Table of Contents

1 INTRODUCTION

This chapter is a part of our software requirement specification for the project "Grocery Management System". In this chapter we will focus on the intended audience for this project.

1.1 PURPOSE

This document briefly describes the Software Requirement Analysis of Grocery Management System. It contains the functional, non-functional and the supporting requirements and establishes a requirement's baseline for the development of the system. The requirements contained in the SRS are independent, uniquely numbered and organized by topics. The SRS serves as an official means 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 report is intended for several audiences including the customers 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. 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 of 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.

2 INCEPTION OF GMS

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

2.1 INTRODUCTION

A grocery store is a retail store that sells daily commodities. They are normally small private business owner. This document will act as a demand list of what the stakeholders of their respective wanted for their anticipated software to be. Like the various features and interfaces of the software which will help them to ease the way of their business.

2.2 INCEPTION OF A GROCERY BUSINESS

At the beginning of our project, we entered the inception stage. This stage includes, how the project will be started and their scope and limitation. The main goal of this phase is to identify the requirements, demand and establish some sort of mutual understanding between the software team and the stakeholders of the groceries. In order to make this phase effective we took the following steps:

- Identifying the client of our project
- Icebreaking
- Identifying the stakeholders of the grocery shop
- Identifying the multiple viewpoints of stakeholder

2.2.1 IDENTIFY THE CLIENT OF OUR PROJECT

At first, we identified the location from where we will start our expedition. Normally the shopkeepers do not always act as a stakeholder. So we have to go through a lengthy process in order to identify them. We have analyzed our requirements with the consent of both of them.

2.2.2 ICEBREAKING

Icebreaking refers to the fact that to diminish the communication barrier between you and the other person. It is a crucial part since it denotes the acceptation of our proposal. We started this face by talking with them with context free languages. Their behavior, respond to our question or willing to take a change in their shops solely depends on this phase.

2.2.3 IDENTIFYING THE STAKE HOLDERS OF THE GROCERY SHOP

Stakeholder refers to any person or group who will be affected directly or indirectly by the system. Stakeholders include end-users who interact with the system and everyone else in an organization who may be affected by its installation. The shops that we visited have limited number of stakeholder. Identification of the stakeholders were done from the information provided by the shopkeepers. Their names are given below:

- Supplier
- Owner
- Employee
- Customer

2.2.4 IDENTIFYING THE MULTIPLE VIEWPOINTS OF THE STAKEHOLDER

Different stakeholders expect different benefits from the system as every person has his own point of view. So, we have to recognize the requirements from multiple viewpoints. Different viewpoints of the stakeholders about the expected software are given below:

Owner's Viewpoint:

- First and foremost, a really friendly user interface
- Desktop based software if affordable
- Provide signal when any product is short
- Store information about people who are working in the shop

- Calculate total amount of sell and total amount of buying goods and end of the month shows profit or loss
- Keep information about late payment
- Commission calculation
- Regular customer list
- Online shopping facility
- Provides SMS to all the regular customers and nearby shops whenever offer/s will be given
- Create a local network of the nearby groceries
- Some automated system providing in the digital weighted machine
- Must have minimal cost

Employee‘s Viewpoint:

- Ease of calculation
- Minimum effort to use the software

Supplier‘s Viewpoint:

- Reduce the time for ordering
- Must be cost effective as well

Customer‘s Viewpoint:

- Want to get the promised services like online shopping of the software by going through an easy-to-understand procedure
- Service must carry out without any unwanted error or failure.

2.3 CONCLUSION

The primary goal of this project is to model and design a software for those people who are related with small scale businesses like grocery shop or pharmacy. For these reasons, the software will be designed in such a way that it won’t be disaster for the client who will use it. The software will be as simple as a person who does not have any idea about software he/she can

be able to maintain it without any annoyance. Otherwise it will not be appreciable by the clients even it may create disturbance with their businesses. The software will be designed in such a way as it takes very little time to manage. To make this software project successful, collaboration with the stakeholders was a main priority that what they want, how the software will work, how it can be more profitable than previous time, how it will save time to maintain the business policy etc.

3 ELICITATION OF GMS

After discussing on the Inception phase, we need to focus on the Elicitation phase. So 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 shopkeeper, owner and other stakeholders. We have faced many difficulties, 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, we have surpassed these problems in an organized and systematic manner.

3.2 ELICITING REQUIREMENTS

We have seen Question and Answer (Q&A) approach in the previous chapter, where the inception phase of requirement engineering has been described. 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. We have finished the following tasks for eliciting requirements-

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

3.2.1 COLLABORATIVE REQUIREMENTS GATHERING

We have met with many stakeholders in the Inception phase such as the shopkeeper, owner and supplier. These meetings created an indecisive state for us to elicit the requirements. To solve this problem, we have met with

the stakeholders (who are acting a vital role in the whole process) few times to elicit the requirements.

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. So, we have followed this methodology to identify the requirements for the project. The requirements, which are given below, are identified successfully by the QFD.

3.2.2.1 NORMAL REQUIREMENTS

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 our project.

- First and foremost, a really friendly user interface
- Desktop based software if affordable
- Store information about people who are working in the shop
- Calculate total amount of sell and total amount of buying goods and end of the month shows profit or loss
- Keep information about late payment
- Must have minimal cost
- Ease of calculation
- Minimum effort to use the software
- Reduce the time for ordering from the owner

3.2.2.2 EXPECTED REQUIREMENTS

These requirements are intrinsic to the product or system and may be so elementary that the customer does not explicitly state them. Their absence will be a cause for significant dissatisfaction. Below the expected requirements for our project are briefly described.

- Storing all shop related information
- Interactive and attractive graphical user interface
- Authentication process

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 our project.

- Provide automatic signal when any product is short.
- Provide automatic signal when any product is expired.
- Create auto-generated financial report.

3.2.3 USAGE SCENARIO

Grocery Management System (GMS) is an automated system for the following purposes:

1. Authentication
2. Shop management
3. Accounting
4. Notification

Authentication

System has two types of users:

- Administrator

- Shopkeeper

At the time of installation, a user should be created as an administrator. Thereafter, more user(s) can create account(s). For creating a shopkeeper account, the approval of an administrator account is needed. For any type of account creation, the following information needs to be given-

- User name
- Full name
- Permanent address
- Current address
- Designation (Two options: Administrator, shopkeeper)
- NID
- Mobile number
- Password (should be at least 8 characters at most 32 characters)

Then user information will be stored. Shopkeeper has salary and commission along with all the attributes mentioned above.

For user name, only alphanumeric characters and underscore (‘_’) are allowed. If a user tries to create an account with empty field or contain invalid input, the system shows an error message and allows him/her to try

again. For administrator account system will provide an administrator id and for shopkeeper system will provide a shopkeeper id. All information regarding accounts should be stored. For login, users will have to provide user name or a valid mobile number and password.

For invalid input or input mismatch, system shows an error message. A user can try to login to the system for three consecutive times with invalid or wrong input at a time. After third login failure, the system will pause for two minutes. While logging out of any account, the system will check for any unsaved file and wants confirmation of logging out. An administrator can modify his own information. But a shopkeeper will need administrator's approval for editing any information.

If a user forget password, he can recover his account. He will enter his mobile number. The system will send his forgotten password in his mobile through contact number.

Shop Management

Shop management subsystem includes:

- HR management
- Product management
- Maintenance

If the administrator wants to fire any shopkeeper, he will first check all information about the shopkeeper and remove him. If any information about any shopkeeper needed to be modified, the administrator can update the list.

In product management section, product information of all the products are stored by the shopkeeper. The following information are stored:

- Product ID (PID)
- Product name
- Quantity
- Price
- delivery date
- expiry date
- company name
- status

User will check expiry date regularly. If the product is exchangeable, the user will contact with supplier(s) and exchange item(s). User will update expiry date of the product. Otherwise shopkeeper will inform administrator about the expired product and the administrator will take necessary steps to manage the product. User will check if inventory level is low or not. If inventory is low, he will inform the administrator. The administrator will either deal with the supplier or collect the products by his own. Then the cash and inventory level will be updated by him. If the administrator does not want to sell any product(s), he will remove it from the list.
If customer suggests any new product then it will be stored by the shopkeeper in a temporary suggested list. Then, the owner will check the list. If he approves the list then the product(s) will be stored. Otherwise, that suggested product will be removed by administrator.

Maintenance details will be stored separately by the user. Electricity bill, phone bill, repairing shop, adding furniture, repairing furniture, repairing fan, switching bulb are elements of maintenance. User will handle maintenance. User can also add new maintenance element. Elements with “done” status will be processed by the system and the expenditure information goes to the management part of the Accounting subsystem.

Accounting

Along with user(s), clients are also involved in this system. Client has client id, name, address, type and contact. Here, types can be supplier, customer and shopkeeper of other shop. Anything related to spending and earning money will be handled by this subsystem. This subsystem is divided into three parts:

1. Transaction
2. Management cost
3. Financial report

Both the shopkeeper or administrator can request transaction with the supplier of certain product(s) and of specified quantity. The supplier(s) will confirm the item delivery. The system will store supplier id, supplier mobile number, company name. After, delivery, transaction between supplier(s) and the user will take place. After payment, that information of product(s) must be kept into inventory. For this purpose, the following information about the transaction will be stored:

- Transaction id
- Transaction date
- Transaction time
- Supplier id
- Product name
- Quantity
- Unit cost
- Total cost
- Total amount

Transaction id will be auto-generated. Total cost will be calculated by the system. All the information will be stored by user. Transaction are of several types such as buying, selling, loan and expenditures.

At the time of buying product from supplier, a buying transaction will be created. If the user can not pay full amount, an agreement will be created as due type. The inventory level will be updated as either new product(s) will be added or quantity of some product(s) will be increased. On the other hand, selling transaction will take place between customer and the user(s). If any customer fails to pay full amount, another type of agreement called late payment will be created. The user can borrow product(s) from nearby shop with an agreement. Again, the inventory level will be updated as the quantity of product(s) will decrease. For an agreement following information must be kept:

- Transaction id
- Type
- Payee Id
- Receiver Id
- Amount
- Status
- Occurrence date
- Return date
- Description

.

If any due type agreement exists, user will check the cash. If he has enough amount to pay, he will complete payment with a transaction and update the cash. Otherwise, he can partially pay and issue a new date.

If any late payment occurs, the user will store information about the agreement. Before updating agreement information the user will check payable amount of the customer. This agreement can be updated in three situations:

- If the customer pays full amount
- If he pays partial amount
- If customer wants to buy on account, the user will check whether his due amount is less than 500 tk.

When agreement is updated transaction will also occur. When a customer asks for product(s), user will search for product(s). If the product(s) is found, user will add the product(s) quantity to receipt or if the customer does not want to buy the product(s) which was previously selected then user will remove the product(s) from the receipt. If desired product(s) are not available in the shop, the shopkeeper can bring those product(s) from nearby shop. If loan is not possible, shopkeeper will ask for next desired product(s). If desired product(s) is found, the user will enter the unit of the products(s). System will calculate the price and add it to the receipt. If the number of product is zero on the receipt, no transaction will take place. Otherwise, system will calculate total amount. Then, the receipt will be printed.

"Management cost" list firstly stores the information about maintenance namely maintenance element and related cost. This list also includes shop rent, loan payment, shopkeeper salary, bonus on different occasions. User will record these expenditure information. He may add, delete and update any element.

There will be a financial report which will contain information about total cash, total inventory, total account payables, total account receivable and capital of a specific time. Administrator can see the financial report.

Notification

There are three types of notification.

Two types of notification are generated by the system and sent to the owner and the shopkeeper. User will get notification(s):

i. If quantity of any product(s) is below a certain threshold.

ii. If expired date of any product exceeds.

When an owner or shopkeeper wants to include some product to the inventory, he may create a notification for the appropriate supplier. The system will send it to the desired supplier.
All the notification will have to store the notification id for each notification and also time, sender, receiver, type and description.

4 SCENARIO BASED MODELING OF "GROCERY MANAGEMENT SYSTEM"

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 we understand how end users (and other actors) want to interact with a system, our software team 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 USE CASE

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 describes a story 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.

4.3 USE CASE DIAGRAMS

Use Case diagrams give the non-technical view of overall system.

4.3.1 LEVEL-0 USE CASE DIAGRAM - GMS

Level-0 GMS
Version 1.1
Created on 11/13/2017. Last modified 11/16/2017

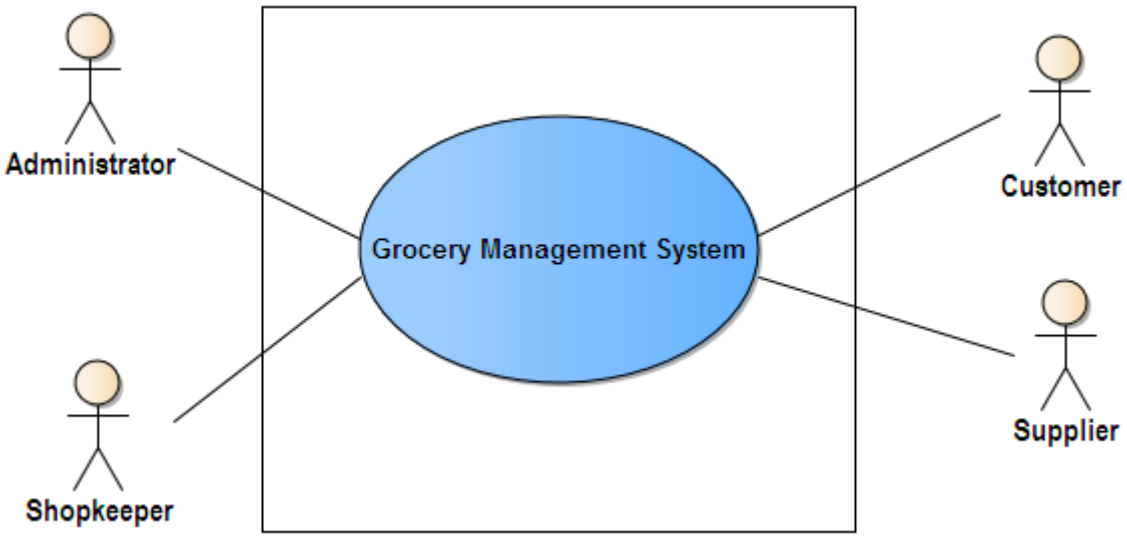


Figure 1: Level-0 GMS

Description of Use case diagram level-0:
After analyzing the user's story, we found 4 actors that directly or indirectly interacts with the system. Primary actors are those who will play action and get reply from the system whereas secondary actors only produce or consume information. The actors are -

- Shopkeeper
- Owner
- Customer
- suppliers

4.3.2 LEVEL-1 USE CASE DIAGRAM-SUB SYSTEM

Level-1 sub-system
Version 1.1
Created on 11/13/2017. Last modified 11/16/2017

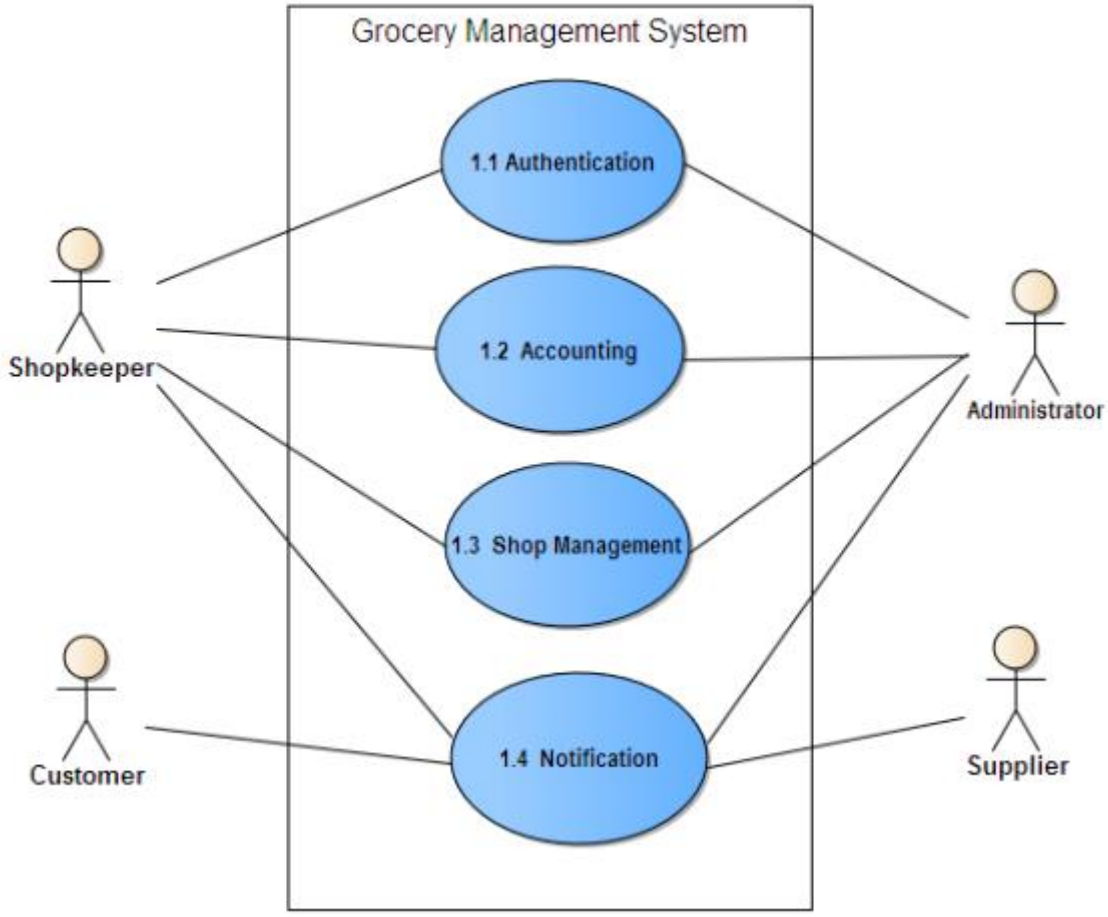


Figure 2: Sub-system

Description of level-1 use case diagram:

There are four subsystems in Grocery Management System. They are as follows:

1. Authentication
2. Accounting
3. Shop Management
4. Notification

The first three subsystems are further decomposed, in level 1.1, 1.2, 1.3 respectively. The description of notification system is given below.
There are five types of notification.
Three types of notification are generated by the system and sent to the owner and the shopkeeper will get notification(s):

i. If quantity of any product(s) is below a certain threshold.

ii. If expired date of any product exceeds.
iii. Immediate day before the loan period is over.

