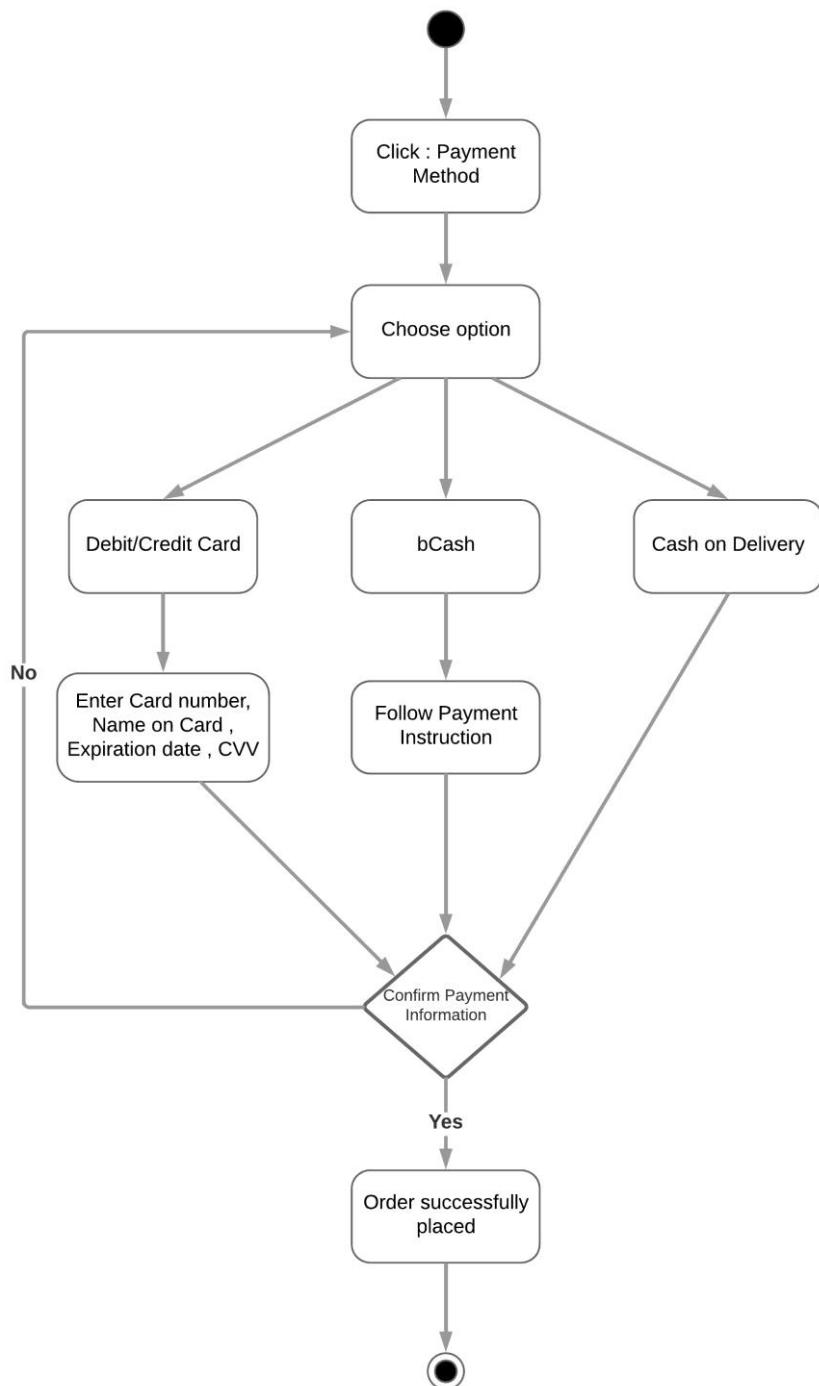


Swimlane Diagram:

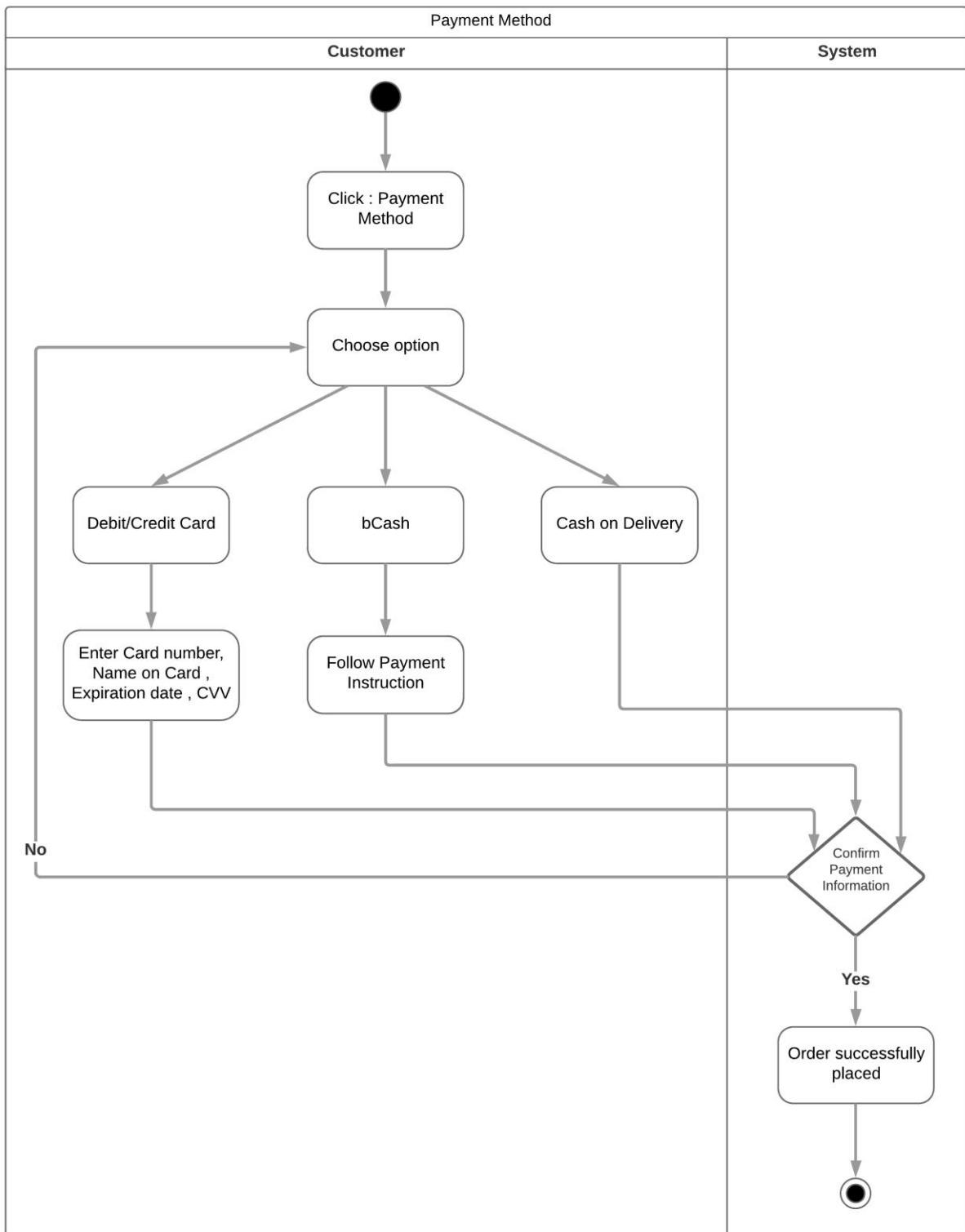


## Use Case 20: Payment System

Activity Diagram:

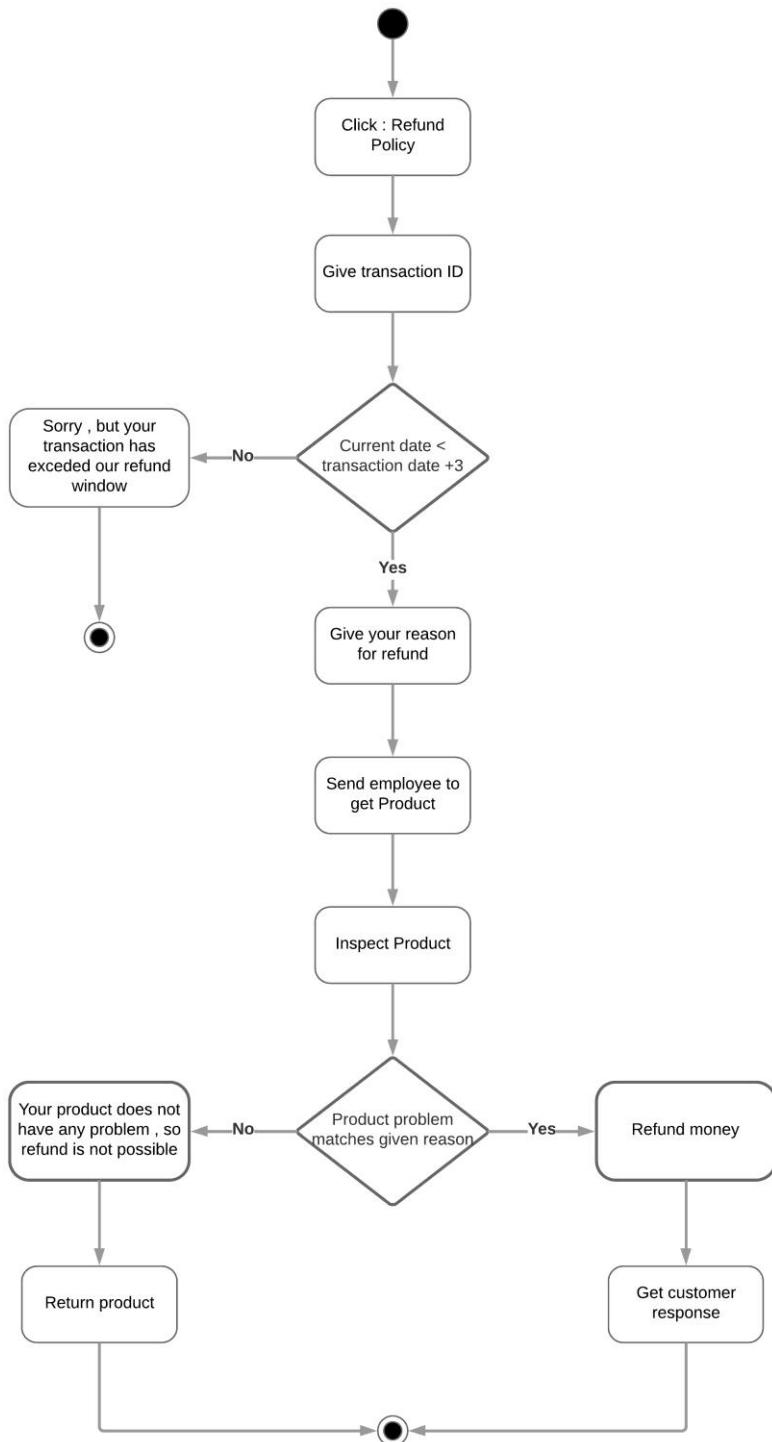


Swimlane Diagram:

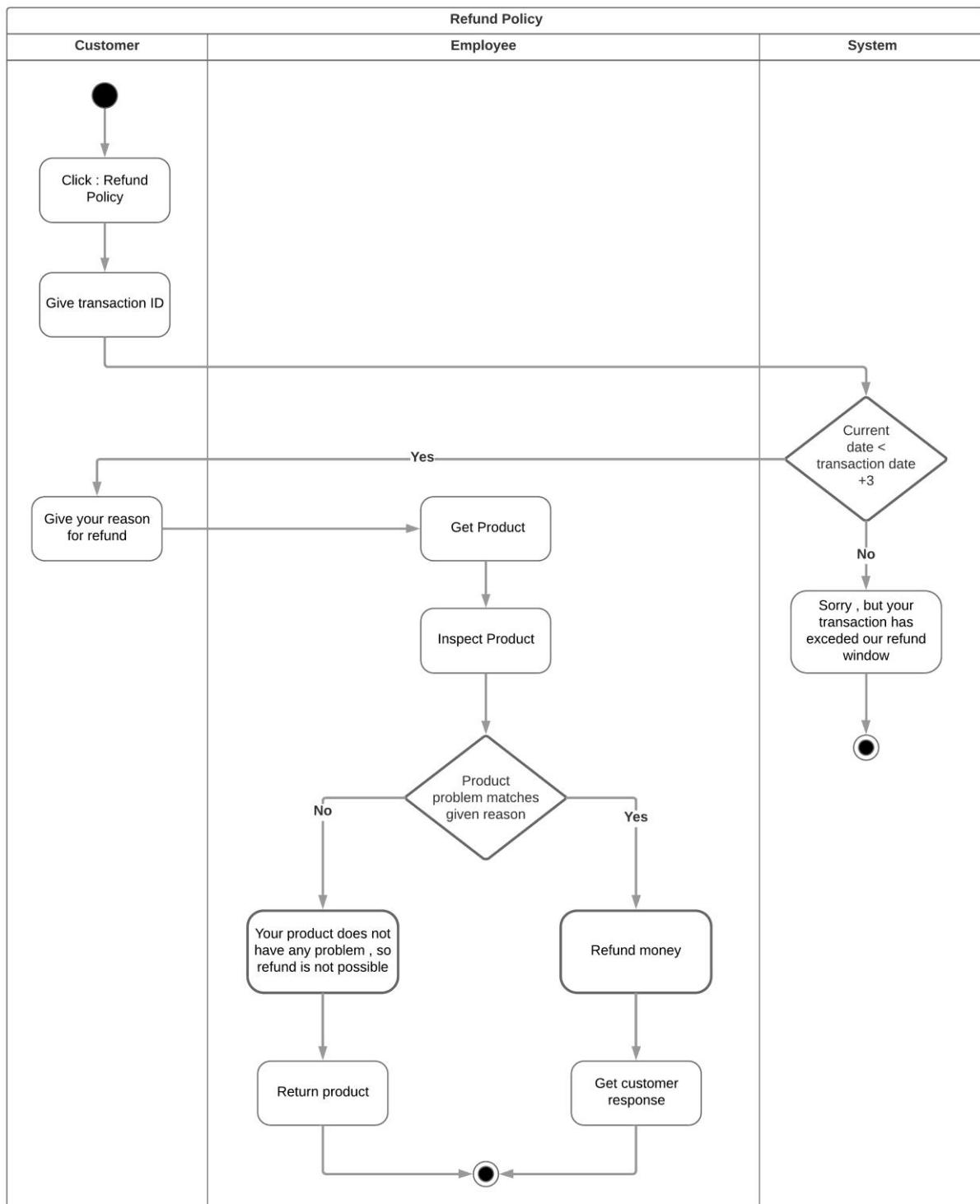


## Use Case 21: Refund policy

Activity Diagram:



Swimlane Diagram:



# Chapter 5

## Data Modeling

### Data modeling concept

If software requirements include the necessity to create, extend or interact with a database or complex data structures need to be constructed and manipulated, then the software team chooses to create data models as part of overall requirements modeling. The entity-relationship diagram (ERD) defines all data objects that are processed within the system, the relationships between the data objects and the information about how the data objects are entered, stored, transformed and produced within the system.

### Data objects

A data object is a representation of composite information that must be understood by the software. Here, composite information means an information that has a number of different properties or attributes. A data object can be an external entity, a thing, an occurrence, a role, an organizational unit, a place or a structure.

### Noun Identification

All the nouns in the scenario were identified.

Table 1: Noun identification for data modeling

No.	Noun	P/S	Attributes of
-----	------	-----	---------------

1	Authentication	S	
---	----------------	---	--

2	Sign Up	S	
3	SSL	P	
4	User	S	1,150
5	System	S	
6	Owner	S	
7	Employee	S	
8	Supplier	S	
9	Customer	S	
10	Account	P	
11	Form	S	
12	First Name	S	4,6,7,8,9
13	Last Name	S	4,6,7,8,9
14	Email id	S	4,6,7,8,9
15	Address	S	4,6,7,8,9
16	Mobile number	S	4,6,7,8,9
17	Password	S	4,6,7,8,9
18	Account type	S	4,6,7,8,9
19	Access	P	
20	Number	P	
21	Punctuation Mark	P	
22	Special Character	P	
23	Length	P	
24	Duplicity	P	
25	Validity	P	

26	Information	P	
27	Verification code	S	1,150
28	Approval	P	
29	Log In	S	
30	Verification	S	
31	Hour	P	
32	Retry	P	
33	Option	P	
34	Count	P	
35	Recover Password	S	
36	Transaction	S	
37	F Commerce Management	P	
38	Activity Log	S	
39	Attributes	P	
40	Transaction id	S	36
41	Transaction type	S	36
42	Item selling	S	36
43	Product buying	S	36
44	Resource buying	S	36
45	Employee salary	S	36,7
46	Discount	S	
47	Product id	S	46,61,120,139,150
48	Expenditure id	S	46
49	Customer id	S	46
50	Employee id	S	46
51	Amount	S	46,68,61,9
52	Date and time	S	1,46,68,61

53	Promotion id	S	46
54	Discount Code	S	46
55	Special Customer id	S	46
56	Quantity	S	46,61,120,139
57	Status	S	46,4,6,7,8,9
58	Discount type	S	46
59	Employee Commission	S	
60	Gadget Type	S	
61	Products	S	
62	Brand	P	
63	Install	P	
64	Notifications	P	
65	Two	P	
66	Three	P	
67	Product type	S	
68	Expenditure	S	
69	Supplier id	S	4,8
70	Repairing	P	
71	Description	S	68,6,120
72	Profit calculation	S	
73	Data	P	
74	Total income	S	72
75	Total expenditure	S	72
76	Profit	S	72,61
77	Date range	S	72,61
78	Product management	S	
79	Stock update	S	

80	Product list	S	61
81	Price	S	61,120
82	Product name	S	61,120,139
83	Discount amount	S	61
84	Fields	P	
85	Supplement	S	61
86	Manufacturing date	S	61,120
87	Date of Delivery	S	61,120
88	Expiration data	S	61,120
89	Discount percentage	S	61,120
90	Amount after discount	S	61
91	Discount validity date	S	61
92	Details	P	
93	Missing products	S	61
94	Missing quantities	S	61
95	Product cost	S	61
96	Stock analysis	P	
97	Tasks	P	
98	Popular products	S	61
99	Interval	P	
100	Transaction data	P	
101	Total products	S	
102	Week	P	
103	Month	P	
104	Profitable products	S	61
105	Total profit	S	72
106	Current Month	P	

