18 November 2017

Pharmacy Management System

Software Requirement Specification and Analysis

IIT

University of Dhaka

Institute of Information Technology
University of Dhaka

Pharmacy Management System

Submitted to

Dr. Kazi Muheymin-Us-Sakib

Professor

Institute of Information Technology

University of Dhaka

Submitted By

Name	Roll
Mahir Mahbub	BSSE0807
Shuvradeb Saha	BSSE0815
S.M. Khayrul Islam	BSSE0822
Obaidur Rahman	BSSE0830
Sefat-E-Mahadi	BSSE0839

Institute of Information Technology

University of Dhaka

LETTER OF TRANSMITTAL

18th November 2017
Dr. Kazi Muheymin-Us-Sakib
Professor
Institute of Information Technology,
University of Dhaka

Subject: Submission of term report on "Pharmacy Management System".

Sir,
With due respect, we are submitting the report on the above topic you assigned to 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

Mahir Mahbub – BSSE0807

Shuvradeb Saha - BSSE0815

S.M. Khayrul Islam – BSSE0822

Obaidur Rahman – BSSE0830

Sefat-E-Mahadi – BSSE0839

2nd Year, 4th Semester, 8th 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 Software Requirements Specification and Analysis(SRS) report on Pharmacy Management System. We would like to thank our course teacher, Dr. Kazi Muheymin-Us-Sakib, Professor, Institute of Information Technology, University of Dhaka, for giving us guidelines about how we can prepare this report. In completing this paper, we have collected various important data and information from various pharmacies beside our university campus. We are thankful to all the owner and employees of the pharmacy shop which we visited.

Purpose

This document initially describes the Software Requirement of an automated small size pharmacy Management System. It contains functional, non-functional and supporting requirements and establishes a requirements baseline for the development of the system.

Table of Contents

Index of Figures

Index of Tables

Chapter 1 INTRODUCTION

1.1 Purpose

This document is the Software Requirements Specification (SRS) for Pharmacy Management System(PMS). It contains detailed functional, non-functional and support requirements. The requirements contained in the SRS are independent, uniquely numbered and organized by topic. The SRS serves as the official means of communicating user requirements to the developer and provides a common reference point for both the developer team and 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 customer, as well as the
project managers, designers, developers, and testers.

- The customer will use this SRS to verify that the developer team has created a product that is acceptable to the customer.
- The 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 continually refer back to this SRS to ensure that the system they are designing will fulfill the customer's needs.
- The developers will use this SRS as a basis for developing the system's functionality.

1.3 CONCLUSION

This analysis 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 and their requirements.

`

Chapter 2 INCEPTION OF PMS

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

2.1 Introduction

Inception is the initial phase of requirements engineering. It defines how a software project gets started and what the scope and nature of the problem to be solved. 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 establish a basic understanding of the problem, the people who want a solution, the nature of the solution that is desired and the effectiveness of preliminary communication and collaborations between the other stakeholders and the software team.

To establish the groundwork, we have worked with the following factors related to the inception phases:

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

2.1.1 List of Stakeholders

Stakeholder 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 everyone else in an organization that may 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 stakeholders are contacted because every stakeholder will be asked:

“Whom else do you think I should talk to?”

To identify the stakeholders, we consulted with pharmacy owner and asked him following questions:

- Who is paying for the project?
- Who will be using the project outcomes?
- Who gets to make the decisions about the project (if this is different from the money source)?
- Who has resources I need to get the project done?
- Whose work will my project affect? (During the project and also once the project is completed).

Stakeholders

Stakeholders are those people who get affected by the overall System directly or indirectly. In this project we initially find out the stakeholders of this project. Stakeholders are:

1. Pharmacy Owner.
2. Manager.
3. Employees.
4. Customer.

2.1.2 Multiple Viewpoints

Owner's Viewpoints:

- User friendly
- Low cost
- Fast and low time to provide services
- Low maintenance
- Better search technique
- Perfect Accounting
- Safe and efficient management of pharmacy

Employee’s Viewpoints:

- Reliable
- User friendly
- Efficient searching (Drag name and group)
- Suggestion list on data entry
- Notification on low stock.
- Long term support
- Dynamic
- Cash memo
- Security
- Making life easier

2.1.3 Working Towards Collaboration

Common Viewpoints:

- User friendly
- Long term support
- Better search option

Conflicting Viewpoints:

- Limited budget
- Cost within budget.
- High security of the system
- Easy access (Different stakeholder wants different type of access)

Final Requirements:

- Generate cash memo
- Storing necessary information of drugs
- Searching Medicine and other Data’s
- Update, delete and edit medicine information.
- Give notification to the user when medicine item is low
- Generate report on medicine
- Strong security

2.1.4 Requirements Questionnaire

We asked the stakeholders some context free questions to understand the project’s overall performances and the goals of the project. Those context free questions have been added to section 2.1.1. These questions have helped us to identify the stakeholders. Then we asked our next question.

2.2 Conclusion

The Inception phase helped us to establish basic understanding about the Pharmacy Management System, identify the stakeholders who will be benefited if this system becomes automated, define the nature of the system and the tasks done by the system, and establish a preliminary communication with our stakeholders.

Chapter 3 ELICITATION OF PMS

3.1 Introduction:

Elicitation is a task that helps the customer to define what is required. To complete the elicitation step, we face many problems like problems of scope, problems of volatility and problems of understanding. However, this is not an easy task. To help overcome these problems, we have worked with the Eliciting requirements activity in an organized and systematic manner.

3.2 Eliciting Requirements

Unlike inception where Q&A (Question and Answer) approach is used, elicitation makes use of a requirements elicitation format that combines the elements of problem solving, elaboration, negotiation, and specification. It requires the cooperation of a group of end-users and developers to elicit requirements. To elicit requirements, we completed following three works.

1. Collaborative Requirements Gathering
2. Quality Function Deployment
3. Usage Scenario

3.2.1 Collaborative Requirements Gathering

Many different approaches to collaborative requirements gathering have been proposed. Each makes use of a slightly different scenario. We completed the following steps to do it:

- The meetings were conducted with Program Chair. He was questioned about their requirements and expectations from the Result Analysis Tool.
- He was asked about the problems he is facing with the current manual system. We also inquired regarding the efficiency of the current process. At last we selected our final requirement list from the meetings.

3.2.2 Quality Function Deployment

Quality Function Deployment (QFD) is a quality management technique that translates the needs of the customer into technical requirement for software. QFD's main aim is understanding that what is valuable to the customer and then deploys these values throughout the engineering process.

There are mainly three types of requirements as QFD:

Normal Requirements: Normal requirements include the objectives and goals that are stated during meeting with customer for a product or system. We found some such objectives and goals during requirement analysis in inception step:

- Generate cash memo
- Storing necessary information of drugs
- Searching Medicine and other Data's
- Alerting pharmacy Data's in the system
- Update, delete and edit medicine information.
- Give alert for the user when medicine item is low
- Generate report on medicine
- Strong security

Expected Requirement: These requirements may not have described by the users but they are so fundamental that the absence of them will cause significant dissatisfaction.

- Daily sales report
- Overall Efficiency
- Error free
- Long term support
- Notification on expire date, low stock
- Admin and User(Employee) authentication
- Transparent transaction and accounting

Exciting Requirement: These requirements are beyond customers expectation but proved to be very exciting.

- Suggestion list and data entry
- Platform independent

3.2.3 Usage Scenario:

Sub-system: Authentication.

Users (owner, manager, employee, and customer) login to the system and after that users interact with the system. When user login to the system, users must provide username (only characters include in the username) and password (must be greater than 5 characters and less than 12 characters contains number and alphabets). System checks the username and password to the database. If username and password match, users will eligible for their task. The system shows the content page to the users. When users enter wrong username or password, system will show wrong password or username message. Users can try maximum three unsuccessful chances to login to the system. In case of users want to try for fourth time to login to the system, system will block users for a certain time (for designed system, it will be 5 minutes) and will try after that time passed. If users forget his/her password, users can change password by forget-password option. In this time users need to enter username, mobile number, email-address. Users select an option (via voice call, via SMS, via email) by which he/she gets the verification code. If users enter the required information successfully, system will send the verification code to the users by the selected option which is chosen by the users. Users enter the verification code successfully. Users can set new password. System store the new password in database. After that users can login to the system. If users fail to enter the correct verification code. System shows message of wrong verification code and an option of send new verification code. Whereas users select "send new verification code", system sends new verification code to the users by the selected option (via voice call, via SMS, via email). System provides maximum three-time chances to the users to send new verification code.

Initially users have no username and password to login to the system. For this reason, users need to sign up the system. For signing up to the system, Users have to provide some information (first name, surname, mobile-number, email, password and date of birth). After signing up to the system a verification code is sent to the users and users enter the verification code and system verifies the user profile and store it to the database for the next use. Users can edit their profile. For editing user’s profile, they need to login to the system. When users want to leave from the system, they need to logout the system. Customers can search any medicine, they can see the result of the search. When a customer wants to order or request for medicine, customer needs to login to the system.

Sub-system: Searching.

Users can search any medicine in the search box of the system. When users write in the search box, system will provide suggestion item name under the search box. Users can search the medicine item based on the item name or medicine group name. System shows for the searched item information in the current page. System also shows the related medicine items along with the searching item. System displays the medicine name, group name, company name, expire date, price, picture, available quantity. If medicine is out of stock system shows all the information and available quantity is replaced by the out of stock statement. If medicine does not exist in the database in this case system will show unavailable and a request option is shown that customer can request for the new medicine.

Admin-users (owner, manager) can search the employee information, edit maintenance information and search the notification of the system. Admin users can search online order list.

Sub-system: Purchase Items.

Online: Before purchasing/ordering any medicine, Customers need to login to the system. Customers search medicine for order. System shows the related medicine in the current page with essential information. For every medicine, "order option'' is available. If a customer chooses to order, he/she needs to enter the quantity of the item and system calculate the price of the order and store all the order in the order list. In between selecting any order, a customer can change or delete item from the order list. Before confirming the order, the whole order list with quantity, price and total price is shown to check whether everything is included. After selecting payment method (cash on delivery, online bank, credit card) by which customer wants to pay for the order. If a customer chooses online bank or credit card payment method in this case he/she enter the account number/ card number and password/ pin number. System check that if customer's account has sufficient balance to order the medicine. In case of account has enough balance, system cut the required balance from the customer's account. System generates an order id and sends the order to the database, with correspond to the order id the transaction id also send to the database. Manager takes the order list and send medicines to the customer's address. When customer receives the delivery, manager selects the order id is successfully delivered. When the balance is not enough, customers can not order items. Instead system will show customers a failure message about transactions.

Shop Service: In the pharmacy shop, employees check the item list from the customer's prescription. Employees' sale the available medicine to the customer on their demand. System generates a cash memo which contains item name, quantity, price, total price, and employee id, employee name (who sale the medicine to the customer). Employees receive money from the customer and deliver the medicine to the customer and select order successfully delivered and payment successful.

Sub-system: Accounting.

In the pharmacy, employees’ sale medicine and generate cash memo. Employees receive money from the customers, deliver the medicine to the customer. Employees confirm successful delivery and payment. If customers want to pay online bank. Employees store the transaction id corresponding to the sales id, employee id and employee name. When employees confirm successful payment, system automatically stores the sales amount, date, time, sales id, employee id, payment method, transaction id. If any due is considered in the pharmacy, employees store the customers’ information (customer name, mobile number, present address, due money, sale id, employee id, date, and time). Employees store Pharmacy shop maintenance cost (purpose of cost, amount, date, time) in the system. Employee’s salary, shop rent is also stored in the system.

In the online system, after confirming the order, customers choose the payment method. Customers choose cash on delivery in the case of payment is pending. If customers choose online payment, customers enter the account number and password, system checks that account has sufficient balance according to the order. If account contains enough balance, system cuts the required amount from the customer’s account to the shop account and stores the transaction id and order id. If balance is insufficient, system shows message to the customer in the screen that you have not enough balance to order the medicine and choose another option. When payment is successful system generate an order to the manager.

Sub-system: Notification Management.

Managers retrieve the online order and deliver the order to the customer's address. If customer's payment is pending, in time of delivery receive the payment and select order successfully delivered and payment successful. If payment is successful in time of order manager sends the ordered medicine to the customer and after receiving the order, managers select order successfully delivered option. If any due payment exists in the system managers send message to the customer to clear the due payment. Managers publish daily or weekly sales report to the owner of the shop. Managers also evaluate sales performance of every individual employee.

Sub-system: Stock Management.

Managers store medicine information (medicine name, group, company, quantity, expire date, picture) in the database. If medicine stock is empty or close to empty. System automatically generates a list of required medicine and divides it according to the company. Managers send the required medicine request to the company email or the company representative. When managers take the requested medicine from the company. Manager stores the medicine information in the database and the shelf. If medicines are previously stored in the database in this case manager updates the quantity only. Manager stores new medicine, delete medicine in the database.

Chapter 4 SCENARIO-BASED MODEL

4.1 Introduction

In this model the system is described from the user's point of view. As this is the first model, it serves as input for creation of other modeling elements.

4.2 Use Case Scenario

As requirements are gathered, an overall vision of system functions and features begins to materialize. To understand how these functions and features will be used by different classes of end users, developers and users create a set of scenarios, called use case scenario, that identify a thread of usage for the system to be constructed.

Use Case:

Use case diagrams are usually referred to as behavior diagrams used to describe a set of actions that some system or sub-systems can perform in collaboration with one or more external users of the system. Each use case should provide some observable and valuable result to the actors or other stakeholders of the system.

The first step in writing a use case is to define the set of actors that will be involved in the story. 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 the system function to achieve a goal. 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.

Use Case Diagram:

Level-0: Use Case Diagram – PMS.

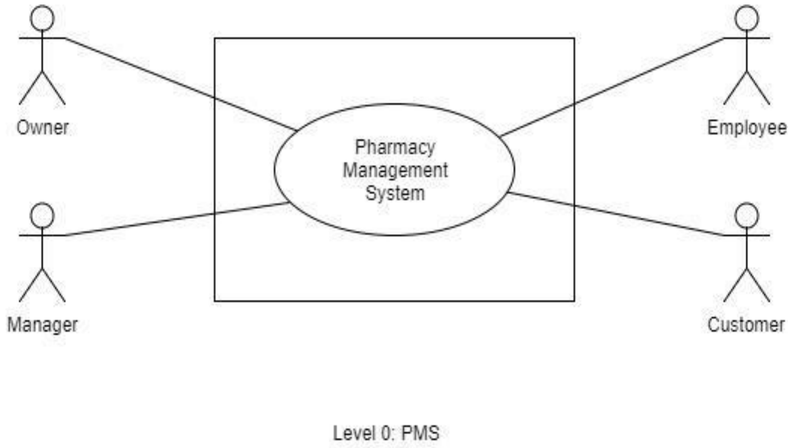

Figure 1: Level-0 Use Case Diagram-PMS.

Level-1: Use Case Diagram – Subsystems.

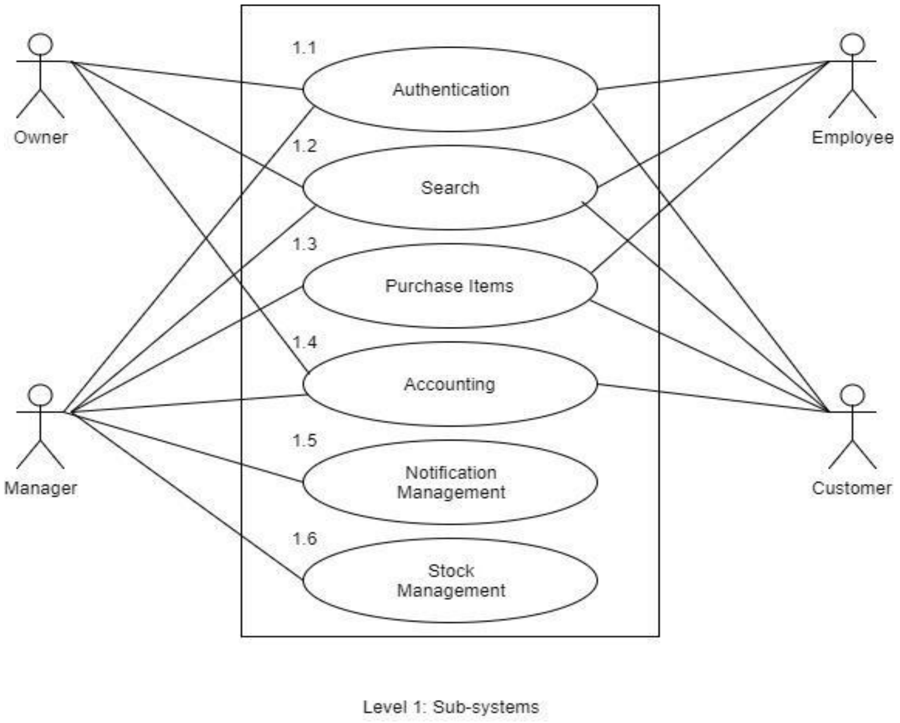

Figure 2: Level-1 Use Case Diagram – Subsystems

Description of Use Case Diagram Level-1:

Primary actor: Owner, manager, Employee, Customer.

Secondary actor: Bank, Medicine Company.

Goal in Context: To set the users to use the pharmacy management system.

Precondition: Users has opened the system for using.

Post condition: Users will successfully use the system.

There are six sub-system in pharmacy management system. They are:

- Authentication.
- Search.
- Purchase items.
- Accounting.
- Notification management.
- Stock management.

Activity Diagram of Level-1:

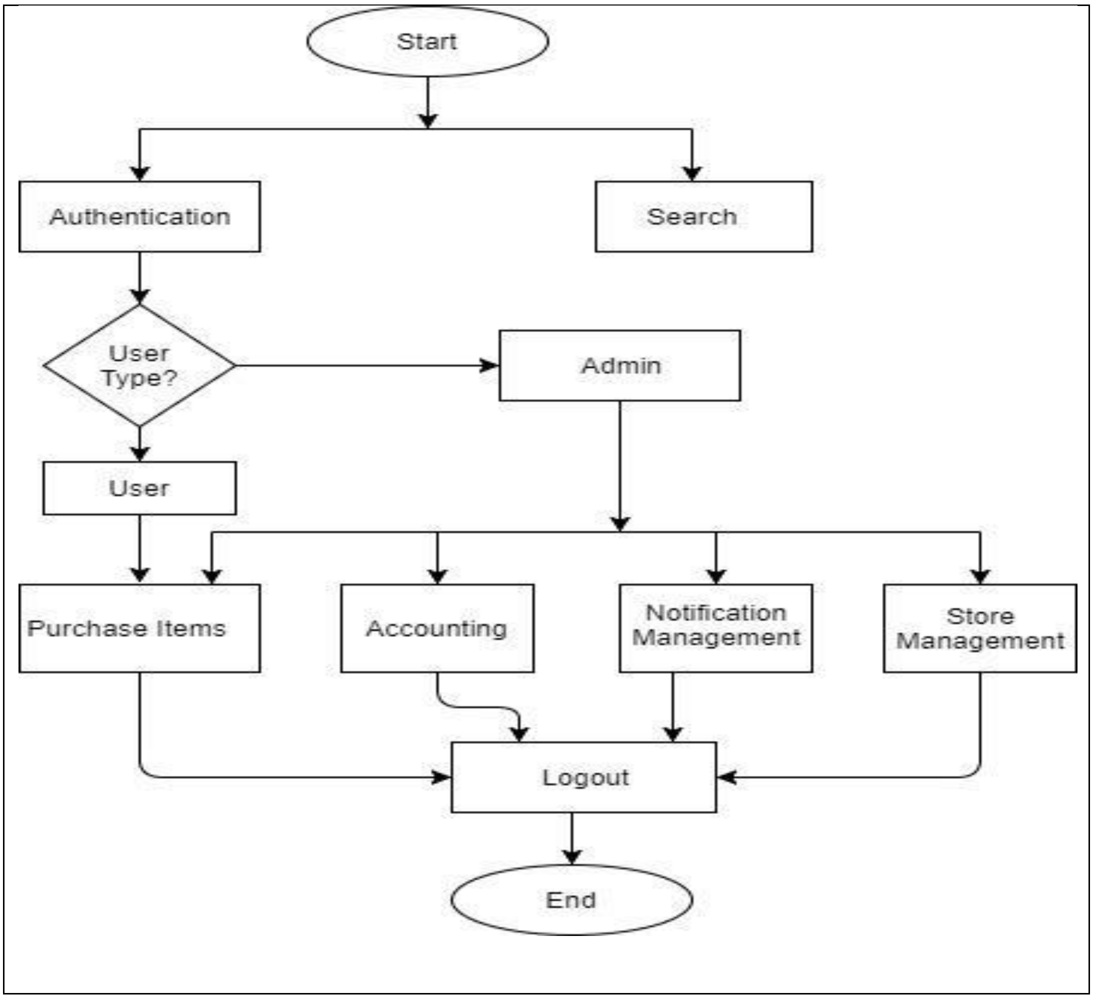

Figure 3: Activity Diagram 1: PMS.

Level-1.1 Use Case Diagram – Authentication.

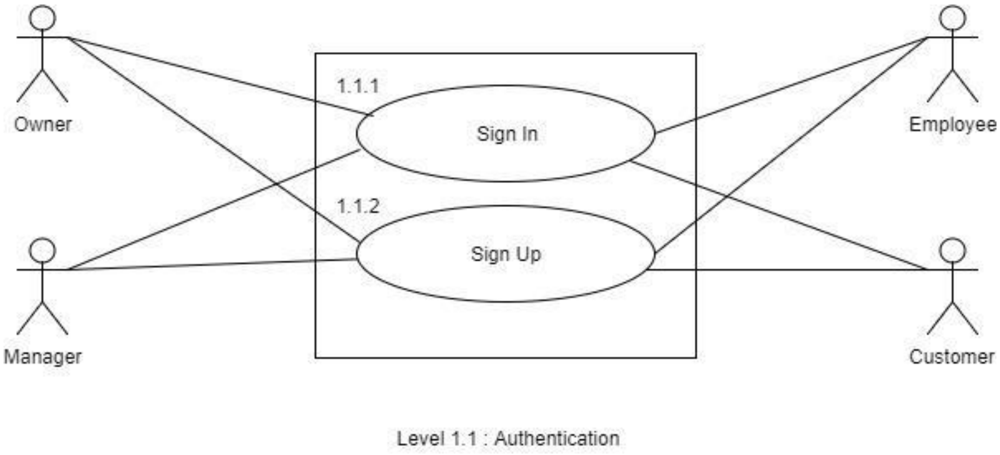

Figure 4: Level-1.1 Use Case Diagram – Authentication.