When an owner or shopkeeper wants to include some product to the inventory, he/she may create a notification for the appropriate supplier. The system will send it to the desired supplier.

Lastly, the owner or shop keeper may create notification for customer who have late payment issue(s). The system will send to the desired customer.

All the notification will have the store a notification id for each notification and also time, sender, receiver, type and description.

4.3.3 LEVEL-1.1 USE CASE DIAGRAM- AUTHENTICATION

1.1 Authentication
Version 1.1
Created on 11/13/2017. Last modified 11/13/2017

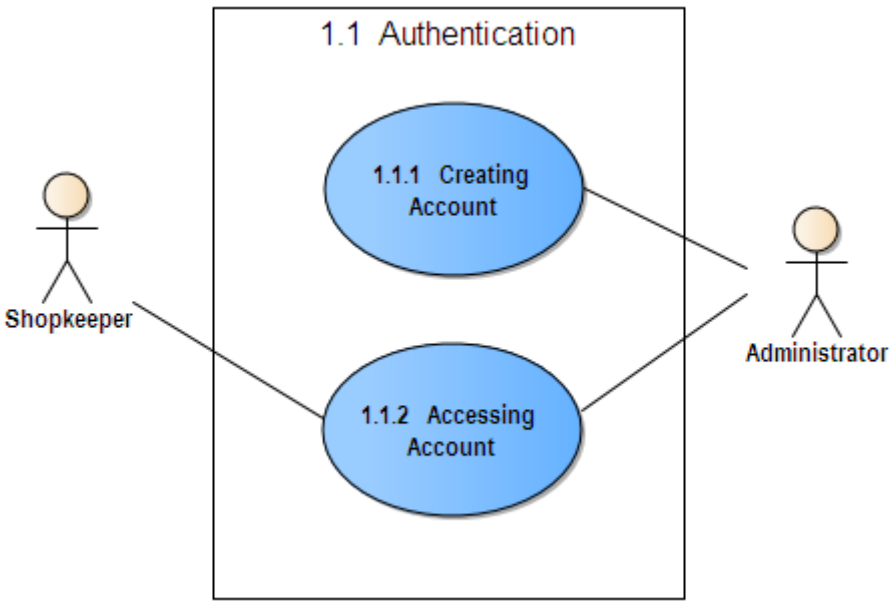


Figure 3: Level 1.1 Authentication

Description of level-1.1 use case diagram:

Authentication is a process in which credentials provided are compared to those on files in a database of authorized user's information. The authentication subsystem can be divided into two parts. They are as follows:

1. Creating Account
2. Accessing Account

4.3.4 LEVEL 1.1.1 CREATING ACCOUNT

Use Case diagram in package 'Use Case Model'

Level 1.1.1
Version 1.1
Created on 10/22/2017. Last modified 11/16/2017

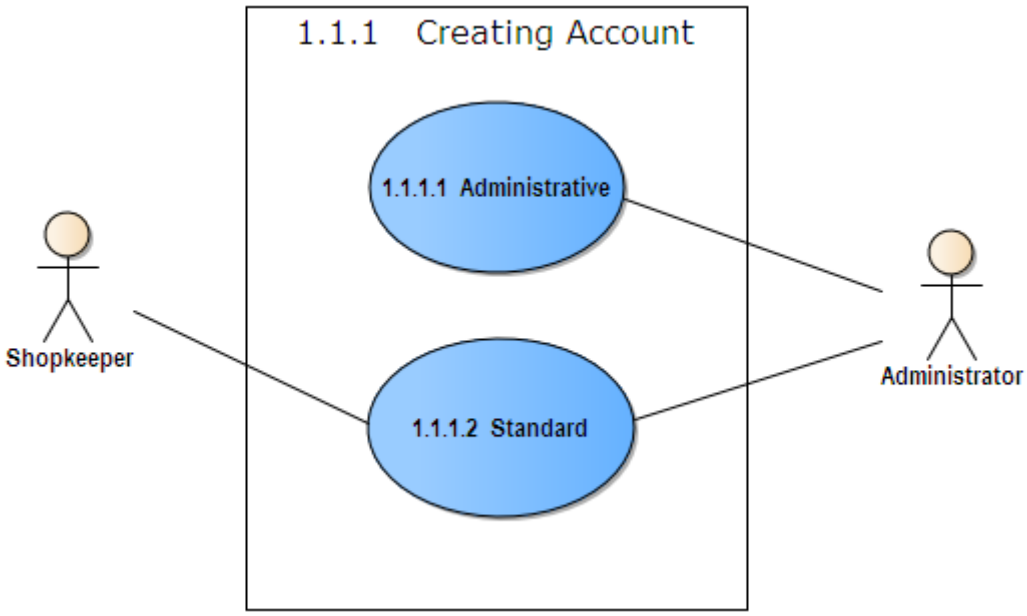


Figure 4: Level 1.1.1 Creating Account

Description of level-1.1.1 use case diagram:

System has two types of users:

- Administrator
- Shopkeeper

At the time of installation, an user should be created as an administrator. Thereafter, more user(s) can create account(s). For creating a shopkeeper account, the approval of an administrator account is needed. For any type of account creation, the following information needs to be given-

- User name
- Full name
- Permanent address

- Current address
- Designation (Two options: Administrator, shopkeeper)
- NID
- Mobile number
- Password (should be at least 8 characters at most 32 characters)

Shopkeeper has salary and commission along with all the attributes mentioned above.

If a user tries to create an account with empty field or contain invalid input, the system shows an error message and allows him/her to try again. For administrator account system will provide an administrator id and for shopkeeper system will provide a shopkeeper id. All information regarding accounts should be stored.

Action-Reply of Use Case Diagram Level 1.1.1:

Administrator:

- A1: Administrator creates an account filling with valid information.
  R1: System creates an administrator account and the account information is stored.
- A2: Administrator creates an account filling with invalid information.
  R2: System allows the administrator to try again for account creation.
- A3: Administrator accepts or delete request(s) for shopkeeper account creation.
  R3: System will work accordingly. If administrator accepts request then system will create and store that account(s).

Shopkeeper:

- A1: Shopkeeper creates an account filling with valid information.
  R1: System waits for administrator’s approval for creating shopkeeper account.

- A2: Shopkeeper creates an account filling with invalid information.
  R2: System allows the shopkeeper to try again for account creation.

4.3.5 LEVEL 1.1.2 USE CASE DIAGRAM-ACCESSING ACCOUNT

Level 1.1.2 Accessing-account
Version 1.1
Created on 10/22/2017. Last modified 11/16/2017

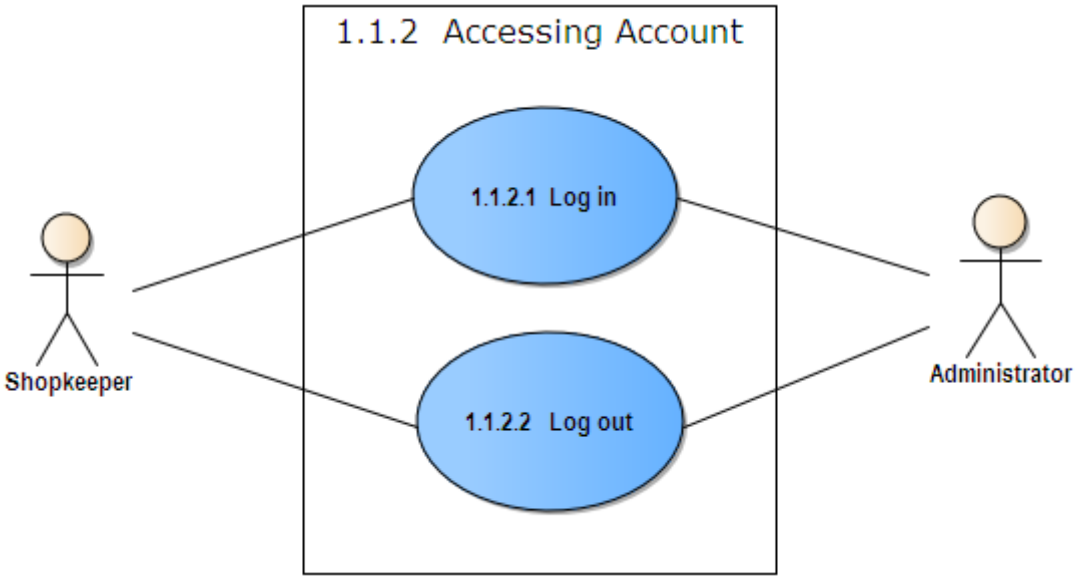


Figure 5: Level 1.1.2

Description of level-1.1.2 use case diagram:
Accessing account involves two tasks:

i. Log in

ii. Log out

For logging into the account, the user must provide mobile number or user name and password. User name should be consisting of alpha numeric characters and underscore. A valid mobile number can also be used in the user name field for login. Correct password should be provided. If the user(s) tries to log into the account with empty field or invalid input or incorrect user name or invalid mobile number, the system will show error message and will allow to try again. After, third time failure, the system will get locked for two minutes. While logging out of any account the system will check for any unsaved file and will want confirmation of logging out.

Action Reply

User (Administrator & shopkeeper)

- A1: User provides user name or valid mobile number and password.

R1: System will check validity. For valid information system will allow user(s) to log into the account.

- A2: User provides invalid information.
  R2: System will show error message and allows to try again.
- A3: User fails to log into the account for third time.
  R3: System will be blocked for two minutes.
- A4: Users will give command for log out.
  R4: System will search for unsaved file(s). If unsaved file(s) is found, system will ask for confirmation whether the account should be logged out or not. Otherwise, account will be logged out automatically.

4.3.6 LEVEL-1.2 USE CASE DIAGRAM-ACCOUNTING

Level 1.2 Accounting
Version 1.1
Created on 11/13/2017. Last modified 11/13/2017

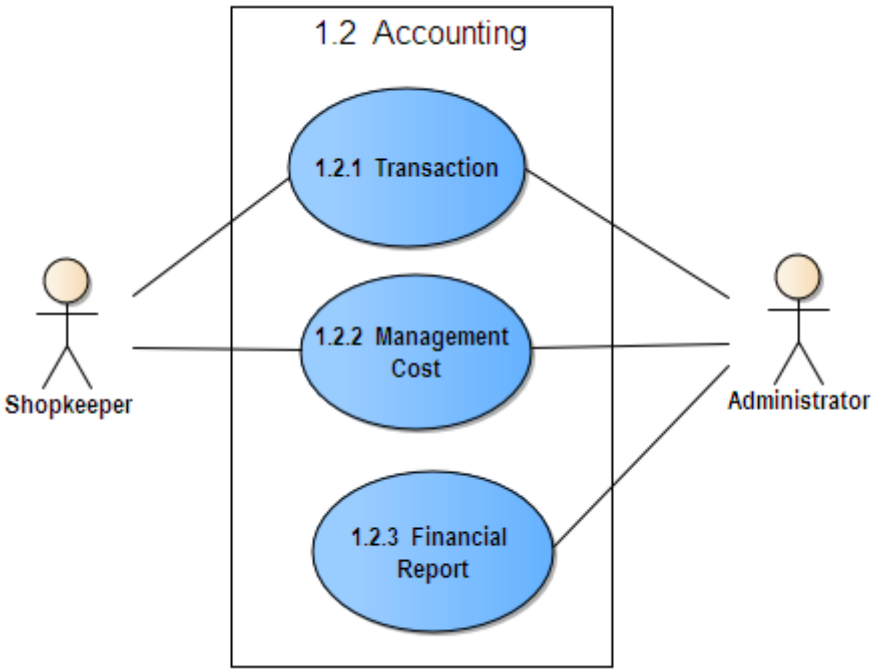


Figure 6: 1.2 Accounting

Description of level 1.2 use case diagram:
Anything related to spending and earing money will be handled by this subsystem. This subsystem is divided into three parts:

i. Transaction
ii. Management cost

iii. Financial report

Transaction consists of dealing, buying and selling things and exchanging. For more details on transaction see level 1.2.1.

“Management cost” list firstly stores the information about maintenance namely maintenance element and related cost. This list also includes shop rent, loan payment, employee salary, bonus for employees. These elements and the related cost are stored by the administrator. Only the administrator will have access into this table. He/she may add or delete any element from the list.

There is a file called “Financial report” that will contain total amount of money spent on buying products and management cost all together, total amount of money earned by selling products. Monthly record will be kept. Monthly and yearly profit will also be calculated and stored in this file. This is only accessed by the owner. He only can observe the records, but can’t edit it.

Action Reply:

Administrator:

- A1: Administrator enter transaction information into the system

  R1: System saves the given information.

- A2: Administrator will provide maintenance cost of every maintenance elements.

  R2: System saves the given information.

- A3: Administrator makes a request to see balance sheet

  R3: System shows the balance sheet.

User (Shopkeeper and administrator):

A1: User give total amount of money spent on buying products and management cost all together, total amount of money earned by selling products into the system.