107	Period	P	
108	Popularity	P	
109	Result	P	
110	Regular check-up	S	5
111	Minimum amount	S	61,139
112	Limit	P	
113	Contacting supplier	P	
114	Needed products	P	
115	Desired products	P	
116	List	S	6
117	Message Box	S	4,6,7,8,9
118	Confirmation message	P	
119	Advertisement	P	
120	Advertisement section	S	
121	Advertised products	S	120,
122	HR management	P	
123	Designation	S	4,6,7,8,9
124	Joined date	S	4,6,7,8,9
125	Account delete date	S	4,6,7,8,9
126	Online	P	
127	Offline	P	
128	Total dealings	S	8
129	Access permission	P	
130	Employment status	S	7
131	Salary Management	P	
132	Advance	P	
133	Reasons	P	

134	Certain percentage	P	
135	Valid Number	P	
136	Profile update	P	
137	Subsystem	P	
138	Type change	S	
139	Online Management	S	
140	Online product modification	P	
141	Online shopping	P	
142	Highest amount	S	139
143	Sale	P	
144	Categories	P	
145	Order	P	
146	First	P	
147	Default	P	
148	Product icon	S	61,139
149	Cart icon	P	
150	Message	S	
151	Total price	S	139
152	Pop-up interface	P	
153	Total cost	S	139
154	Purchase information	P	
155	Subsegments	P	
156	Notification	P	
157	Communication	P	
158	Recipient	P	
159	Message status	S	150
160	Owner-Supplier communication	P	

161	Owner-Employee communication	P	
162	Search	P	
163	User search	P	
164	Product search	P	
165	Transactional search	P	
166	Owner information	P	
167	Supplier information	P	
168	Customer information	P	
169	Substring	P	
170	Search option	P	
171	Description	S	38,150
172	Activity Log Id	S	38
173	Activity log type	S	38
174	Message type	S	150
175	Message Id	S	150

### Potential data objects

- User: 12-18,69,57,117,123,124,125
- Owner: 12-18,57,71,116,117,123-125
- Employee:12-18,45,57,117,123-125,130
- Supplier:12-18,57,69,117,123-125,128
- Customer:12-18,51,57,117,123-125
- Transaction :40-45
- Loan :47-58
- Products: 47,51,52,56,80-83,85-91,93,94,95,98,104,111,148
- Advertisement section: 47,56,71,81,82,86-89
- Online management: 47,56,82,111,142,148,151,153

- Expenditure: 51,52,71
- Profit calculation: 74,75,76,77,105
- Product-trade transaction: 40-45
- Owner-supplier transaction: 40-45
- Customer-employee transaction: 40-45
- Salary: 12-18,57,71,116,117
- Customer-employee loan:47-58
- Owner-supplier loan:47-58
- Activity log:171-173
- Authentication: 4,27,52
- Message:171,27,159,47,174,4,175
- Owner-supplier message:175,4
- Customer-employer message:175,4
- Notification:175,4

### Analysis for finalizing Data objects

- Owner, employee, supplier and customer are users of the grocery management system. So, all four kinds of users can be merged into a user data object.
- All other data objects can be used as data objects as they have enough importance in the system.

### Final Data objects

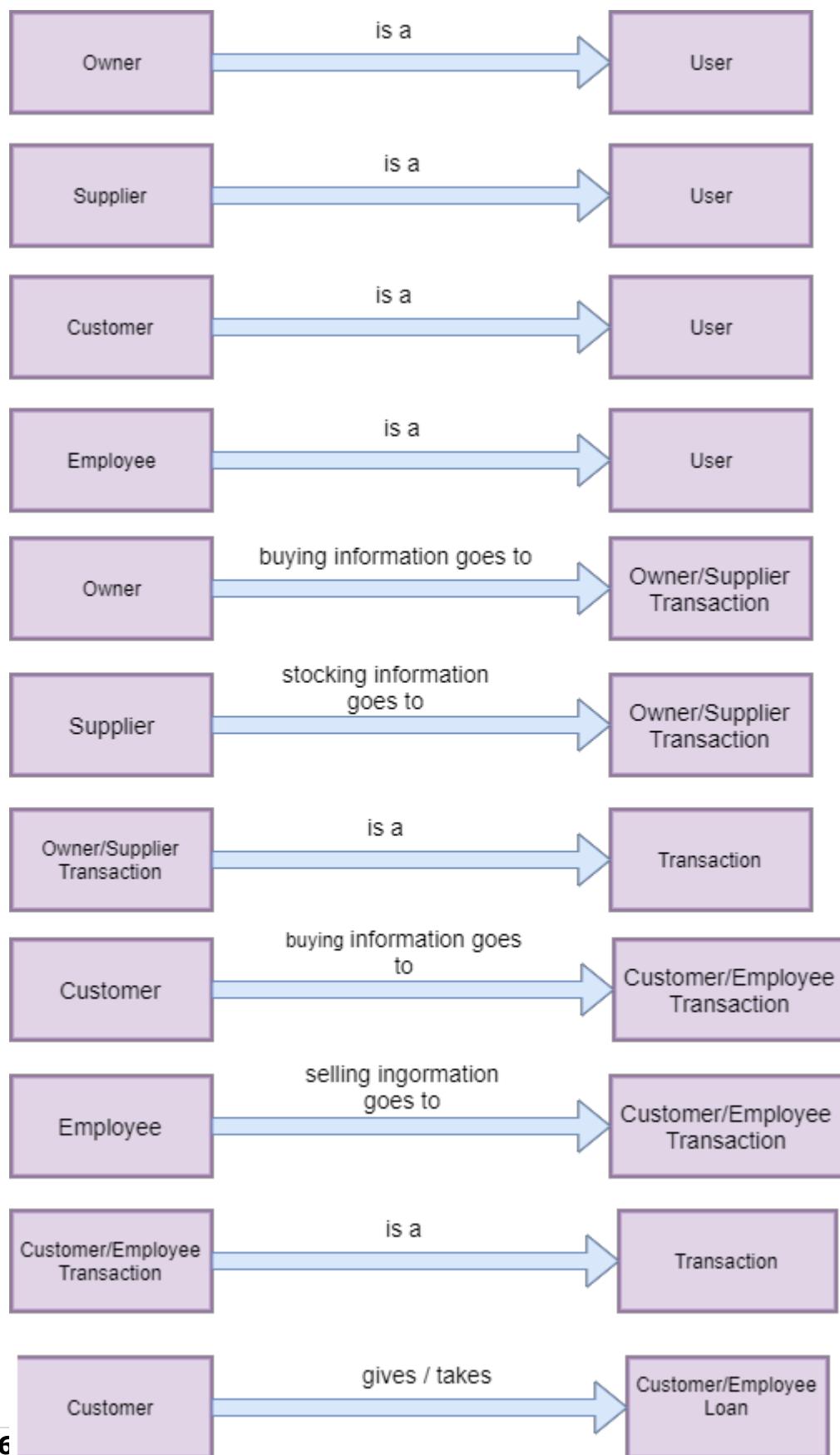
*Table 2: Final Data Objects*

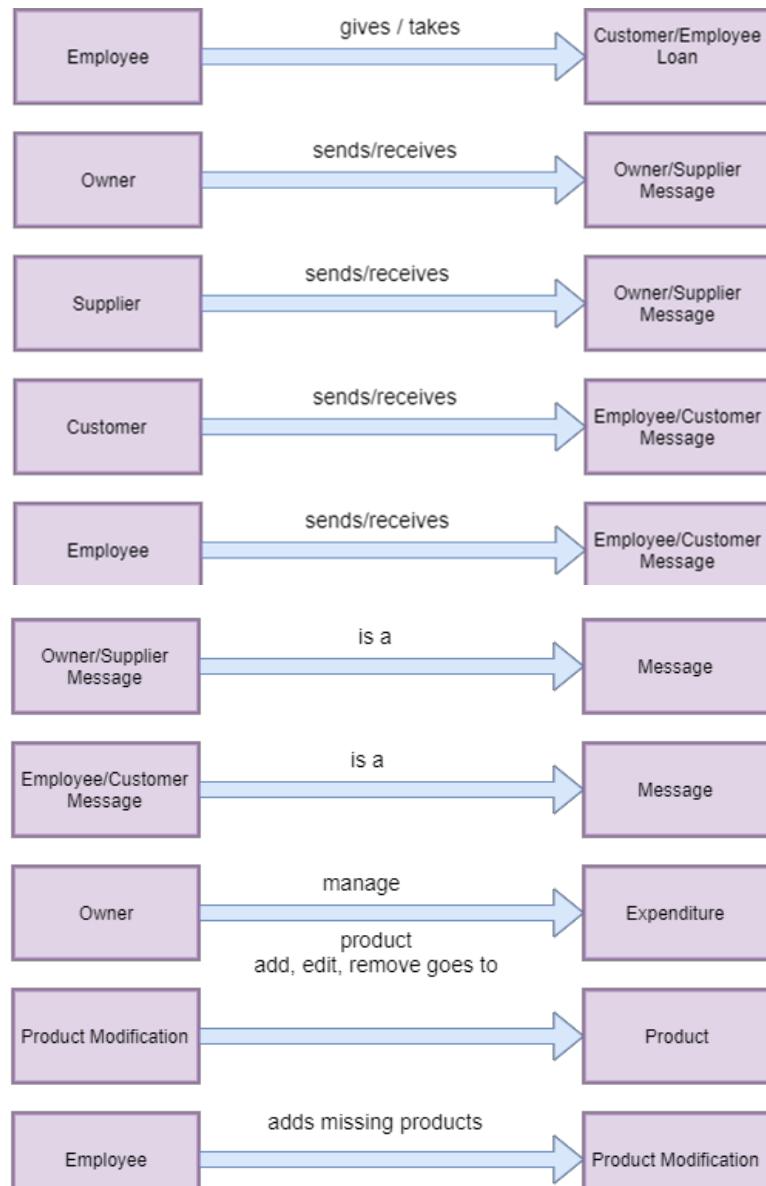
1	User: <u>User id</u> , First name, Last name, Email, Phone, Password, Address, Joined date, Last date, status.
2	Owner: <u>User id</u>

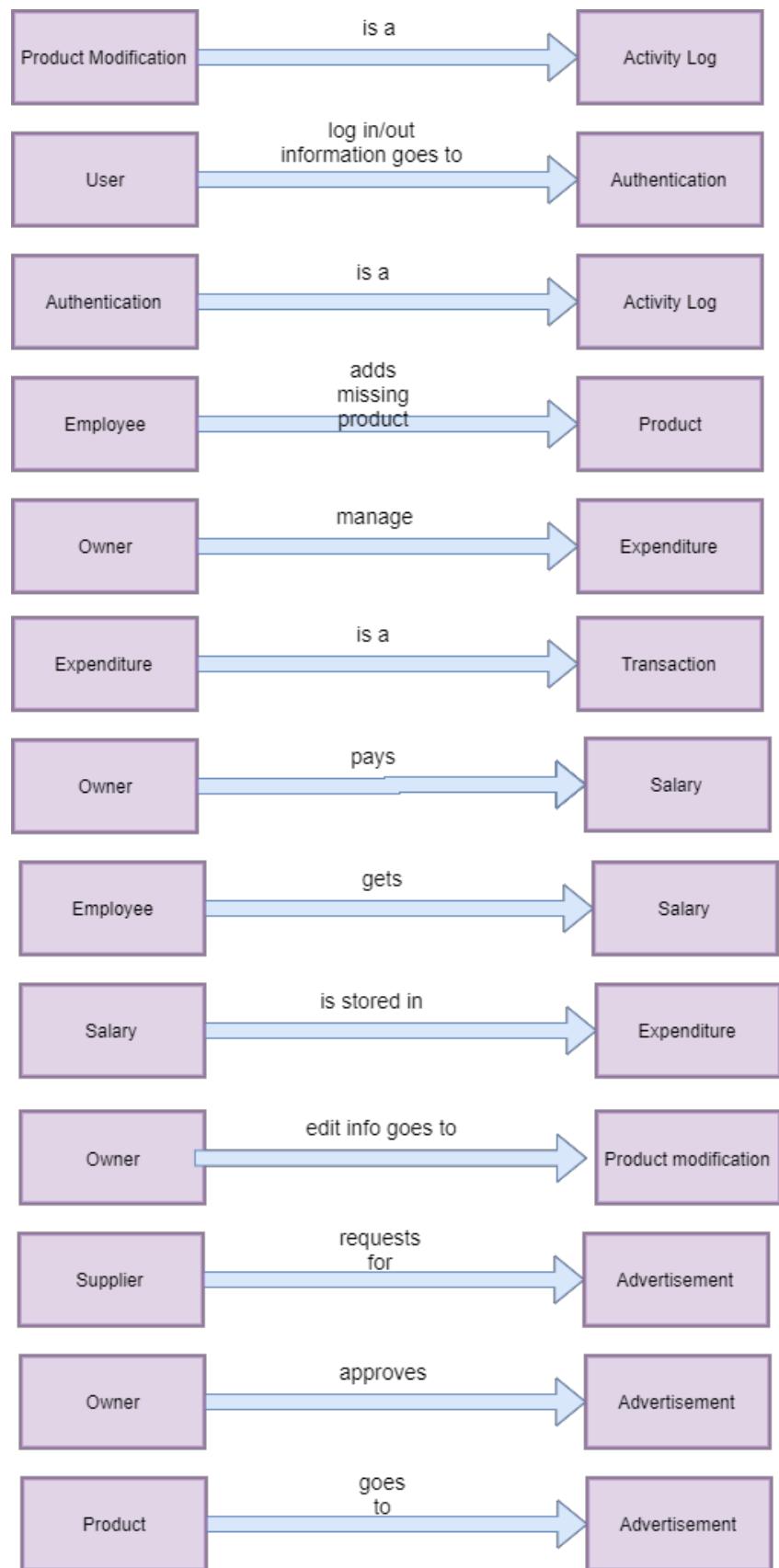
3	Employee: <u>User id</u> , Salary
4	Customer: <u>User id</u> , Type
5	Supplier: <u>User id</u> , Description
6.	Product: <u>Product Id</u> , Product Name, Online Quantity, Offline Quantity, Insufficient Online Quantity, Insufficient Offline Quantity, Price, Is Advertised, Discount
7.	Transaction: <u>Transaction Id</u> , Type, Amount, Total Amount Paid, Date and Time
8	Product trade transaction: <u>Transaction Id</u> , <u>Product Id</u> , Quantity
9	Owner-Supplier Transaction: <u>Transaction Id</u> , Owner id, Supplier id
10	Customer-Employee Transaction: <u>Transaction Id</u> , Employee id, Customer id
11	Expenditure: <u>Transaction Id</u> , Owner id, Description
12	Salary: <u>Salary Id</u> , Expenditure Id, Owner id, Employee Id
13	Commission: <u>Commission Id</u> , Transaction Id, Total Amount, Amount Paid, Date and Time

14	Supply: <u>Supplier Id</u> , <u>Product Id</u>
15	Advertisement: <u>Advertise Id</u> , Supplier Id, Owner Id, Product Id, Date and Time, Status
16	Activity Log: <u>log Id</u> , User Id, Date and Time
17	Authentication: <u>log Id</u> , Type
18	Product Modification: <u>log Id</u> , Product Id, Type
19	Discount: <u>Discount Id</u> , Product Id, Owner Id, Discount From, Discount To
20	Expire: <u>Expire Id</u> , Product Id, Owner Id, Supplied Date, Expired Date, Quantity
21	Message: <u>Message Id</u> , Description, Date, Status
22	Admin-Supplier Message: <u>Message Id</u> , Sender Id, Recipient Id
23	Customer-Employee Message: <u>Message Id</u> , Sender Id, Recipient Id
24	Notification: <u>Message Id</u> , Recipient Id
25	Supply: <u>Supplier Id</u> , <u>Product Id</u>