Description of Use Case Diagram Level-1.1:

Primary actor: Owner, manager, Employee, Customer.

Goal in Context: To set the users to use the pharmacy management system.
Pre-condition: Users has opened the sign in/sign up page for using.
Post-condition: Users will successfully log into the system.

The process or action of verifying the identity of a user. Authentication merely ensures that the individual is who he or she claims to be, but says nothing about the access rights of the individual.

The authentication subsystem of Pharmacy Management System can be divided into two parts. They are:

- Sign in.
- Sign up.

In the Pharmacy Management System users need to sign in to access the system. Initially users have no username and password to login to the system. For this reason, users need to sign up the system. For signing up to the system users have to provide some information (first name, surname, mobile-number, email, password, date of birth). After signing up to the system, a verification code is sent to the users and users enter the verification code and system verifies the users profile and stores it to the database for the next use.

Activity Diagram of Level-1.1:

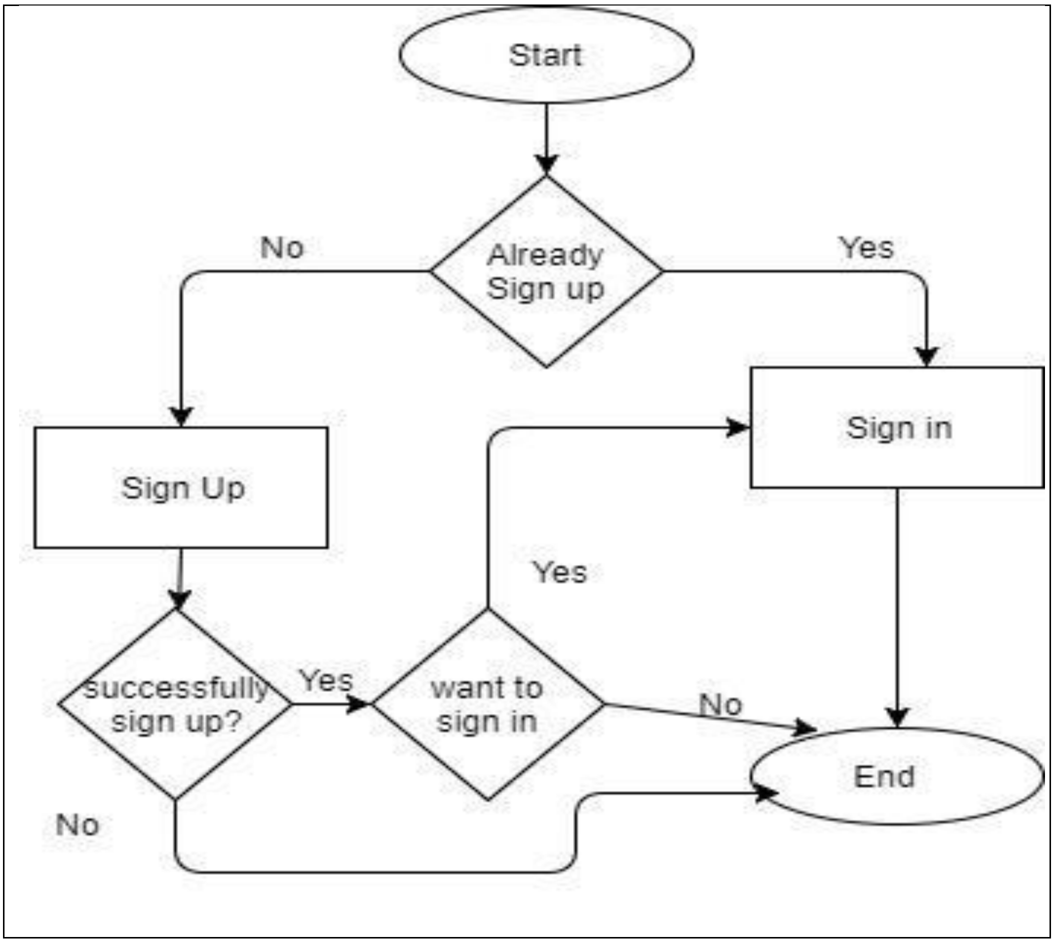

Figure 5: Activity Diagram 1.1: Authentication.

Activity Diagram of Sign UP:

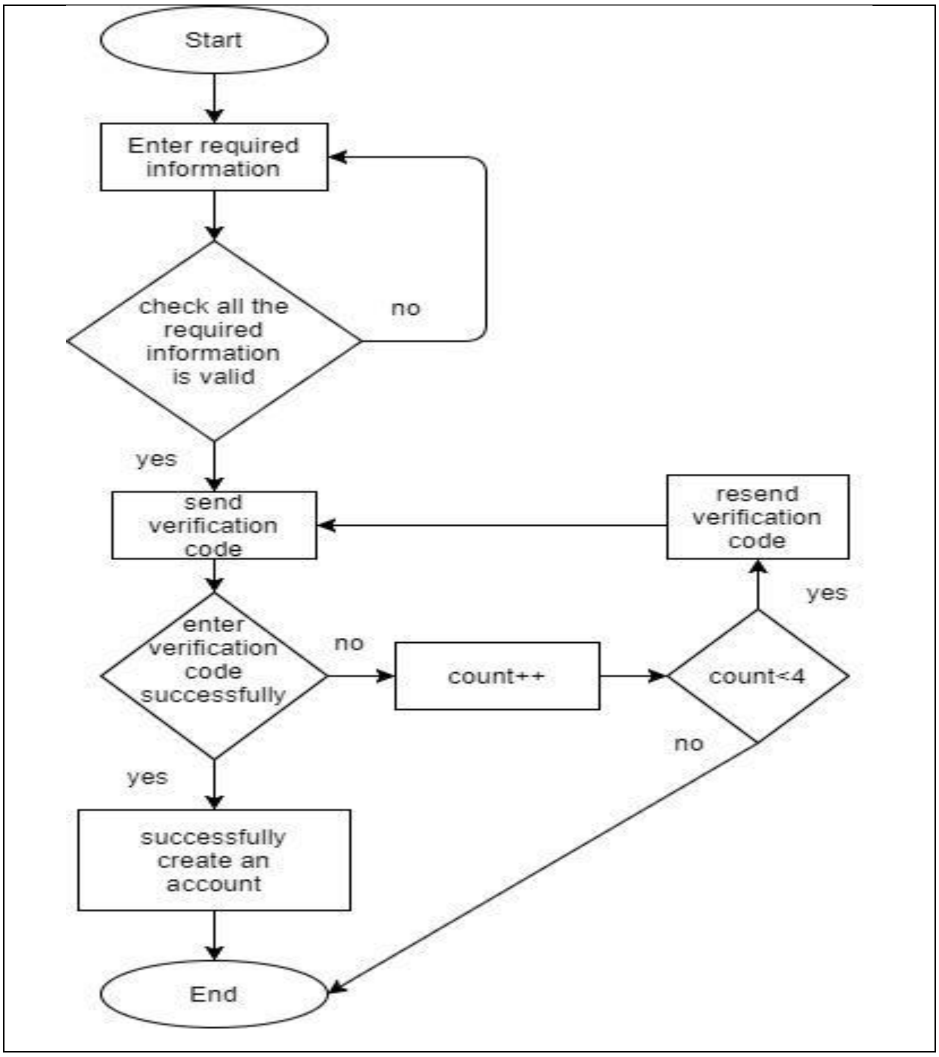
Start
Enter required information
check all the required information is valid
no
yes
send verification code
resend verification code
yes
enter verification code successfully
no
count++
count<4
no
yes
successfully create an account
End

Figure 6: Activity Diagram 1.1: Sign Up

Swim lane Diagram of Level-1.1:

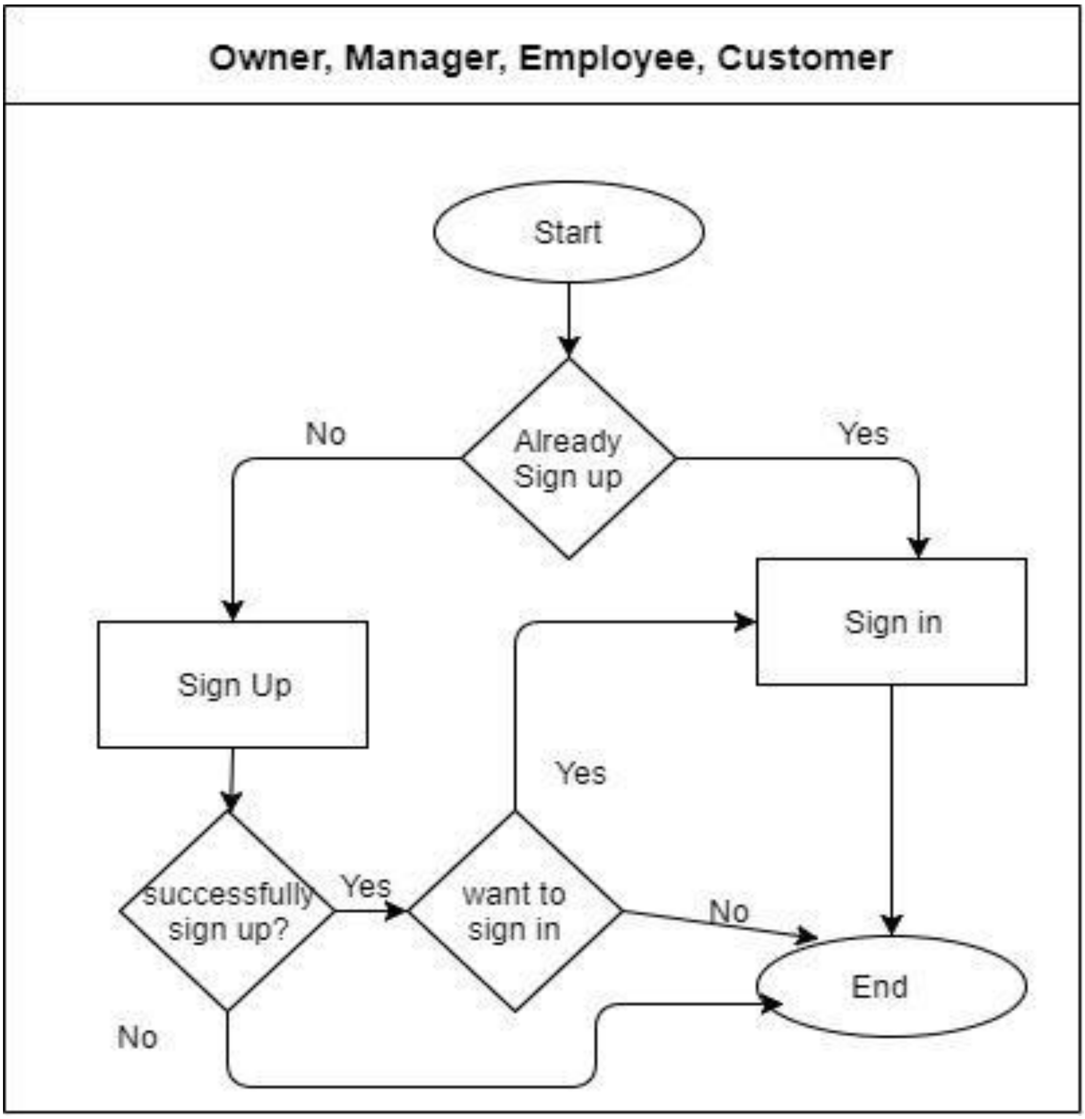

Figure 7: Swim lane Diagram 1.1: Authentication.

Swim lane Diagram of Sign UP:

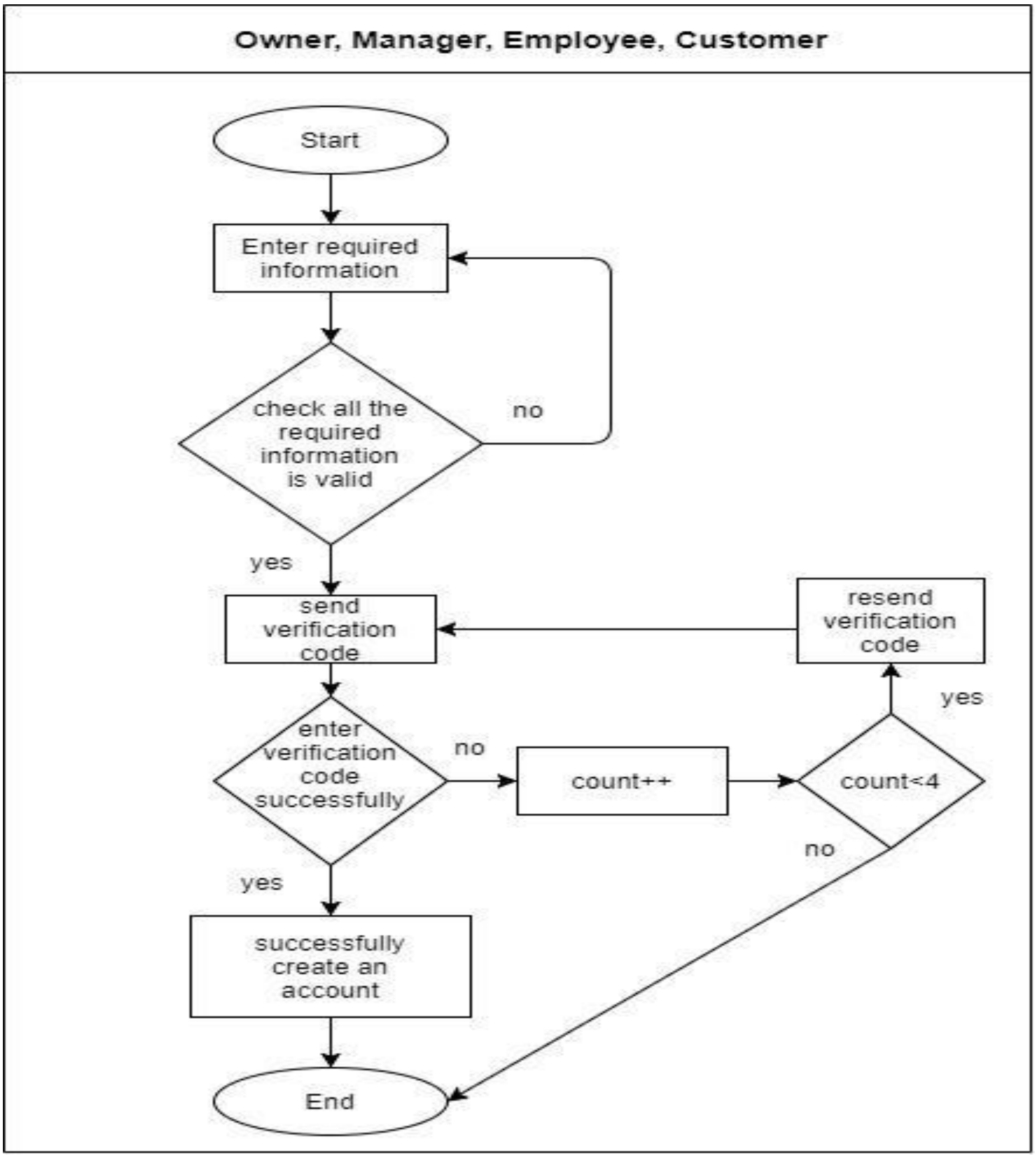
Owner, Manager, Employee, Customer
Start
Enter required information
check all the required information is valid
no
yes
send verification code
resend verification code
yes
enter verification code successfully
no
count++
count<4
no
yes
successfully create an account
End

Figure 8: Swim lane Diagram 1.1.1: Sign Up.

Level-1.1.1 Use Case Diagram – Sign In.

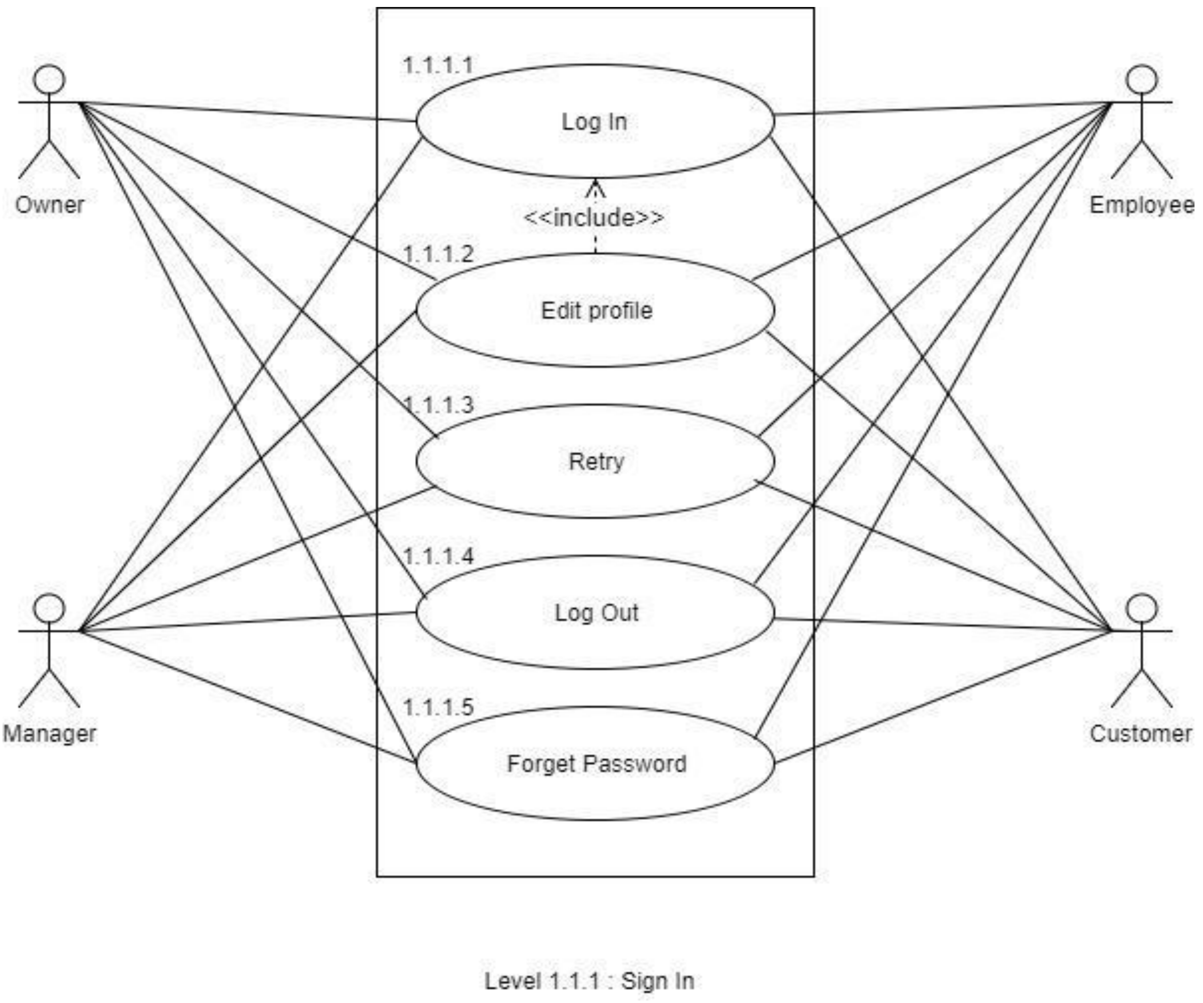

Figure 9: Level-1.1.1 Use Case Diagram – Sign In.

Description of Use Case Diagram Level-1.1.1:

Primary actor: Owner, manager, Employee, Customer.

Sign in sub-system is divided into five parts. They are:

- Log in
- Edit profile
- Retry
- Log out
- Forget password

Initially users (owner, manager, employee, customer) login to the system and after that users interact with the system. When users' login to the system, users must provide username (only characters include in the username) and password (must be greater than 5 characters and less than 12 characters contains number and alphabets). System checks the username and password to the database. If username and password match, users will eligible for their task. The system shows the content page to the users. If users enter wrong username or password system shows users enters wrong username or password. Users can try maximum four unsuccessful chance to login to the system. If users want to try fifth time to login to the system, users will need to wait for a certain time and try after that time. Users can edit their profile. When user's work is done he/she want to leave from the system, he/she needs to log out the system for data security.

Exceptions:

Superscript 1(Blank field or wrong data): No data field for sign up should be blank. If any data field is blank, sign up process will halt and will demand for the data on the blank data field. Same process will be applied for invalid email and date of birth. And continue….

Action-Reply of Use Case Diagram Level 1.1.1:

- **Action:** Users will enter PMS or username and password to login to PMS.
  Reply: if username and password is correct system allow users to enter the system.
- **Action:** Users given Username or password is wrong for the fourth time.
  Reply: Allow him/her to reset or recover the password through forget password option.