R1: System stores all information and calculate monthly and yearly profit and store it in a file

4.3.7 LEVEL 1.2.1 USE CASE DIAGRAM-TRANSACTION

1.2.1 Transaction
Version 1.1
Created on 11/13/2017. Last modified 11/16/2017

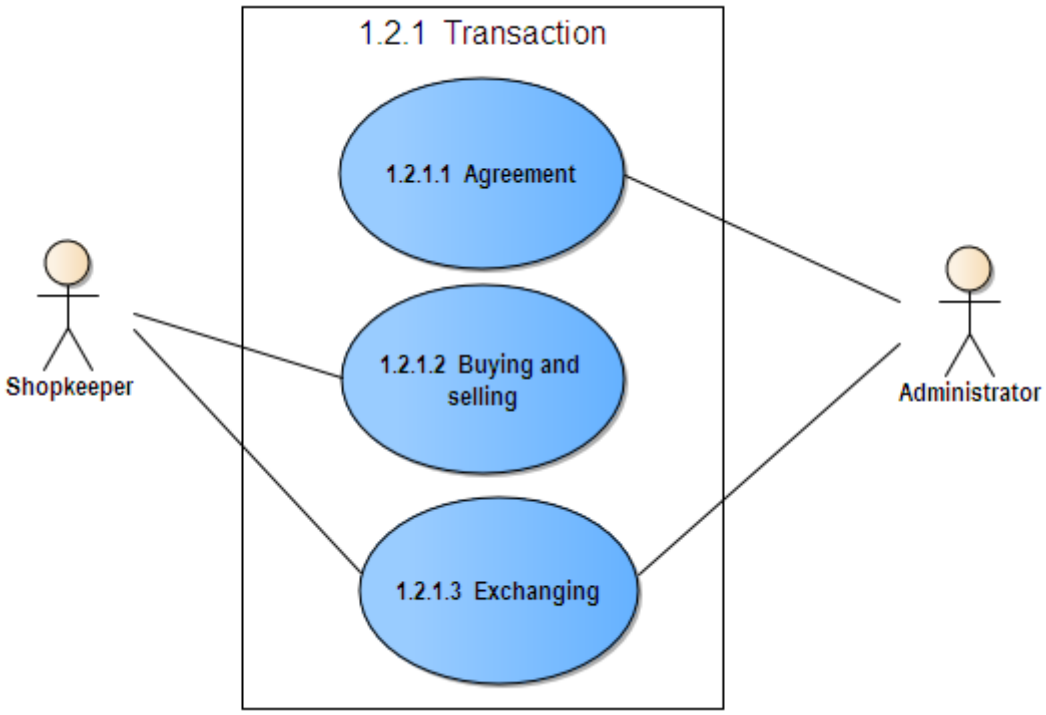


Figure 7: 1.2.1 Transaction

Description of level 1.2.1 use case diagram:
Transaction subsystem holds records of agreement, buying, selling and exchanging products.

Agreement can be established between the owner and suppliers, administrator and customers, shopkeeper and customers and sometimes between owner and shopkeeper for fixing his salary. For more details on dealing see level 1.2.1.1.

When the owner orders one or more products from suppliers he has to pay for those products and that information must be kept. For this purpose, the following information about the transaction will be stored:

- Transaction id
- Transaction date
- Transaction time
- Supplier id
- Product name
- Quantity
- Unit cost
- Total cost

Transaction id, date and time are auto-generated. Total cost is calculated by the system. All the information is stored by the shopkeeper in a temporary list. The list is then approved by the owner then all the information gets into a master list and the information in the temporary list get deleted.
Again, information about transaction with the customer is also stored by the shopkeeper at each transaction. The information that should be filled are:

- Transaction id
- Transaction date
- Transaction time
- Supplier id
- Number of products
- Product names
- Quantities
- Unit costs
- Total cost

If a customer wants to return any product, he will not be allowed to do it. Instead, he will be able to exchange purchased product with another product having the same price. This information is stored in a separate list by the shopkeeper containing name of the returned product and name of the exchanged product.

Action Reply

User (Administrator & shopkeeper)

- A1: User provides information about different expenditure.
  R1: System stores the information.

- A2: User stores transaction information about sold items.
  R2: System stores the information.

4.3.8 LEVEL-1.2.1.1 USE CASE DIAGRAM-AGREEMENT

1.2.1.1 Agreement
Version 1.1
Created on 11/13/2017. Last modified 11/16/2017

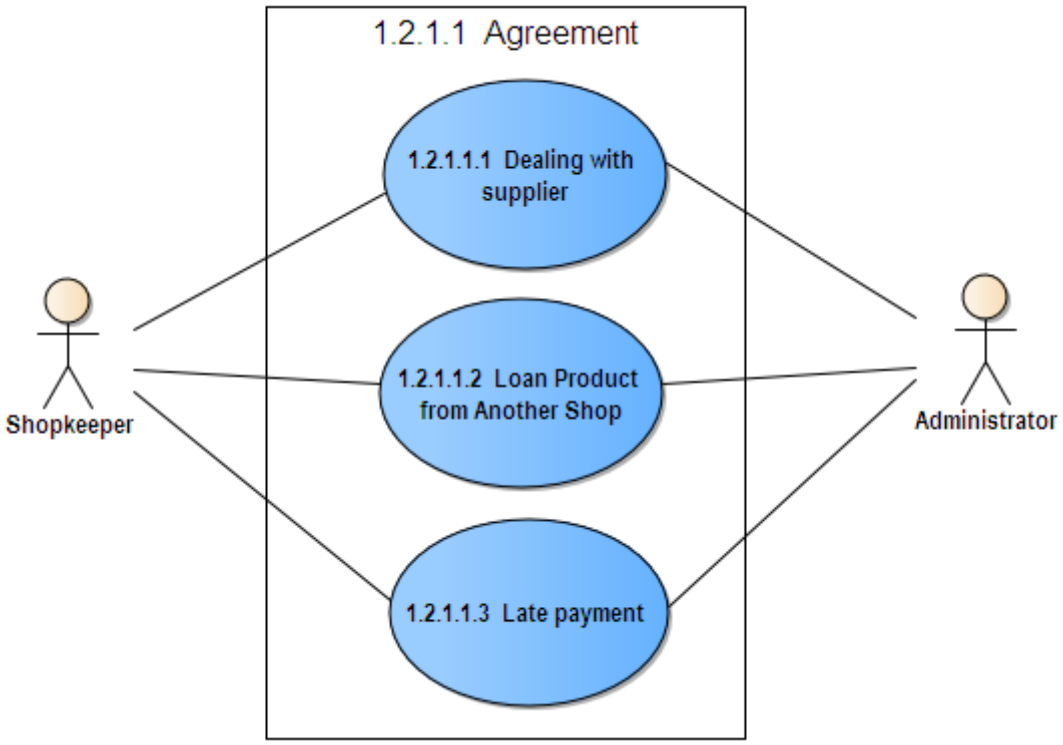


Figure 8: Level -1.2.1.1 Agreement

Description of level 1.2.1.1 use case diagram:

Agreement with suppliers means to buy products from different suppliers. The user deals with suppliers and stores all information about dealing items. Information are given below:

- Transaction id
- Transaction date
- Transaction time
- Supplier id
- Number of products
- Product names
- Quantities
- Unit costs
- Total cost

If desired product(s) are not available in the shop, the shopkeeper brings those product(s)

from any nearby shop or may sometimes borrow money from them. There is a loan table in which name and cost of borrowed product(s) are stored. In case of borrowing money, only the cost column is updated.
Customers may not pay cash instantly. Shopkeeper will store name, address, mobile number, amount and status (paid/not paid) that is to be paid of the customer into a table named “late payment”. When a customer returns money then the shopkeeper updates the status to be ‘paid’.

Action Reply

User

- A1: User stores information about dealing items with suppliers.
  R1: System stores the information.

- A2: User stores information about loan with other shop.
  R2: System stores the information.

- A3: Shopkeeper will give information about late payment of customer.
  R3: System stores the information.

4.3.9 LEVEL-1.3 SHOP MANAGEMENT

1.3 Shop Management
Version 1.0
Created on 11/13/2017. Last modified 11/16/2017

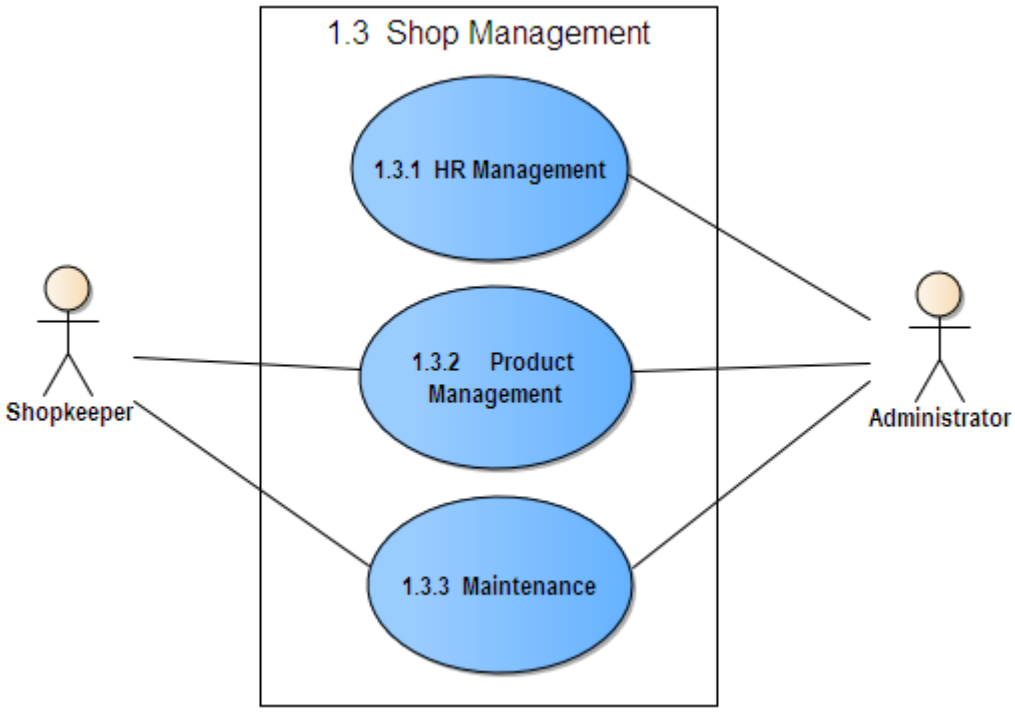


Figure 9: 1.3 Shop Management

Description of level 1.3 use case diagram:

Shop management subsystem manages various tasks of grocery management system. These tasks can be categorized as follows:

1. HR management
2. Product management
3. Maintenance

4.3.10 LEVEL-1.3.2 PRODUCT MANAGEMENT DIAGRAM

1.3.2 Product Management
Version 1.1
Created on 11/13/2017. Last modified 11/16/2017

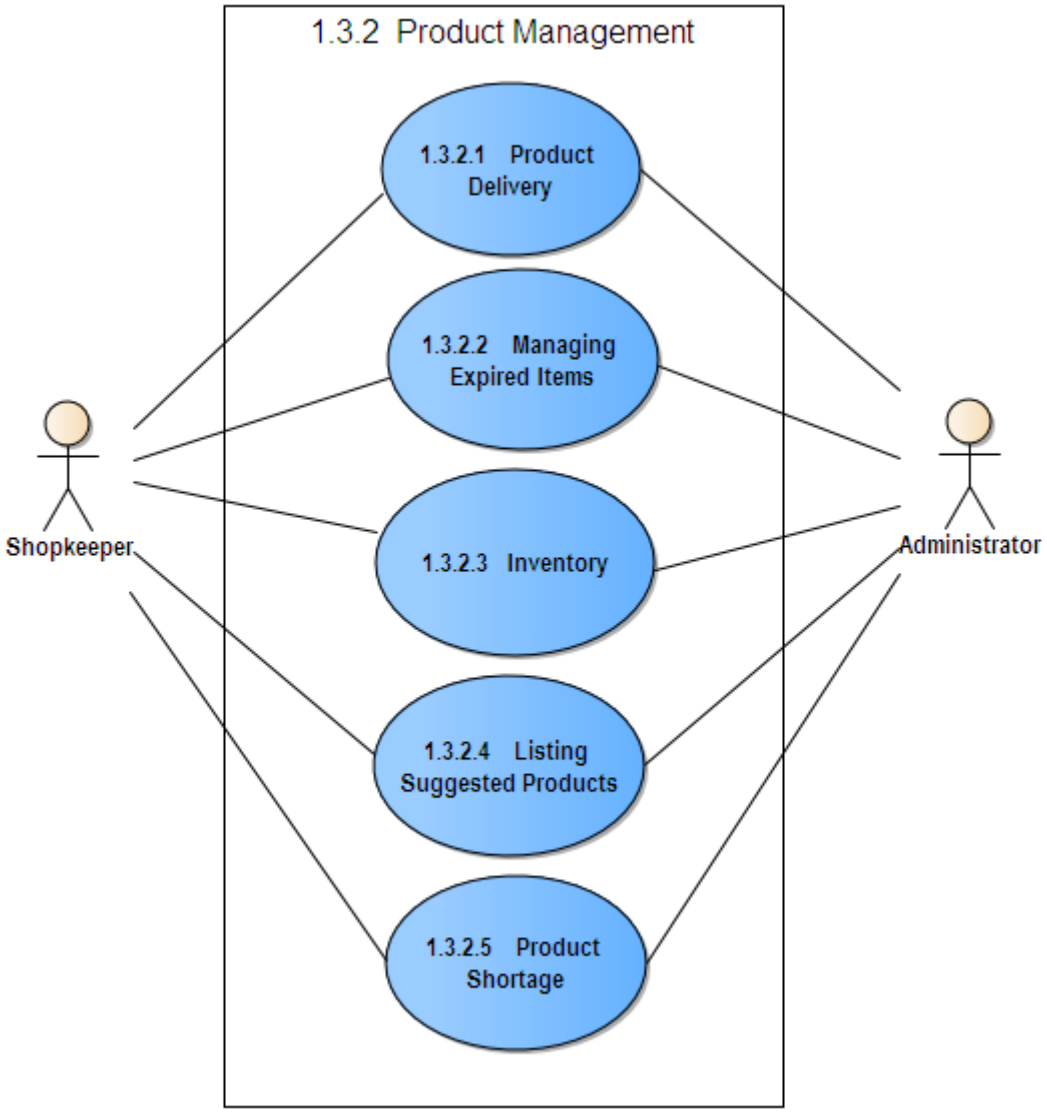


Figure 10: 1.3.2 Product Management

Description of level 1.3.2 use case diagram:

Product management subsystem plays a significant role in the grocery management system. This subsystem is related with managing all sorts of product item to be sale, tracking inventory level and stock of the product supply, overseeing expired items, ensuring product delivery in time and listing suggested products which are under consideration to be included.
This subsystem is divided into several subsystem which are as follows:

- Product delivery

- Managing Expired items

- Inventory
- Listing suggested products
- Product shortage

Through the “Product Delivery” subsystem shortage of product is managed to recover in due time to cope with customer need. Supplier and shopkeeper are engaged with this subsystem. While “Managing Expired Item” is responsible for the management of products having expired. Shopkeeper gets notification and take steps to resolve the situation. Sometimes customer or other external person may suggest products that are on demand. “Listing suggested Product” provides with the scope of keeping a list of this kind of merchandises.

Action Reply:

User:

- A1: User inputs product delivery information.

  R1: System prompts with messages and saves information at the various stages of the execution of product delivery subsystem.

- A2: User manages expired item.

  R2: System acts accordingly.

- A3: User updates a list of suggested products.

  R3: System saves the list.

4.3.11 LEVEL-1.3.2.3 INVENTORY

1.3.2.3 Inventory
Version 1.1
Created on 11/13/2017. Last modified 11/16/2017

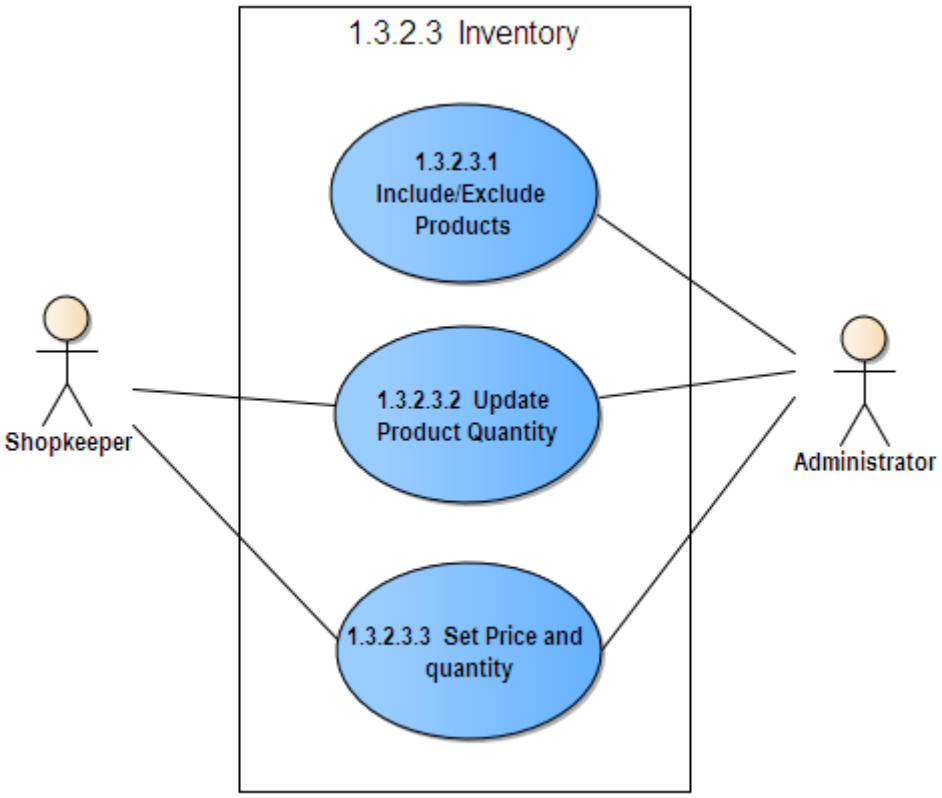


Figure 11: Level-1.3.2.3 Inventory

Description of level 1.3.2.3 use case diagram:

Inventory subsystem is responsible for tracking inventory levels, sales and deliveries as well as controlling and supervising the ordering and flow of products. This subsystem consists of three basic part. They are:

1. Include/Exclude product
2. Update product quantity
3. Set price and quality

Owner can include new product item on the basis of current market demand and also exclude existing product item. When a particular product item falls short shopkeeper can update product quantity after product delivery and receiving new products increasing inventory level adequately. The task of setting the price of each product on the market value is performed by the owner that executed in this subsystem.