## Data Object Relations







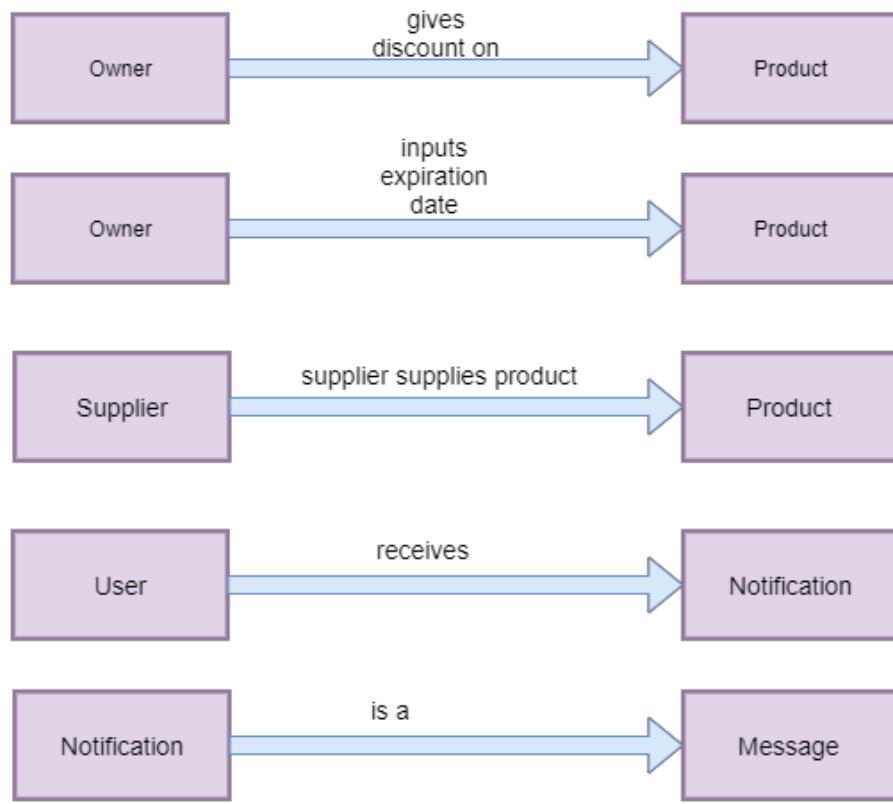


Figure : Relation Between Data Objects

## Entity Relationship Diagram

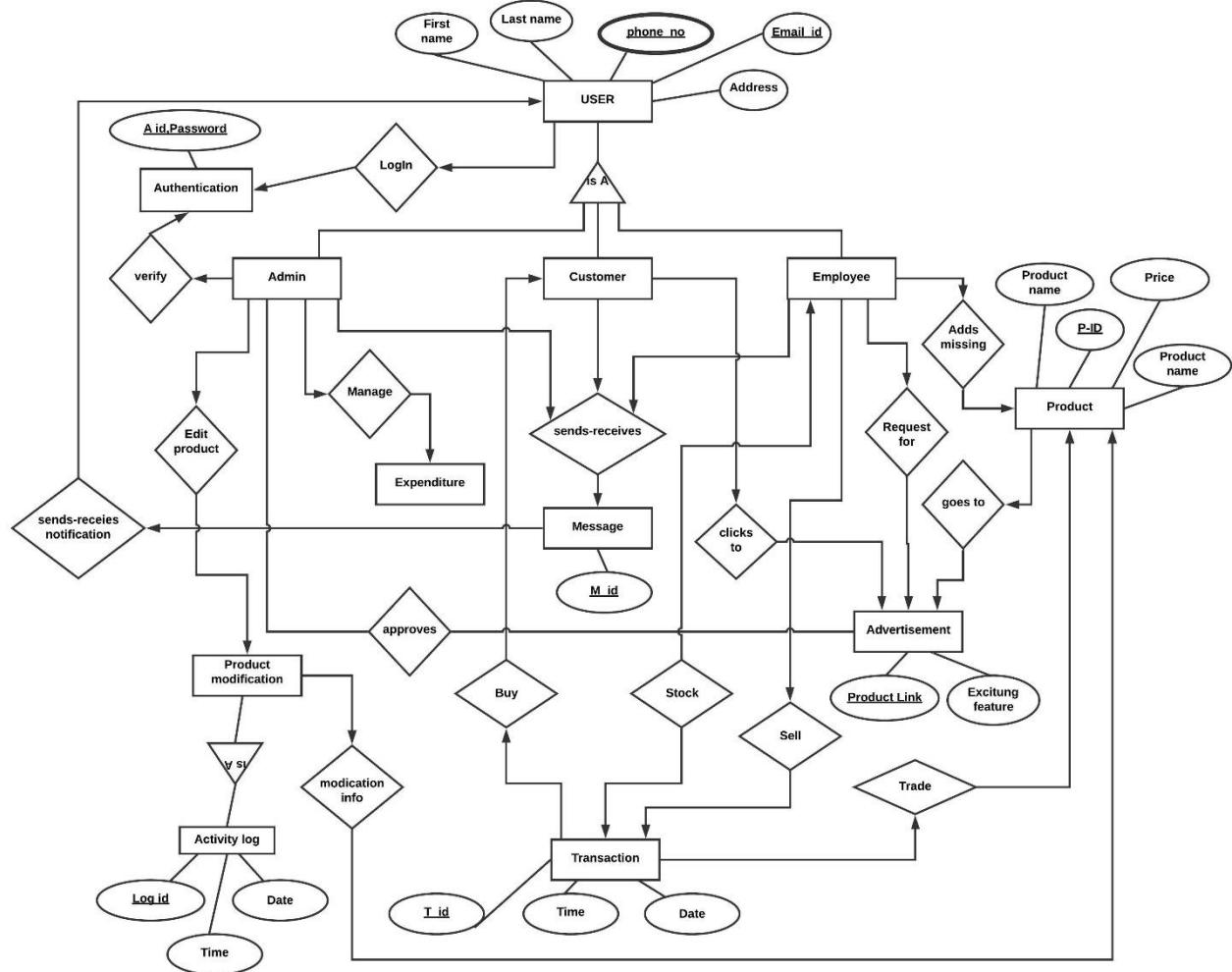


Figure : ER diagram of FCMS

# Chapter 6

## Class Based Modeling

This chapter is intended to describe class based modeling of the Grocery Management System.

### Class Based Modeling Concept

Class-based modeling represents the objects that the system will manipulate, the operations that will be applied to the objects, relationships between the objects and the collaborations that occur between the classes that are defined.

### General Classification

To identify the potential classes, nouns were selected from the solution space of the story. These were then characterized in seven general classifications. The seven general characteristics are as follows:

1. External entities
2. Things
3. Events
4. Roles
5. Organizational units
6. Places
7. Structures

Following are the specifications of the nouns according to the general classifications.

Table 16: General Classification

No.	Noun	GC
1	Account	3,5
2	Sign up	3
3	Customer	4,5,7
4	System	4,5
5	Owner	4,5,7
6	Employee	4,5,7
7	Admin	4,5,7
8	User	4,5,7
9	Sign Up Form	
10	Name	
11	Owner	
12	Email id	
13	Address	
14	Mobile number	
15	Password	
16	Account type	
17	Log In	
18	Verification	
19	Recover Password	

20	Transaction	2,3,5,7
21	Transaction id	
22	Payment system	
23	Item selling	
24	Item buying	
25	Stock Update	
26	Employee salary	
27	Discount	2,3,5,7
28	Item id	
29	Cost	
30	Customer id	
31	Employee id	
32	Amount	
33	Date and time	
34	Payment id	
35	Promotional Code	
36	Brand id	
37	Quantity	
38	Status	
39	Owner	
40	Sales Commission	

41	Order ID	
42	Product	2,5,7
43	Expenditure	3,5,7
44	Activity log	2,5,7
45	Customer id	
46	Details	
47	Profit calculation	
48	Net income	
49	Total expenditure	
50	Profit	
51	Date range	
52	Product management	
53	Stock update	
54	Product list	
55	Price	
56	Product name	
57	Discount amount	
58	Supplement	
59	Manufacturing date	
60	Date of Delivery	
61	Warranty Expiration data	

62	Discount percentage	
63	Amount after discount	
64	Discount validity date	
65	Missing products	
66	Missing quantities	
67	Product cost	
68	Popular products	
69	Total products	
70	Profitable products	
71	Total profit	
72	Product Categories	
73	Minimum amount	
74	List	
75	Message	2,3,5,7
76	Advertisement section	
77	Advertised products	
78	Employee	
79	Joined date	
80	Account delete date	
81	Total dealings	
82	Employment status	

83	Type change	
84	Online Shopping	
85	Highest amount	
86	Product icon	
87	Total price	
88	Communication	
89	Notification	2,3,5,7
90	Search	2,3
91	Database	1,5,6
92	Cookie	
93	History	
94	Block Account	
95	Order	
96	Facebook ID	
97	Facebook Log in	
98	Gmail Log in	
99	Gadget Brand	
100	Management Cost	
101	Management Expenditure	
102	Order	
103	Delivery	
104	Interface	
105	Search	

## **Selection Criteria**

The potential classes were then selected as classes by six Selection Criteria. A potential class becomes a class when it fulfills all six characteristics.

1. Retained Information
2. Needed Services
3. Multiple Attributes
4. Common attributes
5. Common operations
6. Essential requirements

No.	Noun	AC
1	Account	1,2,3
2	user	1,2,3,4,5
3	owner	1,2,3,4,5
4	employee	1,2,3,4,5
5	Admin	1,2,3,4,5
6	customer	1,2,3,4,5
7	system	1,2,3
8	transaction	3,4,5
9	Cost	3,4,5
10	Management Cost	3,4,5
11	Financial Report	2,3,4,5
12	Order	3,4,5

13	notification	2,3,4,5
14	message	2,3,4,5
15	Delivery	2,3,4,5
16	database	6
17	search	1,2,3
18	verification	1,2,3
19	Interface	1,6
20	Item	1,2,3

### Associate Noun and Verb Identification

The nouns and the verbs associated with the potential classes are identified to find out the attributes and methods of each class.

No	Potential Class	Noun	Verb
1	Account	Verification code , user, time	Log in, sign up, log out, automatically log out, lock account, send verification code
2	User	Id, first name, last name, email, phone number, address, joined date, designation, The date he/she was fired/blocked, messa ge, status	Requires an account, can log in to the system, sends message, receives message, search, update information, recover his/her password
3	Admin		Approves request, can add product, can edit product, remove product, order product, receive product, pay the supplier late, pay the supplier before he/she gets the product, advertise product, remove advertise product,

			manage employee salary, raise employee salary, pay him/her advance, remove employee, remove supplier, block customer, change type, inputs type of shop management, identifying the needed products
4	Employee	salary	Adds missing product, edit missing product, removes missing product, sells product, borrow product
5	Customer	Type, amount paid online, total amount paid	View or buy products online
6	Owner		Can approve order, can ask for advertising product
7	System	Product, user	Show popular product, sort out profitable product, regular check up on expiry date, send notifications to the owner (if it is an owner-supplier loan) or the employee (if it is an employee-customer loan) every three days until the loan is completely paid or received, calculate profit, calculate total price.
8	Transaction	Id, type, Product, date, total amount paid, total amount, employee id, customer id	Print receipt, confirm purchase, add product, remove product

9	Cost	Id, type, date, product, seller id	
10	Management Cost	Type, sector, date	Calculate Cost
11	Management Expenditure	Date, date	Calculate Expenditure
12	Delivery	ItemID,date,Address	Print receipt
13	Item	Id, name, quantity, price	Quantity will be updated, editing product's details
		discount  Insufficient quantity	
14	Interface	Log id, date  Log type, user id, product id	Will be displayed
15	Notification	Id, receiver, type, description, date, status	Will be sent
16	Message	Id, sender, receiver, type, description, date, status	Status will be seen, will be sent, received

17	Order	Id, Customer id, amount bought, products, date	Order forwarded to delivery
18	Database	User, product, transaction	Will be stored, will be updated
19	Search	User, product, transaction, date	Search by week, Search by month, Search by year, Based on a given interval
20	Verification	email phone password	will be duplicity and validity (syntax) checking for email and mobile number, will be verified

## Attribute Selection

Table 30: Attribute Selection of Classes

No.	Noun	Attributes
1	Account	Id firstName lastName email phone address facebook ID type lastDate status
2	user	Id Name Email phone address

		status
3	owner	Status
4	employee	Salary Commission
5	Admin	1,2,3,4,5
6	customer	totalAmountShopping customerType
7	system	item account amount
8	transaction	Id type date item totalAmount amountPaid OrderId status
9	Cost	Productid Quantity Amount Status Type date
10	Management Cost	Sector id
11	Management Expenditure	Purpose cost
12	Order	OdrerId Address Quantity
13	notification	Id Receiver Type Description date status discount insufficientQuantity
14	message	Id Sender receiver type itemId description

		date status
15	Delivery	CustomerId OderId Address
16	database	user product transaction
17	search	account item transaction date
18	verification	User verificationCode time
19	Interface	status
20	Item	Id Name Quantity onlineStock offlineStock isAdvertised price discount insufficientQuantity

### Methods Identification:

Table 31: Methods of Classes

No.	Noun	Methods
1	Account	<ul style="list-style-type: none"> <li>• signUp()</li> <li>• validateInput()</li> <li>• setUser()</li> <li>• login()</li> <li>• findUser()</li> <li>• GetUser()</li> <li>• ChangeData()</li> <li>• RecoveryData()</li> </ul>
2	user	<ul style="list-style-type: none"> <li>• getId ()</li> <li>• getFirstName()</li> <li>• setFirstName()</li> <li>• getLastName()</li> <li>• setLastName()</li> </ul>

		<ul style="list-style-type: none"> <li>• getEmail()</li> <li>• setEmail()</li> <li>• getPhone()</li> <li>• setPhone()</li> <li>• getAddress()</li> <li>• setAddress()</li> <li>• getJoinedDate()</li> <li>• setJoinedDate()</li> <li>• getDesignation()</li> <li>• setDesignation()</li> <li>• getLastDate()</li> <li>• setLastDate()</li> <li>• getStatus ()</li> <li>• setStatus ()</li> <li>• signUp()</li> <li>• showOptions()</li> <li>• receiveMessage()</li> <li>• search()</li> <li>• sendMessage()</li> <li>• updateInfo()</li> <li>• recoverPassword()</li> <li>• logIn()</li> </ul>
3	owner	<ul style="list-style-type: none"> <li>• approveRequest()</li> <li>• addProduct()</li> <li>• editProduct()</li> <li>• removeProduct()</li> <li>• orderProduct()</li> <li>• receiveProduct()</li> </ul>
4	employee	<ul style="list-style-type: none"> <li>• getSalary()</li> <li>• setSalary()</li> <li>• sellProduct()</li> <li>• getCommision()</li> <li>• DeliveryProduct()</li> </ul>
5	Admin	<ul style="list-style-type: none"> <li>• approveAccount()</li> <li>• addAdmin ()</li> <li>• addEmployee()</li> <li>• removeAdmin ()</li> <li>• removeEmployee()</li> <li>• updateAdminInformation()</li> <li>• updateEmployeeInformation()</li> <li>• includeProduct()</li> <li>• exludeProduct()</li> <li>• viewReport()</li> <li>• provideSalary()</li> <li>• announcePromotions()</li> <li>• BlockCustomer()</li> <li>• OrderAnalysis()</li> </ul>

6	customer	<ul style="list-style-type: none"> <li>• validateInput()</li> <li>• verifyUser()</li> <li>• recoverAccount()</li> <li>• createNotification()</li> <li>• recordTransaction()</li> <li>• recordDealingInformation()</li> <li>• updateStock()</li> <li>• logout()</li> <li>• getAccountType()</li> <li>• setCurrentAddress()</li> <li>• setContact()</li> <li>• getFullName()</li> <li>• getFacebookID()</li> <li>• getEmail()</li> <li>• onlinePurchase()</li> </ul>
7	system	<ul style="list-style-type: none"> <li>• connectDatabase()</li> <li>• accessDatabase()</li> <li>• showInterface()</li> <li>• createInterface()</li> <li>• deleteData()</li> <li>• saveFiles()</li> <li>• pause()</li> <li>• DataAnalyze()</li> <li>• createNotification()</li> <li>• disconnectDatabase()</li> </ul>
8	transaction	<ul style="list-style-type: none"> <li>• generateId()</li> <li>• createReceipt()</li> <li>• calculatePrice()</li> <li>• createTransaction()</li> <li>• paidAmount()</li> <li>• determineDealingAmount()</li> <li>• updateCash()</li> <li>• updateFinancialReport()</li> <li>• processDealingInformation()</li> <li>• updateStock()</li> </ul>
9	Cost	<ul style="list-style-type: none"> <li>• addProduct()</li> <li>• removeProduct()</li> <li>• calculateAmount()</li> <li>• printReceipt()</li> <li>• getProduct()</li> <li>• setAmount()getDescription ()</li> <li>• setDescription ()</li> <li>• getDate ()</li> <li>• setDate ()</li> </ul>

		<ul style="list-style-type: none"> <li>• sendMessage()</li> <li>• getStatus()</li> <li>• changeStatus()</li> <li>• sendVerificationCode()</li> <li>• sendPassword()</li> </ul>
10	Management Cost	<ul style="list-style-type: none"> <li>• getId()</li> <li>• getSender ()</li> <li>• setSender ()</li> <li>• getReceiver()</li> <li>• setReceiver()</li> <li>• getType ()</li> <li>• setType ()</li> </ul>
11	Financial Report	<ul style="list-style-type: none"> <li>• showReport()</li> <li>• getCash()</li> <li>• getReceivableAmount()</li> <li>• getPayableAmount()</li> <li>• calculateProfit()</li> <li>• calculateMonthlyProfit()</li> <li>• getDate()</li> </ul>
12	Order	<ul style="list-style-type: none"> <li>• getCash()</li> <li>• getPayableAmount()</li> <li>• getDate()</li> <li>• getItemID()</li> <li>• getAddress()</li> <li>• setPaymentMethod()</li> </ul>
13	notification	<ul style="list-style-type: none"> <li>• sendMessage()</li> <li>• receiveMessage()</li> <li>• storeMessage()</li> <li>• removeMessage()</li> </ul>
14	message	<ul style="list-style-type: none"> <li>• getId()</li> <li>• getSender ()</li> <li>• setSender ()</li> <li>• getReceiver()</li> <li>• setReceiver()</li> <li>• getType ()</li> <li>• setType ()</li> <li>• getDescription ()</li> <li>• setDescription ()</li> <li>• getDate ()</li> <li>• setDate ()</li> <li>• sendMessage()</li> <li>• getStatus()</li> <li>• changeStatus()</li> </ul>
15	Delivery	<ul style="list-style-type: none"> <li>• getCash()</li> <li>• getReceivableAmount()</li> <li>• getDate()</li> </ul>

		<ul style="list-style-type: none"> <li>• getOrderID()</li> <li>• getAddress()</li> </ul>
16	database	<ul style="list-style-type: none"> <li>• insert()</li> <li>• update()</li> <li>• delete()</li> <li>• search()</li> </ul>
17	search	<ul style="list-style-type: none"> <li>• searchByName()</li> <li>• searchByDateRange()</li> <li>• searchByWeek()</li> <li>• searchByMonth()</li> <li>• searchByYear()</li> </ul>
18	verification	<ul style="list-style-type: none"> <li>• checkAvailability()</li> <li>• isValidInput()</li> <li>• userNameVerification()</li> <li>• passwordVerification()</li> <li>• emailVerification()</li> <li>• phoneVerification()</li> <li>• matchEmail()</li> <li>• matchPhone()</li> <li>• matchPassword()</li> </ul>
19	Interface	<ul style="list-style-type: none"> <li>• showAuthenticationAction()</li> <li>• showMenu()</li> <li>• getInput()</li> <li>• getAction()</li> <li>• authenticate()</li> </ul>
20	Item	<ul style="list-style-type: none"> <li>• generateProductId()</li> <li>• searchProduct()</li> <li>• addProduct()</li> <li>• removeProduct()</li> <li>• calculatePrice()</li> <li>• getProductID()</li> <li>• getProductName()</li> <li>• getPrice()</li> <li>• getQuantity()</li> <li>• getExpiryDate()</li> <li>• getDeliveryDate()</li> <li>• getBrandName()</li> <li>• getTotalPrice()</li> <li>• setProductID()</li> <li>• setProductName()</li> <li>• setPrice()</li> <li>• setQuantity()</li> <li>• setExpiryDate()</li> <li>• setDeliveryDate()</li> <li>• setCompanyName()</li> <li>• setTotalPrice()</li> </ul>

## **Finalizing Classes**

### **Class Cards**

To identify the final classes, it was required to check if there can be any hierarchies, merges, additional attributes, methods or classes. These identifications are given below:

- a. There are four types of account in the system: Customer,admin,Employee,Owner . They all have common attributes and methods of account. So, the account class can be the parent class of the classes Customer,Admin, Employee
- b. Cost class has same attributes and methods as the classes Management Cost and Management Expenditure. Also, Cost class has more attributes and methods than the classes Management Cost and Management Expenditure. So, Management Cost and Management Expenditure can be merged with Transaction class.
3. As, Transactional information will be stored in Database, Transaction class will have an additional method (`insertInfo()`) to communicate with the Database class.
4. Message class has same attributes and methods as and more than the Notification class. So, Notification class can be merged with the Message class.
5. Verification class needs regular expression for verifying name, email, phone and password. So, `regexUsername`, `regexEmail`, `regexPhone` and `regexPassword` attributes are added.
6. As the Message class will be used to communicate with other Account, Authentication class will also send verification code and send password (recovery) through the Message class. So, Message call will have two additional methods: `sendVerificationCode()` and `sendPassword()`.
7. System and Interface classes will have an additional method (`viewOptions()`) so that it can be used to get to other methods.