Activity Diagram of Level 1.1.1:

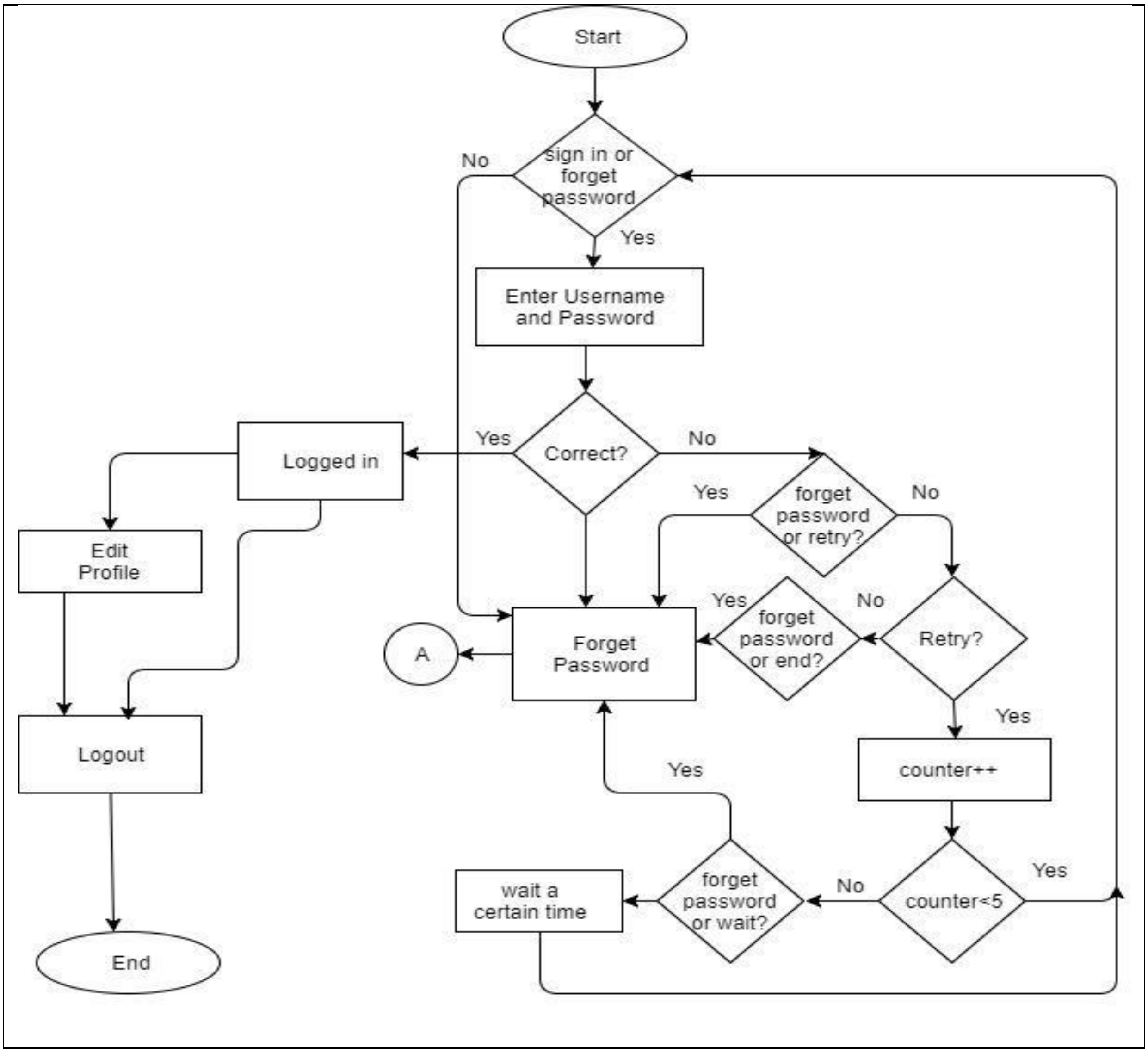

Figure 10: Activity Diagram 1.1.1: Log in

Swim lane Diagram of 1.1.1:

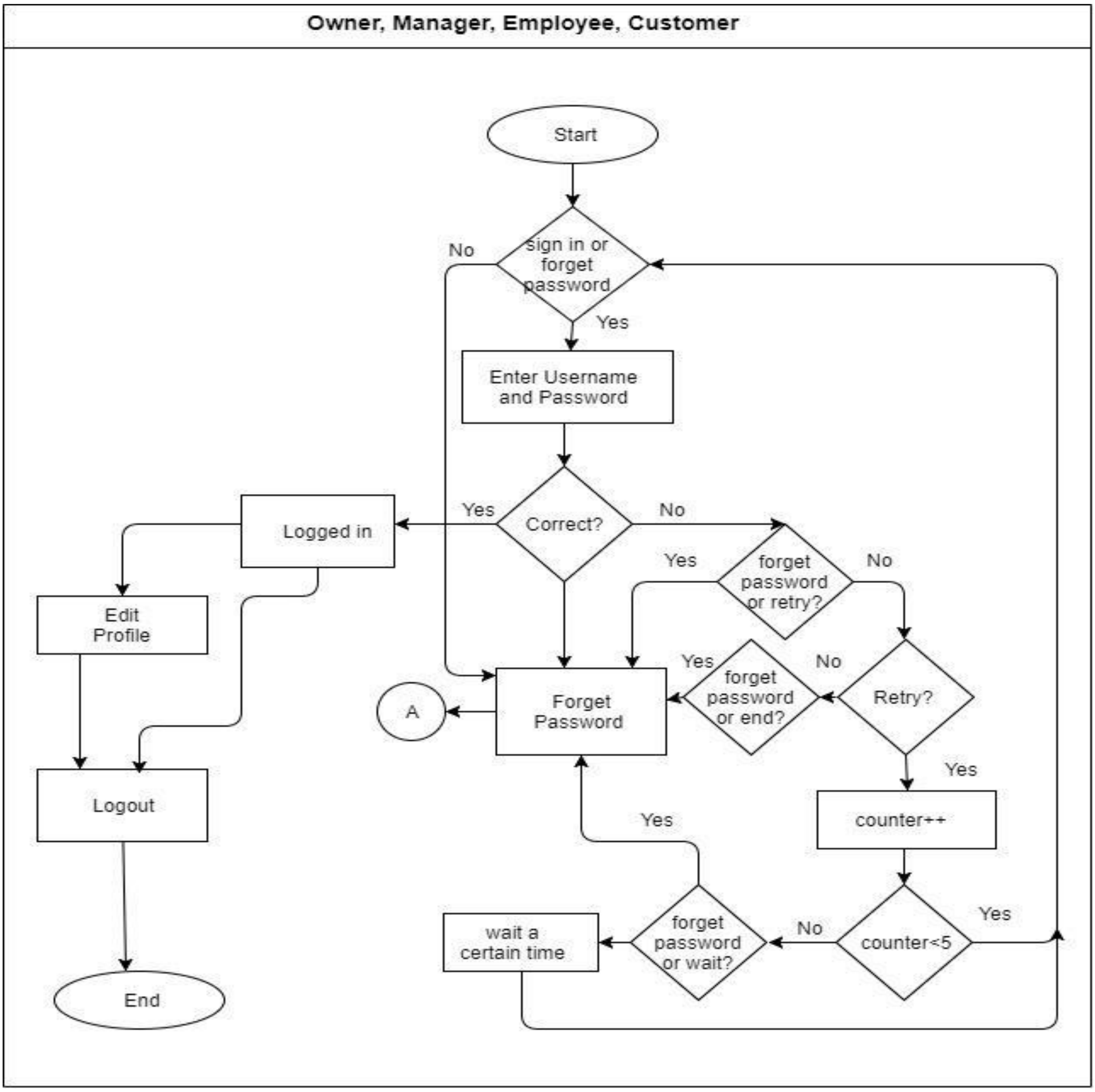

Figure 11: Swim lane Diagram 1.1.1: Log in

Level-1.1.1.5 Use Case Diagram – Forget Password

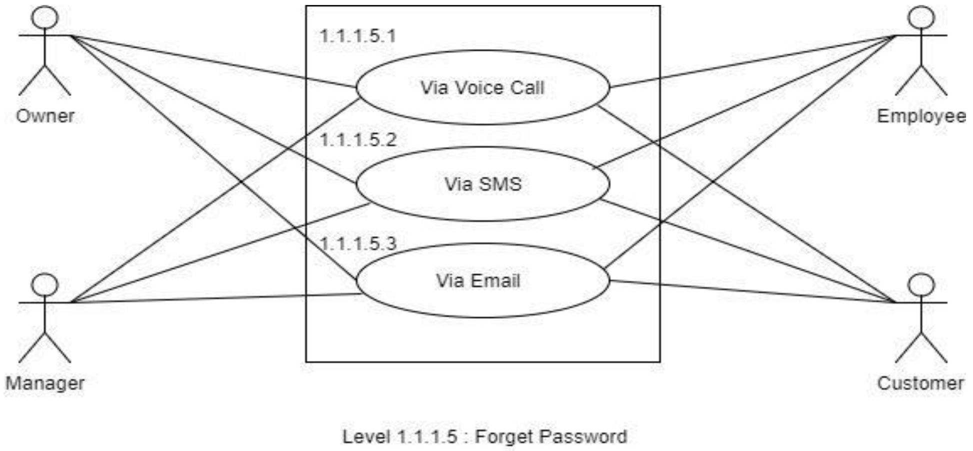

Figure 12:Level-1.1.1.5 Use Case Diagram – Forget Password.

Description of Use Case Diagram Level-1.1.1.5:

Primary actor: Owner, manager, Employee, Customer.

Goal in context: To set the users to reset the password and providing safety.

Pre-condition: The users should have some way for verification.

Post-condition: The users will be acknowledging about the successful resetting of password.

Forget password sub-system is divided into three parts depends on the verification code sending system. They are:

- Via voice call.
- Via SMS.
- Via email.

If users forget password, users can change password by forget-password option. In this time users' need to enter username, mobile number, email-address and new password. After that users select an option (via voice call, via SMS, via email) to get a verification code. If users enter the required information successfully, system sends the verification code to the users by the selected option which is chosen by the users. If users enter the verification code successfully, system will log in automatically. If users fail to enter the correct verification code. System will show message of wrong verification code and an option of send new verification code. If users' select send new verification code, system sends new verification code to the users by the selected option (via voice call, via SMS, via email) which is previously chosen by the users. System provides maximum three times chance to the users to send new verification code. If users unable to enter the correct verification code within these three times chance, system will block the user id.

Exception:

Superscript 1: If no verification code is sent, the users can select alternative verification process. A verification code has a life time of 3 hours from its generation. After that it will be invalid. And continue….

Action-Reply of Use Case Diagram Level 1.1.1.5:

- **Action:** Enter username, mobile number, email-address and new password.
  Reply: if all the input is correct then go to the next step.
- **Action:** select an option for getting verification code.
  Reply: system sends verification code through the chosen option.

Activity Diagram of 1.1.1.5:

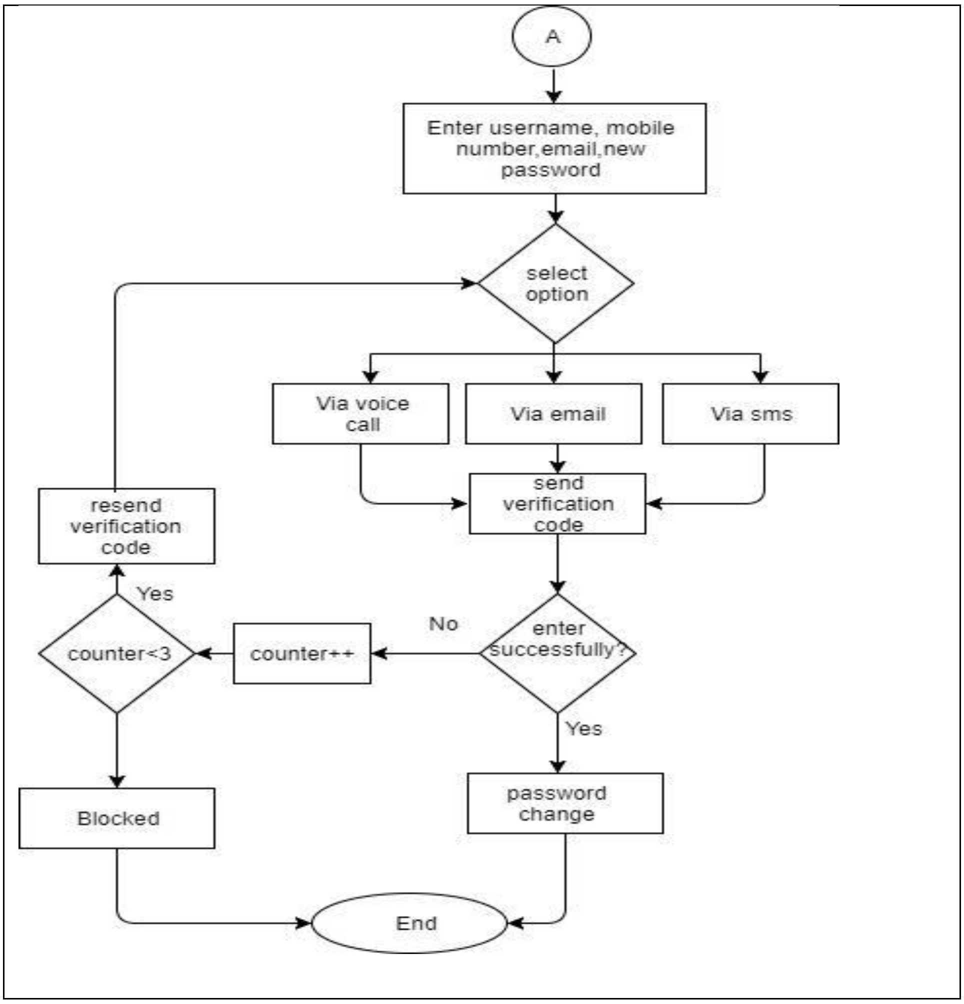

Figure 13: Activity Diagram 1.1.1.5: Forget Password.

Swim lane Diagram of 1.1.1.5:

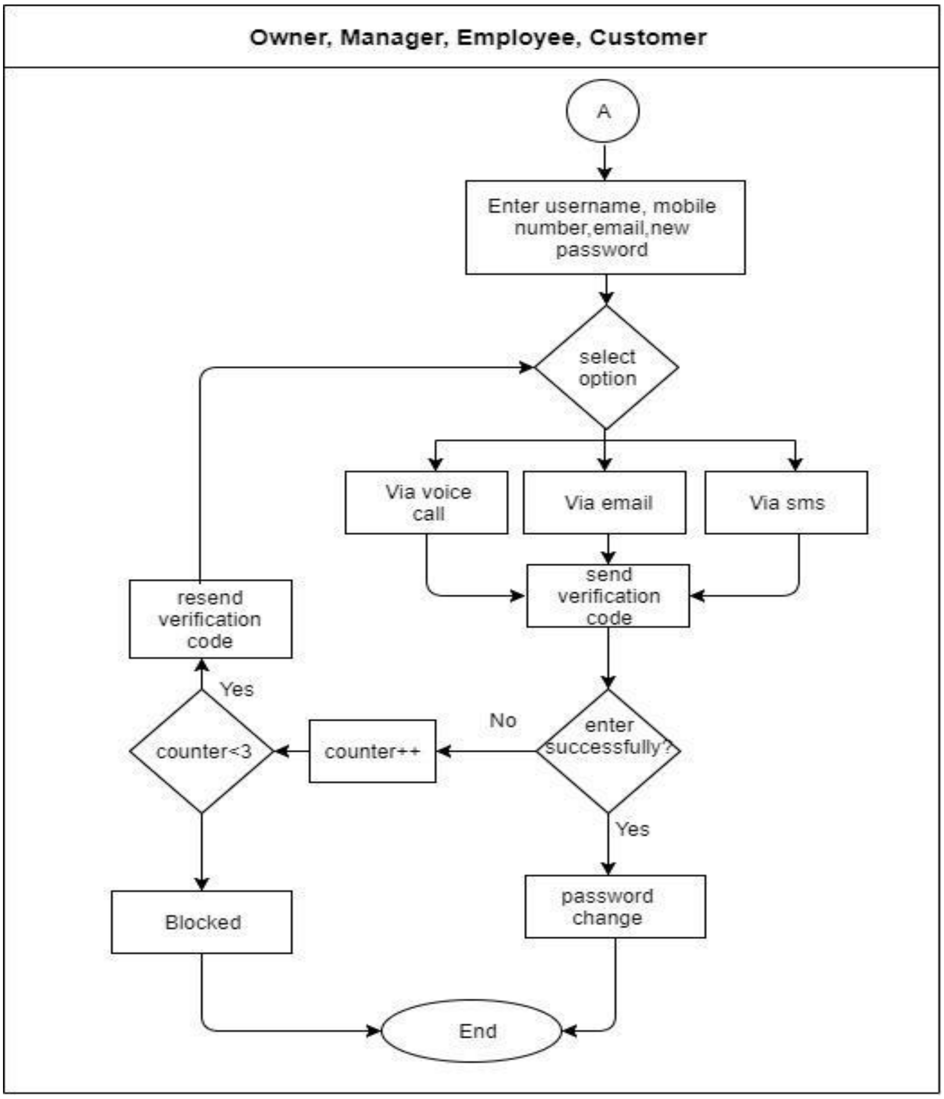

Figure 14: Swim lane Diagram 1.1.1.5: Forget Password.

Level-1.2 Use Case Diagram – Search.

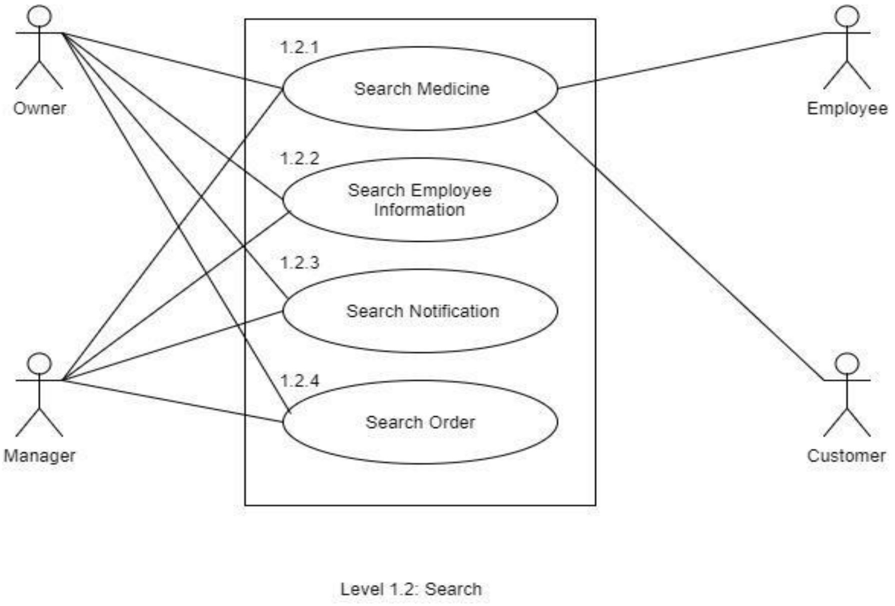

Figure 15: Level-1.2 Use Case Diagram – Search.

Description of Use Case Diagram Level-1.2:

Primary actor: Owner, manager, Employee, Customer.

Goal in context: To query the desire information from the system.

Pre-condition: Database must be active/open. Manager can access employee and purchased item information. Customer can access only item information.

Post-condition: Users will get desire query value to accomplish its further processing and work.

Users can search any medicine in the search box of the system. When users write in the search box, system will provide suggestion item name under the search box. Users can search the medicine item based on the item name or medicine group name. System shows for the searched item information in the current page. System also shows the related medicine items along with the searching item. System displays the medicine name, group name, company name, expire date, price, picture, available quantity. If medicine is out of stock system shows all the information and available quantity is replaced by the out of stock statement. If medicine does not exist in the database in this case system will show unavailable and a request option is shown that customer can request for the new medicine. In case of a customer wants to request for the medicine he/she provides the information of the requested medicine (medicine name, group, company, quantity) and the due date customer wants the delivery.

Admin-users (owner, manager) can search the employee information, edit the information and search the notification of the system. Admin users can search available order, pending order, delivered order list.

Exception:

Superscript 1 (No Value Found): Users requests for information. System returns nearly matched information when such information is not available or "Not Found" message.

Activity Diagram of 1.2:

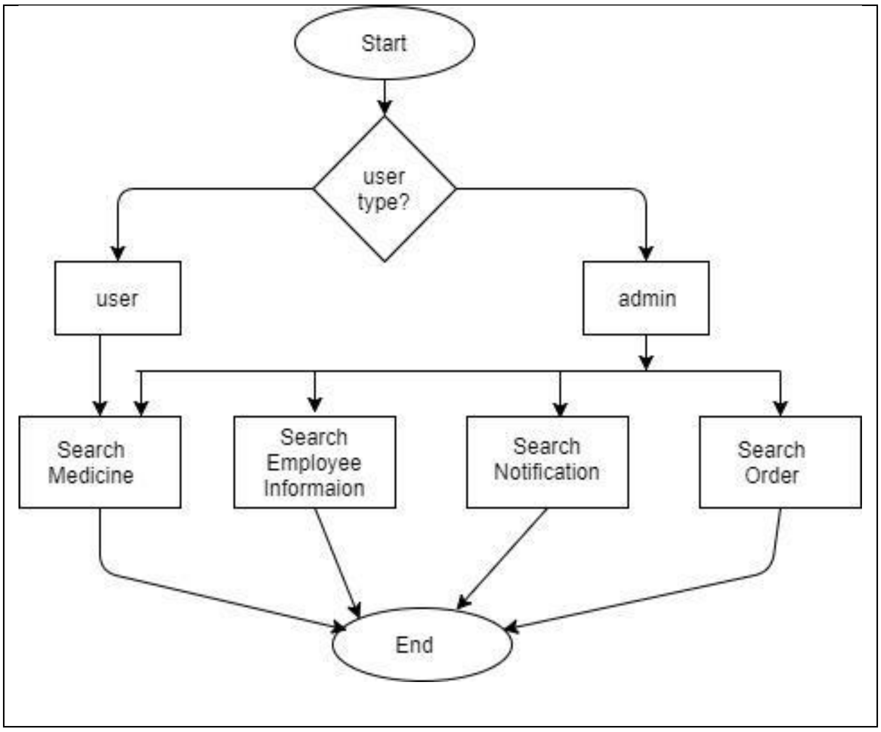

Figure 16: Activity Diagram 1.2: Search.