Action Reply:
User

- A1: User includes or excludes product from product list
  R1: System receives command and execute accordingly.

- A2: User updates price and other information about a product.

R2: System stores the information.

4.4 ACTIVITY DIAGRAM OF GMS

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency.

4.4.1 GROCERY MANAGEMENT SYSTEM

Grocery Management System
Version 1.0
Created on 11/16/2017. Last modified 11/16/2017

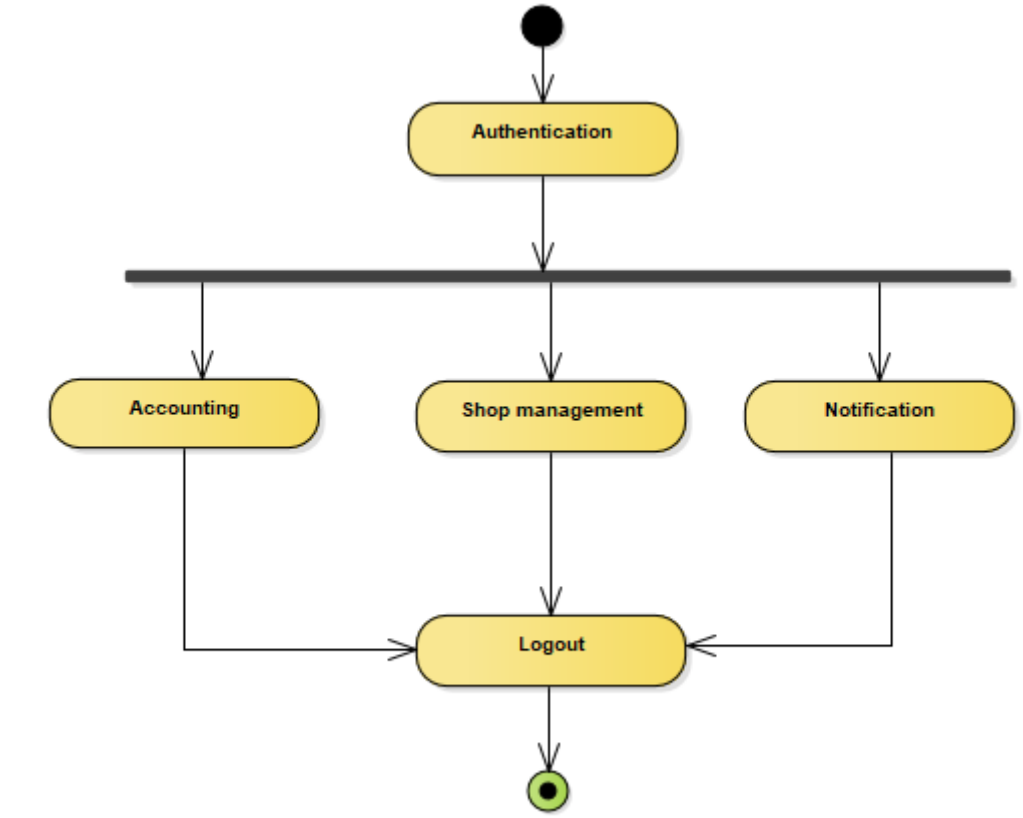


Figure 12: Main

4.4.2 SIGN UP

Sign Up
Version 1.2
Created on 10/13/2017. Last modified 11/16/2017

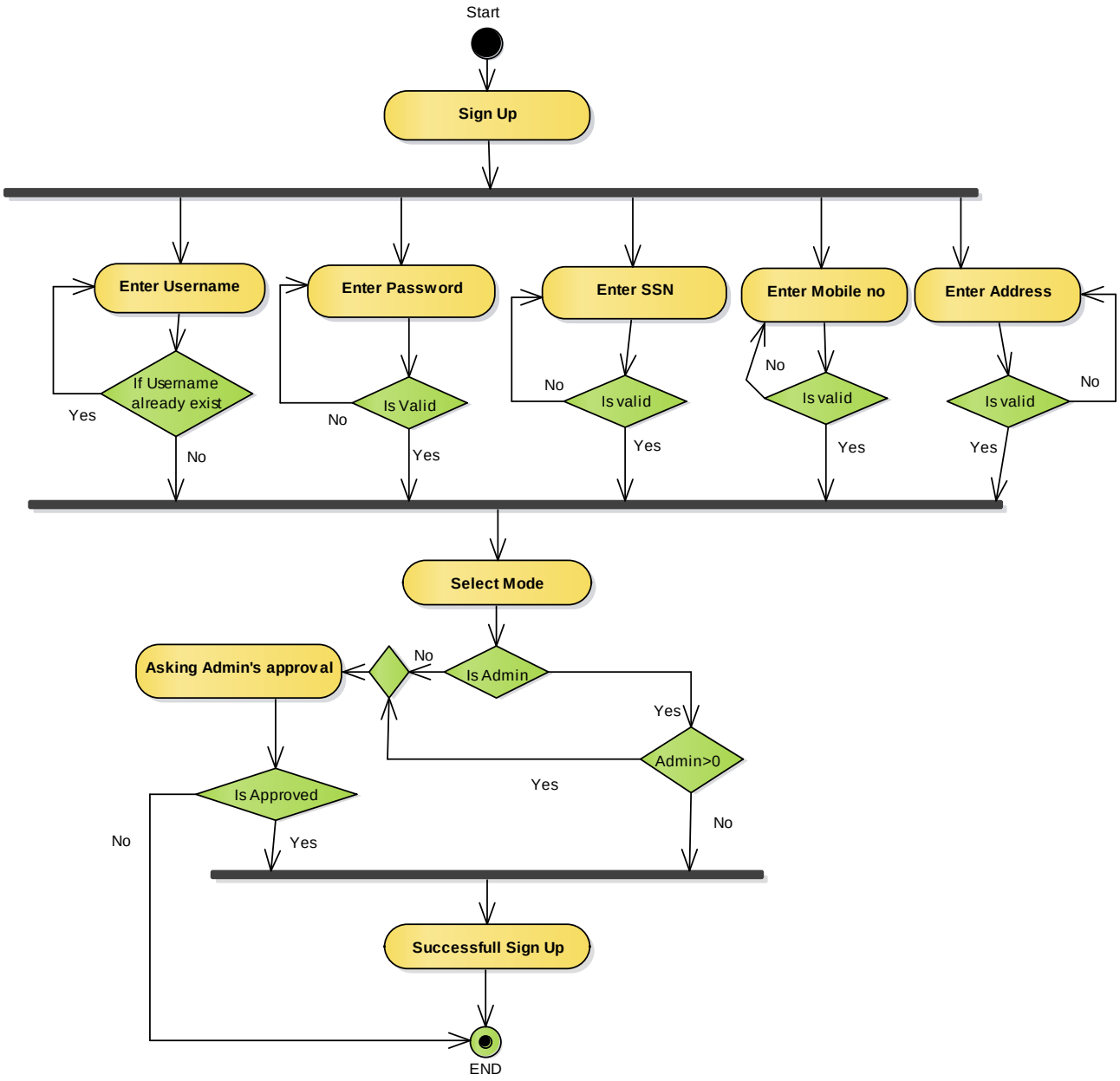


Figure 13: Sign Up

4.4.3 LOGIN ACTIVITY DIAGRAM

Login Activity
Version 1.2
Created on 10/12/2017. Last modified 10/28/2017

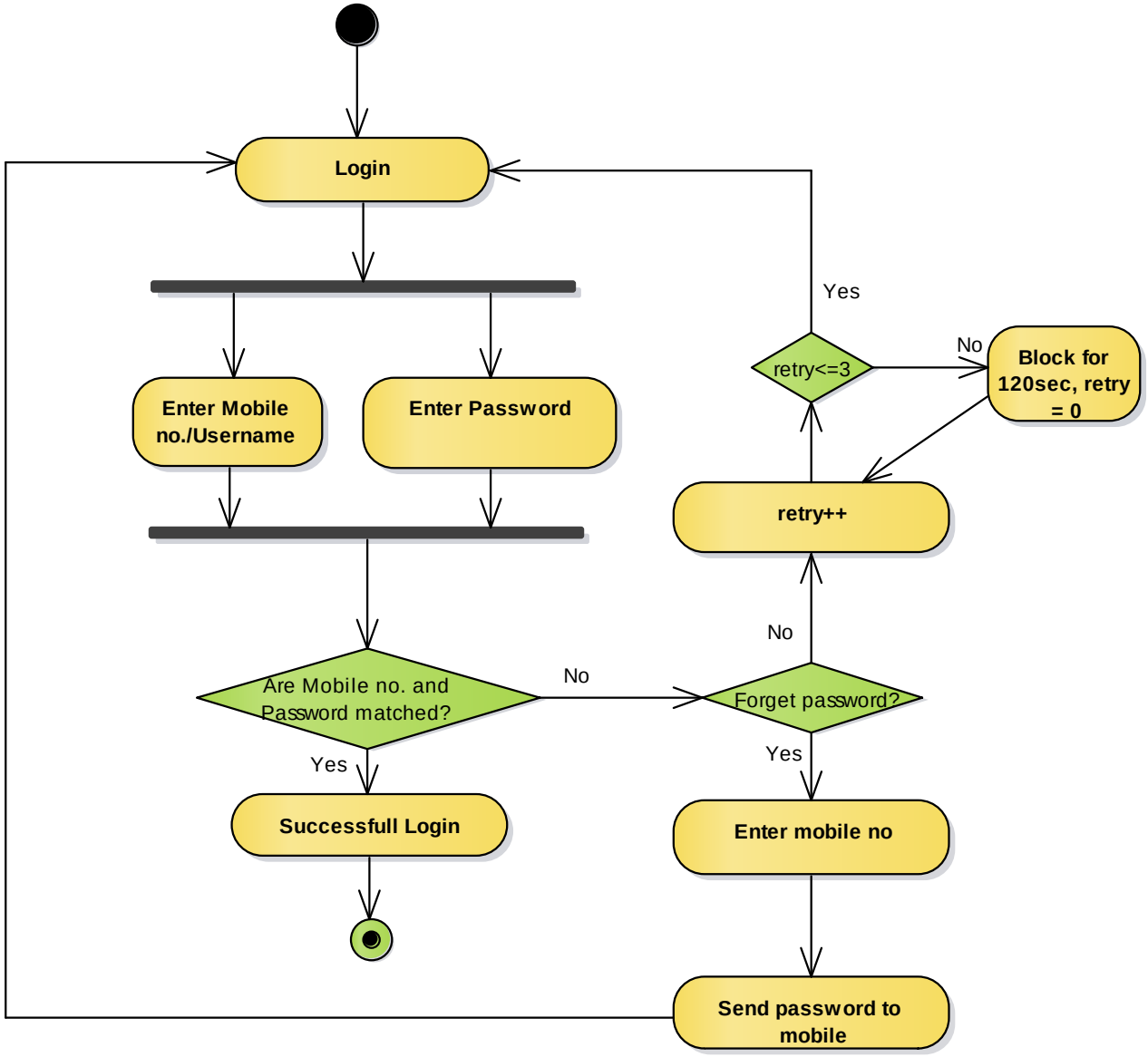


Figure 14: Login Activity

4.4.4 ACCOUNTING ACTIVITY DIAGRAM

Accounting activity
Version 1.0
Created on 10/12/2017. Last modified 10/28/2017

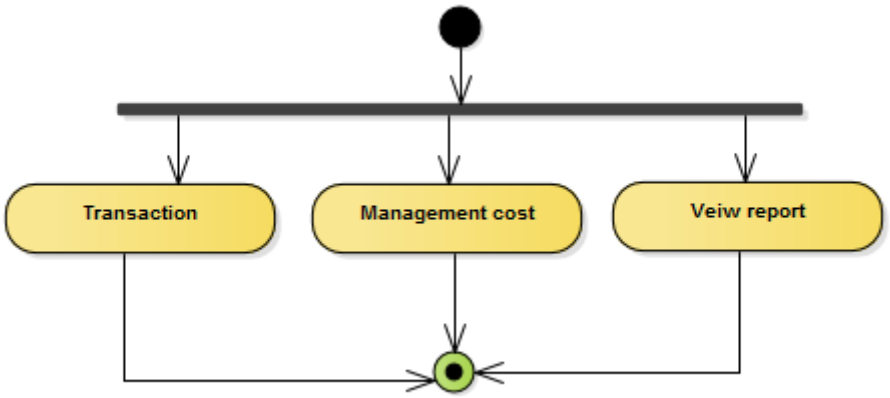


Figure 15: Login Activity

4.4.5 TRANSACTION ACTIVITY DIAGRAM

Transaction Activity
Version 1.0
Created on 10/13/2017. Last modified 11/16/2017

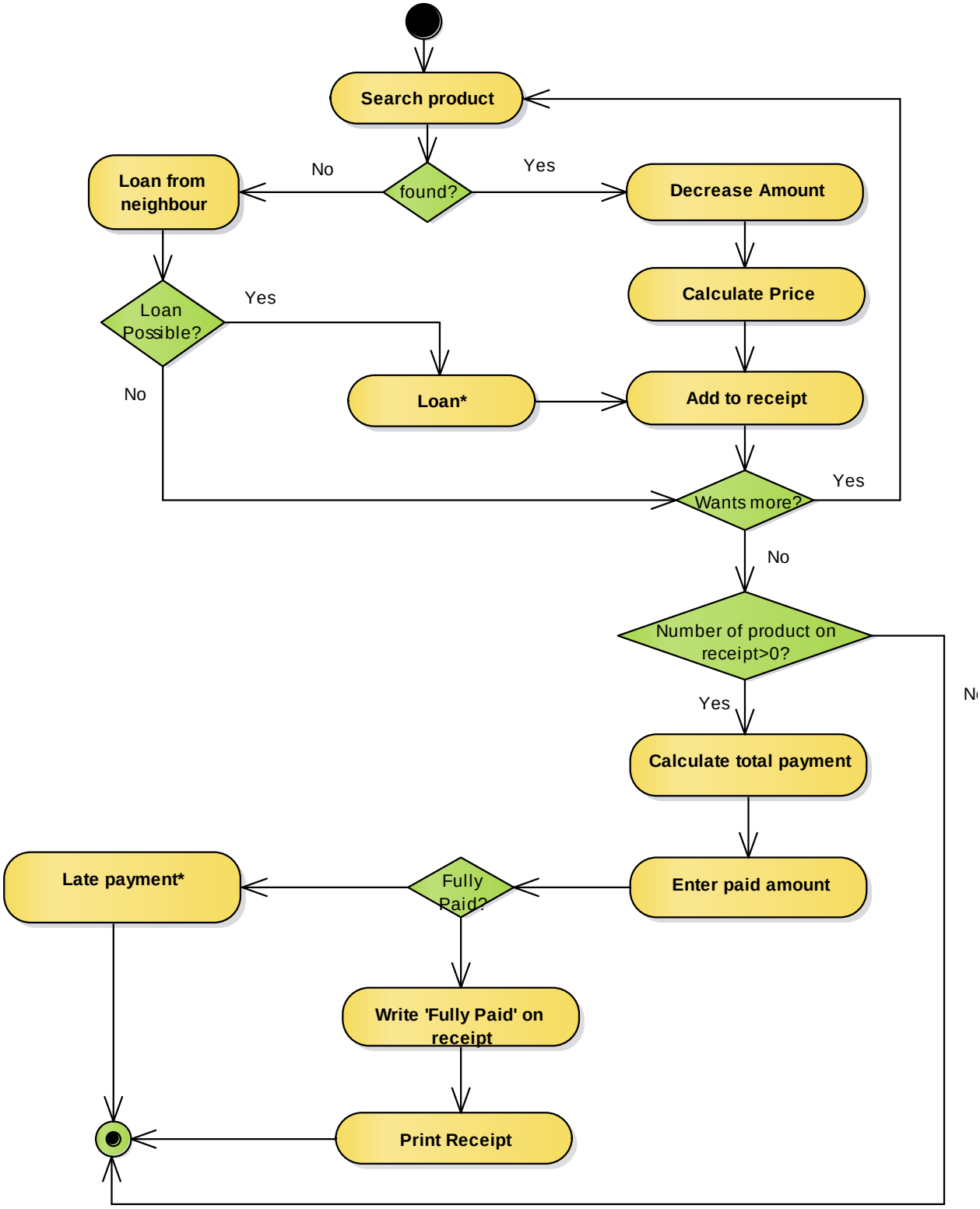


Figure 16: Transaction Activity

4.4.6 LATE PAYMENTS DIAGRAM

Late Payments
Version 1.0
Created on 10/13/2017. Last modified 10/27/2017

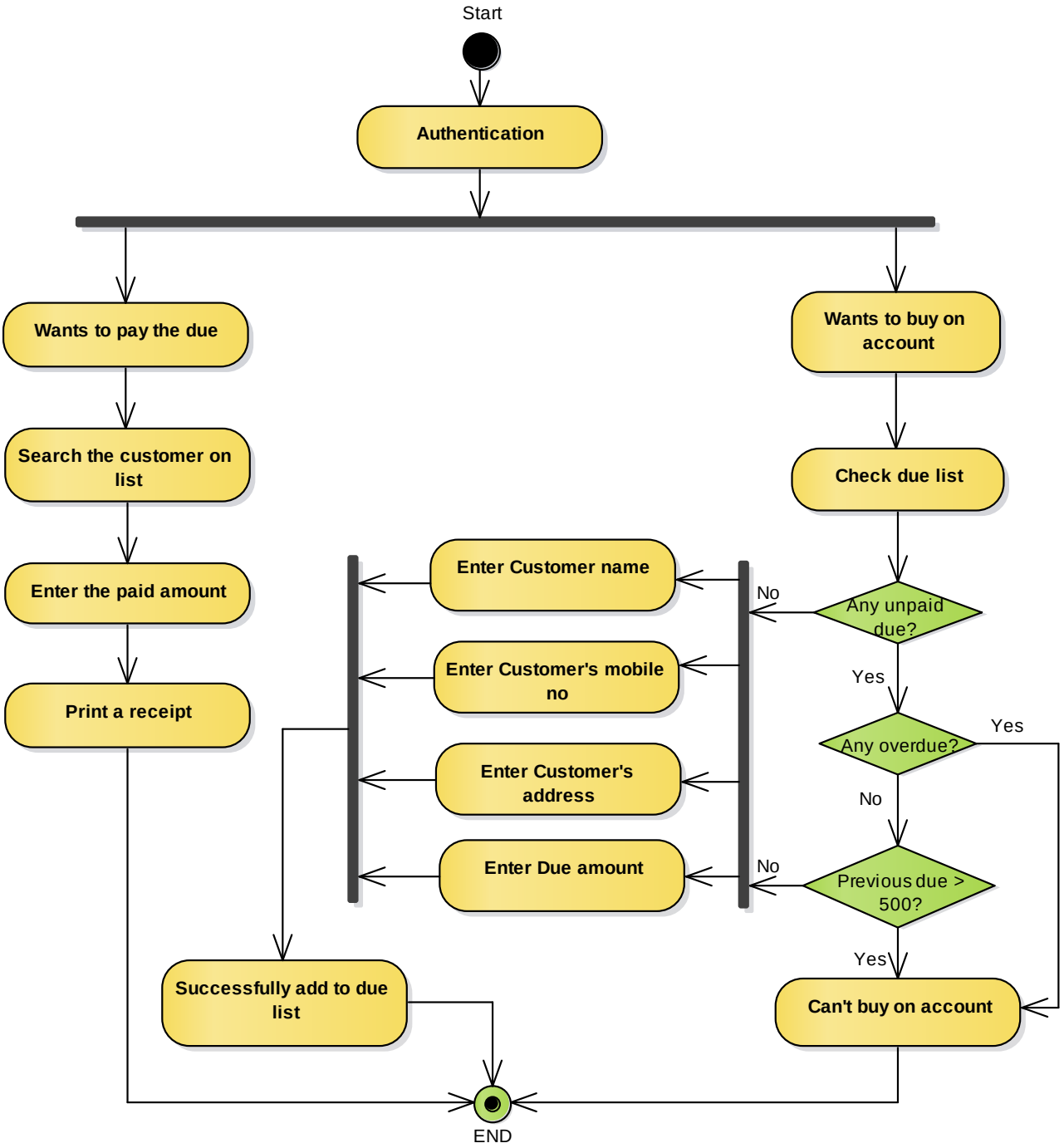


Figure 17: Late Payments

4.4.7 DEALING WITH CLIENT ACTIVITY DIAGRAM

Dealing with supplier
Version 1.0
ABDULLAH AL JUBAER created on 10/13/2017. Last modified 11/16/2017