Table 1: Class Card for System Class

System	
Attribute	Method
<ul style="list-style-type: none"> <li>• Database</li> <li>• Interface</li> </ul>	<ul style="list-style-type: none"> <li>• connectDatabase()</li> <li>• accessDatabase()</li> <li>• showInterface()</li> <li>• createInterface()</li> <li>• deleteData()</li> <li>• saveFiles()</li> <li>• pause()</li> <li>• DataAnlyze()</li> <li>• createNotification()</li> <li>• disconnectDatabase()</li> </ul>
Responsibilities	Collaboration
<ul style="list-style-type: none"> <li>• Handling database</li> <li>• Creating interface</li> <li>• Analyzing Data</li> </ul>	Account Database

Table 2: Class Card for Account Class

Account	
Attribute	Method
<ul style="list-style-type: none"> <li>• Database</li> <li>• Interface</li> <li>• Customer</li> <li>• Admin</li> </ul>	<ul style="list-style-type: none"> <li>• signUp()</li> <li>• validateInput()</li> <li>• setUser()</li> <li>• login()</li> <li>• findUser()</li> <li>• GetUser()</li> <li>• ChangeData()</li> <li>• RecoveryData()</li> </ul>
Responsibilities	Collaboration
<ul style="list-style-type: none"> <li>• Account Creation</li> <li>• Getting Input for sign up &amp; sign in</li> <li>• Log Out</li> <li>• Recovery Data</li> <li>• Change Info</li> </ul>	Customer Admin Database

Table 3: Class Card for Customer Class

Customer	
Attribute	Method
<ul style="list-style-type: none"> <li>• fullName</li> <li>• currentAddress</li> <li>• Occupation</li> <li>• Contact</li> <li>• Facebook ID</li> <li>• Email</li> <li>• Customer Type</li> <li>• Online Shopping</li> </ul>	<ul style="list-style-type: none"> <li>• validateInput()</li> <li>• verifyUser()</li> <li>• recoverAccount()</li> <li>• createNotification()</li> <li>• recordTransaction()</li> <li>• recordDealingInformation()</li> <li>• updateStock()</li> <li>• logout()</li> <li>• getAccountType()</li> <li>• setCurrentAddress()</li> <li>• setContact()</li> <li>• getFullName()</li> <li>• getFacebookID()</li> <li>• getEmail()</li> <li>• onlinePurchase()</li> </ul>
Responsibilities	Collaboration
<ul style="list-style-type: none"> <li>• Online Shopping</li> <li>• Browsing Inventory</li> <li>• Customer information Storing</li> </ul>	Database Admin

Table 4: Class Card for Admin Class

Admin	
Attribute	Method
<ul style="list-style-type: none"> <li>• fullName</li> <li>• permanentAddress</li> <li>• currentAddress</li> <li>• Branch</li> <li>• NID</li> <li>• Contact</li> <li>• Account</li> <li>• userID</li> <li>• EmailID</li> </ul>	<ul style="list-style-type: none"> <li>• approveAccount()</li> <li>• addAdmin ()</li> <li>• addEmployee()</li> <li>• removeAdmin ()</li> <li>• removeEmployee()</li> <li>• updateAdminInformation()</li> <li>• updateEmployeeInformation()</li> <li>• includeProduct()</li> <li>• excludeProduct()</li> <li>• viewReport()</li> <li>• provideSalary()</li> <li>• announcePromotions()</li> <li>• BlockCustomer()</li> <li>• OrderAnalysis()</li> </ul>
Responsibilities	Collaboration
<ul style="list-style-type: none"> <li>• Managing customer Info</li> <li>• Supervising product Info</li> <li>• Managing Inventory</li> <li>• Transaction Management</li> <li>• Authentication</li> </ul>	Item Cost Financial Report Verification Database

Table 5 : Class Card for Transaction Class

Transaction	
Attribute	Method
<ul style="list-style-type: none"> <li>• TId</li> <li>• Amount</li> <li>• Cost</li> <li>• receipt</li> <li>• transactionType</li> <li>• userId</li> </ul>	<ul style="list-style-type: none"> <li>• generateId()</li> <li>• createReceipt()</li> <li>• calculatePrice()</li> <li>• createTransaction()</li> <li>• paidAmount()</li> <li>• determineDealingAmount()</li> <li>• updateCash()</li> <li>• updateFinancialReport()</li> <li>• processDealingInformation()</li> <li>• updateStock()</li> </ul>
Responsibilities	Collaboration
<ul style="list-style-type: none"> <li>• Recording Transactional Info</li> <li>• updating cash</li> <li>• Payment Procedure</li> </ul>	Financial report Item Order Delivery

Table 6: Class Card for Cost Class

Cost	
Attribute	Method
<ul style="list-style-type: none"> <li>• Amount</li> <li>• Quantity</li> <li>• Payment System</li> </ul>	<ul style="list-style-type: none"> <li>• addProduct()</li> <li>• removeProduct()</li> <li>• calculateAmount()</li> <li>• printReceipt()</li> <li>• getProduct()</li> <li>• setAmount()</li> </ul>
Responsibilities	Collaboration
<ul style="list-style-type: none"> <li>• calculating amount of each item</li> <li>• containing information of sold products</li> </ul>	Item Database Financial Report

Table 7: Class Card for Employee Class

<b>Employee</b>	
Attributes	Methods
<ul style="list-style-type: none"> <li>• Salary</li> <li>• Commission</li> <li>• Sales</li> </ul>	<ul style="list-style-type: none"> <li>• getSalary()</li> <li>• setSalary()</li> <li>• sellProduct()</li> <li>• getCommision()</li> <li>• DeliveryProduct()</li> </ul>
Responsibilities	Collaboration
<ul style="list-style-type: none"> <li>• Sell Product</li> <li>• Product Delivery</li> <li>• Maintain Customers Need</li> </ul>	Admin Item

Table 9: Class Card for Delivery Class

Delivery	
Attribute	Method
<ul style="list-style-type: none"> <li>• orderID</li> <li>• amount</li> <li>• date</li> <li>• Contact</li> </ul>	<ul style="list-style-type: none"> <li>• getCash()</li> <li>• getReceivableAmount()</li> <li>• getDate()</li> <li>• getOrderID()</li> <li>• getAddress()</li> </ul>
Responsibilities	Collaboration
<ul style="list-style-type: none"> <li>• Delivery items to customers</li> </ul>	Customer Order

Table 10: Class Card for Financial Report Class

Financial Report	
Attribute	Method
<ul style="list-style-type: none"> <li>• cashID</li> <li>• amount</li> <li>• date</li> <li>• profit</li> </ul>	<ul style="list-style-type: none"> <li>• showReport()</li> <li>• getCash()</li> <li>• getReceivableAmount()</li> <li>• getPayableAmount()</li> <li>• calculateProfit()</li> <li>• calculateMonthlyProfit()</li> <li>• getDate()</li> </ul>
Responsibilities	Collaboration
<ul style="list-style-type: none"> <li>• calculating profit</li> <li>• providing report on daily transaction</li> </ul>	Admin System

Table : Class Card for Financial Database Class

<b>Database</b>	
Attribute	Method
<ul style="list-style-type: none"> <li>• isConnected</li> <li>• tableName</li> </ul>	<ul style="list-style-type: none"> <li>• insert()</li> <li>• update()</li> <li>• delete()</li> <li>• search()</li> </ul>
Responsibilities	Collaboration
<ul style="list-style-type: none"> <li>• Insert element</li> <li>• Update element</li> <li>• Delete element</li> <li>• Search element</li> </ul>	System

Table 11: Class Card for Order Class

<b>Order</b>	
Attribute	Method
<ul style="list-style-type: none"> <li>• itemID</li> <li>• amount</li> <li>• date</li> <li>• Contact</li> <li>• Payment Method</li> </ul>	<ul style="list-style-type: none"> <li>• getCash()</li> <li>• getPayableAmount()</li> <li>• getDate()</li> <li>• getItemID()</li> <li>• getAddress()</li> <li>• setPaymentMethod()</li> </ul>
Responsibilities	Collaboration
<ul style="list-style-type: none"> <li>• Getting Orders</li> </ul>	Customer Delivery

Table 12: Class Card for Item Class

Item	
Attribute	Method
<ul style="list-style-type: none"> <li>• productid</li> <li>• productName</li> <li>• price</li> <li>• quantity</li> <li>• deliveryDate</li> <li>• brandName</li> <li>• totalPrice</li> </ul>	<ul style="list-style-type: none"> <li>• generateProductId()</li> <li>• searchProduct()</li> <li>• addProduct()</li> <li>• removeProduct()</li> <li>• calculatePrice()</li> <li>• getProductID()</li> <li>• getProductName()</li> <li>• getPrice()</li> <li>• getQuantity()</li> <li>• getExpiryDate()</li> <li>• getDeliveryDate()</li> <li>• getBrandName()</li> <li>• getTotalPrice()</li> <li>• setProductID()</li> <li>• setProductName()</li> <li>• setPrice()</li> <li>• setQuantity()</li> <li>• setExpiryDate()</li> <li>• setDeliveryDate()</li> <li>• setCompanyName()</li> <li>• setTotalPrice()</li> </ul>
Responsibilities	Collaboration
<ul style="list-style-type: none"> <li>• Calculating total involve amount in a transaction</li> </ul>	

Table 13: Class Card for Search Class

Notification	
Attribute	Method
<ul style="list-style-type: none"> <li>notificationId</li> <li>Time</li> <li>Message</li> <li>Contact</li> </ul>	<ul style="list-style-type: none"> <li>sendMessage()</li> <li>receiveMessage()</li> <li>storeMessage()</li> <li>removeMessage()</li> </ul>
Responsibilities	Collaboration

Search	
Attribute	Method
<ul style="list-style-type: none"> <li>User</li> <li>Product</li> <li>Transaction</li> <li>Date</li> </ul>	<ul style="list-style-type: none"> <li>searchByName()</li> <li>searchByDateRange()</li> <li>searchByWeek()</li> <li>searchByMonth()</li> <li>searchByYear()</li> </ul>
Responsibilities	Collaboration
<ul style="list-style-type: none"> <li>Information providing</li> </ul>	

Table 13: Class Card for Search Class

Search	
Attribute	Method
<ul style="list-style-type: none"> <li>User</li> <li>Product</li> <li>Transaction</li> <li>Date</li> </ul>	<ul style="list-style-type: none"> <li>searchByName()</li> <li>searchByDateRange()</li> <li>searchByWeek()</li> <li>searchByMonth()</li> <li>searchByYear()</li> </ul>
Responsibilities	Collaboration
<ul style="list-style-type: none"> <li>Information providing</li> </ul>	

Table 14: Class Card for Notification Class

Notification	
Attribute	Method
<ul style="list-style-type: none"> <li>notificationId</li> <li>Time</li> <li>Message</li> <li>Contact</li> </ul>	<ul style="list-style-type: none"> <li>sendMessage()</li> <li>receiveMessage()</li> <li>storeMessage()</li> <li>removeMessage()</li> </ul>
Responsibilities	Collaboration
<ul style="list-style-type: none"> <li>managing notification system</li> <li>storing message</li> </ul>	System Database Customer Admin

Table 15: Class Card for Verification Class

Verification	
Attributes	Methods
<ul style="list-style-type: none"> <li>regexUsername</li> <li>regexEmail</li> <li>regexPhone</li> <li>regexPassword</li> <li>email</li> <li>phone</li> <li>password</li> </ul>	<ul style="list-style-type: none"> <li>checkAvailability()</li> <li>isValidInput()</li> <li>userNameVerification()</li> <li>passwordVerification()</li> <li>emailVerification()</li> <li>phoneVerification()</li> <li>matchEmail()</li> <li>matchPhone()</li> <li>matchPassword()</li> </ul>
Responsibilities	Collaborative class
<ul style="list-style-type: none"> <li>Input Validation</li> <li>Availability Checking</li> <li>User Verification</li> </ul>	Database System

# Chapter 7

## Behavioral Modeling

### 7.1 Data Flow Diagram

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated.