Swim lane Diagram of 1.2:

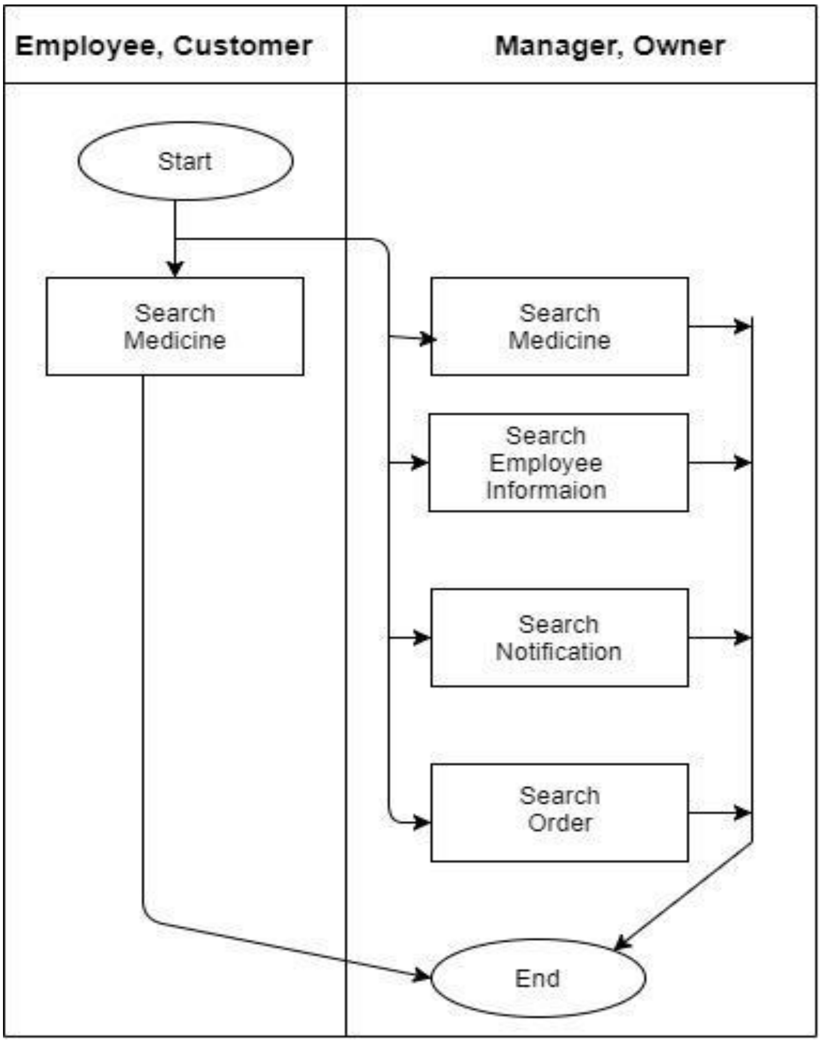

Figure 17: Swim lane Diagram 1.2: Search.

Level-1.2.1 Use Case Diagram – Search Medicine.

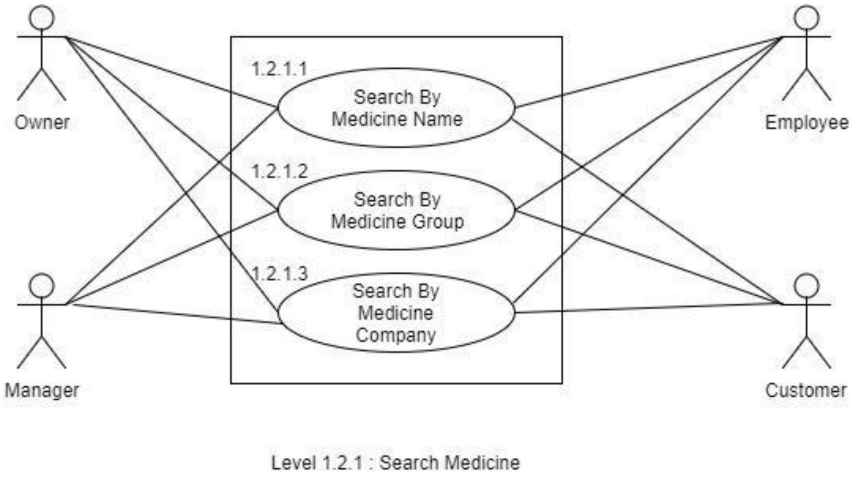

Figure 18: Level-1.2.1 Use Case Diagram – Search Medicine.

Description of Use Case Diagram Level-1.2.1:

Primary actor: Owner, manager, Employee, Customer.

Goal in context: To set users for having desired item/items list.

Pre-condition: Users must select a category before searching.

Post-condition: Users will be delivered with desired search result.

Users can search any medicine in the search box of the system. When users write in the search box, system will provide suggestion item name under the search box. Users can search the medicine item based on the item name or medicine group name. System shows for the searched item information in the current page. System also shows the related medicine items along with the searching item. System displays the medicine name, group name, company name, expire date, price, picture, available quantity. If medicine is out of stock system shows all the information and available quantity is replaced by the out of stock statement. If medicine does not exist in the database in this case system will show unavailable and a request option is shown that customer can request for the new medicine.

Exception:

Superscript 1 (No Value Found): Users requests for information. System returns nearly matched information when such information is not available. If no suggestion is found, system will show "No value found" message and a request option is shown that customer can request for the new medicine.

Superscript 2 (Out of Stock): If medicine is out of stock system shows all the information and available quantity is replaced by the out of stock statement

Action-Reply of Use Case Diagram Level-1.2.1:

- **Action:** Owner/ manager/ Employee/ Customer search medicine.

  Reply: Display the searched item with related medicine item list.

Level-1.3 Use Case Diagram – Purchase Item.

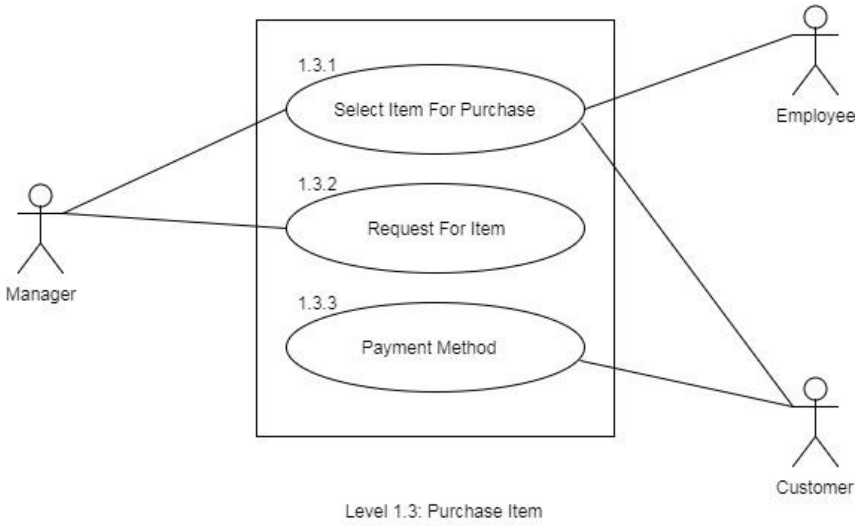

Figure 19: Level-1.3 Use Case Diagram – Purchase Item.

Description of Use Case Diagram Level-1.3:

Primary actor: Manager, Employee, Customer.

Goal in context: To set the customer to purchase items and purchase confirmation.

Precondition: Customer has added at least one item to shopping cart.

Post condition: The order will be placed in the system. The order will have a tracked by the system.

Online:

Before purchasing/ordering any medicine, Customers need to login to the system. Customers search medicine for order. System shows the related medicine in the current page with essential information. For every medicine, ''order option'' is available. If a customer chooses to order, he/she needs to enter the quantity of the item and system calculate the price of the order and store all the order in the order list. In between selecting any order, a customer can change the order list. Before confirming the order, the whole order list with quantity, price and total price is shown to check whether everything is included. After selecting payment method (cash on delivery, online bank, credit card) by which customer wants to pay for the order. If a customer chooses online bank or credit card payment method in this case he/she enter the account number/ card number and password/ pin number. System check that if customer's account has sufficient balance to order the medicine. In case of account has enough balance, system cut the required balance from the customer's account. System generates an order id and sends the order to the database, with correspond to the order id the transaction id also send to the database. When manager approves the order request, order is in pending mode. Manager takes the order list and send medicines to the customer's address. When customer receives the delivery, manager selects the order id is successfully delivered. When the balance is not enough, customers can not order items. Instead system will show customers a failure message about transactions.

Shop Service:

In the pharmacy shop, employees check the item list from the customer's prescription. Employees' sale the available medicine to the customer on their demand. System generates a cash memo which contains item name, quantity, price, total price, and employee id, employee name (who sale the medicine to the customer). Employees receive money from the customer and deliver the medicine to the customer and select order successfully delivered and payment successful.

Action-Reply of Use Case Diagram Level-1.3:

- **Action:** Customer / Employee / Manager select item for order.
  Reply: System store the item in the order list and calculate the price of the individual item and total price.
- **Action:** Payment for online bank or credit card.
  Reply: Store the transaction id with the order id in the database.
- **Action:** Manager request for medicine.
  Reply: Representative sends acknowledgement message.

Activity Diagram of 1.3:

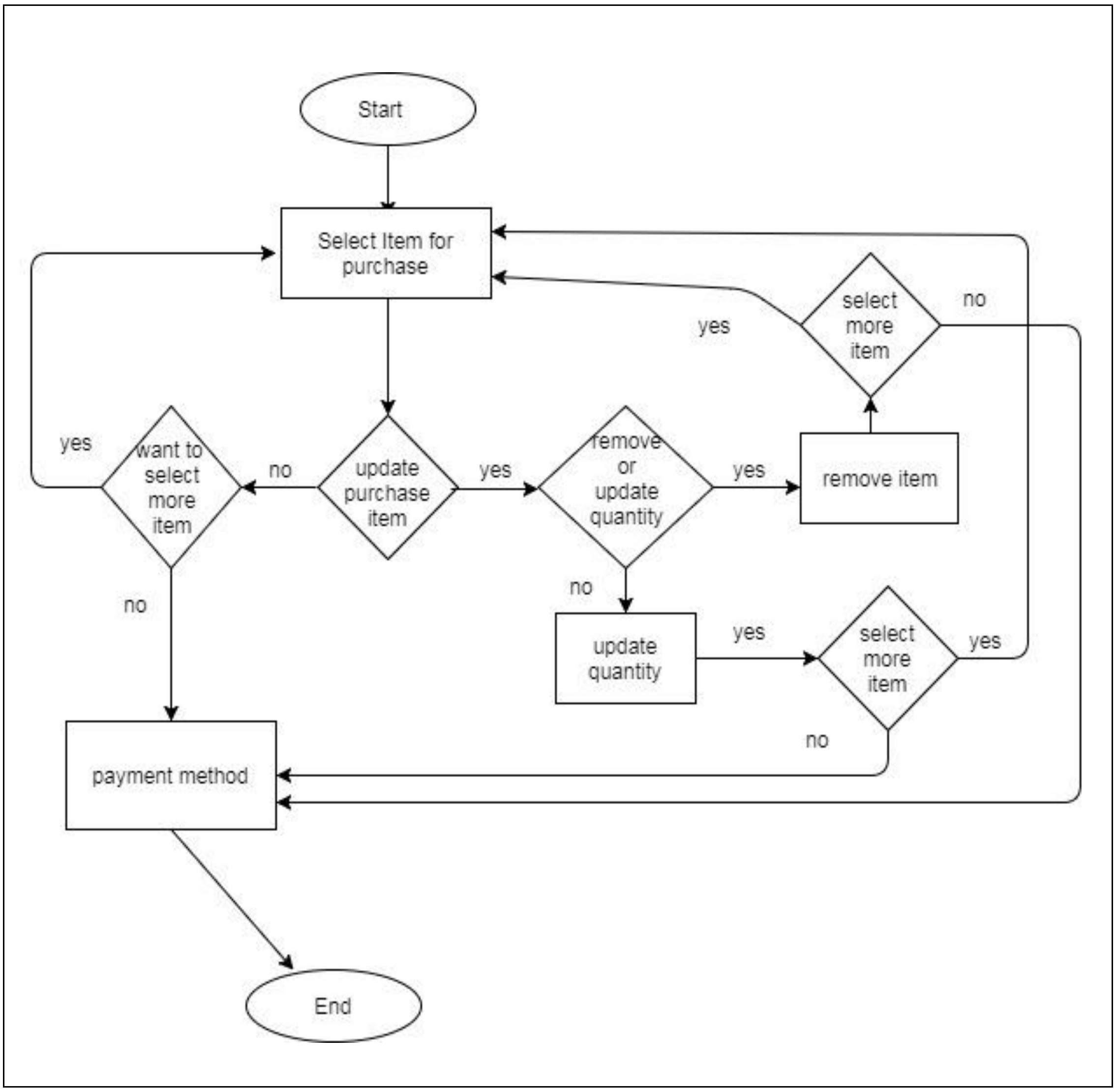

Figure 20: Activity Diagram 1.3: Purchase Item.

Swim lane Diagram of 1.3:

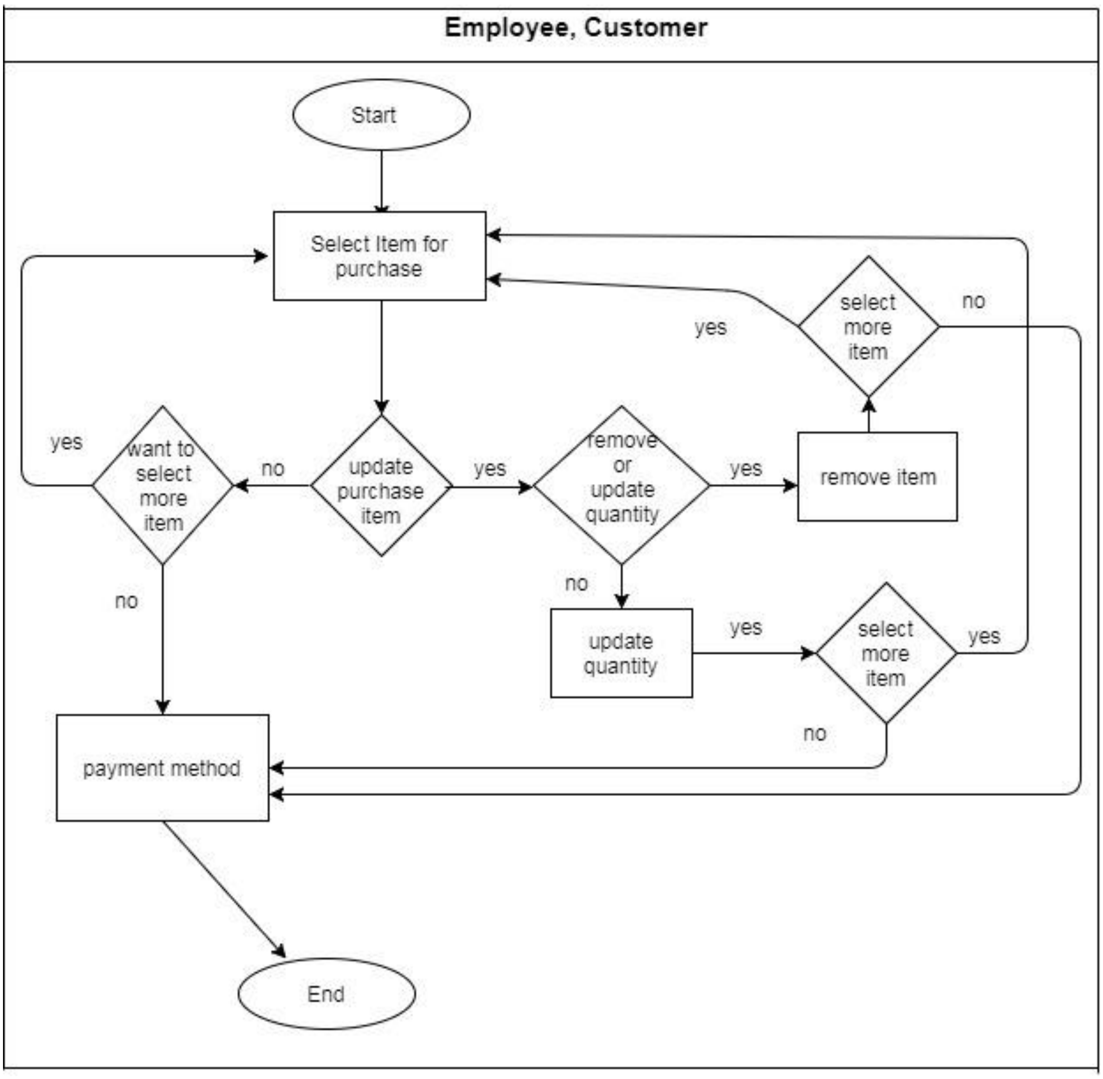

Figure 21: Swim lane Diagram 1.3: Purchase Items.

Level-1.3.3 Use Case Diagram – Payment Method.

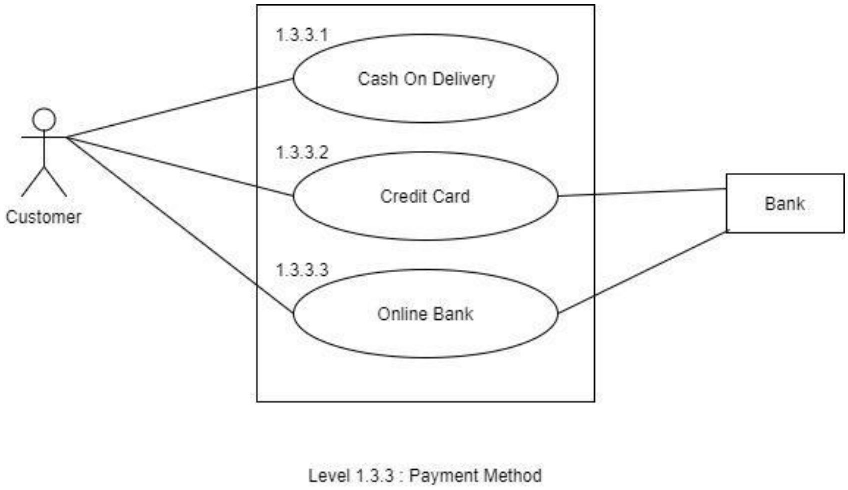

Figure 22: Level-1.3.3 Use Case Diagram – Payment Method.

Description of Use Case Diagram Level-1.3.3:

Primary actor: Customer.

Secondary actor: Bank.

Goal of context: Reliable payment method and confirmation.

Pre-condition: Must have some transaction method.

Post-condition: Payment confirmation will be delivered to the users.

Select the payment method (cash on delivery, online bank, credit card) by which customer wants to pay for the order. If a customer chooses online bank or credit card payment method in this case he/she enter the account number/ card number and password/ pin number. System checks that if customer's account has sufficient balance to order the medicine. If account has enough balance, system cut the required balance from the customer's account.

Exception:

Superscript 1(Out of Balance): If the balance is not sufficient on account in case of online transaction, System will show "Not enough balance" message and Request for select another payment method.

Action-Reply of Use Case Diagram Level-1.3.3:

- **Action:** Check balance validity.
  Reply: If account has sufficient balance to order the medicine system cut the balance else show the message of insufficient balance to order medicine.

Level-1.4 Use Case Diagram – Accounting.

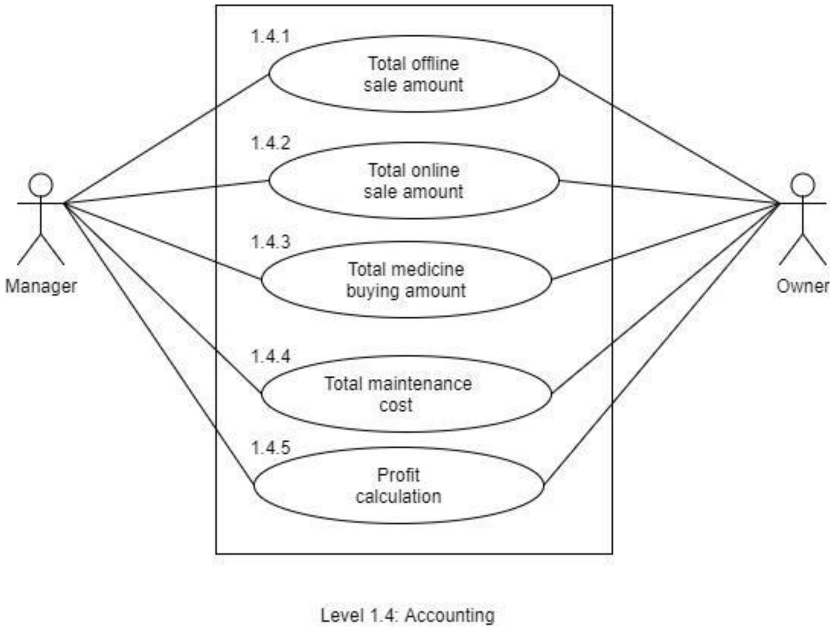

Figure 23: Level-1.4 Use Case Diagram – Accounting.

Description of Use Case Diagram Level-1.4:

Primary actor: Owner, Manager.

Goal of context: To set manager for managing accounting information

Pre-condition: Database system must be active.

Post-condition: The system will store all the information regarding accounting.

In the pharmacy, employees' sale medicine and generate cash memo. Employees receive money from the customers, deliver the medicine to the customer. Employees confirm successful delivery and payment. If customers want to pay online bank. Employees store the transaction id corresponding to the sales id, employee id and employee name. When employees confirm successful payment, system automatically stores the sales amount, date, time, sales id, employee id, payment method, transaction id. If any due is considered in the pharmacy, employees store the customers' information (customer name, mobile number, present address, due money, sale id, employee id, date, and time). Employees store Pharmacy shop maintenance cost (purpose of cost, amount, date, time) in the system. Employee’s salary, shop rent is also stored in the system.