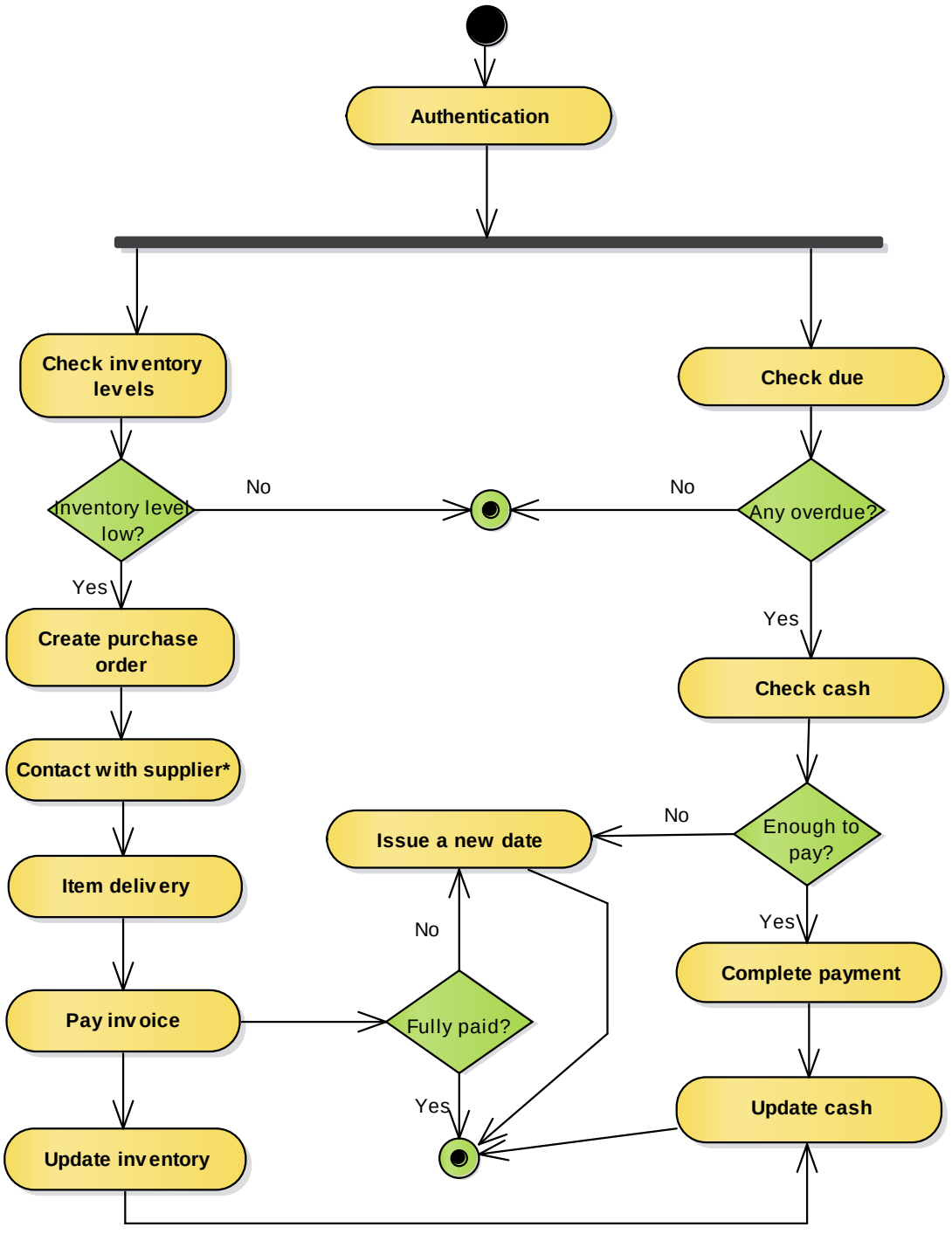


Figure 18: Dealing with client

4.4.8 MANAGEMENT COST DIAGRAM

Management cost
Version 1.1
Created on 10/27/2017. Last modified 10/27/2017

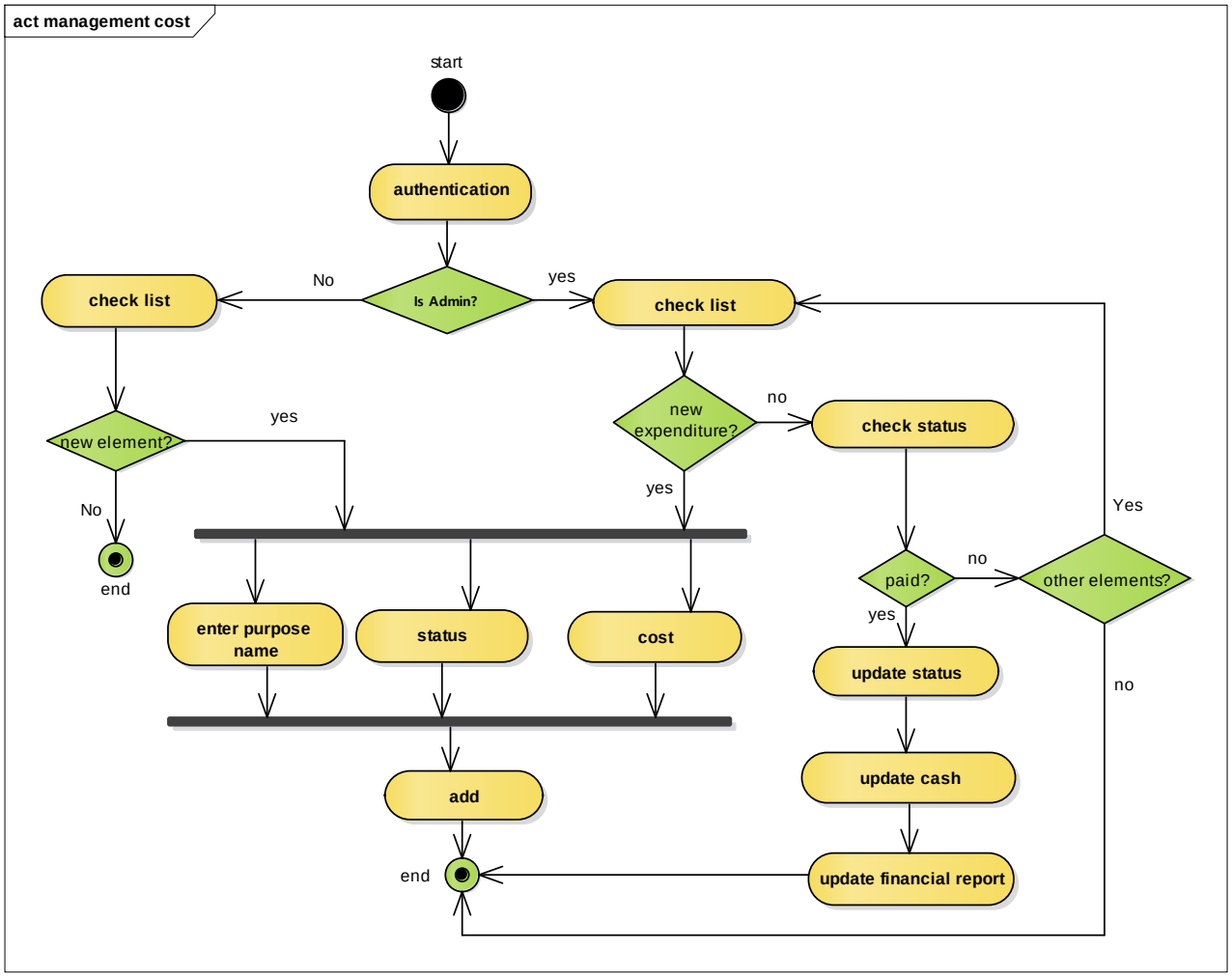


Figure 19: Management Cost

4.4.9 HR MANAGEMENT DIAGRAM

HR Management
Version 1.0
Created on 10/15/2017. Last modified 11/16/2017

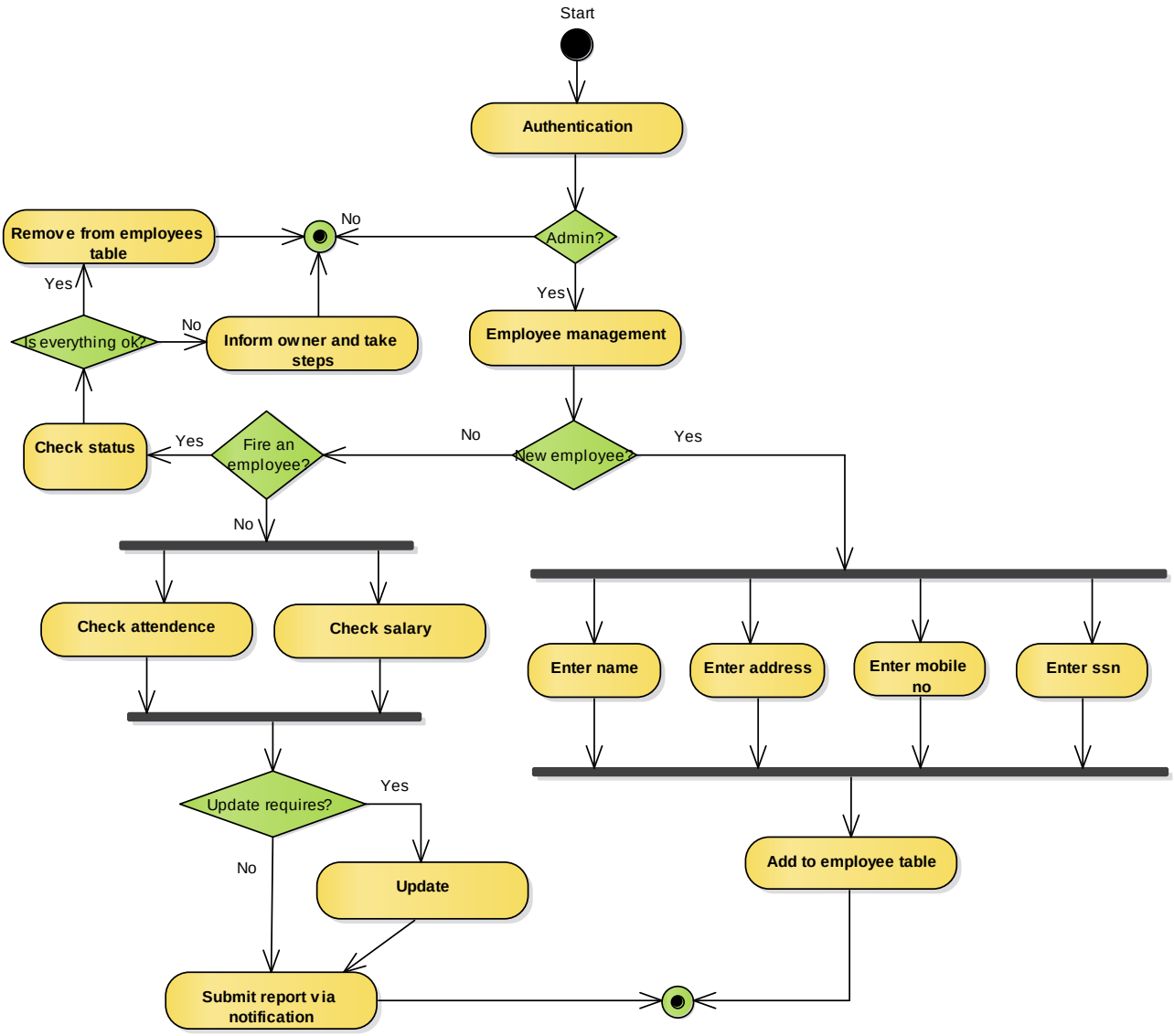


Figure 20: HR Management

4.4.10 PRODUCT MANAGEMENT DIAGRAM

HR Management
Version 1.2
Created on 10/15/2017. Last modified 11/16/2017

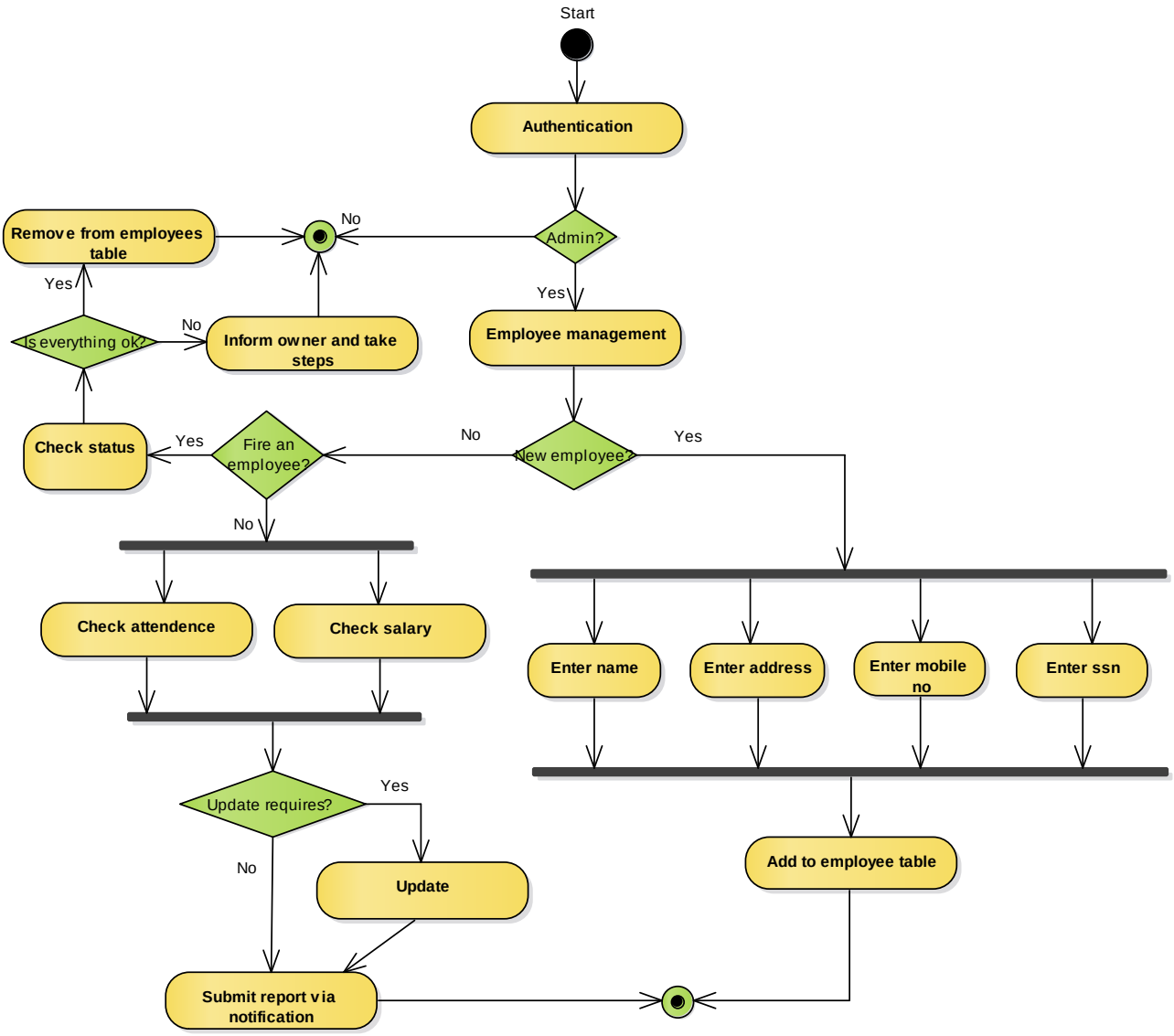


Figure 21: HR Management