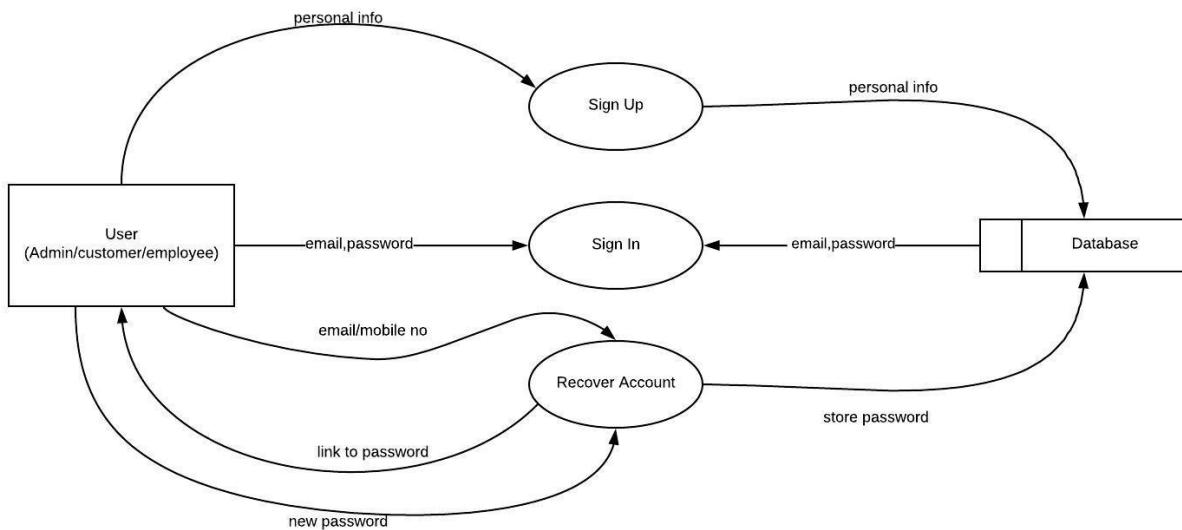


Figure: 2.1

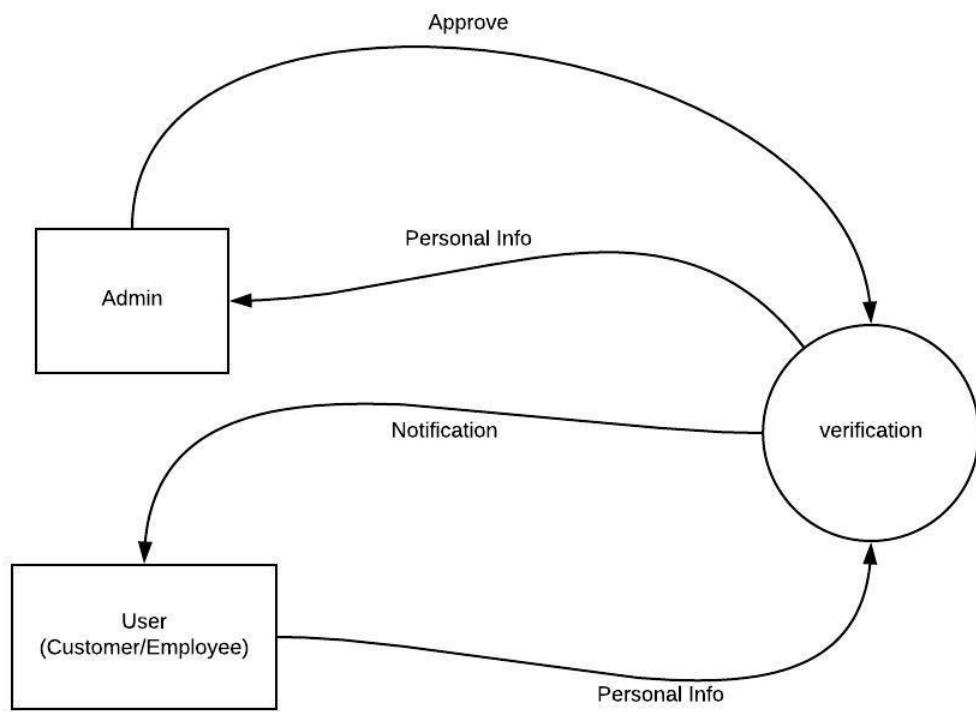


Figure : 2.1.1

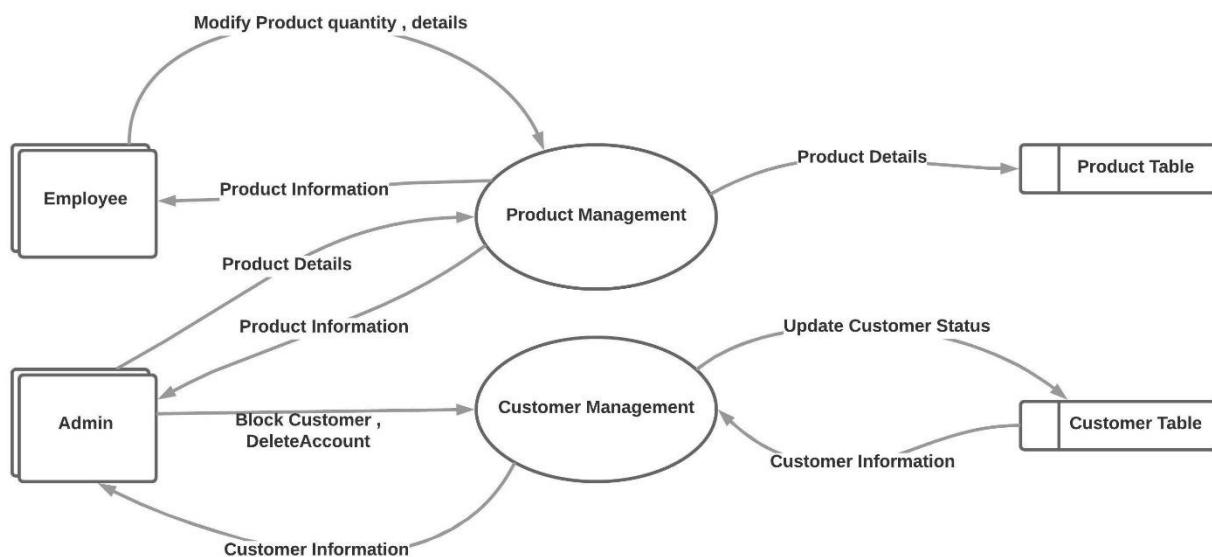
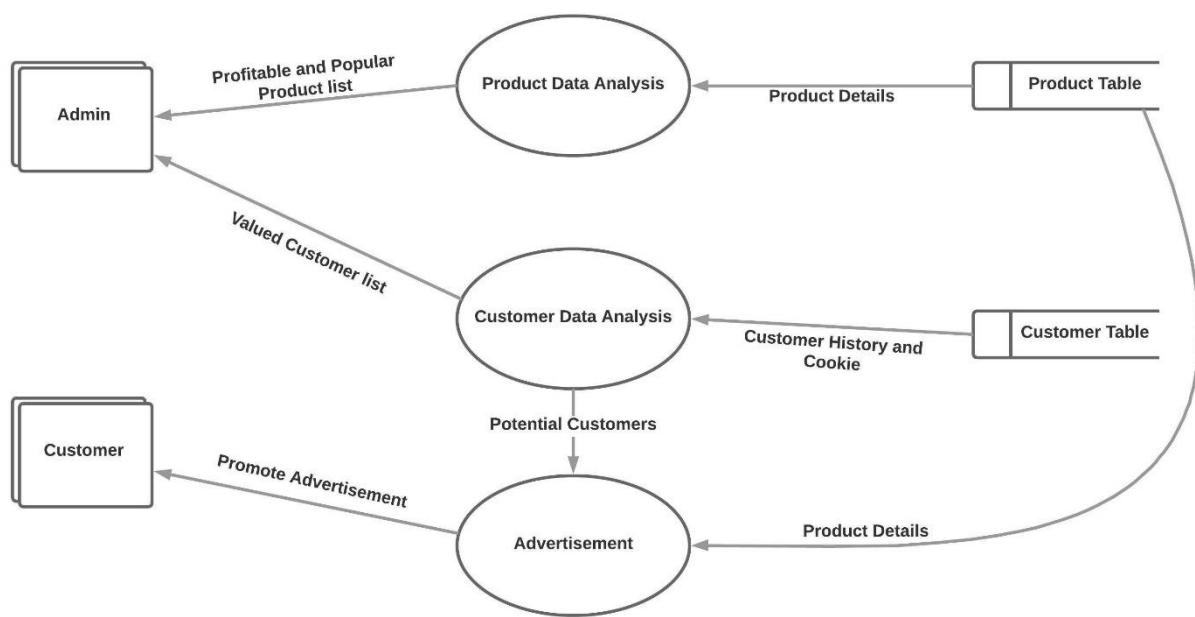
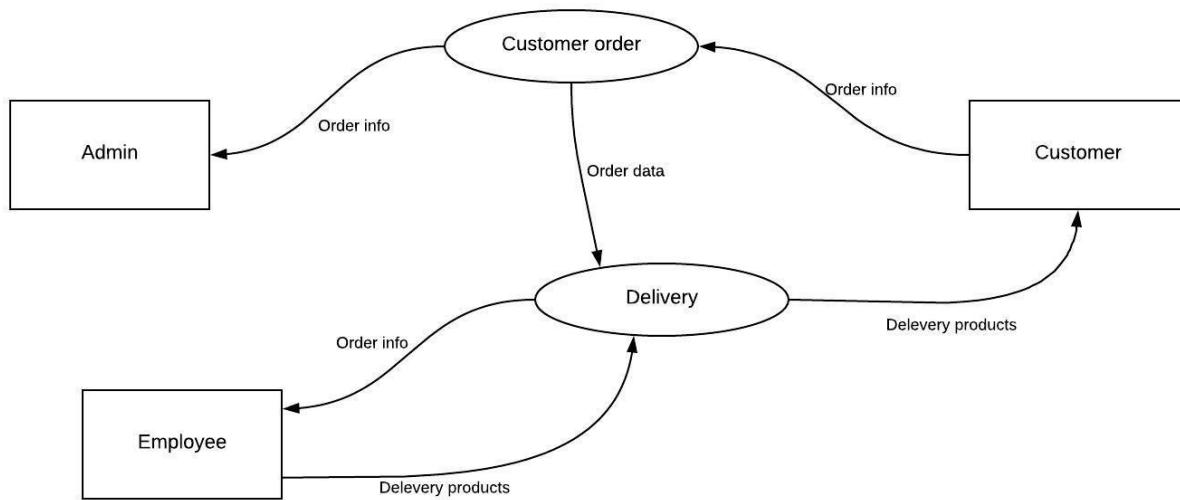


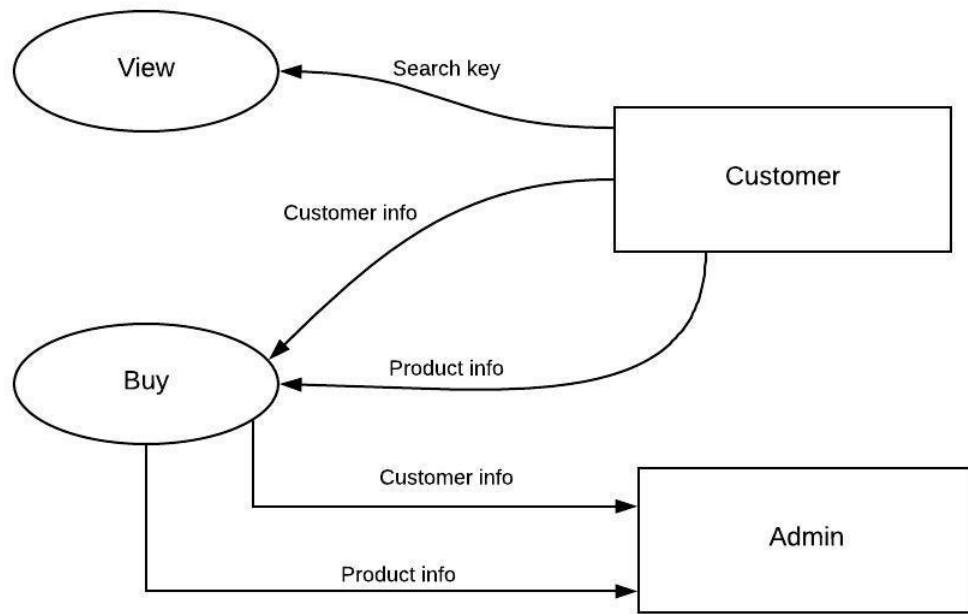
Figure : 2.2



**Figure: 2.3**



**Figure: 2.4**



**Figure: 2.4.1**

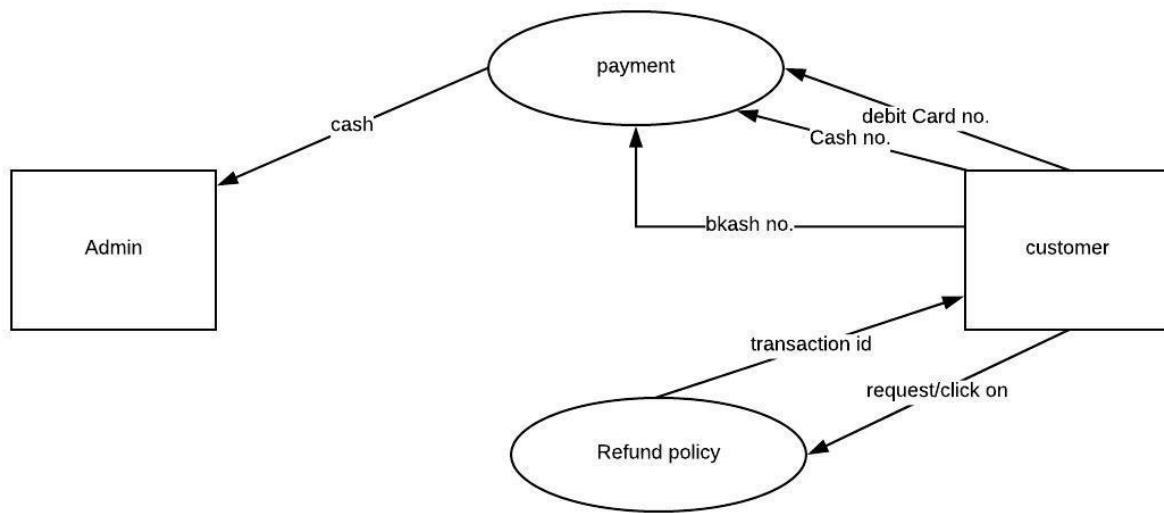


Figure : 2.5

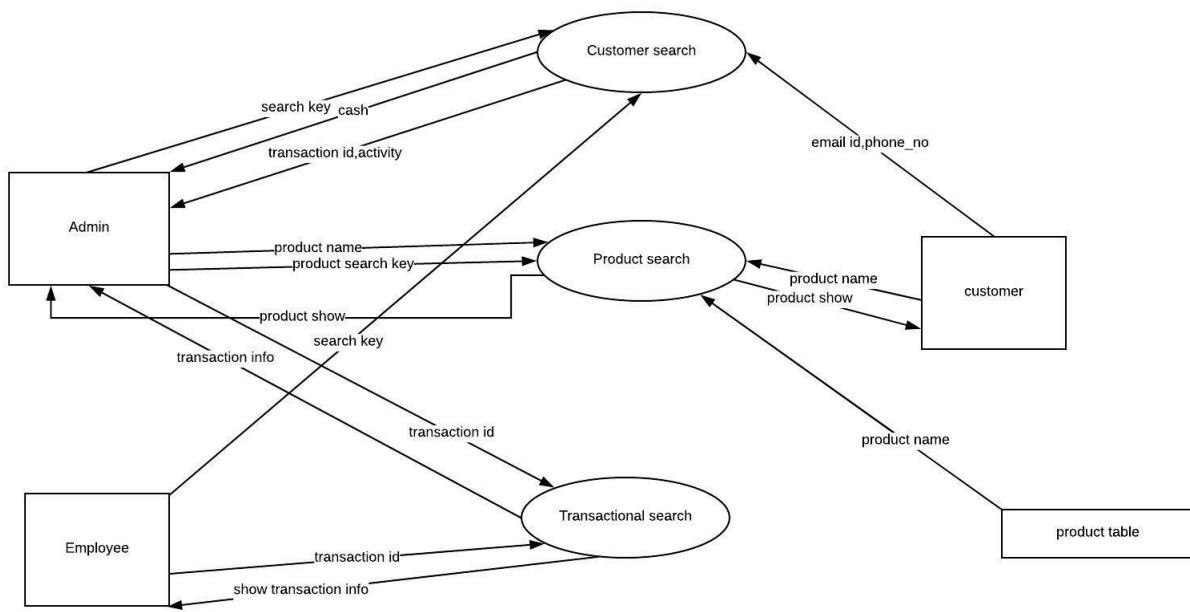


Figure : 2.6

## 7.2 State Transition Diagram

State diagram represents active states for each class the events (triggers). For this we identified all the events, their initiators and collaborators