In the online system, after confirming the order, customers choose the payment method. Customers choose cash on delivery in the case of payment is pending. If customers choose online payment, customers enter the account number and password, system checks that account has sufficient balance according to the order. If account contains enough balance, system cuts the required amount from the customer’s account to the shop account and stores the transaction id and order id. If balance is insufficient, system shows message to the customer in the screen that you have not enough balance to order the medicine and choose another option. When payment is successful system generate an order to the manager.

Action-Reply of Use Case Diagram Level-1.4:

- **Action:** Manager generate offline sales report.
  Reply: Display the offline sale amount list based on the date.
- **Action:** Manager generate online sales report.
  Reply: Display the online sale amount list based on the date and compare with previous month.
- **Action:** Manager generates medicine buying cost.
  Reply: Show the medicine buying cost with medicine name, company, quantity, individual price, date.
- **Action:** Manager generates maintenance cost report.
  Reply: Display the list of cost purpose and cost with correspond to date and month.
- **Action:** Manager generates profit/loss report.
  Reply: show the profit or loss on the current date or corresponding date or month.

Activity Diagram of 1.4:

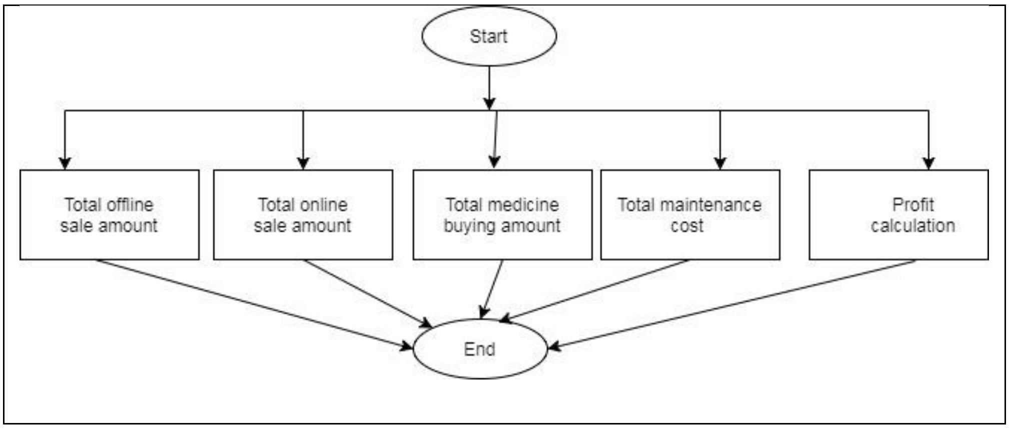

Figure 24: Activity Diagram 1.4: Accounting.

Swim lane Diagram of 1.4:

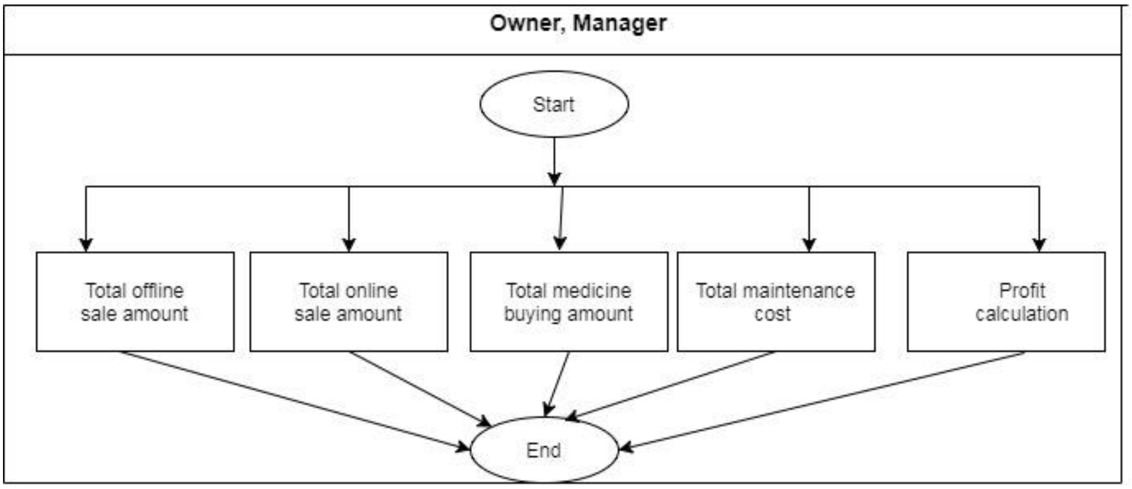

Figure 25: Swim lane Diagram 1.4: Accounting.

Level-1.5 Use Case Diagram – Notification Management

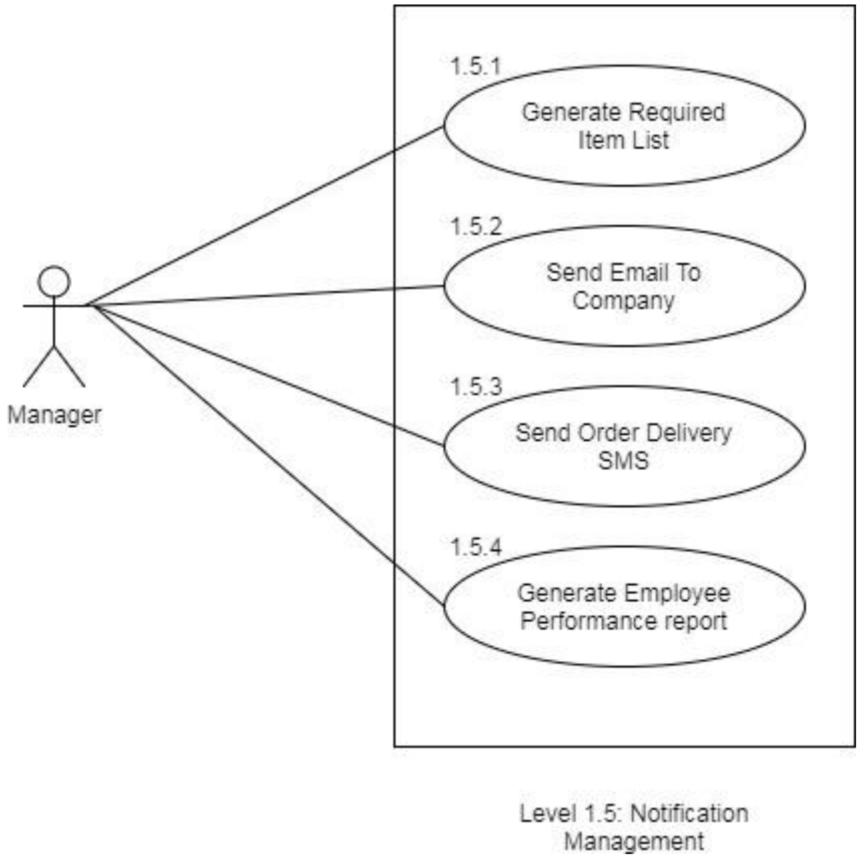
1.5.1
Generate Required Item List
1.5.2
Send Email To Company
1.5.3
Send Order Delivery SMS
1.5.4
Generate Employee Performance report
Manager
Level 1.5: Notification Management

Figure 26: Level-1.5 Use Case Diagram–Notification Management.

Description of Use Case Diagram Level-1.5:

Primary actor: Manager.

When an online order is reached to the manager, the order is approved by the manager. The order is pending until the customer has received the medicine. Manager retrieves the pending order and delivers the order to the customer's address. If customer's payment is pending, in time of delivery will receive the payment and select order successfully delivered and payment successful. If payment is successful in time of order manager sends the ordered medicine to the customer and after receiving the order manager select order successfully delivered. If any due payment exists in the system, managers send message to the customer to clear the due payment. Managers publish daily or weekly sales report to the owner of the shop. Managers also evaluate sales performance of every individual employee.

Exception:

Superscript 1(Notification mode turn off): If notification mood is off. Notification will not generate. So, notification mode must be on.

Action-Reply of Use Case Diagram Level-1.5:

- **Action:** Manager generate required medicine list.
  Reply: Display the list of medicine which are unavailable or close to empty.
- **Action:** Manager approve the online medicine order.
  Reply: System sends the approval sms/email to the customer.
- **Action:** Manager generate employee performance.
  Reply: system display employee sales history and compare with the previous one.

Activity Diagram of 1.5:

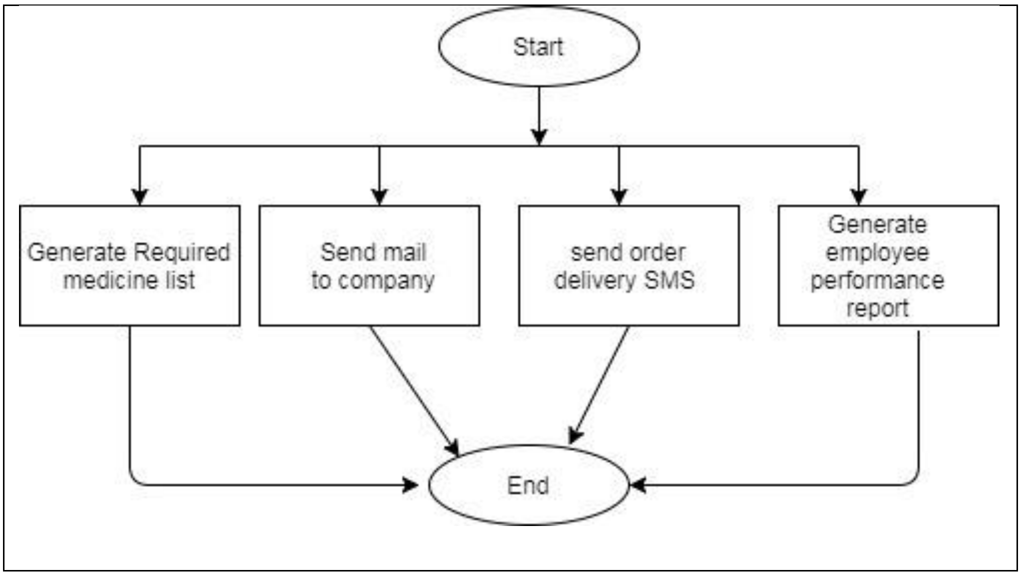
Start
Generate Required medicine list
Send mail to company
send order delivery SMS
Generate employee performance report
End

Figure 27: Activity Diagram 1.5: Notification Management.

Swim lane Diagram of 1.5:

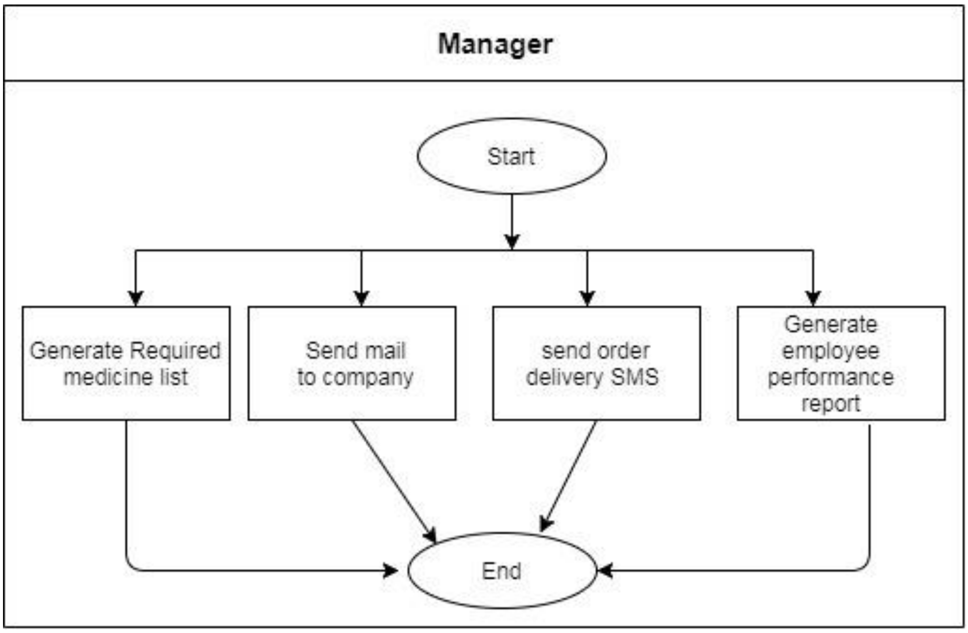
Manager
Start
Generate Required medicine list
Send mail to company
send order delivery SMS
Generate employee performance report
End

Figure 28: Swim lane Diagram 1.5: Notification Management.

Level-1.6 Use Case Diagram – Stock Management

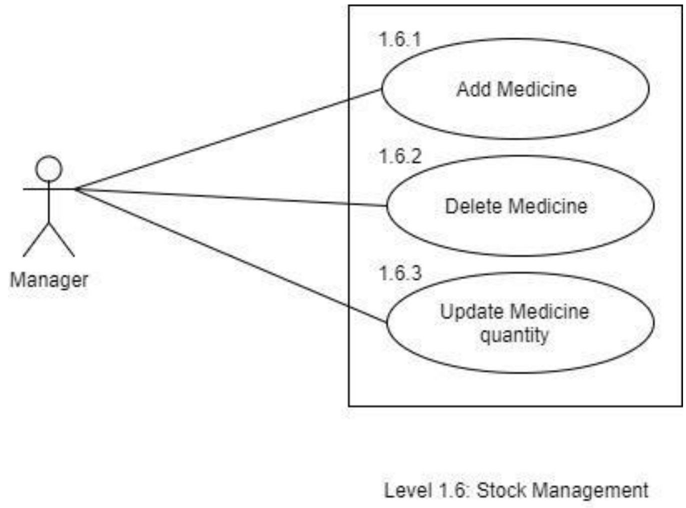

Figure 29: Level-1.6 Use Case Diagram – Stock Management.

Description of Use Case Diagram Level-1.6:

Primary actor: Manager.

Secondary actor: Company.

Pre-condition: Manager must log in to the system.

Post-condition: Database will be updated automatically.

Managers store medicine information (medicine name, group, company, quantity, expire date, picture) in the database. If medicine stock is empty or close to empty. System automatically generates a list of required medicine and divide it according to the company. Managers send the required medicine request to the company email or the company representative. When managers take the requested medicine from the company. Managers store the medicine information in the database and the shelf. Whereas medicine is previously stored in the database in this case manager updates the quantity only. Manager stores new medicine, delete medicine in the database.

Action-Reply of Use Case Diagram Level-1.6:

- **Action:** Manager sends the required medicine request to the company.
  Reply: Company sends the confirmation email of the request.
- **Action:** Manager add or delete medicine.
  Reply: System add or delete medicine to/from the database.
- **Action:** Manager update existing medicine quantity.
  Reply: System update medicine quantity in the database.

Activity Diagram of 1.6:

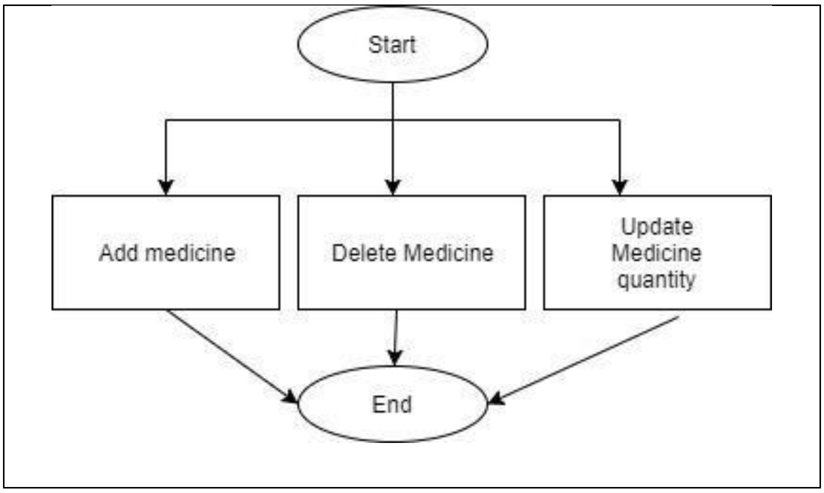

Figure 30: Activity Diagram 1.6: Store Management.

Swim lane Diagram of 1.6:

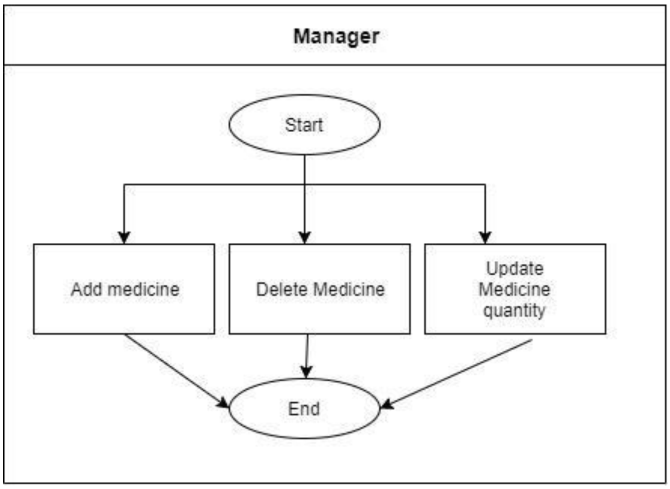

Figure 31: Swim lane Diagram 1.6: Store Management.

Chapter 5 DATA MODEL

5.1 Introduction

Software requirements include the need to create, extend, or interface with a database or if complex data structures must be constructed and manipulated, the software team may choose to create a data model as part of overall requirements modeling.

5.2 Data Object Selection

A data object is a representation of information which has different properties or attributes that must be understood by software. Here is the table of potential data objects.

5.2.1 Noun Identification

No.	Noun	Problem/ Solution Space	Attribute
1	Pharmacy Management System	P	
2	User	S	8-14,76
3	Owner	S	8-14,19,76
4	Manager	S	8-14,72,99,84,76
5	Employee	S	8-14,72,99,83,76
6	Customer	S	8-14,76,79
7	System	P	
8	Username	S	

9	Password	S	
10	Email	S	
11	First name	S	
12	Mobile number	S	
13	Surname	S	
14	Date of birth	S	
15	SMS	S	
16	Verification code	S	
17	Information	P	
18	Chance	P	
19	Owner id	S	
20	Database	S	
21	Authentication	S	
22	Sign Up	S	
23	Voice call	S	
24	Profile	P	
25	Logout	S	
26	Shelf	S	
27	Medicine	S	43-50
28	Result	S	
29	Searching	S	
30	Login	S	
31	Characters	P	
32	Alphabet	P	
33	Task	P	
34	Page	P	

35	Option	P	
36	Maximum	P	
37	Security	P	
38	Due	S	
39	Message	S	
40	Search box	P	
41	Suggestion	P	
42	Searched item	S	
43	Medicine id	S	
44	Medicine group name	S	
45	Quantity	S	
46	Medicine name	S	
47	Company name	S	
48	Expire date	S	
49	Price	S	
50	Picture	S	
51	Related searched item	S	
52	Stock	S	
53	Statement	S	
54	Screen	P	
55	Due date	S	
56	Admin	S	
57	Notification	S	
58	Order	S	
59	Request option	P	
60	Delivery	S	

61	Available order	P	
62	Online order	P	
63	List	P	
64	Shop	P	
65	Order list	P	
66	Payment method	S	
67	Cash on delivery	S	
68	Online bank	P	
69	Credit card	P	
70	Account number	S	
71	Card number	S	
72	Join Date	S	
73	Balance	S	
74	Order id	S	
75	Transaction id	S	
76	Address	S	
77	Transaction	S	45,46,49,70,71,75, 82,83,89,90,91
78	Item list	S	
79	Customers id	S	
80	Cash memo	S	
81	Item name	S	
82	Total price	S	
83	Employee id	S	
84	Manager id	S	
85	Maintenance	S	94-98
86	Accounting	P	

87	Sales id	S	
88	Sales amount	S	
89	Sales date	S	
90	Sales time	S	
91	Due money	S	
92	Shop account	P	
93	Maintenance cost	S	
94	Purpose of cost	S	
95	Amount of cost	S	
96	Date of cost	S	
97	Time of cost	S	
98	Shop rent	S	
99	Employee salary	S	
100	Money	S	
101	Company name	S	
102	Customers account	P	
103	Medicine List	S	
104	Representative	S	101,10
105	Report	S	119-123
106	Due payment	S	
107	Time	P	
108	Employee performance	S	
109	Required medicine	S	45-47
110	Forget Password	S	
111	Number	P	
112	Content	P	

113	Page	P	
114	Suggestion	P	
115	Current	P	
116	Purchase	P	
117	Prescription	P	
118	Demand	P	
119	Online revenue	S	
120	Offline revenue	S	
121	Total due	S	
122	Total maintenance cost	S	
123	Report id	S	

Table 1: *Table 1: Noun Identification*

5.2.2 Potential Data Objects

Owner: 8-14,19,76

Employee: 8-14,72,98,83,76

Manager: 8-14,72,98,84,76

Customer: 8-14,76,70,71,79

Representative: 10,101

Medicine: 43-50

Transaction: 45,46,49,70,71,75,82,83,89,90,91

Report: 119-123

Maintenance: 93-98

Required Medicine: 45-47

5.2.3 Analysis for finalizing Data Objects

1) **Manager** is an extension of employee including additional attribute manager id so manager object can extend employee object forming an is a relationship.

2) A **transaction** table stores each and every transaction details both online and offline.

3) A **transaction** can be paid or can be due so an additional attribute 'type' has been added into the transaction entity.

4) Account number of a customer is needed to store during an online **transaction** that's why transaction entity can contain attributes account number and card number.

5) A **request** table contains every request sent by manager to medicine manufacturing company representative.

6) A **receive** table contains every supply received by the manager from representative.

7) **Manager** generates a daily report can be identified uniquely by date keeps track of daily all kind of transaction records.

8) A **representative** table contains information of every representative who provides medicine supply to the manager.