4.4.11 INVENTORY ACTIVITY DIAGRAM

Inventory Activity
Version 1.1
Created on 10/15/2017. Last modified 10/15/2017

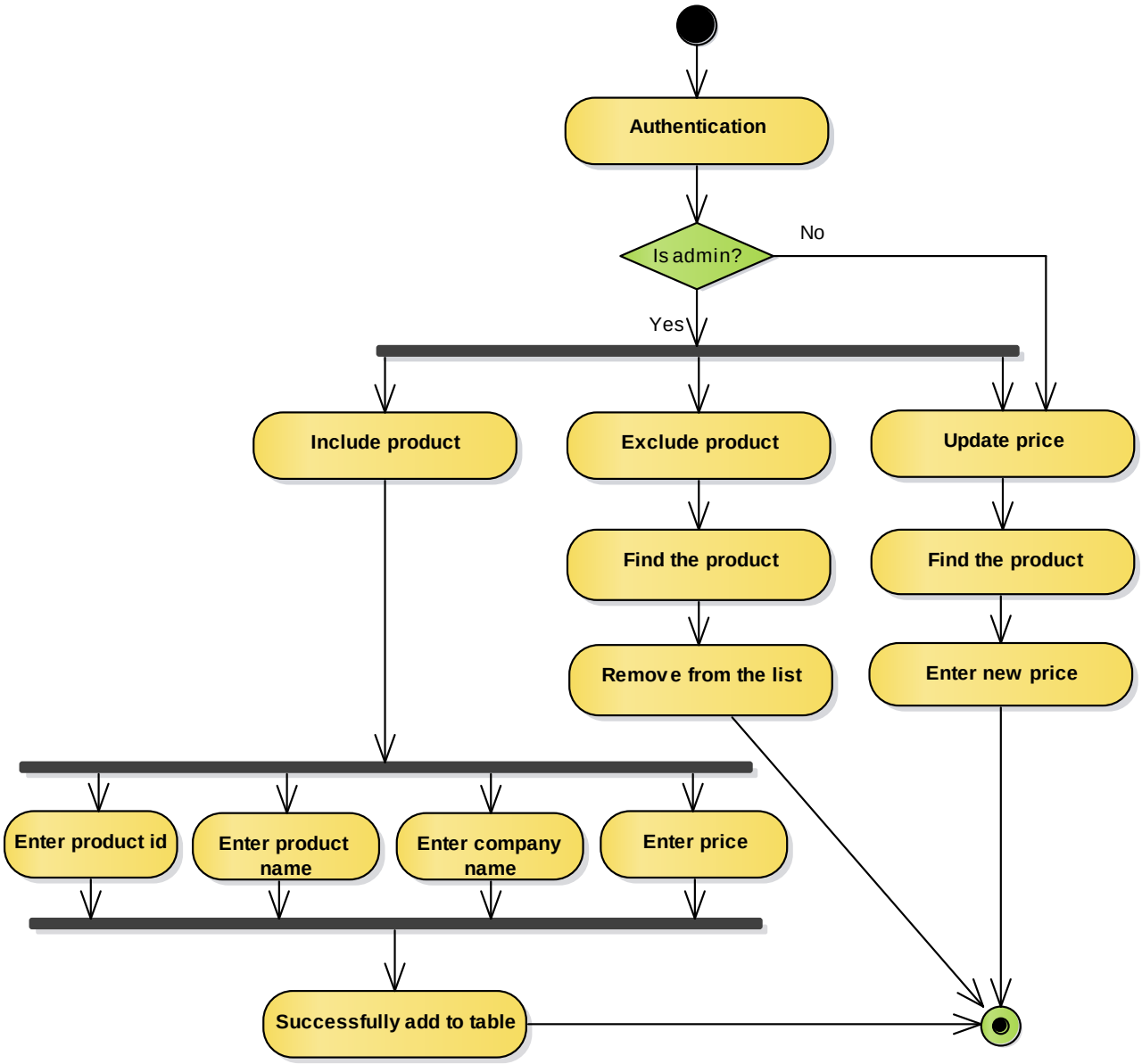


Figure 22: Inventory

4.4.12 NOTIFICATION ACTIVITY DIAGRAM

Notification Diagram
Version 1.1
Created on 10/15/2017. Last modified 11/16/2017

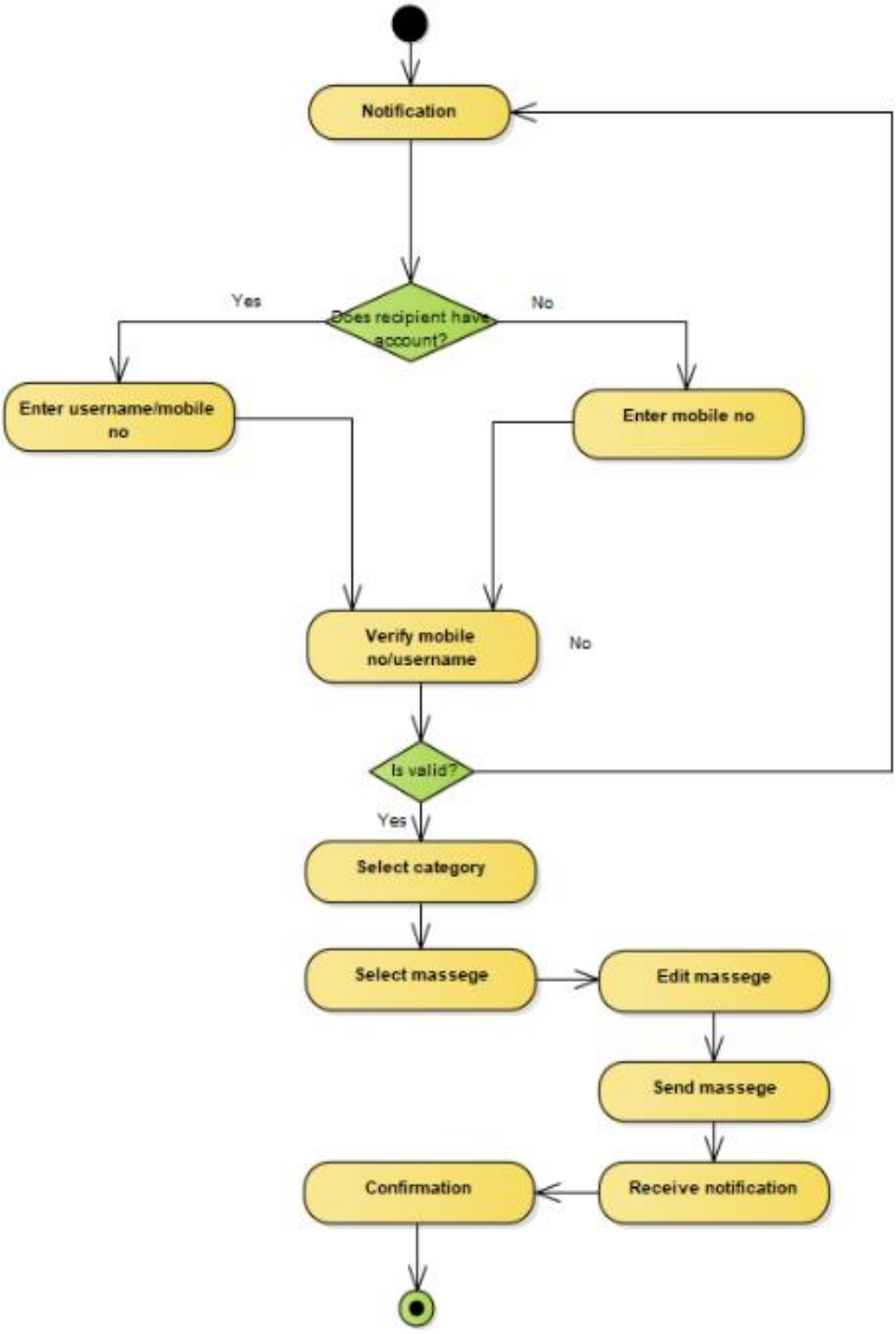


Figure 23: Notification

4.5 SWIMLANE DIAGRAMS OF GMS

4.5.1 SIGN UP SWIMLANE

signup swimlane
Version 1.1
Created on 25-10-17. Last modified 15-11-17

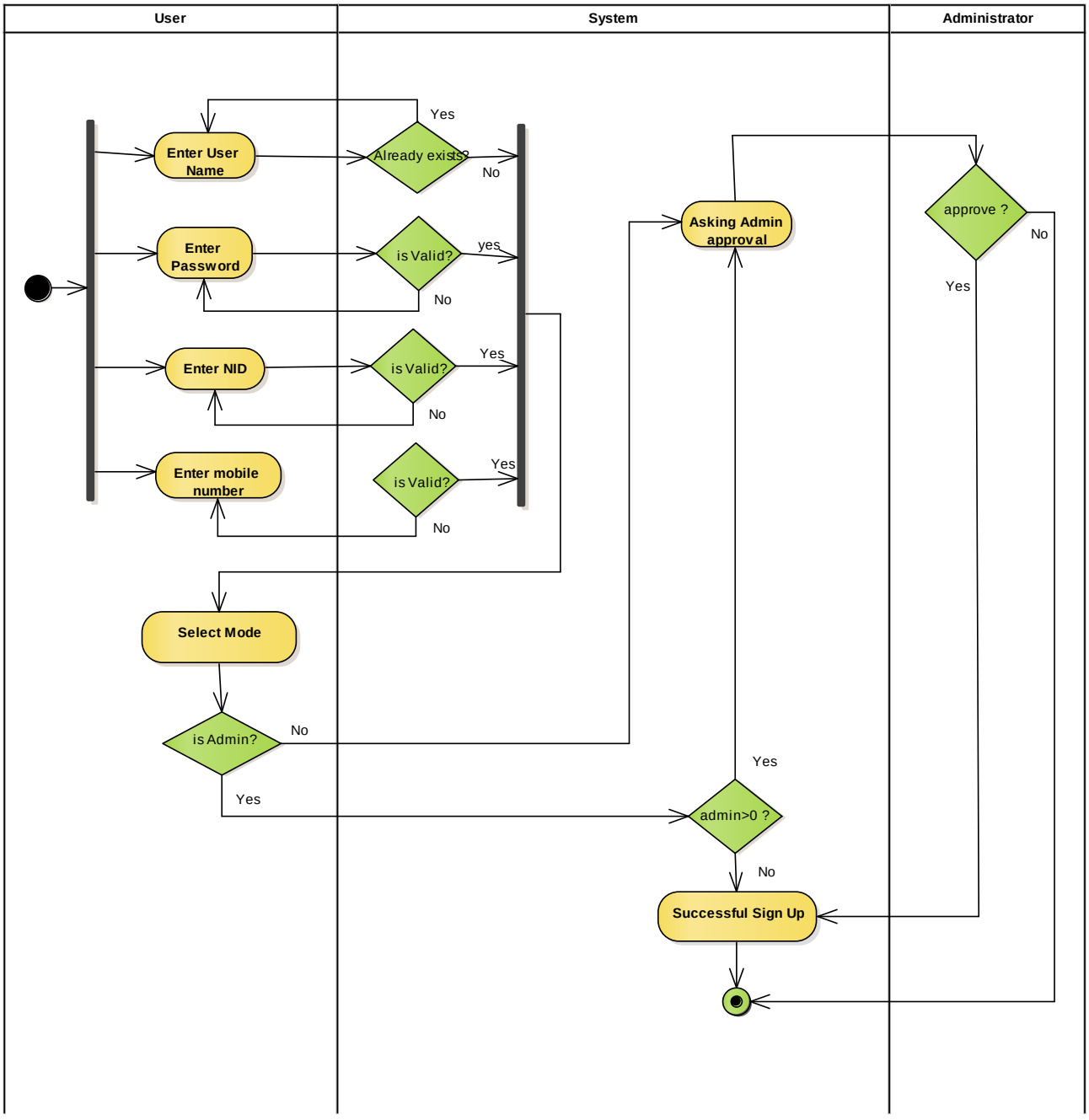


Figure 24: signup

4.5.2 LOGIN SWIMLANE

login swimlane
Version 1.1