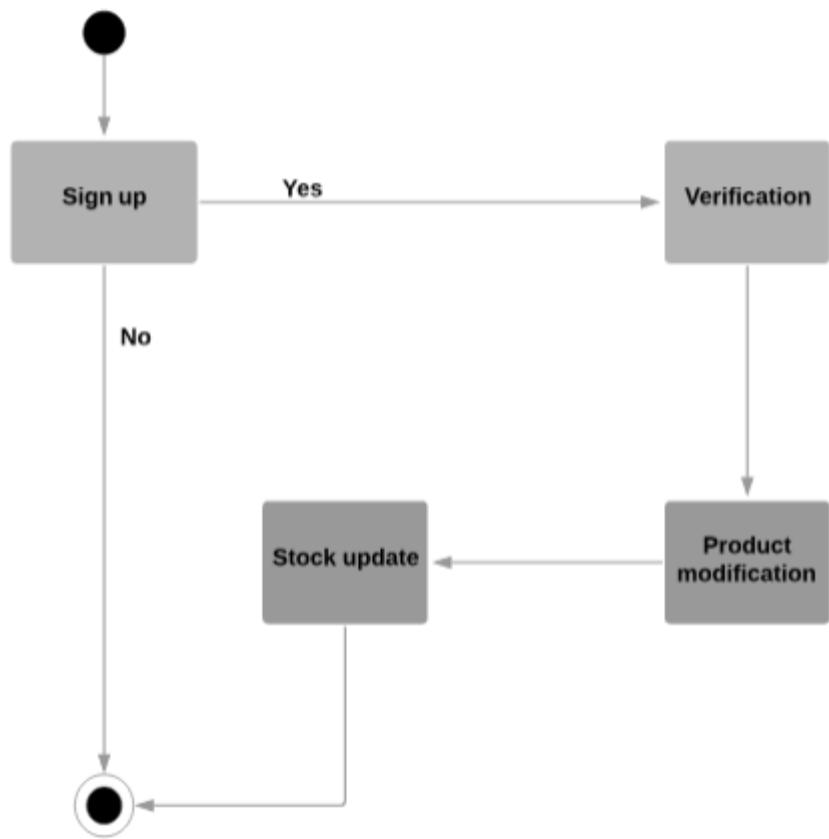


Figure: Admin Class

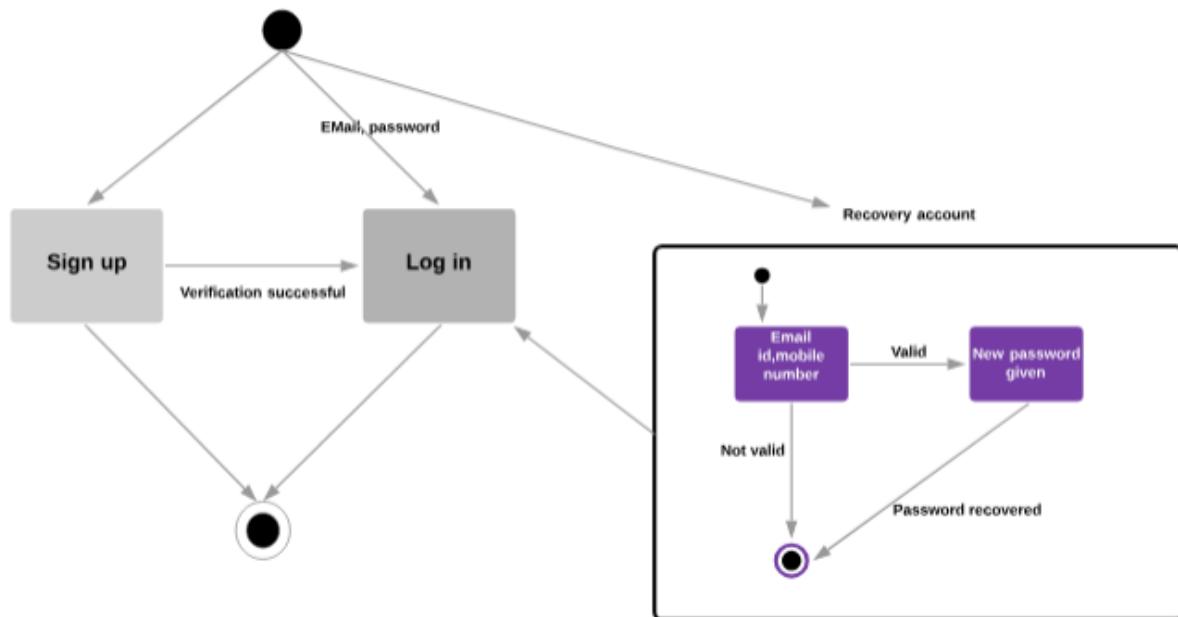


Figure: Authentication

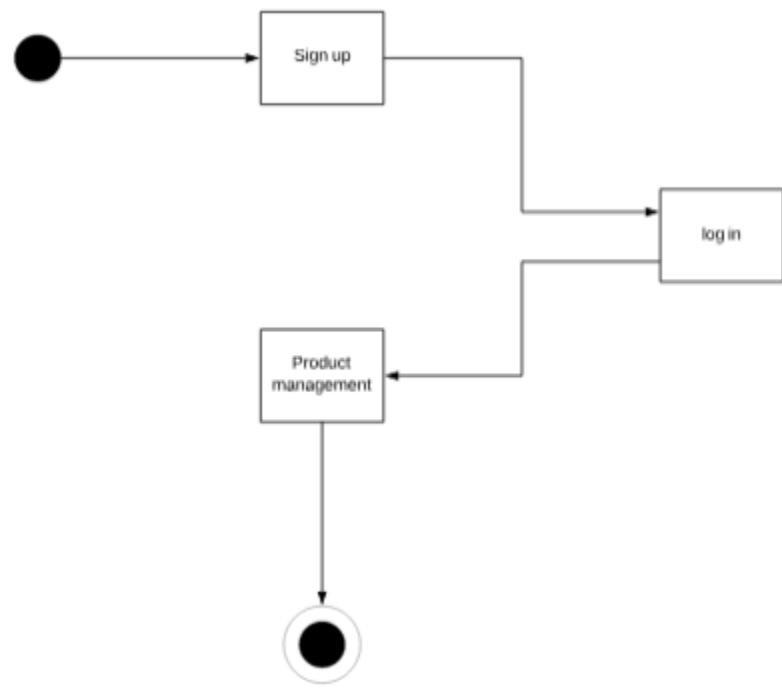


Figure: Customer

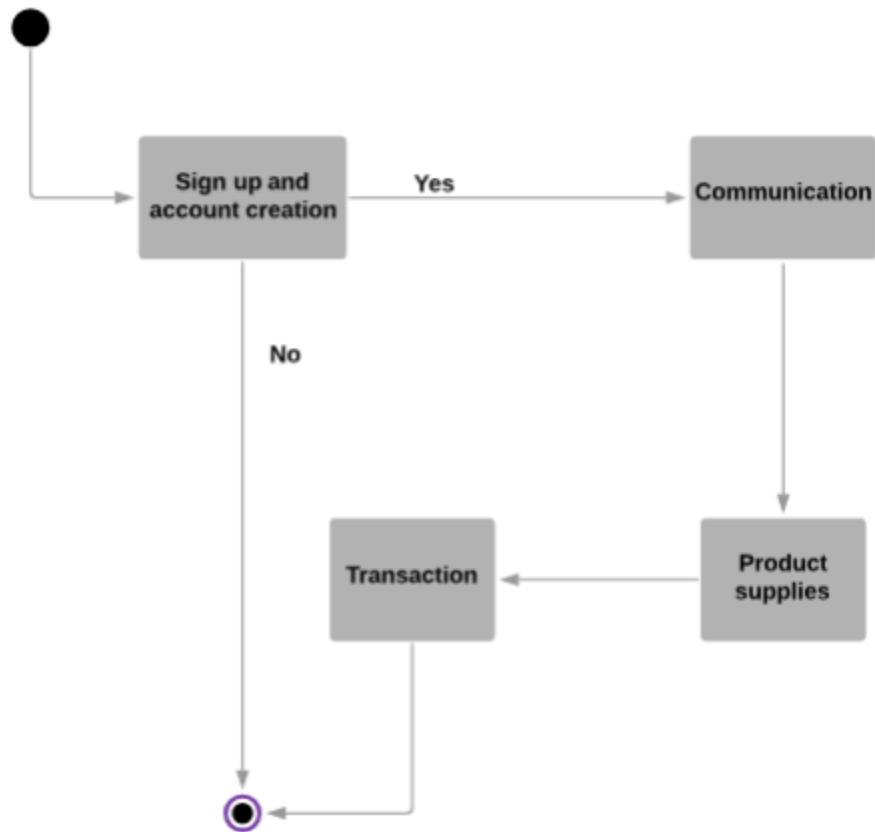


Figure: Employee

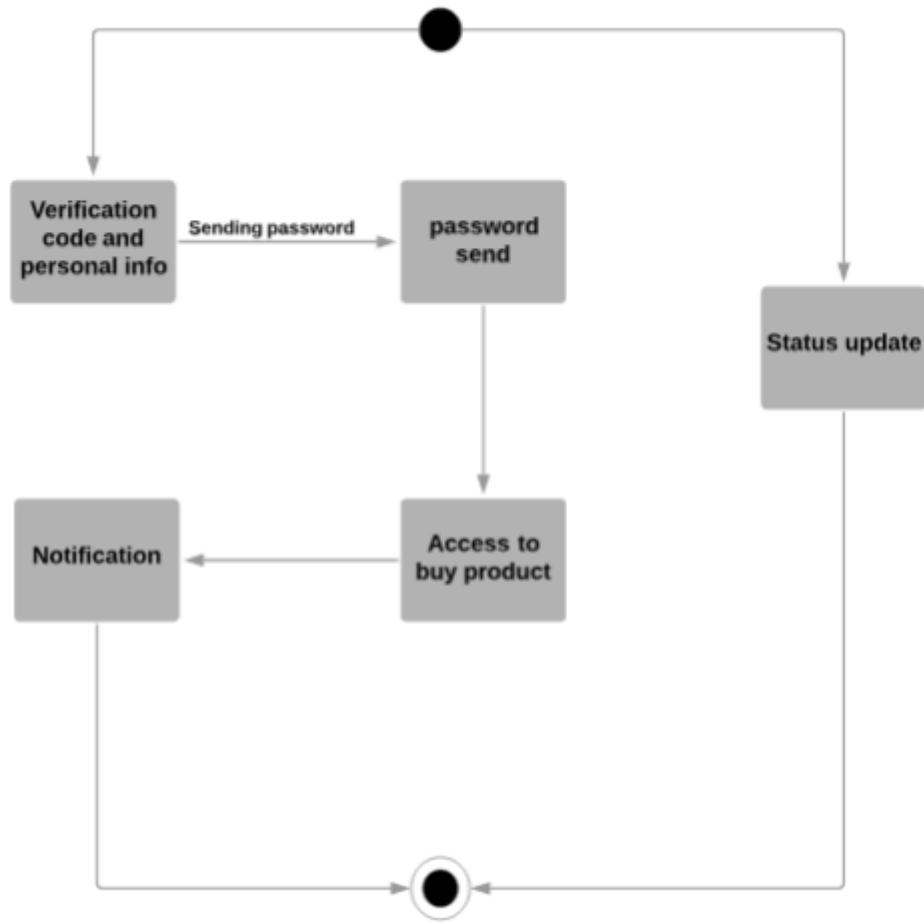


Figure: Message

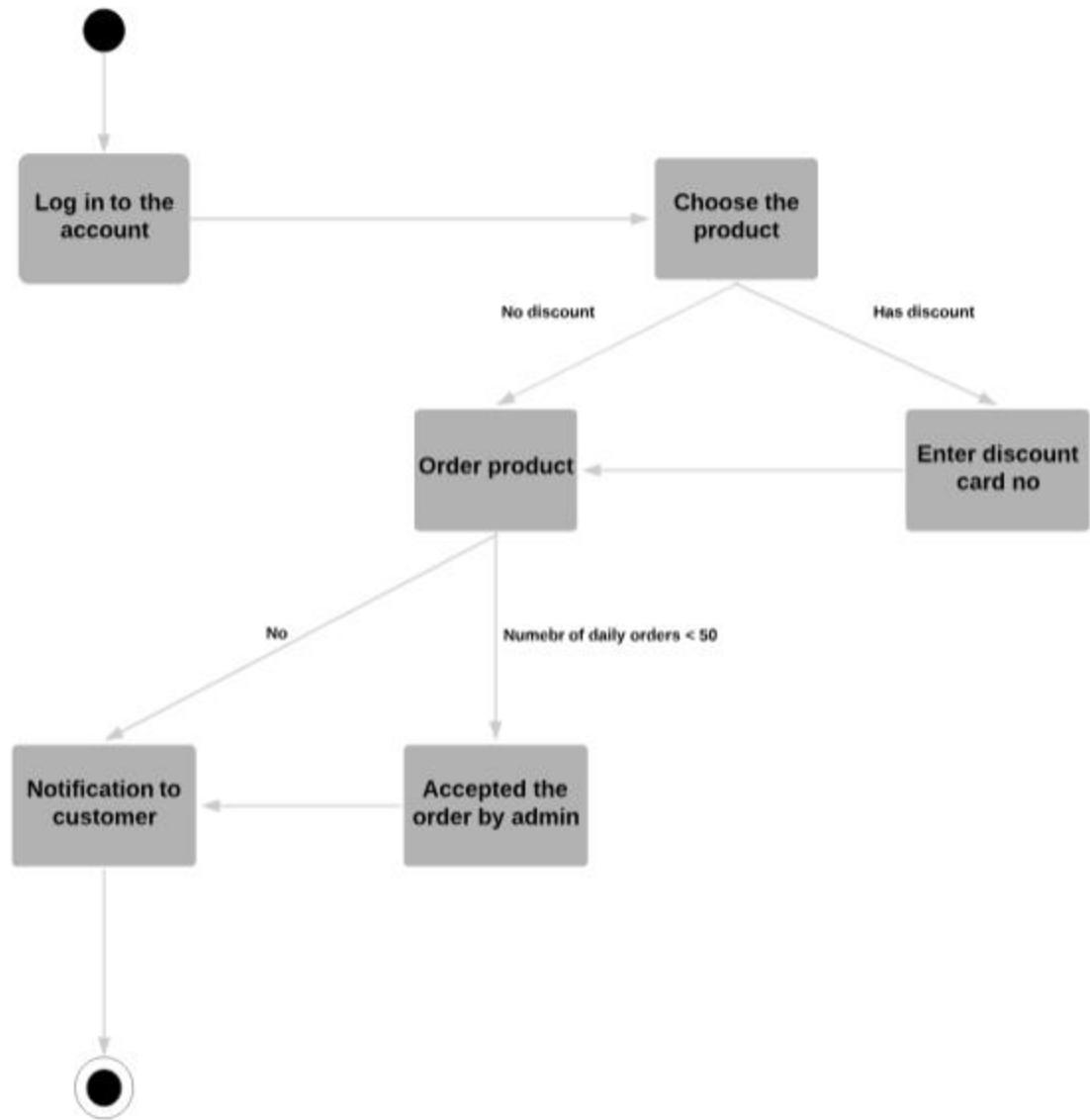


Figure: Order Product

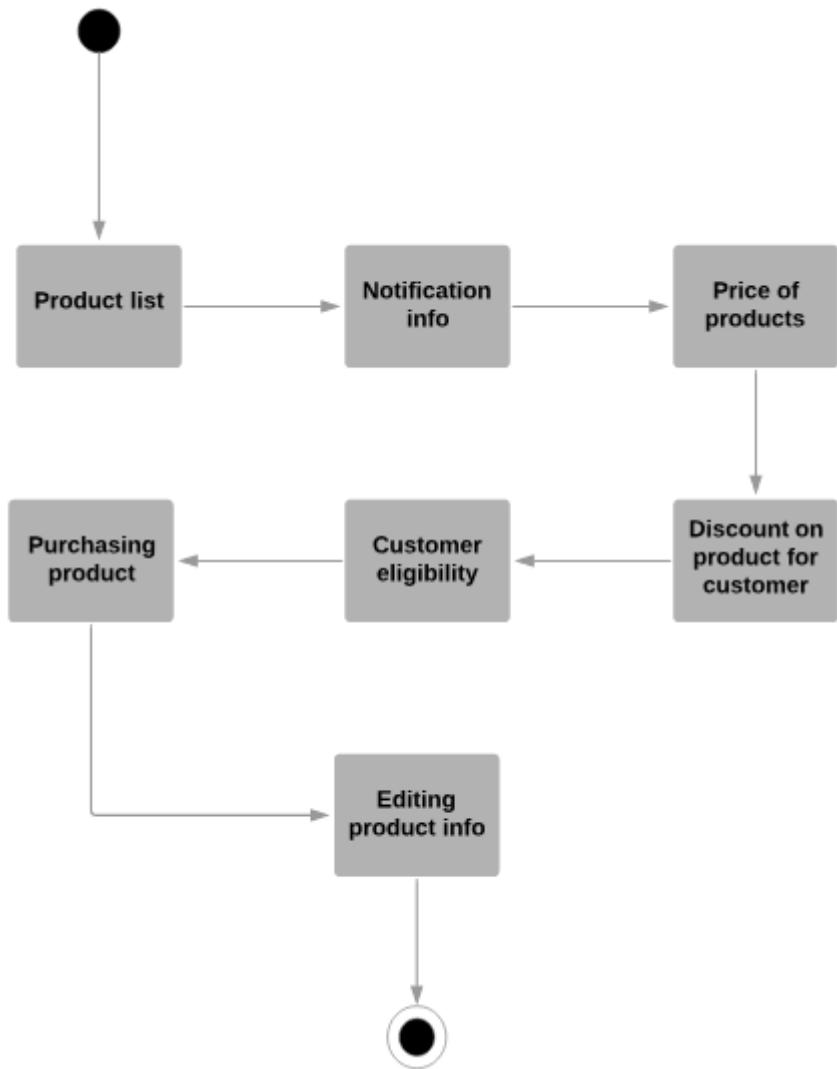


Figure: Product

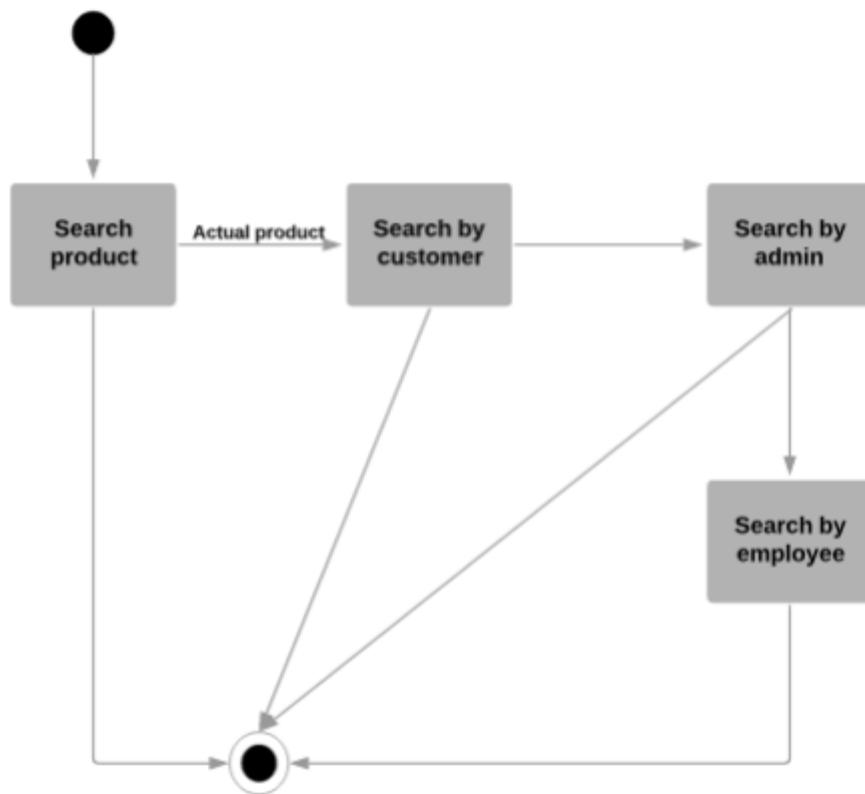


Figure: Sear

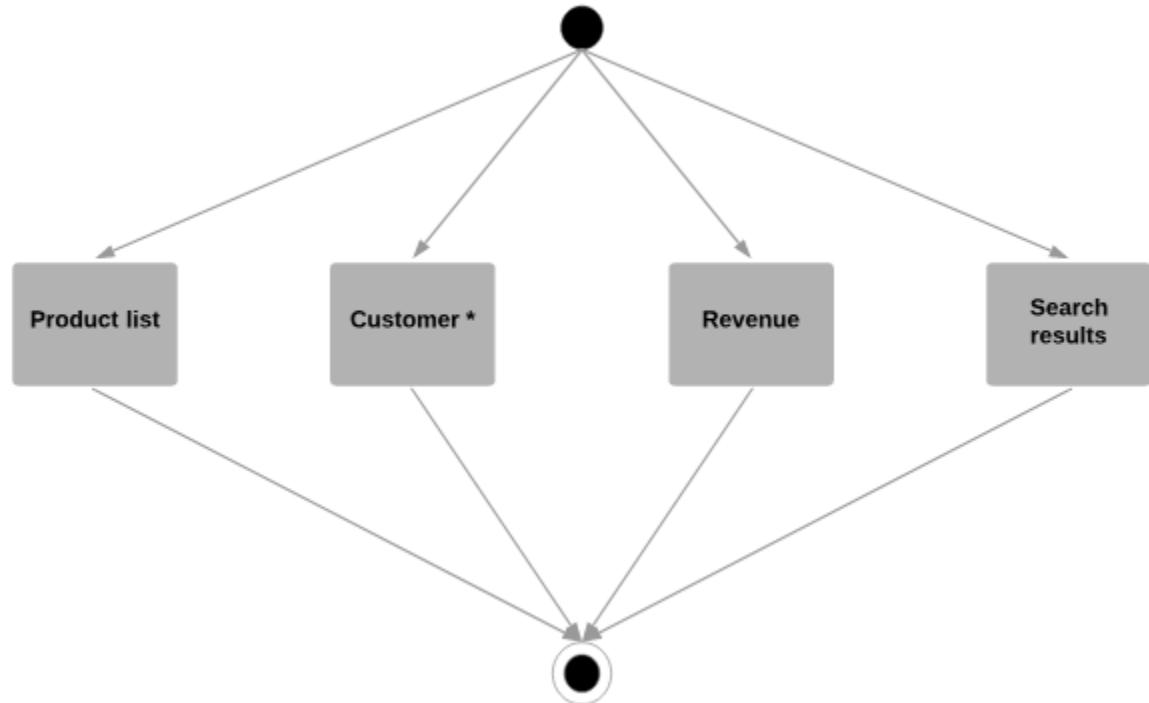
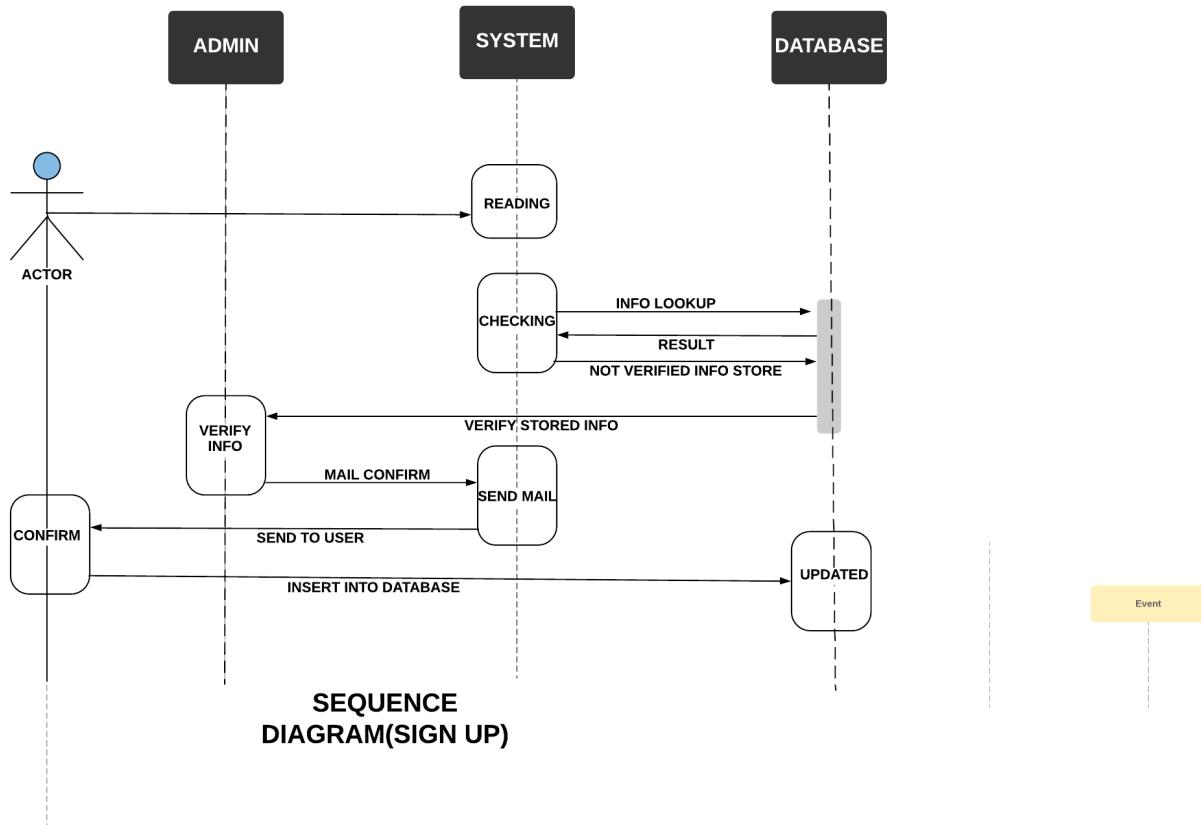
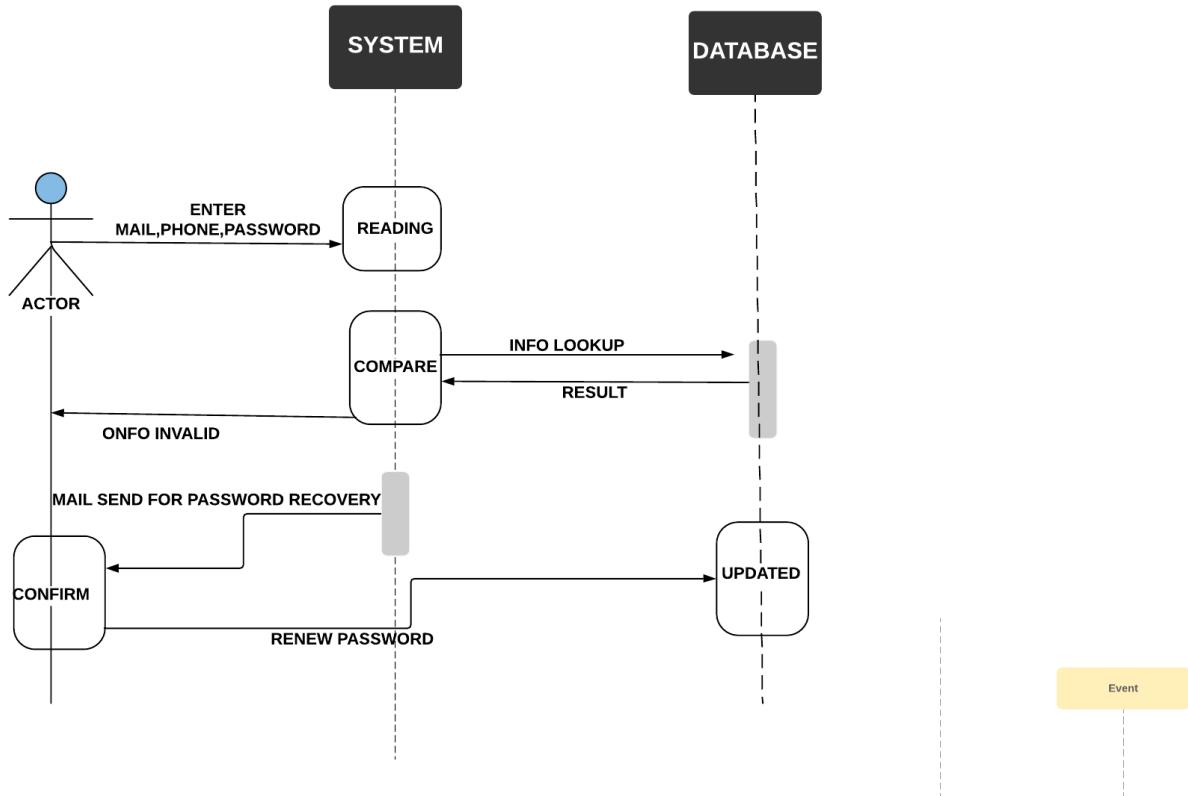
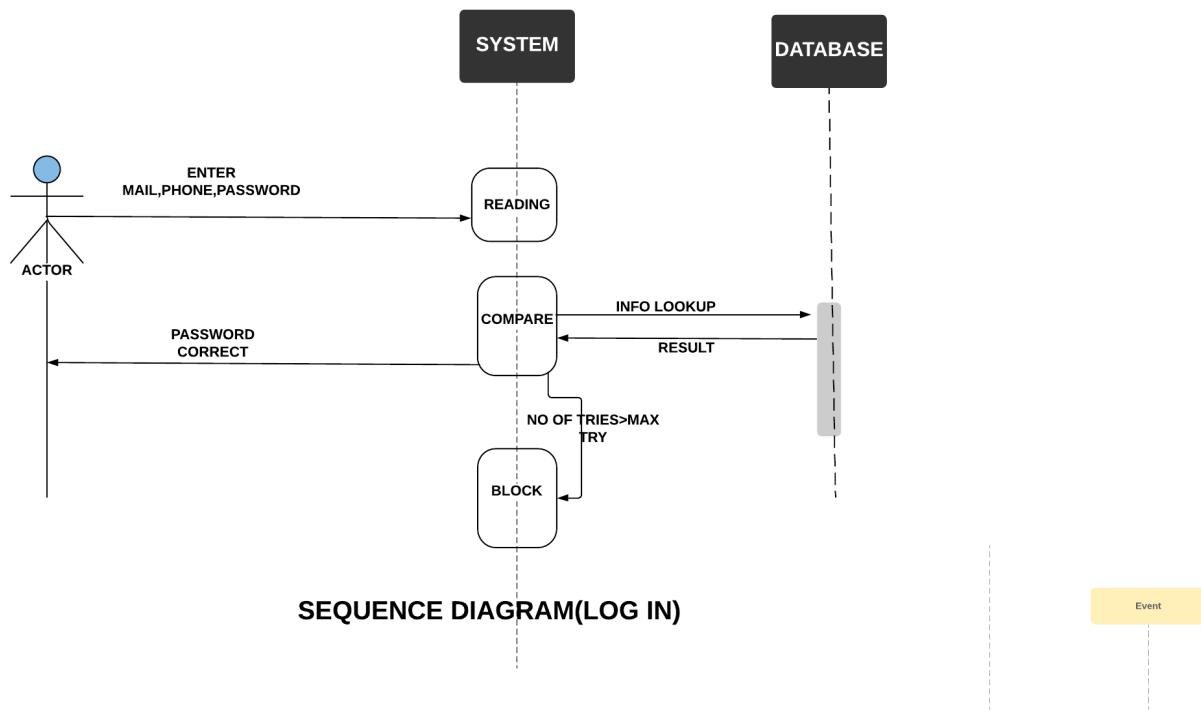