5.2.4 Final Data Objects

No	Entity	Attributes
1	Owner	owner id, username, password, email, mobile number, surname, date of birth, address
2	Employee	employee id, username, password, email, mobile number, surname, date of birth, address, employee salary, join date
3	Manager	manager id, username, password, email, mobile number, surname, date of birth, address, employee salary, join date
4	Customer	customer id, username, password, email, mobile number, surname, date of birth
5	Medicine	medicine id, medicine name, medicine group name, quantity, company name, expire date, price, picture.
6	Representative	Email, company name
7	Report	Report id, online revenue, offline revenue, total due, total maintenance cost
8	Maintenance	purpose of cost, amount of cost, date of cost, time of cost
9	Transaction	Transaction id, medicine name, quantity, price, account number, card number, employee id, sales date, total price, sale time, due money

Table 2: Find Data Objects

5.3 Data Objects Relation:

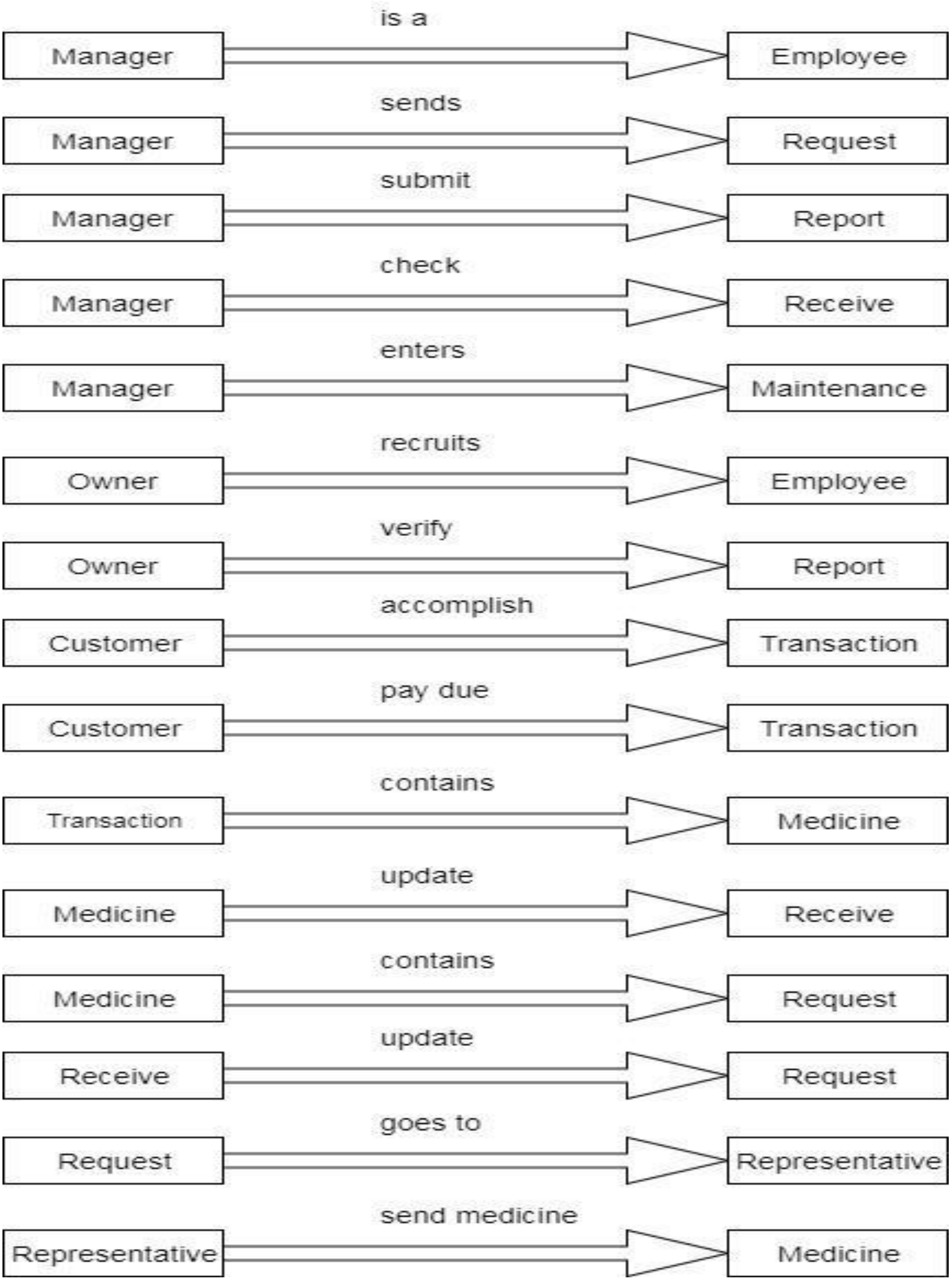

Figure 32: Data object relation.

5.4 Entity Relationship Diagram:

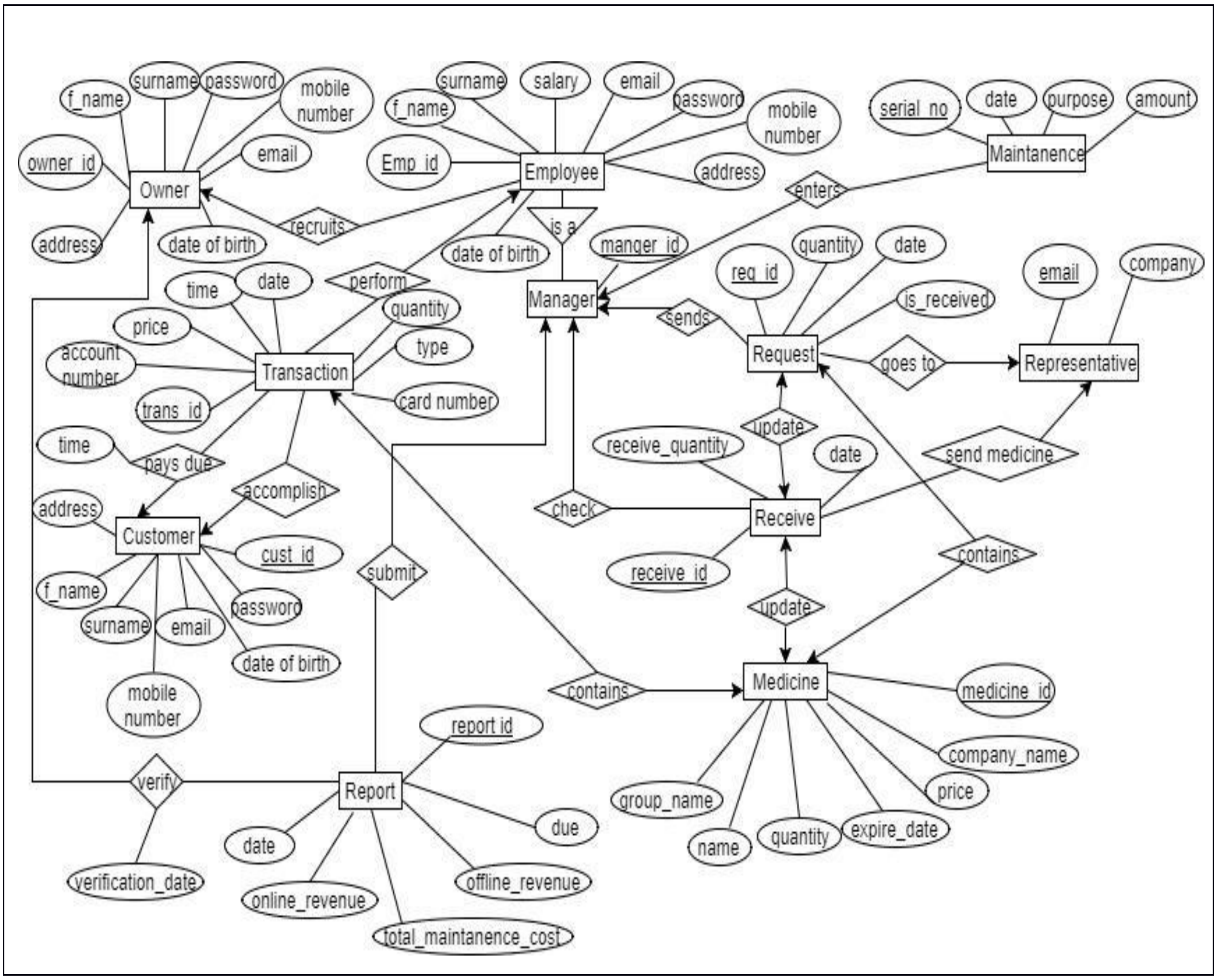

Figure 33: Database ER Diagram.

5.5 Schema Diagram:

	Owner	
Attributes	**Type**	**Size**
Owner id	VARCHAR2	20
username	VARCHAR2	20
date of birth	DATE	20
First name	VARCHAR2	30
Surname	VARCHAR2	30
Email	VARCHAR2	50
Password	VARCHAR2	10
Address	VARCHAR2	50

Table 3: Schema Diagram(Owner)

	Maintenance	
Attribute	**Type**	**Size**
Serial number	VARCHAR2	20
Manager id	VARCHAR2	20
Date	DATE	15
Purpose	VARCHAR2	50
Amount	NUMBER	10
time	VARCHAR2	20

Table 4: Schema Diagram(Maintenance)

	Employee	
Attributes	**Type**	**Size**
Employee_id	VARCHAR2	20
First_name	VARCHAR2	30
Surname	VARCHAR2	30
Email	VARCHAR2	50
Password	VARCHAR2	10
Address	VARCHAR2	50
Salary	NUMBER	5
Join_date	DATE	15
Date of birth	DATE	15
Owner_id	VARCHAR2	20

Table 5: Schema Diagram(Employee)

	Representative	
Attribute	**Type**	**Size**
Email	VARCHAR2	50
Company name	VARCHAR2	20

Table 6: Schema Diagram(Representative)

	Customer	
Attribute	**Type**	**Size**
Customer id	VARCHAR2	20
First_name	VARCHAR2	20
Surname	VARCHAR2	20
Email	VARCHAR2	50
Mobile number	VARCHAR2	20
Password	VARCHAR2	20
Date of birth	DATE	15
Address	VARCHAR2	50

Table 7: Schema Diagram (Customer)

	Receive	
Attribute	**Type**	**Size**
Receive id	VARCHAR2	20
Manager id	VARCHAR2	20
Request id	VARCHAR2	20
Representative email	VARCHAR2	20
Date	DATE	15
Receive quantity	NUMBER	10
Medicine id	VARCHAR2	20

Table 8: Schema Diagram(Receive)

	Request	
Attribute	**Type**	**Size**
Request id	VARCHAR2	20
Manager id	VARCHAR2	20
Medicine id	VARCHAR2	20
Representative email	VARCHAR2	50
Date	DATE	15
Is_received	VARCHAR2	10
Quantity	NUMBER	15

Table 9: Schema Diagram(Request)

	Medicine	
Attribute	**Type**	**Size**
Medicine id	VARCHAR2	20
Medicine name	VARCHAR2	30
Price	NUMBER	15
Expire_date	DATE	15
Quantity	NUMBER	10
Company_name	VARCHAR2	20
Group_name	VARCHAR2	20
picture	VARCHAR2	50

Table 10: Schema Diagram(Medicine)

	Report	
Attribute	**Type**	**Size**
Report id	VARCHAR2	20
Date	DATE	15
Manager id	VARCHAR2	20
Owner id	VARCHAR2	20
due	NUMBER	20
Online_reveneu	NUMBER	15
Offline_revenue	NUMBER	15
Maintenance_cost	NUMBER	15
Verification_date	DATE	15

Table 11: Schema Diagram(Report)

	Pays due	
Attribute	**Type**	**Size**
Customer id	VARCHAR2	20
Transaction id	VARCHAR2	20
time	VARCHAR2	30

Table 12: Schema Diagram (Pays due)

	Transition	
Attribute	**Type**	**Size**
Transition id	VARCHAR2	20
Medicine id	VARCHAR2	20
Customer id	VARCHAR2	20
Employee id	VARCHAR2	20
account number	VARCHAR2	20
Card number	VARCHAR2	20
Date	DATE	15
Time	VARCHAR2	10
Quantity	NUMBER	10
Type	VARCHAR2	15
Price	NUMBER	10

Table 13: Schema Diagram(Transition)

	Manager	
Attribute	**Type**	**Size**
Manager id	VARCHAR2	20
Employee id	VARCHAR2	20

Table 14: Schema Diagram(Manager)

Chapter 6 CLASS-BASED MODEL

This Chapter is intended to describe class based modeling of Pharmacy Management System.

6.1 CLASS BASED MODELING CONCEPT

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

6.2 General Classifications

To identify the potential class, we have to first select the nouns 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:

No	Noun	Problem/Solution Space	General Classifications
1	Pharmacy Management System	P	
2	User	S	4, 5, 7
3	Owner	S	4, 5
4	Manager	S	4, 5 ,7
5	Employee	S	4, 5, 7
6	Customer	S	4, 5, 7
7	System	P	
8	Username	S	
9	Password	S	
10	Database	S	1
11	Authentication	S	3,1
12	Mobile number	S	
13	Email	S	
14	Voice call	S	2
15	SMS	S	2
16	Verification code	S	2
17	Information	P	
18	Chance	P	
19	User id	S	
20	Sign up	S	3, 1
21	First name	S	

22	Surname	S	
23	Date of birth	S	
24	Profile	P	7, 2
25	Logout	S	3, 1
26	Shelf	S	
27	Medicine	S	2, 5, 7
28	Result	S	
29	Searching	S	3
30	Login	P	3
31	Characters	P	
32	Alphabet	P	
33	Task	P	
34	Page	P	
35	Option	P	
36	Maximum	P	
37	Security	P	
38	Due	S	2
39	Message	S	3
40	Search box	P	
41	Suggestion	P	
42	Item name	S	
43	Medicine group name	S	
44	Searched item	S	
45	Related searched item	S	

46	Medicine name	S	
47	Company name	S	
48	Expire date	S	
49	Price	S	
50	Picture	S	
51	Quantity	S	
52	Stock	S	
53	Statement	S	
54	Screen	P	
55	Due date	S	
56	Admin	S	4, 5, 7
57	Notification	S	2, 3, 7
58	Order	S	3
59	Request option	P	
60	Delivery	S	3
61	Available order	P	
62	online order	P	
63	List	P	
64	Shop	P	
65	Order list	P	
66	Payment method	S	3
67	Cash on delivery	S	3
68	Online bank	P	1
69	Credit card	P	

70	Account number	S	
71	Card number	S	
72	Pin number	S	
73	Balance	S	
74	Order id	S	
75	Transaction id	S	
76	Address	S	
77	Pharmacy shop	P	
78	Item list	S	2, 7
79	Customers prescription	P	
80	Cash memo	S	2, 7
81	Item name	S	
82	Total price	S	
83	Employee id	S	
84	Employee name	S	
85	Maintenance	S	2, 7
86	Accounting	P	
87	Sales id	S	
88	Sales amount	S	
89	Sales date	S	
90	Sales time	S	
91	Due money	S	
92	Shop account	P	

93	Maintenance cost	S	
94	Purpose of cost	S	
95	Amount of cost	S	
96	Date of cost	S	
97	Time of cost	S	
98	Employee salary	S	
99	Shop rent	S	
100	Money	S	
101	Company	S	
102	Customers account	S	
103	Medicine List	S	2, 5, 7
104	Company representative	P	1, 7
105	Sales report	S	2, 5, 7
106	Due payment	S	3, 2, 7
107	Time	P	
108	Employee performance	S	2, 7
109	List of required medicine	S	2, 5, 7
110	Forget Password	S	1,2,3
111	Number	P	
112	Content	P	
113	Page	P	
114	Suggestion	P	
115	Current	P	
116	Purchase	P	
117	Prescription	P	
118	Demand	P	

Table 15: General Classification

6.3 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	Potential class	Attributes
1	User	1,2,3,4,5
2	Owner	1,2,3,4,5
3	Manager	1,2,3,4,5
4	Employee	1,2,3,4,5
5	Customer	1,2,3,4,5
6	Authentication	1,3,4,5
7	Database	1,6
8	Verification code	
9	Voice call	
10	SMS	
11	Admin	1,2,3,4,5
12	Cash memo	1,3,4
13	Message	
14	Profile	1,2,3,4,5
15	Logout	1,2,3
16	Medicine	1,2,3,4,5
17	Log in	1,2,3
18	Item list	

19	Maintenance	1,2,3,4
20	Sales report	2,3
21	Notification	1,3,4
22	Order	1,3,4
23	Delivery	
24	Payment method	1,2,3,4
25	Cash on delivery	
26	Online bank	1,6
27	Employee performance	1,3,4
28	Pharmacy Shop	
29	Searching	1,2,3,4,5
30	Due payment	1,3,4,2
31	Medicine list	
32	Company representative	1,3,4,6
33	Forget Password	1,2,3,4,5

Table 16: Selection Criteria

6.4 Associated Noun and Verb Identification

No.	Potential Class	Associated Nouns	Associated verbs
1	Owner	First name, surname, email, pharmacy, date of birth, password, admin	owns, verifies, recruits
2	Manager	First name, surname, email, phone, date of birth, manager id, employee id, salary, joining date	Enters, generates, submits, permits, performs, authenticates
3	Employee	First name, surname, email, phone, date of birth, employee id	Permits, performs, authenticates
4	Customer	Customer id, first name, surname, email, password, address	Performs, pays, authenticates
5	Registration	First name, surname, phone number, password, address, date of birth	receive application, check, application status, check applicant type, send application, receive payment verification, send notification, notify manager, notify owner
6	User	First name, surname, phone number, password, address, date of birth	Gets, sets, updates

7	Medicine	Name, quantity, price, expire date, company name	Gets, sets, updates
8	Maintenance	Date, time, purpose, amount	Gets, sets, updates
9	Report	Online revenue, offline revenue, due, maintenance cost	Get, set, updates
10	Representative	Email, company name	Gets notification, supplies medicine, provides medicine supply
11	Authentication	Name, password, email, phone number, code, voice, sms, email	Log in, log out, forgot, code generation, retry, block
12	Transaction	Date, time, type, customer id, employee id, medicine name, transaction id, quantity, price	records transaction, get employee id, get customer id, get price, get type, get id, set employee id, set customer id, set price, set type, set id
13	Database	Name, user, password	Store, retrieve, query, insert, update, delete

Table 17: Noun and Verb Identification

6.5 Attributes selection

No.	Potential Class	Attributes
1	Owner	First name surname email phone date of birth
2	Manager	First name surname email phone date of birth manager id employee id salary
3	Employee	First name surname email phone date of birth employee id salary

4	Customer	Customer id first name surname email password address
5	Registration	First name surname phone number password address date of birth
6	User	First name surname phone number password address date of birth
7	Medicine	Name id quantity price

		expire date company name
8	Maintenance	Date time purpose amount
9	Report	Online revenue offline revenue due maintenance cost
10	Representative	Email company name
11	Transaction	Date time type customer id employee id medicine name transaction id quantity price transaction type

12	Database	DB Name password DB User
13	Authentication	user_id password verificationCode blockTime

Table 18: Attribute Selection

6.6 Method Selection

No.	Potential Class	Methods

1	Owner	verifyReport()
		recruitEmployees()
		getOwnerId()
		setOwnerId()
		storeEmployeeInformationInDatabase()
		getFirstName()
		setFirstName
		getSurname()
		setSurName
		getPhoneNumber()
		setPhoneNumber()

		getPassword()
		setPassword()
		getEmail()
		setEmail()
		getDateOfBirth()
		setDateOfBirth()
		receiveVerifictionCode()

2	Manager	inputMaintainenceInformation()
		updateMedicineInformation()
		generateReport()
		sendMedicineRequest()
		performTransaction()
		getManagerId()
		setManagerId()
		receiveNotificationFromRepresentative()
		getFirstName()
		setFirstName()
		getSurname()
		setSurName
		getPhoneNumber()
		setPhoneNumber()
		getPassword()

		setPassword()
		getEmail()
		setEmail()
		getDateOfBirth()
		setDateOfBirth()
		receiveVerifictionCode()

3	Employee	performTransaction()
		getEmployeeId()
		setEmpoyeeId()
		getFirstName()
		setFirstName()
		getSurname()
		setSurName
		getPhoneNumber()
		setPhoneNumber()
		getPassword()
		setPassword()
		getEmail()
		setEmail()
		getDateOfBirth()
		setDateOfBirth()
		receiveVerifictionCode()

4	Customer	performTransaction()
		payForTransaction()
		receiveOrderDeliveryNotification()
		getFirstName()
		setFirstName
		getSurname()
		setSurName
		getPhoneNumber()
		setPhoneNumber()
		getPassword()
		setPassword()
		getEmail()
		setEmail()
		getDateOfBirth()
		setDateOfBirth()
		receiveVerifictionCode()

6	Registration	setFirstName()
		setSurname()
		setPhoneNumber()
		setEmail()
		setDateOfBirth()
		setAddress()

		setPassword()
		getFirstName()
		getSurname()
		getPhoneNumber()
		getEmail()
		getDateOfBirth()
		getAddress()
		getPassword()
		checkFirstName()
		checkSurname()
		checkPhoneNumber()
		checkEmail()
		checkDateOfBirth()
		checkAddress()
		storeUserInformationInDatabase()

7	User	getFirstName()
		setFirstName
		getSurname()
		setSurName
		getPhoneNumber()
		setPhoneNumber()
		getPassword()

		setPassword()
		getEmail()
		setEmail()
		getDateOfBirth()
		setDateOfBirth()
		receiveVerifictionCode()
		searchMedicine()

8	Medicine	getMedicineName()
		getQuantity()
		getPrice()
		getCompanyName()
		getProductionDate()
		setProductionDate()
		setMedicineName()
		decreaseMedicineQuantity()
		setQuantity()
		setPrice()
		setCompanyName()
		setProductionDate()
		setProductionDate()
		generateMedicineId()
		showMedicineInformation()

		updateMedicineQuantity()
		updateMedicinePrice()
		deleteMedicine()

9	Maintenance	getMaintanenceCost()
		getMaintanencePurpose()
		getDate()
		setMaintanenceCost()
		setMaintanencePurpose()
		setDate()
		updateMaintanenceCost()
		updateMaintanencePurpose()

10	Report	getOnlineRevenue()
		getOfflineRevenue()
		getDue()
		getReceivedDue()
		setReceiverDue()
		setMaintanceCost()
		setOnlineRevenue()
		setOfflineRevenue()
		setDue()
		setMaintanceCost()
		totalDueCalculation()

		calculateOnlineRevenue()
		calculateOfflineRevenue()
		calculateTotalMaintanenceCost()
		viewReport()
		storeReportInfrormation()

11	Representative	getNotification()
		getMedicineRequest()
		getEmail()
		setEmail()
		getCompany()
		setCompany()

12	Transaction	SetTransactionId()
		getTransactionId
		setCustomerId()
		getCustomerId()
		setEmployeeId()
		getEmployeeId()
		setMedicineId()
		getMedicineId()
		setIsDue()
		getIsDue()
		setDate()

		getDate()
		setTime()
		getTime()
		setQuantity()
		getQuantity()
		setonline_banking_transaction_number()
		getonline_banking_transaction_number()
		setPrice()
		getPrice()
		setTransactionType()
		getTransactionType()
		changeTransactionType()
		viewMedicineInformation()
		payForTransaction()
		updateTransactionQuantity()
		storeTransactionInformation()

13	Database	storeUserInformationInUserTable()
		storeMedicineInformationInMedicineTable()
		updateMedicineInformationInMedicineTable()
		storeMaintenanceInformationInMain

		tenanceTable()
		getUserInformationFromUserTable()
		setUserInformationInUserTable()
		getTransactionInformationFromTransactionTable()
		setTransactionInformationInTransactionTable()
		updateTransactionStatus()
		setInPaysDueTable()
		geInPaysDueTable()
		getRequestMedicineInformation()
		setRequestMedicineInformation()
		storeRepresentativeInformation()
		getRepresentativeInformation()
		updateRepresentativeInformation()
		setReceivedMedicineInformation()
		getReceivedMedicineInformation()
		getReportInformation()
		storeReportInformation()
		setReportInformation()
		searchMedicine()

Table 19: Method Selection

6.7 Finalizing Classes

1) ‘log in’, ’log out’, forget password’ are merged as methods into a class named authentication.