Created on 17-10-17. Last modified 15-11-17

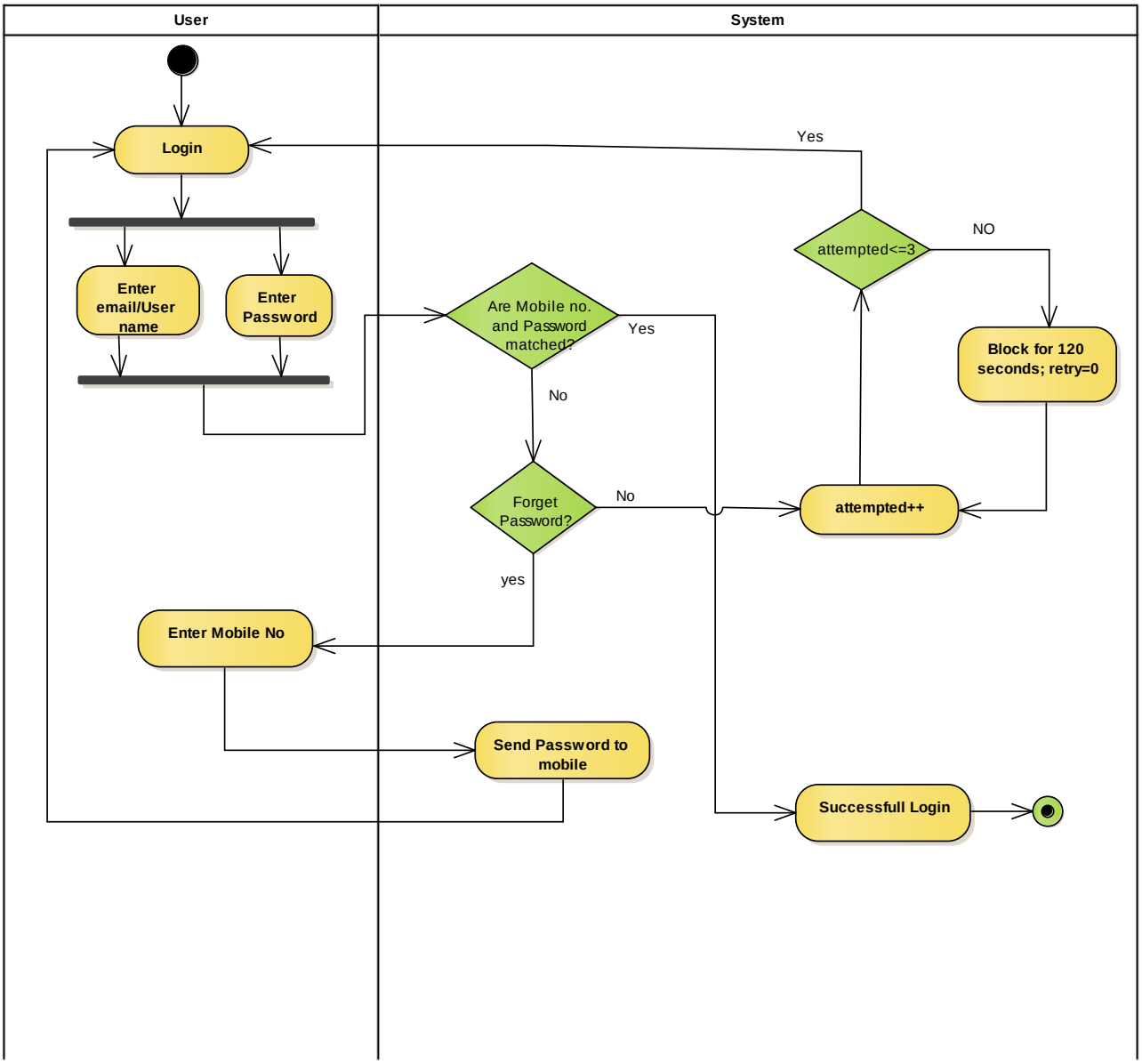


Figure 25: login

4.5.3 TRANSACTION SWIMLANE DIAGRAM

Transaction Swimlane
Version 1.1
Created on 25-10-17. Last modified 16-11-17

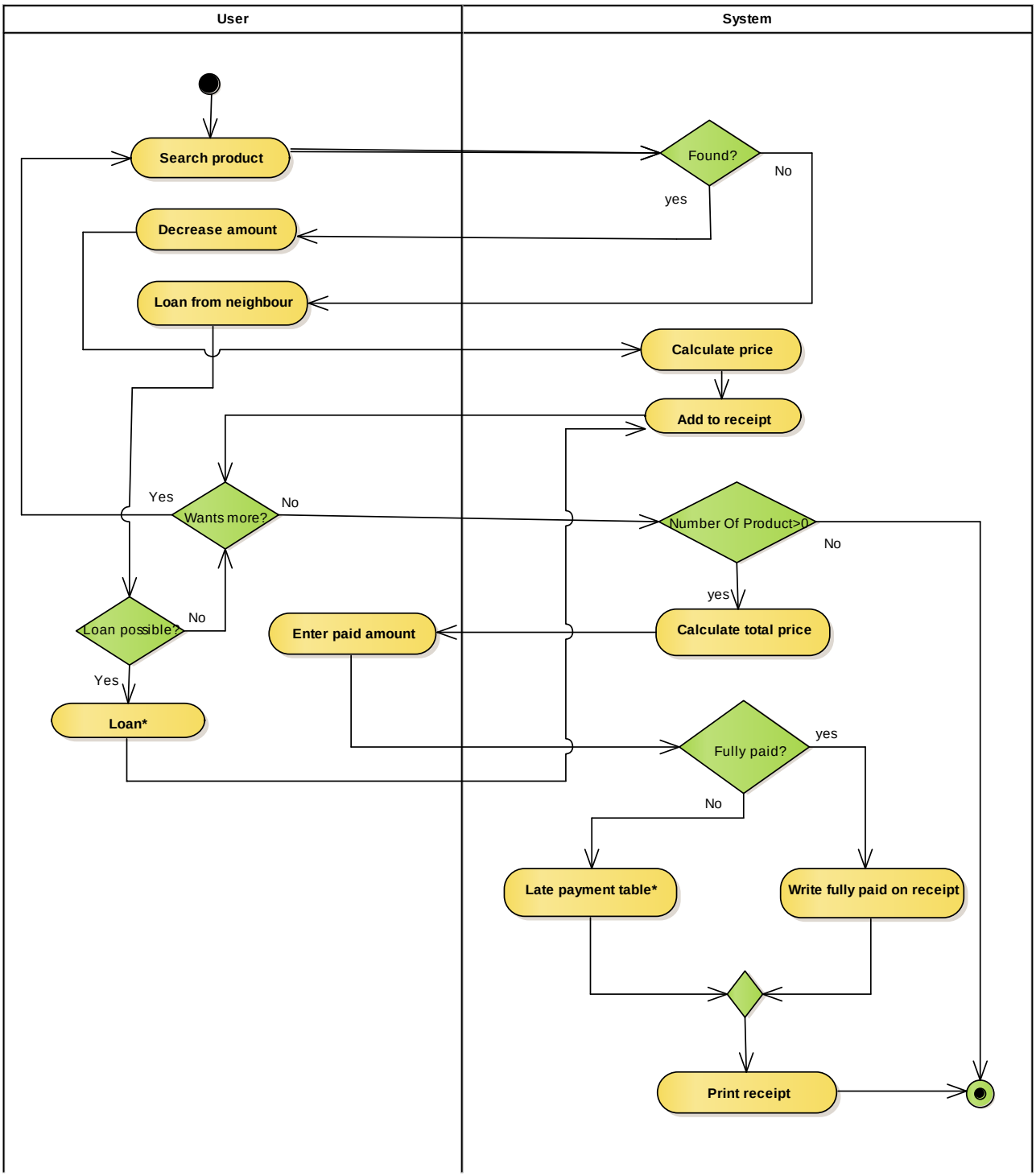


Figure 26: Transaction

4.5.4 LATE PAYMENT SWIMLANE

Late payment swimlane
Version 1.1
Created on 26-10-17. Last modified 16-11-17

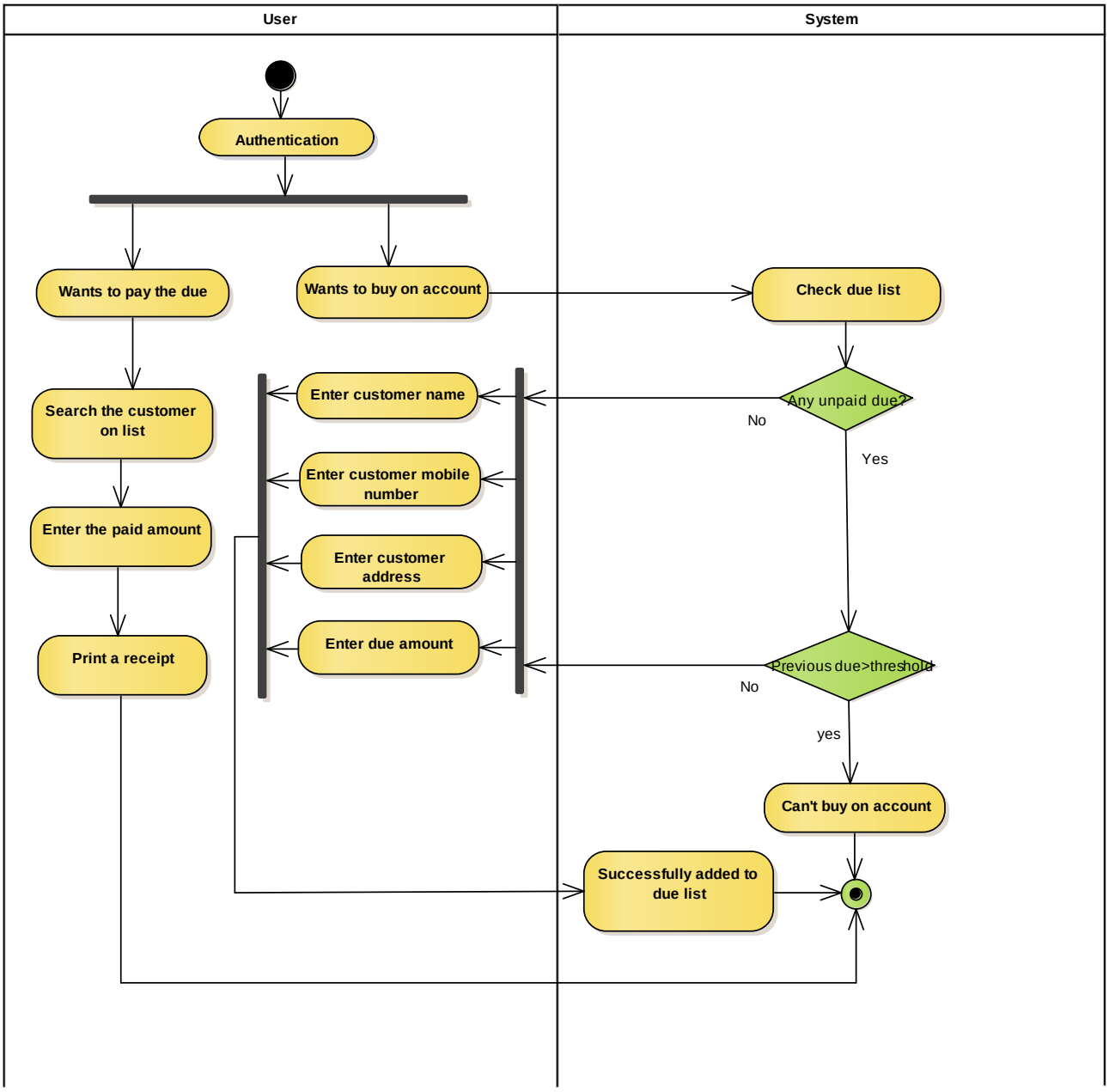


Figure 27: Late payment swimlane

4.5.5 AGREEMENT WITH CLIENT SWIMLANE

Agreement with client swimlane
Version 1.1
Created on 26-10-17. Last modified 16-11-17