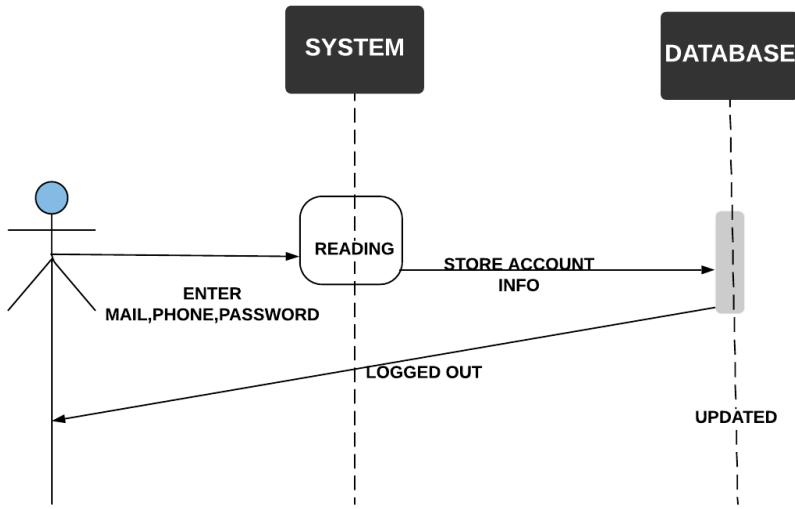


Figure: System

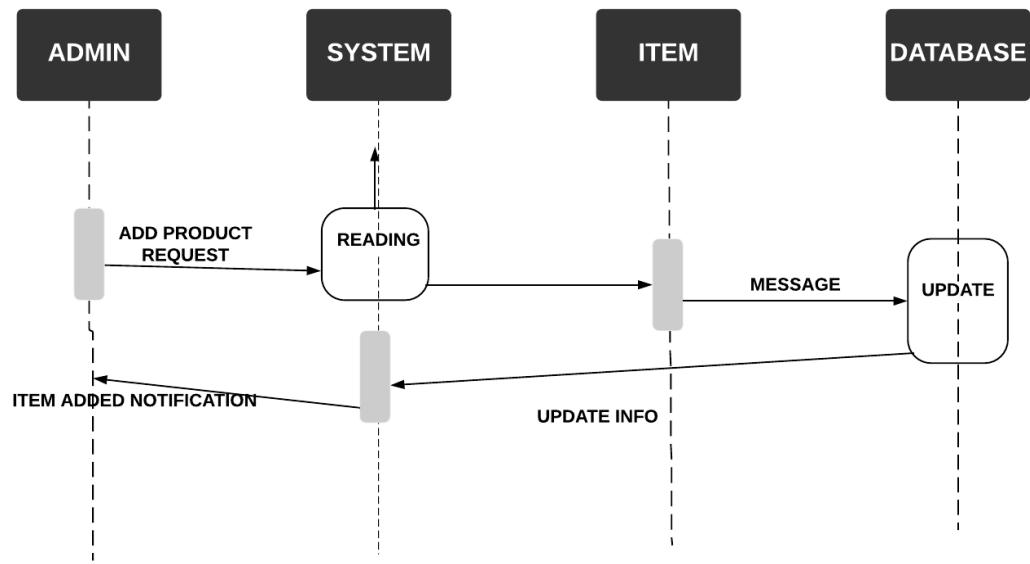
### 7.3 Sequence Diagram of f-commerce Management System





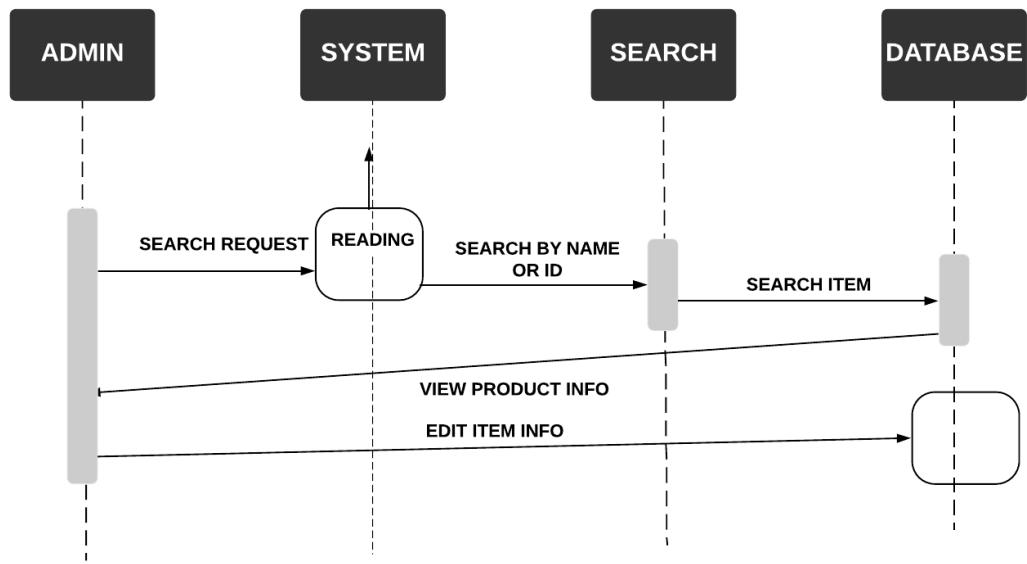


SEQUENCE DIAGRAM(SIGN OUT)



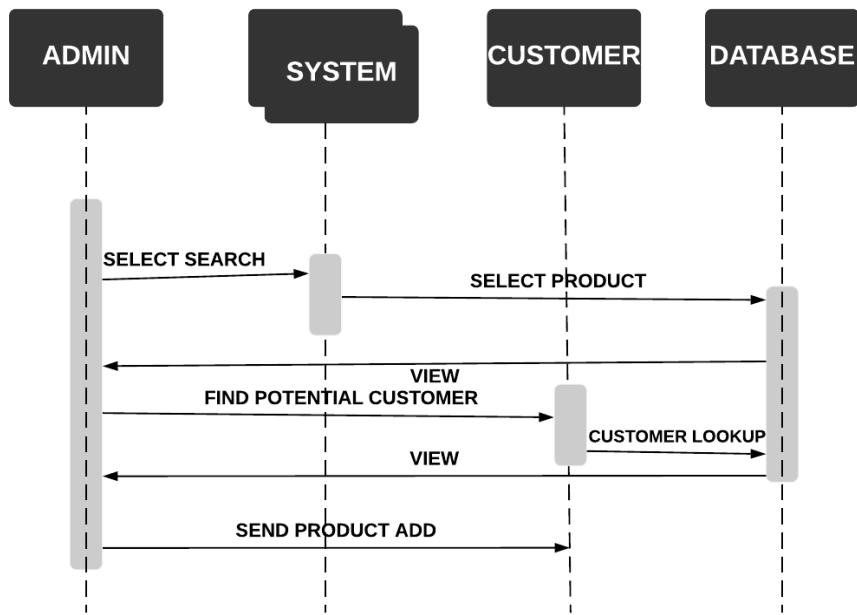
**SEQUENCE DIAGRAM(ADD ITEM)**

Event

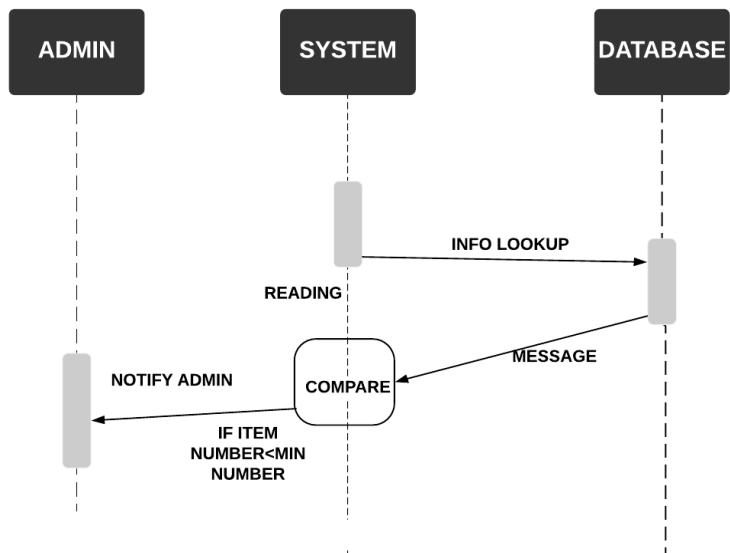


### SEQUENCE DIAGRAM (EDIT ITEM)

Event

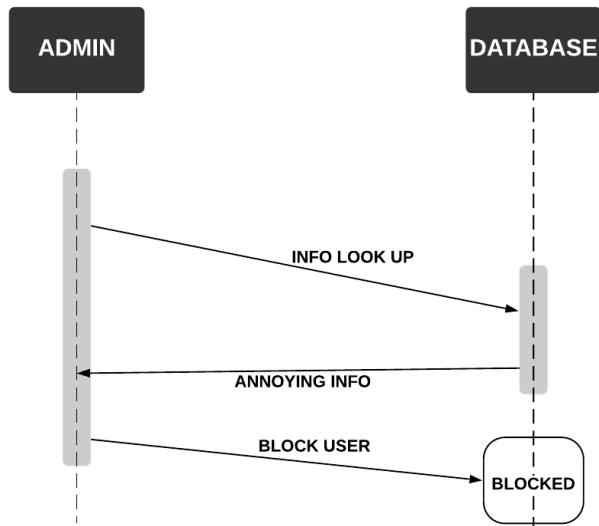


SEQUENCE DIAGRAM (ADD PROMOTION)



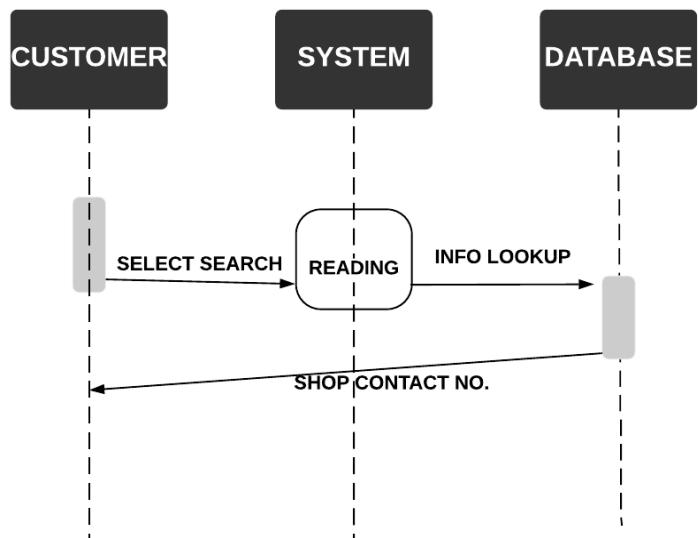
**SEQUENCE DIAGRAM (STOCK UPDATE)**

Event

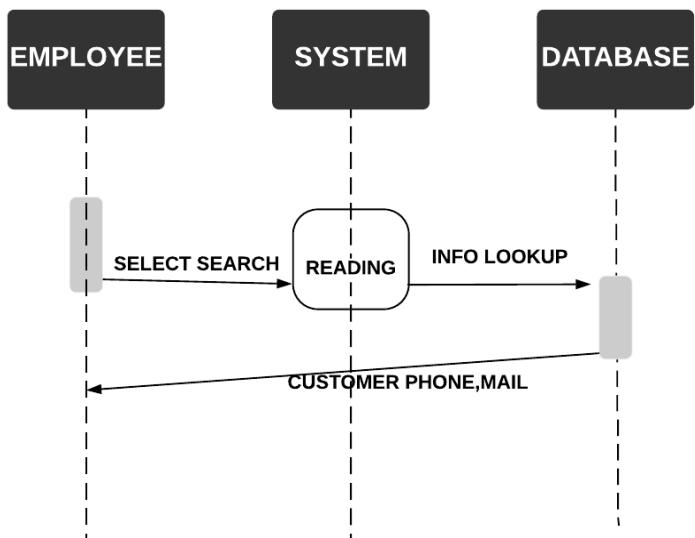


SEQUENCE DIAGRAM (BLOCK CUSTOMER)