2) As manager class contains all of the attributes and methods of employee class, so it has been implemented as a child class of employee

3) To edit, get and set information of owner, employee, manager and customer classes a new class named ‘User’ has been established and these classes have been implemented as child class of profile class

4) A new attribute ‘transaction id’ has been added into payment class to keep track of customer who paid for a transaction

5) A new method ‘changeTransactionType ()’ has been implemented inside Transaction class to change a due transaction into a paid transaction and update it into database for a customer who has paid for a due transaction

6.8 Class Cards

After identifying our final class, we have generated the following class cards.

Owner	
Attributes	Methods

Owner_id
Employee
	verifyReport()
	recruitEmployees()
	getOwnerId()
	setOwnerId()
	storeEmployeeInformationInDatabase()

Responsibilities	Collaborative class
Employee recruitment	Employee
Searching	Manager
Report validation	Report
Authentication	Authentication

Table 20: Class Card of Owner

Manager	
Attributes	Methods
Salary	inputMaintainenceInformation()
Joining_date	updateMedicineInformation()
Manager_id	generateReport()
	sendMedicineRequest()
	performTransaction()
	getManagerId()
	setManagerId()
	receiveNotificationFromRepresentative()
Responsibilities	Collaborative class
Transaction	Employee
Report generation	Report
Notification generation	Representative
Authentication	Authentication

Request and receive medicine Searching Maintenance cost management Medicine information management Profile editing	Representative Maintenance Medicine

Table 21: Class Card of Manager

Employee	
Attributes	Methods

Salary
joiningDate
employee_id
	performTransaction()
	getEmployeeId()
	setEmpoyeeId()

Responsibilities	Collaborative class
Authentication Transaction Searching Profile editing Receive payment	Authentication Transaction User

Table 22: Class Card of Employee

Medicine	
Attributes	Methods

medicine_Name
medicine_id
quantity
price
production date
expire date
company name
	getMedicineName()
	getQuantity()
	getPrice()
	getCompanyName()
	getProductionDate()
	setProductionDate()
	setMedicineName()
	decreaseMedicineQuantity()
	setQuantity()
	setPrice()
	setCompanyName()
	setProductionDate()
	setProductionDate()
	generateMedicineId()
	showMedicineInformation()
	updateMedicineQuantity()
	updateMedicinePrice()
	deleteMedicine()

Responsibilities	Collaborative class
Medicine id generation	

Table 23: Class Card of Medicine

Customer	
Attributes	Methods

customer_id	performTransaction()
	payForTransaction()
	receiveOrderSendNotification()

Responsibilities	Collaborative class
Authentication Transaction Searching Profile editing Receive notification Pay for due	Authentication Transaction User Database

Table 24: Class Card of Customer

Report	
Attributes	Methods

online_revenue
offline_revenue
due
	getOnlineRevenue()
	getOfflineRevenue()
	getDue()

maintenance cost
received_due
totalSaleAmount
	getReceivedDue()
	setReceiverDue()
	setMaintanceCost()
	setOnlineRevenue()
	setOfflineRevenue()
	setDue()
	setMaintanceCost()
	totalDueCalculation()
	calculateOnlineRevenue()
	calculateOfflineRevenue()
	calculateTotalMaintanenceCost()
	viewReport()
	storeReportInfrormation()

Responsibilities	Collaborative class
Generate weekly or monthly sales report Calculate total due amount	

Table 25: Class Card of Report

Transaction	
Attributes	Methods
transaction_id	SetTransactionId()

customer_id
employee_id
medicine_id
isDue
transaction_date
transaction_time
transaction_type
quantity
price
online_banking_transaction_number
	getTransactionId
	setCustomerId()
	getCustomerId()
	setEmployeeId()
	getEmployeeId()
	setMedicineId()
	getMedicineId()
	setIsDue()
	getIsDue()
	setDate()
	getDate()
	setTime()
	getTime()
	setQuantity()
	getQuantity()
	setonline_banking_transaction_number()
	getonline_banking_transaction_number()
	setPrice()
	getPrice()
	setTransactionType()
	getTransactionType()
	changeTransactionType()
	viewMedicineInformation()

	payForTransaction()
	updateTransactionQuantity()
	storeTransactionInformation()

Responsibilities	Collaborative class
Transaction id generation Price calculation Change of transaction type Generate cash memo Search medicine Pay for transaction	Database Report Employee Customer

Table 26: Class Card of Transaction

User	
Attributes	Methods

FirstName
surName
phone number
password
email
dateOfBirth
	getFirstName()
	setFirstName
	getSurname()
	setSurName
	getPhoneNumber()
	setPhoneNumber()
	getPassword()
	setPassword()

	getEmail()
	setEmail()
	getDateOfBirth()
	setDateOfBirth()
	receiveVerifictionCode()
	searchMedicine()

Responsibilities	Collaborative class
Update own attributes	Database

Table 27: Class Card of User

Maintenance	
Attributes	Methods

Date
time
purpose
amount
	getMaintanenceCost()
	getMaintanencePurpose()
	getDate()
	setMaintanenceCost()
	setMaintanencePurpose()
	setDate()
	updateMaintanenceCost()
	updateMaintanencePurpose()

Responsibilities	Collaborative class
Store maintenance cost and other	Database

releated informations	

Table 28: Class Card of Maintenance

Representative	
Attributes	Methods

Email
company name
	getNotification()
	getMedicineRequest()
	getEmail()
	setEmail()
	getCompany()
	setCompany()

Responsibilities	Collaborative class
Receive medicine request Deliver requested medicine	Manager

Table 29: Class Card of Representative

Registration	
Attributes	Methods

First name
surname
phone number
	setFirstName()
	setSurname()
	setPhoneNumber()

password
address
date of birth
	setEmail()
	setDateOfBirth()
	setAddress()
	setPassword()
	getFirstName()
	getSurname()
	getPhoneNumber()
	getEmail()
	getDateOfBirth()
	getAddress()
	getPassword()
	checkFirstName()
	checkSurname()
	checkPhoneNumber()
	checkEmail()
	checkDateOfBirth()
	checkAddress()
	storeUserInformationInDatabase()

Responsibilities	Collaborative class
Check information’s validity Store user information in database	Database

Table 30: Class Card of Registration

Database	
Attributes	Methods
DBName	storeUserInformationInUserTable()
password	storeMedicineInformationInMedicineTable()
DB_User	updateMedicineInformationInMedicineTable()
	storeMaintenanceInformationInMaintenanceTable()
	getUserInformationFromUserTable()
	setUserInformationInUserTable()
	getTransactionInformationFromTransactionTable()
	setTransactionInformationInTransactionTable()
	updateTransactionStatus()
	setInPaysDueTable()
	geInPaysDueTable()
	getRequestMedicineInformation()
	setRequestMedicineInformation()
	storeRepresentativeInformation()
	getRepresentativeInformation()
	updateRepresentativeInformation()
	setReceivedMedicineInformation()

	getReceivedMedicineInformation()
	getReportInformation()
	storeReportInformation()
	setReportInformation()
	searchMedicine()

Responsibilities	Collaborative Class
Store all the data Search data	

Table 31: Class Card of Database

Authentication	
Attributes	Methods

user_id
password
verificationCode
blockTime
	logInUser()
	checkUserIdPassword()
	retryLogin()
	checkBlockTime()
	logOut()
	passwordRecovery()
	generateVerificationCode()
	sendVerificationCode()
	getAndValidateVerificationCode()

	receivePasswordAndUptdate()
	validateUserInformarion()
	disConnectDatabase()
	disConnectProfile()

Responsibilities	Collaborative Class
Check profile information Generate, send, receive verification code Secure disconnection	Database User

Table 32: Class Card of Authentication

Chapter 7 BEHAVIOR MODEL

7.1 Introduction

Behavior modeling is also referred to as State modeling, State machines. Behavior modeling is when one thinks of his ideas in terms of states and transitions. This requires both identifying all of the interesting states of being that software or its components are likely to be in. And also, at a high level, abstracting what events are likely to cause software or its components to change between states of being.

7.2 Identifying Events

Here we have identified events from the **Usage Scenario** and listed their corresponding initiators & collaborators.

Events	Primary Object	Collaborative Objects	Invoked method
Input log in information	Owner	Authentication	logIn()
Input log in information	Manager	Authentication	logIn()
Input log in information	Employee	Authentication	logIn()

Input log in information	Customer	Authentication	logIn()
Check information validity	Authentication	Database	getUserInformation()
Input information	Registration		
Checks information validity	Registration		
Stores information into database	Registration	Database	setUserInformation()
Checks in database	Authentication	Database	getUserInformation()
Input email to get a new password	Authentication	User	getEmail()
Input phone number to get a new password	Authentication	User	getPhoneNumber()
Generate and send verification code	Authentication		
Search medicine	Customer	Database	getMedicineInformation()
Search medicine	Employee	Database	getMedicineInformation()
Select medicine	Customer	Medicine	selectMedicine()
Select medicine	Employee	Medicine	selectMedicine()

Selected medicine store in database	Medicine	Transaction	setTransaction()
Submit order	Customer	Order	storeTransactionInDatabase()
Pay for transaction	Customer	Transaction	paymentMethod()
Pay for due transaction	Customer	Transaction	paymentMethod()
Update due transaction	Customer	Transaction	paymentMethod()
Update due status	Transaction	Database	updateTransactionStatus()
Receive message for due transaction	Notification	Customer	receieveNotification()
Receive notification for requested medicine supply	Representative	Manager	receiveNotification()

Input maintenance cost	Manager	Maintenance	getMaintanenceCost()
			getMaintanenecePurpose()
			getDate()
			setMaintanenceCost()
			setMaintanenecePurpose()
			setDate()

Store maintenance	Maintenance	Database	setMaintenanceInformatio

information			n()
Request for medicine supply	Manager	Representitive	getMedicineRequest()
Generate report	Manager	Report	viewReport()
Submit report	Manager	Owner	receiveReport()
Recruits employee	Owner		
Log out	User	Authentication	LogOut()
View medicine information	Transaction	Medicine	ViewMedicineInformation( )
Store transaction information	Transaction	Database	storeTransactionInformation()
Store online order	Transaction	Database	storeTransactionInformation()
Store online payment transaction number	Transaction	Database	
Receive notification for medicine supply	Representative	Manager	getNotification()
Claculate online revenue	Report	Database	calculateOnlineRevenue()
Claculate offline revenue	Report	Database	calculateOfflineRevenue()
Claculate total due amount	Report	Database	totalDueCalculation()
Calculate total	Report	Database	calculateTotalMaintanenence

maintenance cost			Cost()
Store report information	Report	Database	storeReportInfrormation()
Receive order approval message	Customer	Manager	receiveOrderApprovalNotification()
Update medicine quantity	Medicine	Database	updateMedicineQuantity()
Delete medicine information	Medicine	Database	deleteMedicine()
Update medicine price	Medicine	Database	updateMedicinePrice()
Generate medicine id	Medicine	Database	generateMedicineId()
Stores employee information	Owner	Database	setUserInformation()

Table 33: Identifying Events

7.3 State Transition Diagram

State Transition Diagram represents active states for each class and the events(triggers) that cause changes between these active states. Here we have provideddiagram for each of the actors.