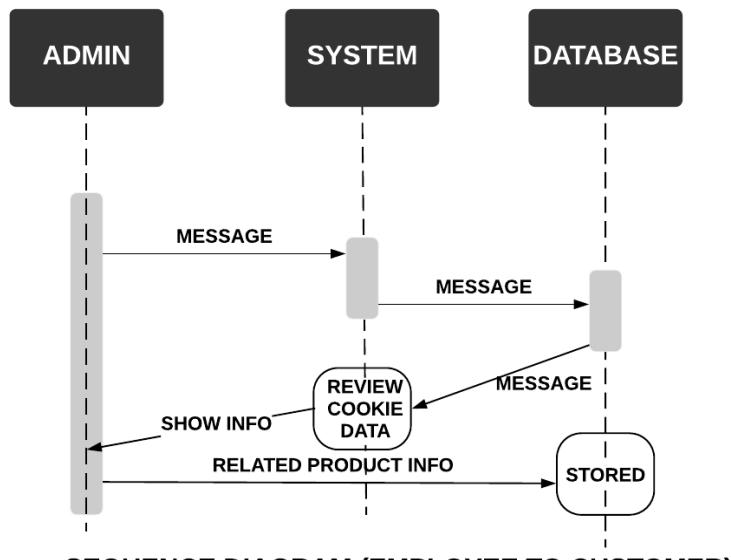




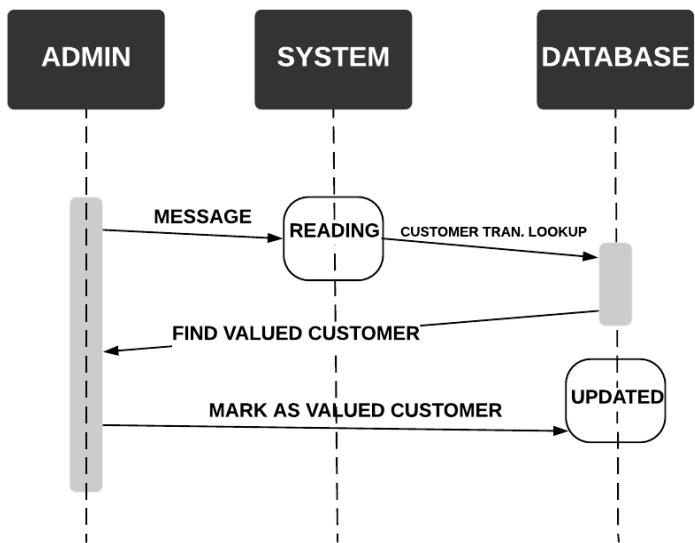
SEQUENCE DIAGRAM (CUSTOMER TO EMPLOYEE)



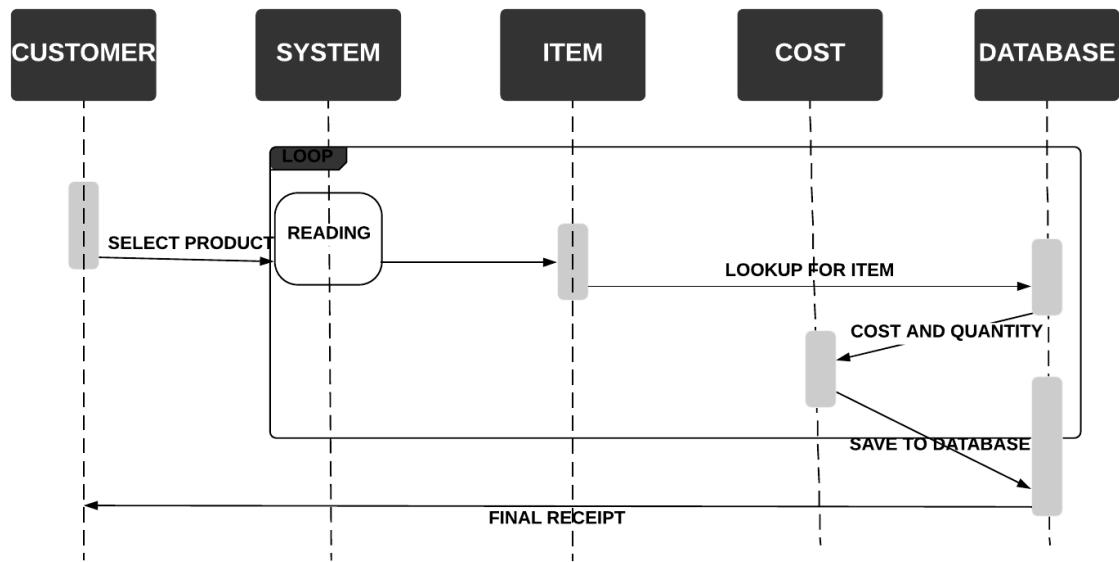
SEQUENCE DIAGRAM (EMPLOYEE TO CUSTOMER)



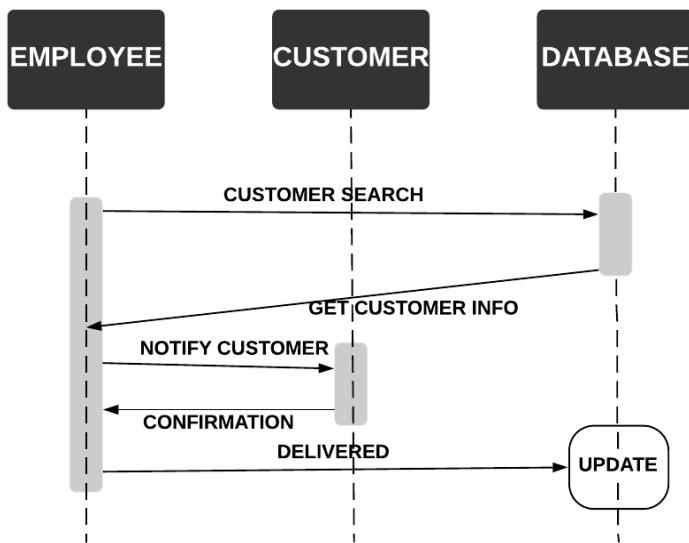
SEQUENCE DIAGRAM (EMPLOYEE TO CUSTOMER)



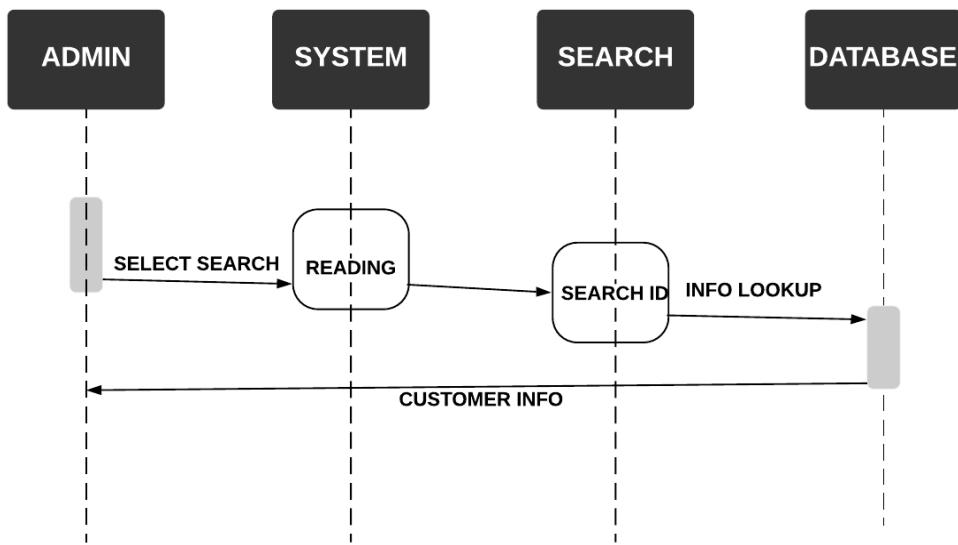
SEQUENCE DIAGRAM(CUSTOMER HISTORY ANALYSIS)



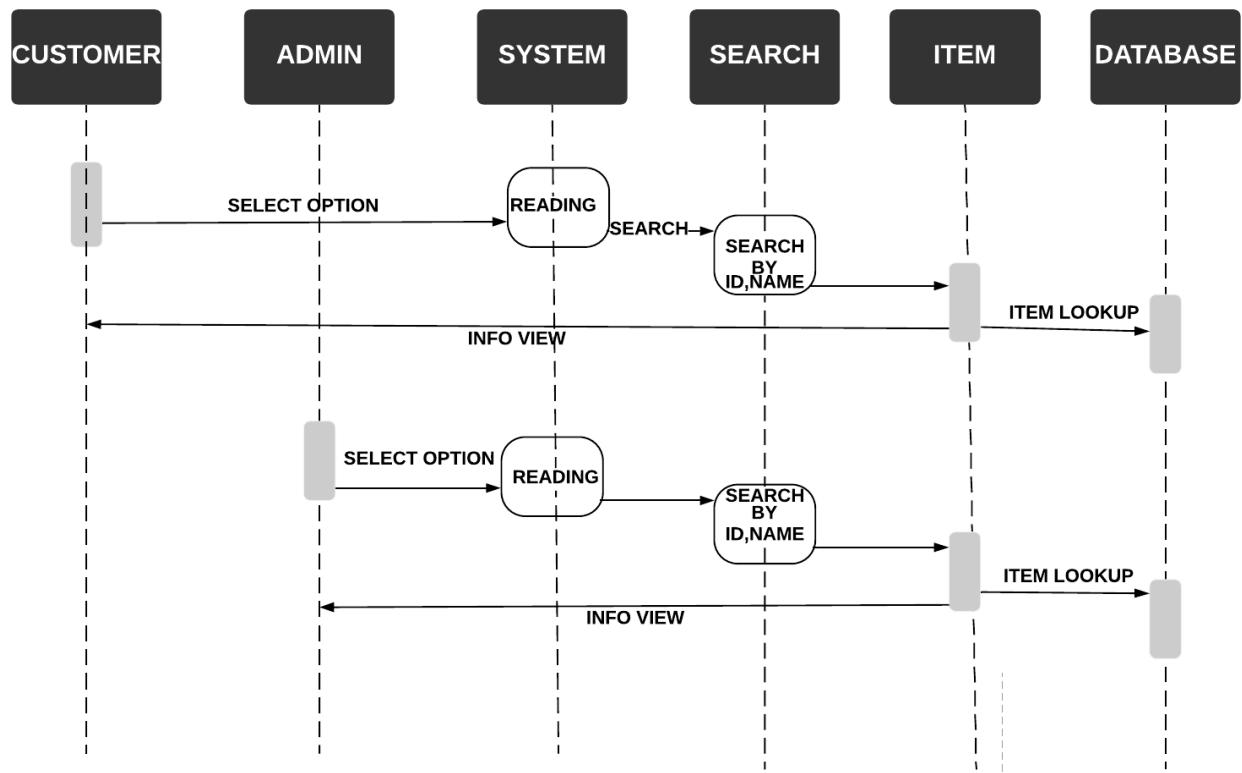
Event



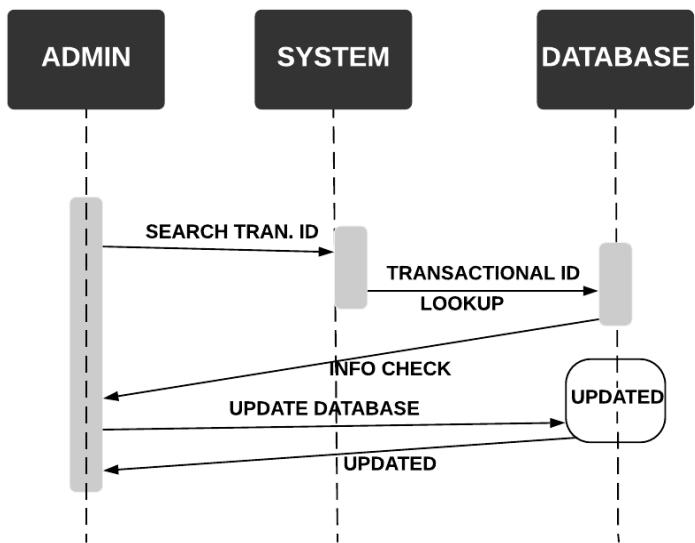
SEQUENCE DIAGRAM (EMPLOYEE TO CUSTOMER)



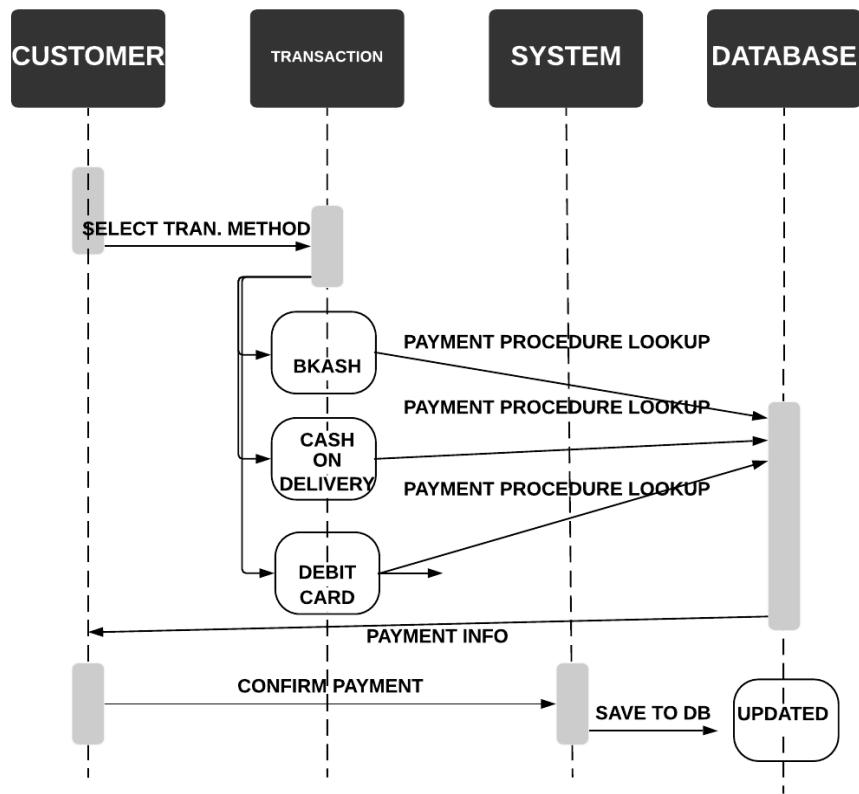
SEQUENCE DIAGRAM (CUSTOMER SEARCH)



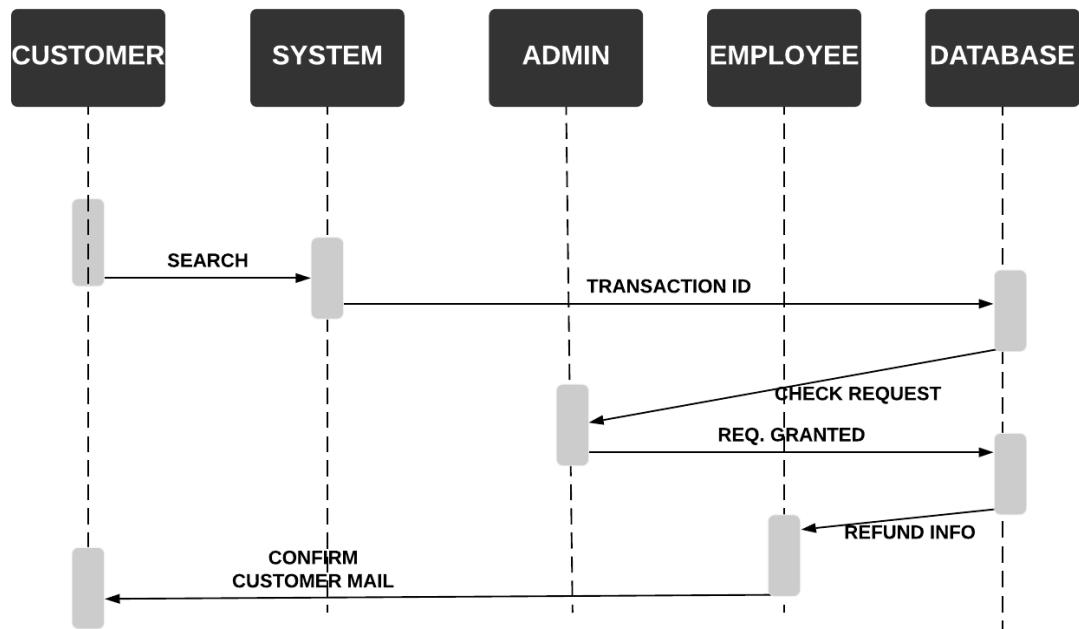
SEQUENCE DIAGRAM (PRODUCT SEARCH)



SEQUENCE DIAGRAM(TRANSACTIONAL SEARCH)



**SEQUENCE DIAGRAM (PAYMENT METHOD)**



**SEQUENCE DIAGRAM (REFUND POLICY)**

# **Chapter8**

## **Conclusion**

From this SRS report on f-commerce Management System, the readers will get a clear and easy view of the overall system of management system of the regular facebook-based online shops. This SRS document can be used effectively to maintain the software development cycle. It will be very easy to conduct the whole project using SRS. Hopefully, this document can also help the junior BSSE students. We tried best to remove all dependencies and make an effective and fully designed SRS.

# Chapter 9

## References

Pressman, Roger S. Software Engineering: A practitioner's Approach (7<sup>th</sup> Edition)