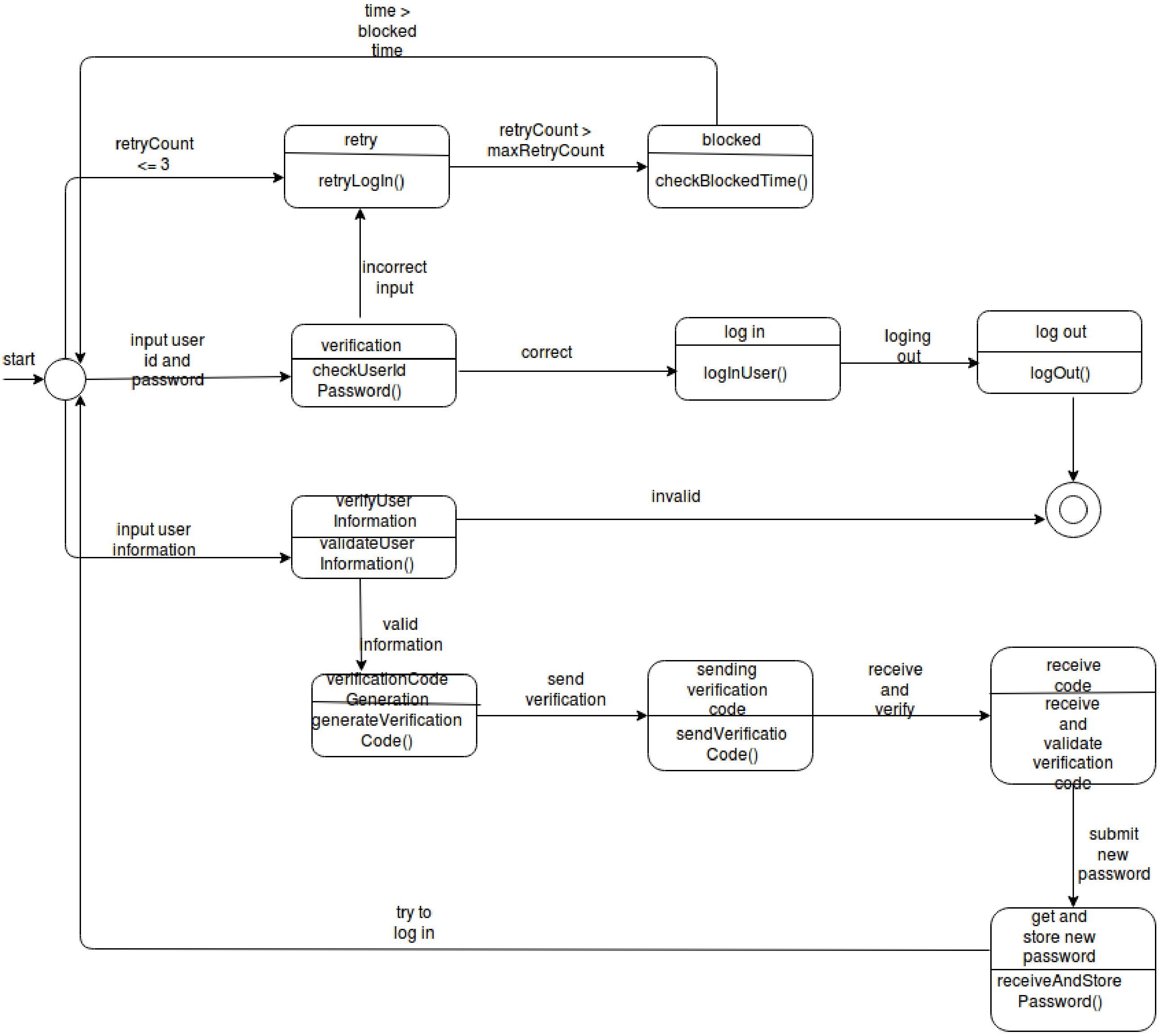

Figure 34: State transition diagram of authentication class

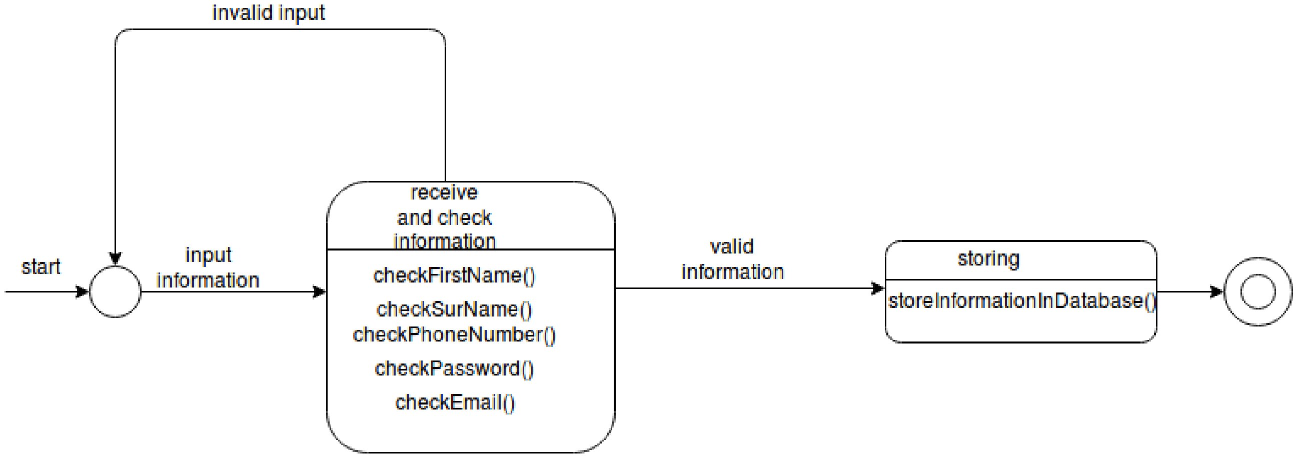

Figure 35: State transition diagram of registration class

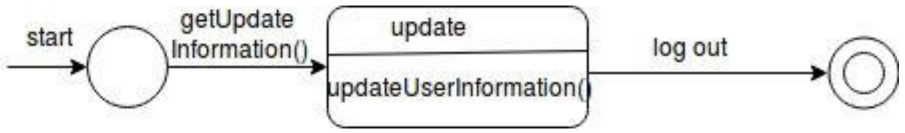

Figure 36: State transition diagram of user class

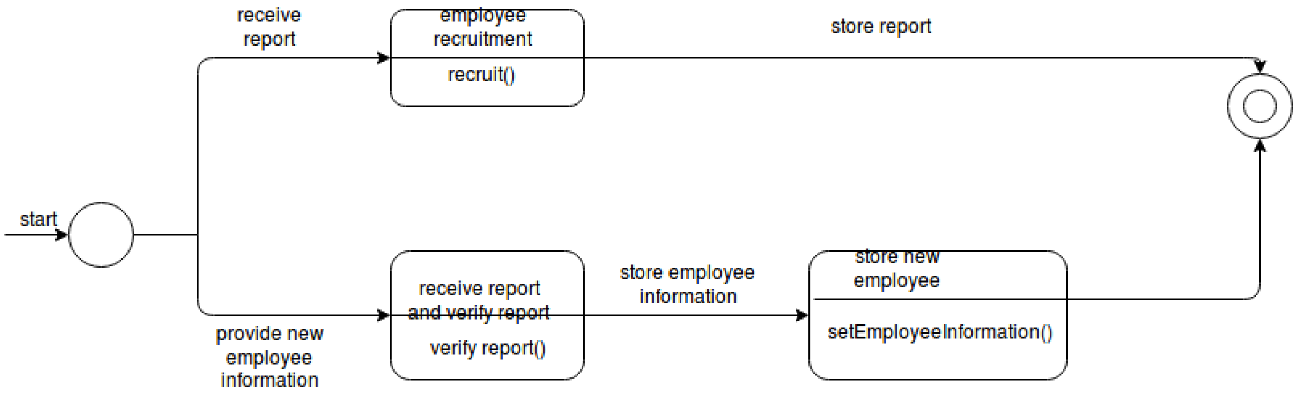

Figure 37: State transition diagram of owner class

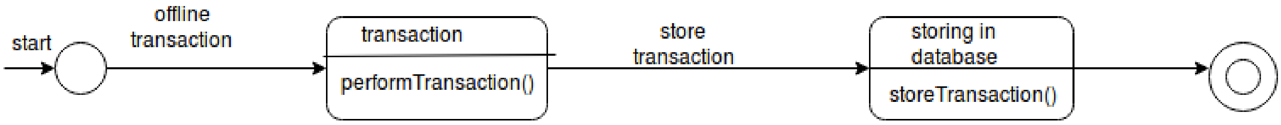

Figure 38: State transition diagram of employee class

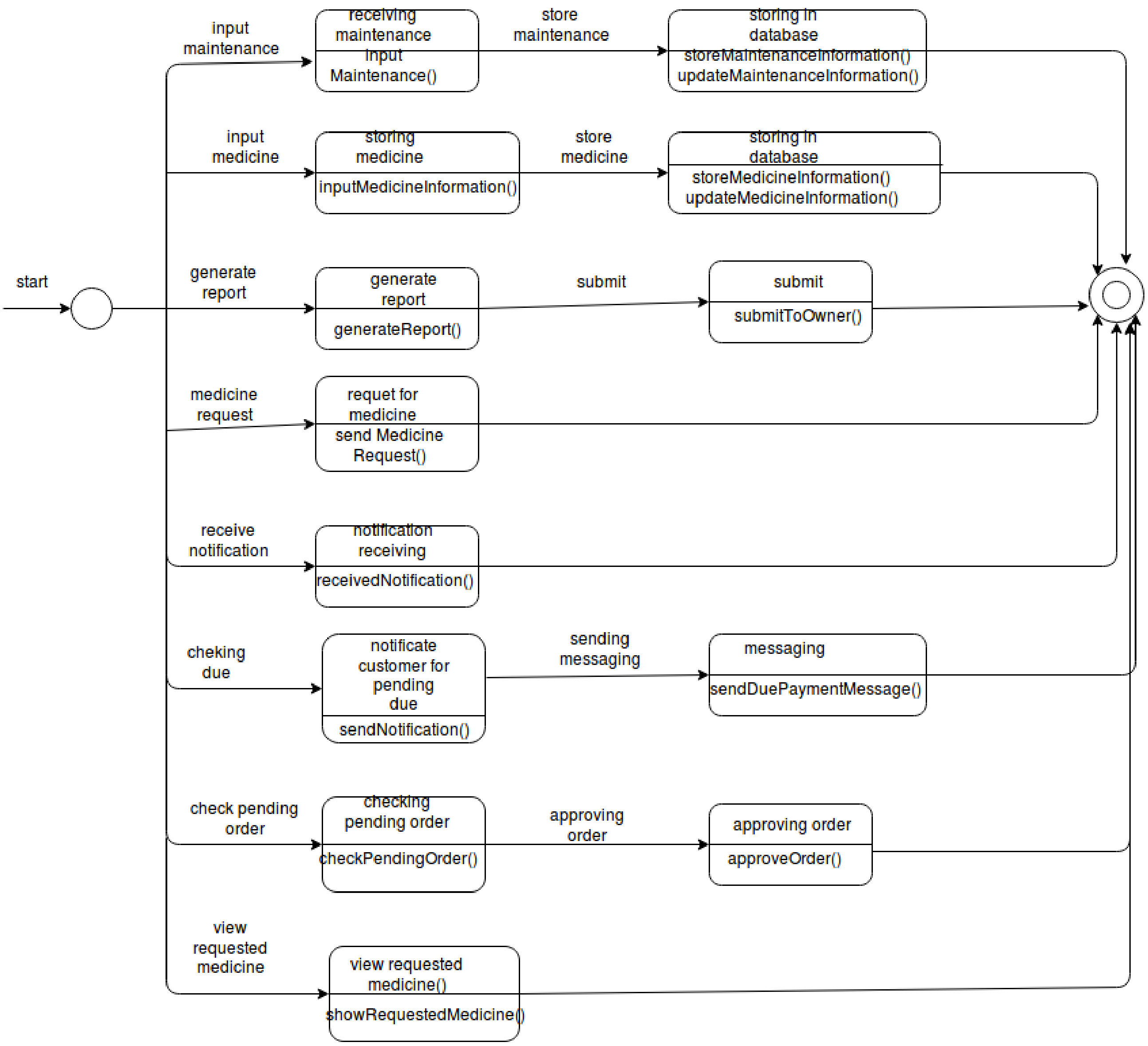

Figure 39: State transition diagram of manager class

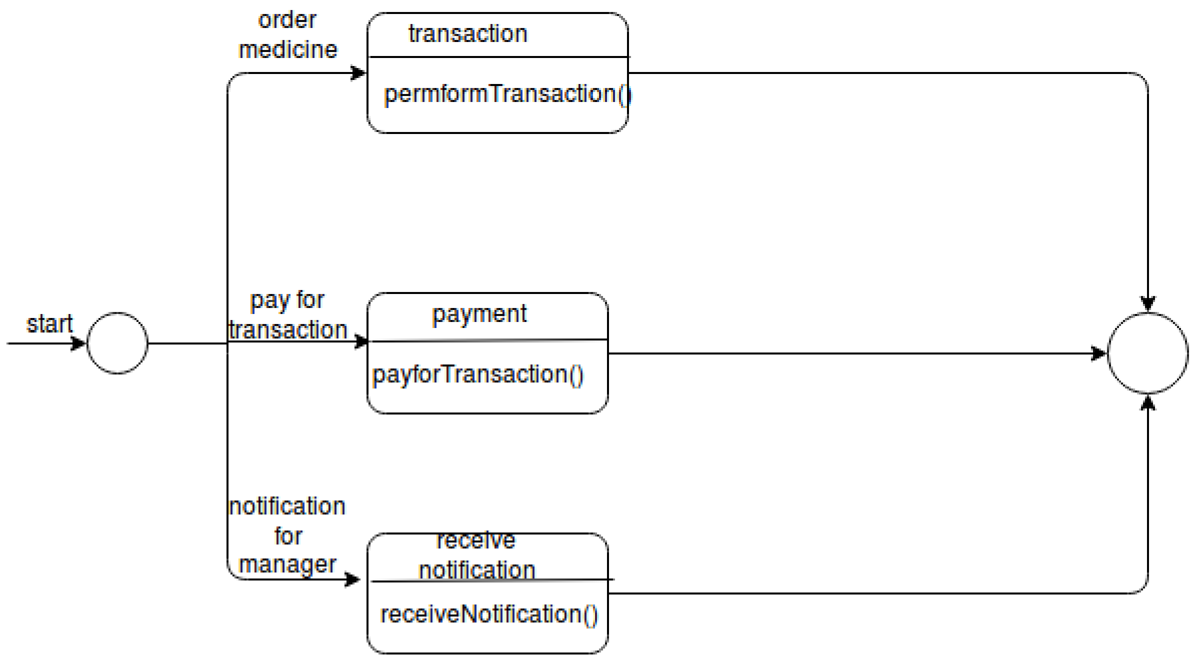

Figure 40: State transition diagram of customer class

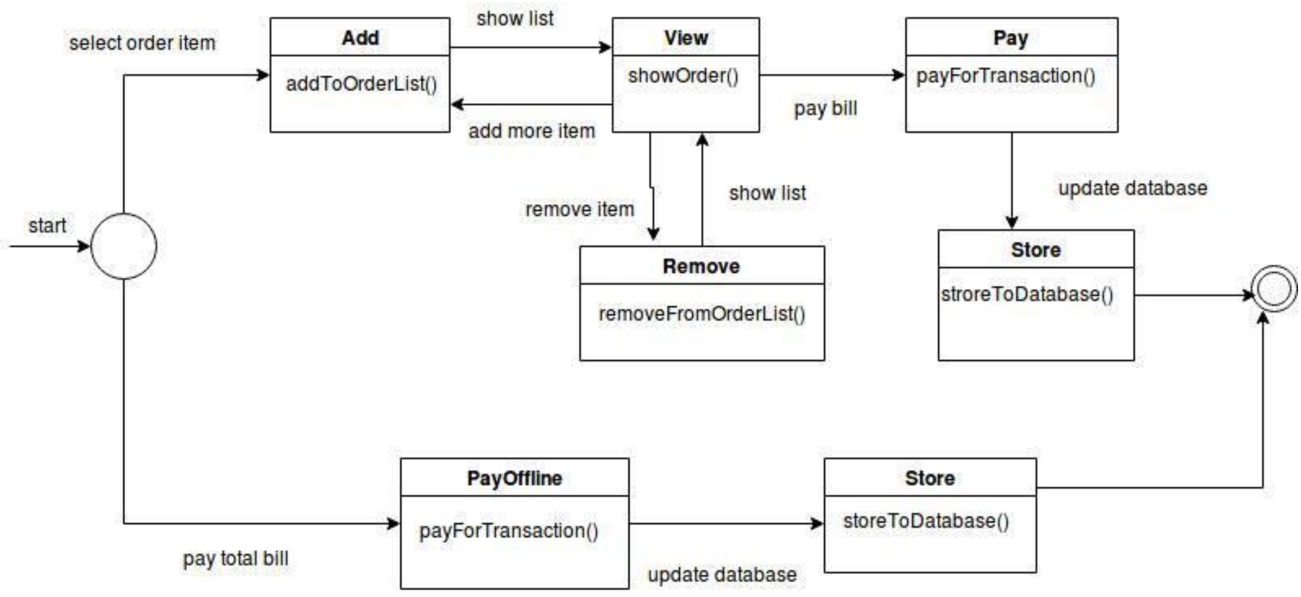

Figure 41: State transition diagram of transaction class

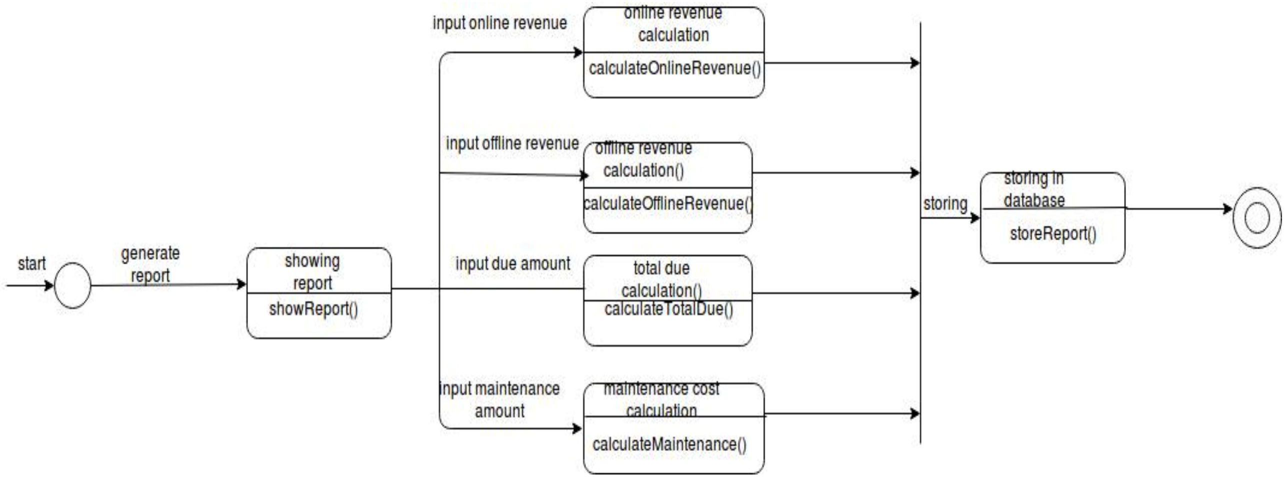

Figure 42: State transition diagram of report class

7.3 SEQUENCE DIAGRAM OF PMS

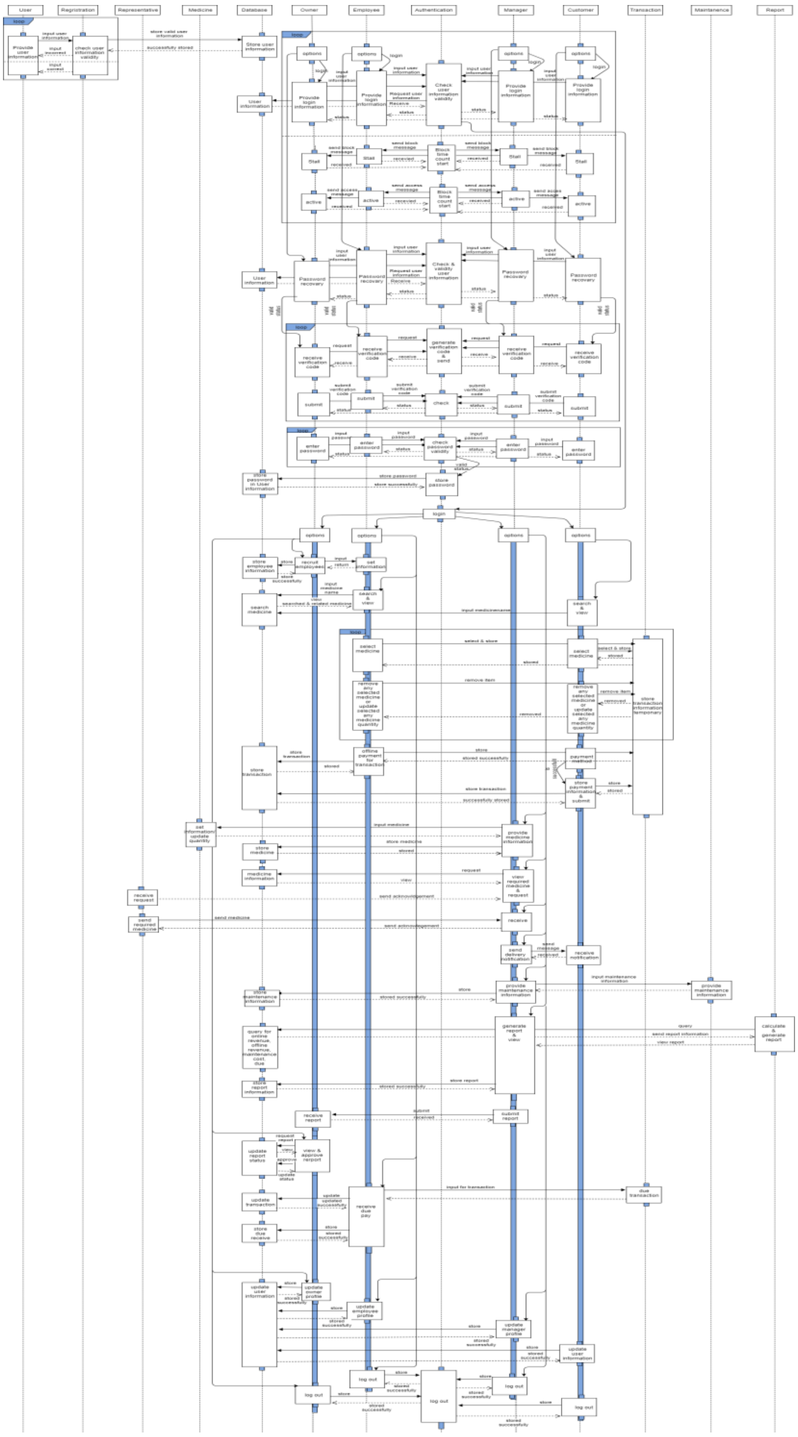

Figure 43: Sequence Diagram

reference

- Pressman, Roger S. Software Engineering: A Practitioner's Approach (7th Edition)

Chapter 8 Appendix

Chapter 9 Group Meeting

Meeting report-1

Meeting Information

Date: 16-07-2017
Location: IIT, DU, Lab406
Time: 4:30-5:00 pm

Attendance:

ROLL	NAME
0807	Mahir Mahbub
0815	Shuvradeb Saha
0822	S. M. Khayrul Islam
0839	Sefat-E-Mahadi

Discussion Agenda

- Introducing with each other among in the group.
- Discussing about the basic of our project topic.
- Selecting pharmacies to visit for the project.

Achievements

- Came to the point that we would visit the medium and large size pharmacies to identify the stakeholders.
- Selected the visiting areas.

Next Meeting

Date: 18-07-2017

Group 6

Meeting report-2

Meeting Information

Date: 18-07-2017
Location: IIT, DU, Lab406
Time: 4:30-5:00 pm

Attendance:

ROLL	NAME
0807	Mahir Mahbub
0815	Shuvradeb Saha
0822	S. M. Khayrul Islam
0830	Obaidur Rahman
0839	Sefat-E-Mahadi

Discussion Agenda

- Identifying all the stakeholders of the pharmacies.
- Identifying contribution of every stakeholders to maintain a pharmacy.
- Learning about their business policies, rules, regulations and techniques.

Achievements

1. Started conversation with the stakeholders with context free questions.
2. Discussed with them about the problems they face to maintain the pharmacy with trivial system.
3. Knew about facilities they expect from the automated software system.
4. Offered them more user-friendly features that the system should contain to help them with conveniences.

Next Meeting

Date: 23-07-2017

Group 6

Meeting report-3

Meeting Information

Date: 23-07-2017
Location: IIT, DU, Lab406
Time: 4:00-5:00 pm

Attendance:

ROLL	NAME
0807	Mahir Mahbub
0815	Shuvradeb Saha
0822	S. M. Khayrul Islam
0830	Obaidur Rahman
0839	Sefat-E-Mahadi

Discussion Agenda

1. Finding out the problem face by the stakeholders in the trivial system by asking questionnaires.
2. Focusing on the needs of the stakeholders.
3. Discussing with them about the easiness and features they may expect in an automated system to deal with customers.
4. Offering them extra additional features to them.
5. Elaborating on quality function deployment.

Achievements

1. Recognized multiple view points of the stakeholders.
2. Worked towards collaboration by identifying common requirements, conflicting requirements and finalize requirements
3. Finalized quality function deployment.

Next Meeting

Date: 01-8-2017

Group 6

Meeting report-4

Meeting Information

Date: 01-08-2017
Location: IIT, DU, Lab406
Time: 4:15-5:00 pm

Attendance:

ROLL	NAME
0807	Mahir Mahbub
0815	Shuvradeb Saha
0822	S. M. Khayrul Islam
0839	Sefat-E-Mahadi

Discussion Agenda

- Identifying the sub systems.
- Specifying the use case scenarios.
- Initiating Use Case.
- Translating to Use Case Diagram

Decisions

- Gathered the raw requirements of stakeholders and analyze those.
- Finalized the sub systems
- Determined the basic use cases and will elaborate them.
- Use case has to be written per instruction.
- Leraned basic methodology of use case diagram.

Next Meeting

Date: 06-08-2017

Group 6

Meeting report-5

Meeting Information

Date: 06-08-2017
Location: TSC, University of Dhaka
Time: 3:00-4:30 pm

Attendance:

ROLL	NAME
0807	Mahir Mahbub
0815	Shuvradeb Saha
0822	S. M. Khayrul Islam
0839	Sefat-E-Mahadi

Discussion Agenda

- Swim line diagram and activity diagram

Decisions

- Structure of Swim line diagram and activity diagram.
- Methodology to draw Swim line and activity diagram.

Next Meeting

Date: 10-09-2017

Group 6

Meeting report-6

Meeting Information

Date: 24-09-2017
Location: IIT, DU LAB 406
Time: 3:15-4:30 pm

Attendance:

ROLL	NAME
0807	Mahir Mahbub
0815	Shuvradeb Saha
0822	S. M. Khayrul Islam
0839	Sefat-E-Mahadi

Discussion Agenda

Identifying potential entities
Finalizing entities and attributes of entities
Identifying relationship between entities
Developing the ER diagram
Constructing schema tables

Decisions

- Identified potential entities
- Finalized entities and attributes of entities
- Identified relationship between entities
- Developed the final ER diagram
- Constructed schema tables

Next Meeting

Date: 24-10-2017

Group 6

Meeting report-7

Meeting Information

Date: 10-09-2017
Location: Fazlul Haque Muslim Hall, University of Dhaka
Time: 3:15-4:30 pm

Attendance:

ROLL	**NAME**
0807	Mahir Mahbub
0815	Shuvradeb Saha
0822	S. M. Khayrul Islam
0839	Sefat-E-Mahadi

Discussion Agenda

- Post vacation re-union
- Noun identification for data modeling

Decisions

- Basic Methodology of Noun identification
- Noun parsing from story

Next Meeting

Date: 24-09-2017

Group 6

Meeting report-8

Meeting Information

Date: 24-10-2017
Location: IIT, DU LAB 406
Time: 3:15-4:30 pm

Attendance:

ROLL	NAME
0807	Mahir Mahbub
0815	Shuvradeb Saha
0822	S. M. Khayrul Islam
0839	Sefat-E-Mahadi

Discussion Agenda

- Performing general classification
- Performing selection criteria
- Identifying potential classes
- Identifying associated nouns and verbs of each potential classes
- Identifying attributes of each potential classes
- Identifying methods of each potential classes
- Finalizing classes for the entire system
- Constructing class cards
- Completing the CRC diagram of the system

Decisions

- Performed general classification
- Performed selection criteria
- Identified potential classes

- Identified associated nouns and verbs of each potential classes
- Identified attributes of each potential classes
- Identified methods of each potential classes
- Finalized classes for the entire system
- Constructed class cards

9. Completed the CRC diagram of the system

Next Meeting

16-11-2017

Group 6

Meeting report-9

Meeting Information

Date: 16-11-2017
Location: IIT, DU LAB 406
Time: 3:15-4:30 pm

Attendance:

ROLL	NAME
0807	Mahir Mahbub
0815	Shuvradeb Saha
0822	S. M. Khayrul Islam
0839	Sefat-E-Mahadi

Discussion Agenda

- Constructing a table that includes all events, initiator classes of events, collaborator classes of events and associated methods.
- Constructing the state transition diagram of the entire system to be developed

Decisions

- Constructed a table that includes all events, initiator classes of events, collaborator classes of events and associated methods.
- Constructed the state transition diagram of the entire system to be developed

Next Meeting

Date: 17-11-2017

Group 6

Meeting report-10

Meeting Information

Date: 17-11-2017
Location: IIT, DU LAB 406
Time: 3:15-4:30 pm

Attendance:

ROLL	**NAME**
0807	Mahir Mahbub
0815	Shuvradeb Saha
0822	S. M. Khayrul Islam
0839	Sefat-E-Mahadi

Discussion Agenda

- Writing and finalizing reports
- Drawing posters for the final presentation

Decisions

- Wrote and finalized reports
- Drawn posters for the final presentation