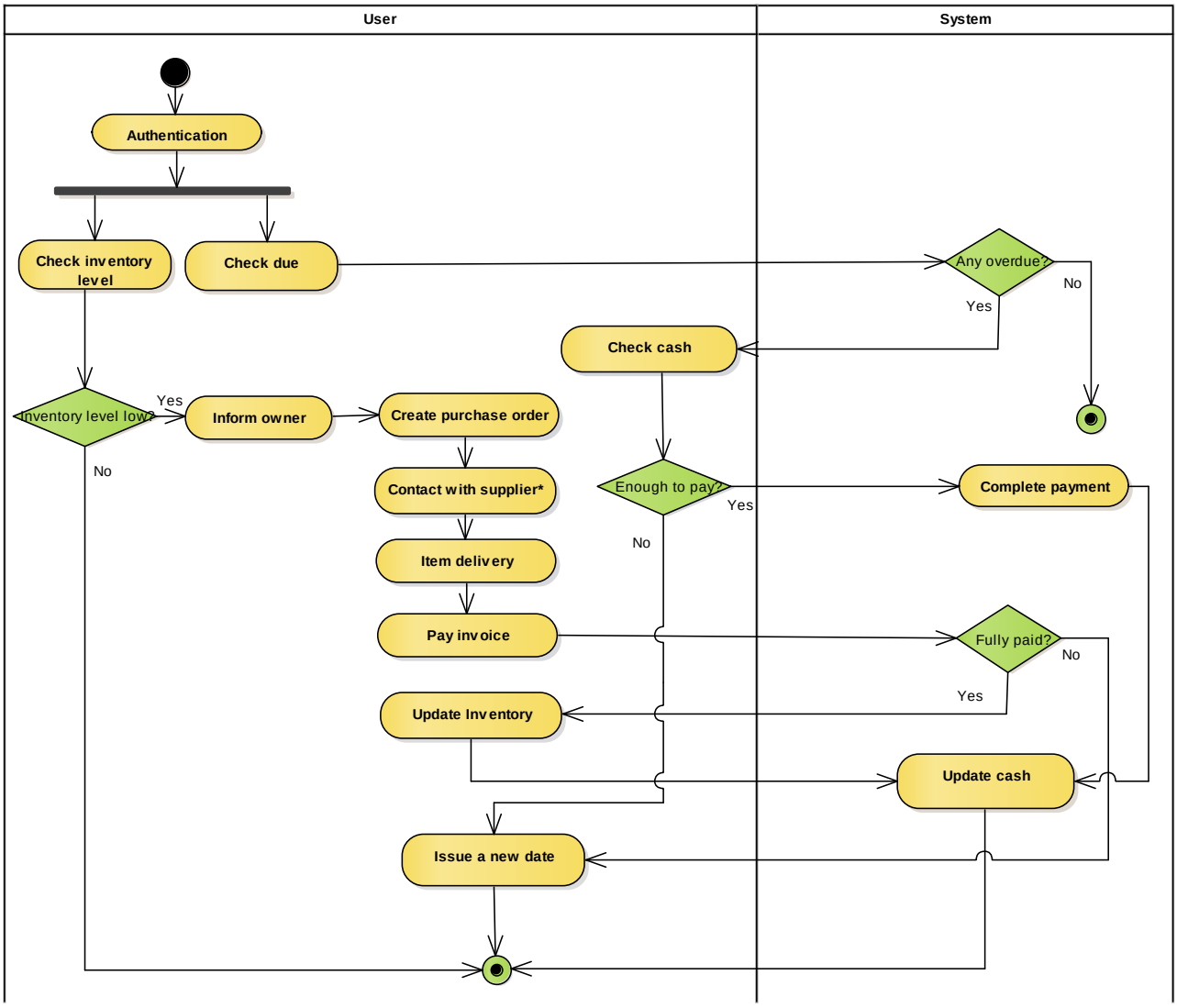


Figure 28: Agreement with client swimlane

4.5.6 MANAGEMENT COST SWIMLANE

Management cost swimlane
Version 1.1
Created on 09-11-17. Last modified 16-11-17

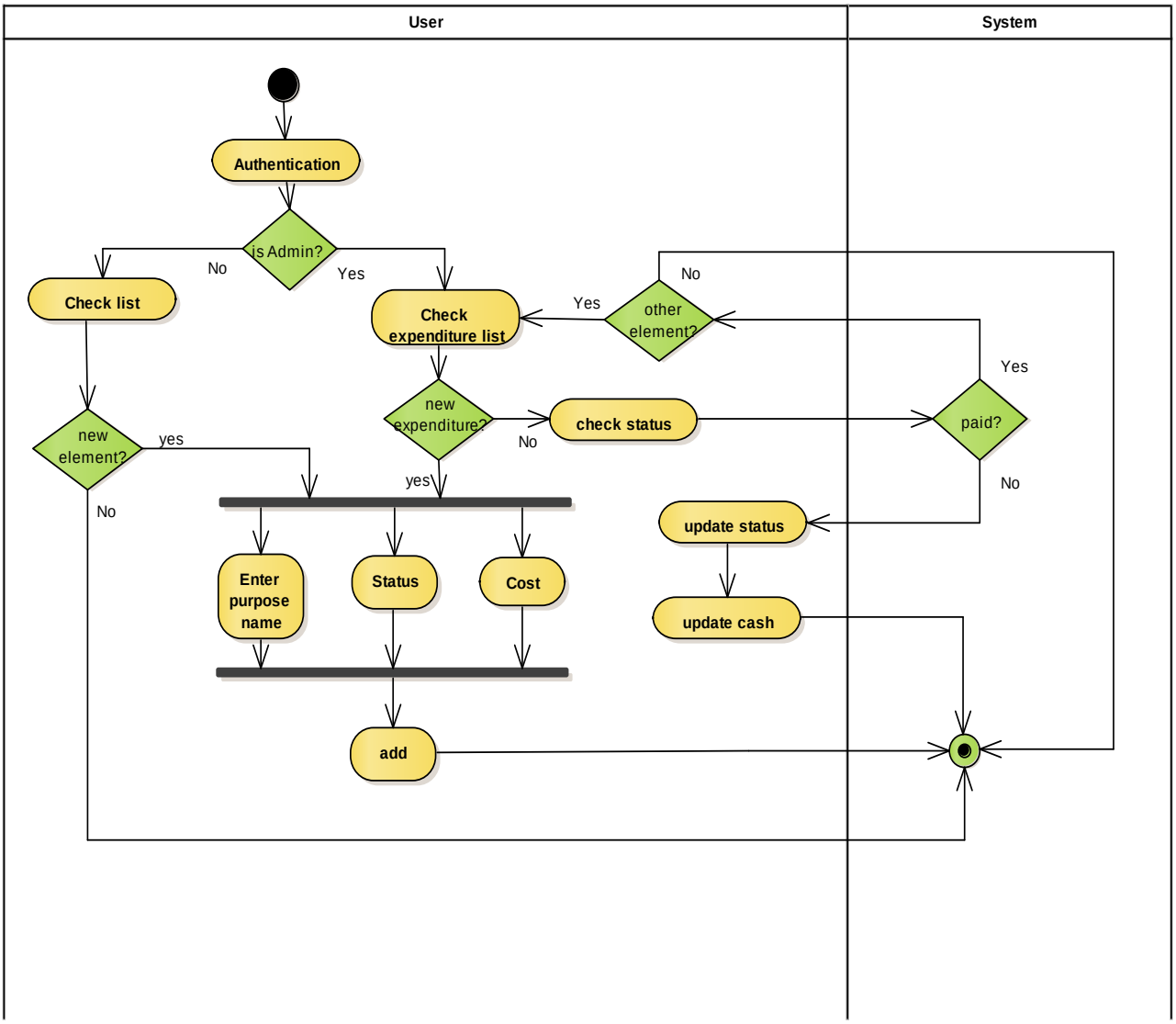


Figure 29: management

4.5.7 HR MANAGEMENT SWIMLANE

HR management
Version 1.1
Created on 27-10-17. Last modified 17-11-17

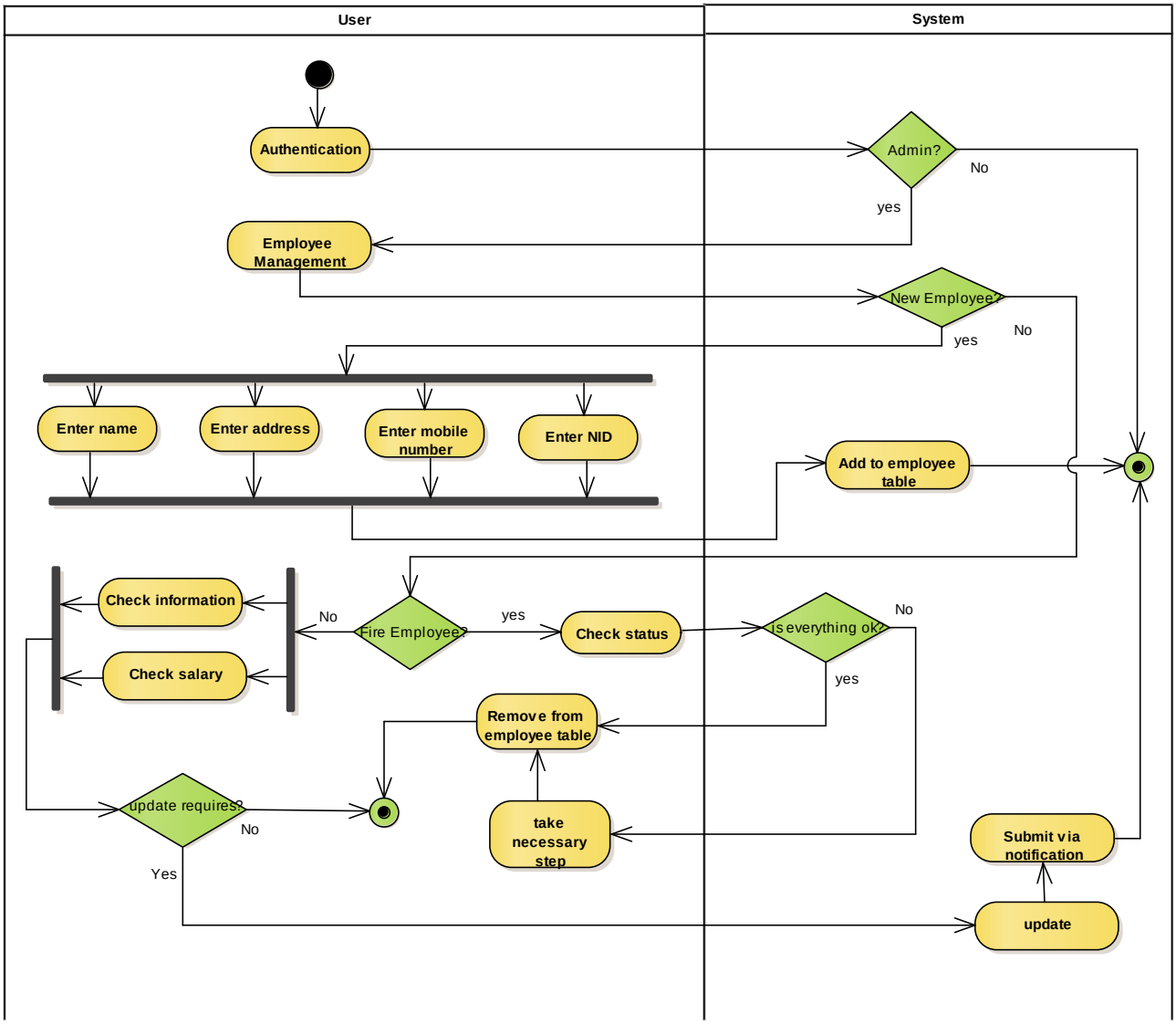


Figure 30: HR management

4.5.8 PRODUCT MANAGEMENT SWIMLANE

Product Management swimlane
Version 1.1
Created on 26-10-17. Last modified 16-11-17

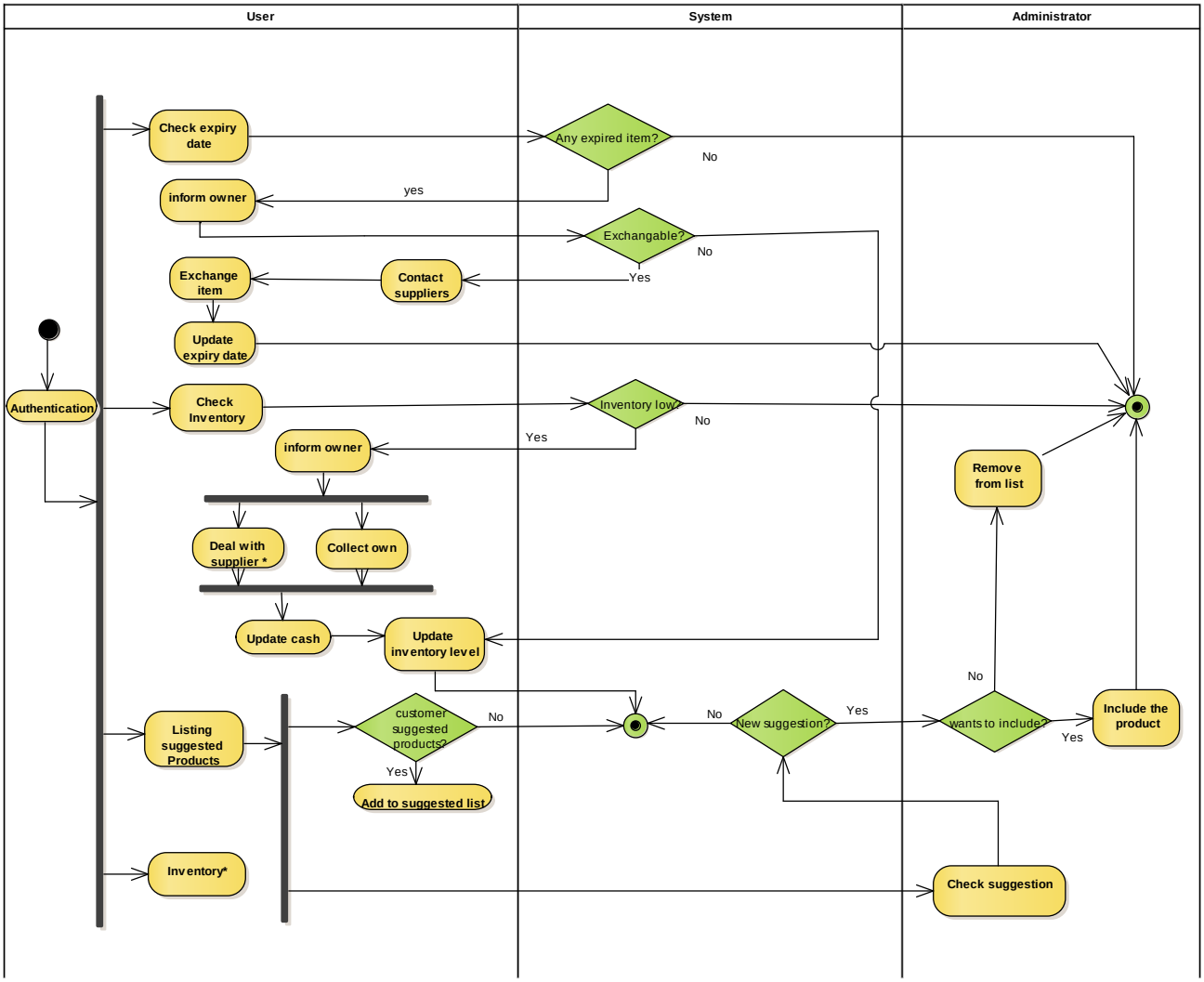


Figure 31: Product Management

4.5.9 INVENTORY SWIMLANE

Inventory Swimlane
Version 1.0
Created on 04-11-17. Last modified 17-11-17

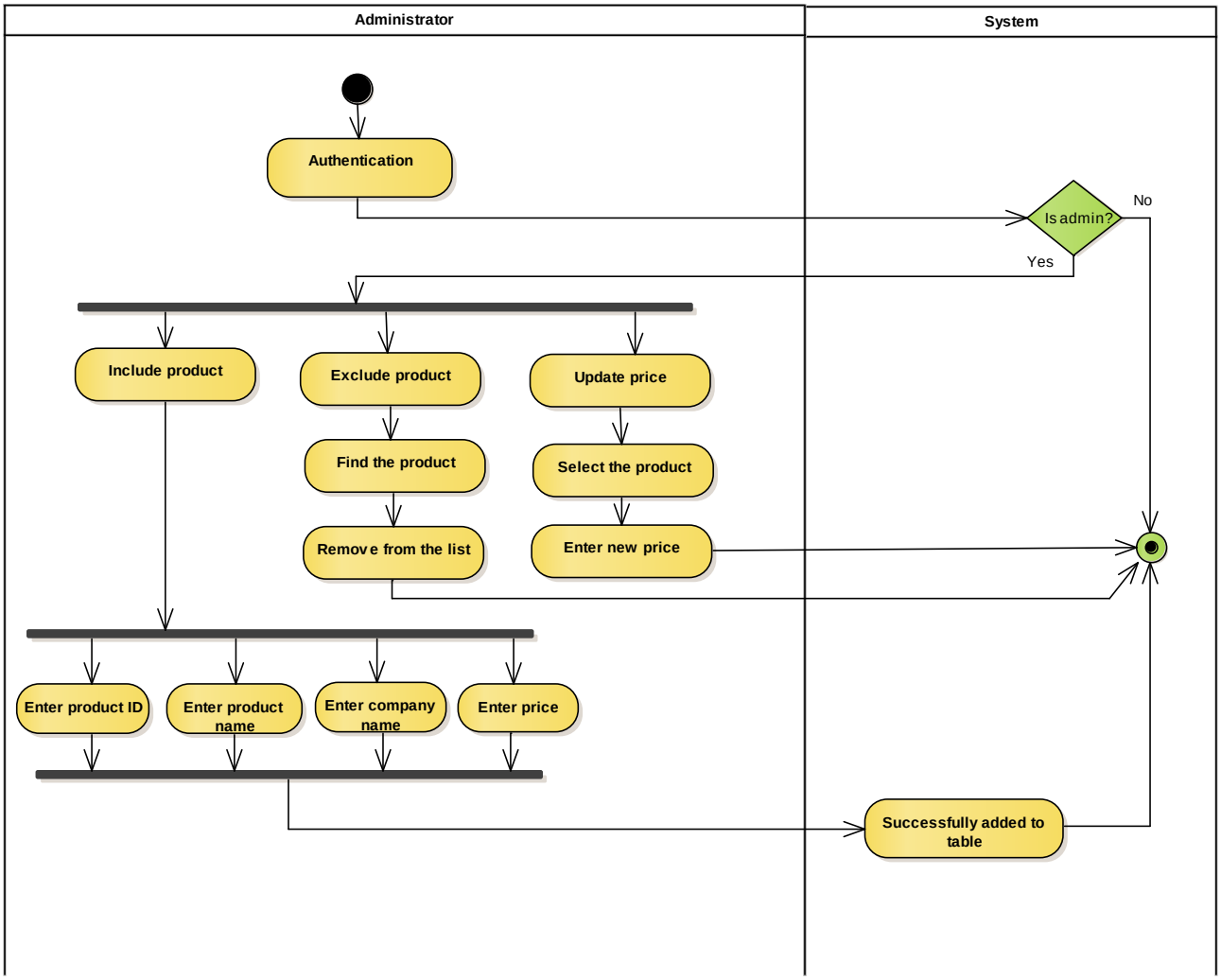


Figure 32: Inventory swimlane

4.5.10 NOTIFICATION SWIMLANE

notification
Version 1.1
Created on 09-11-17. Last modified 17-11-17

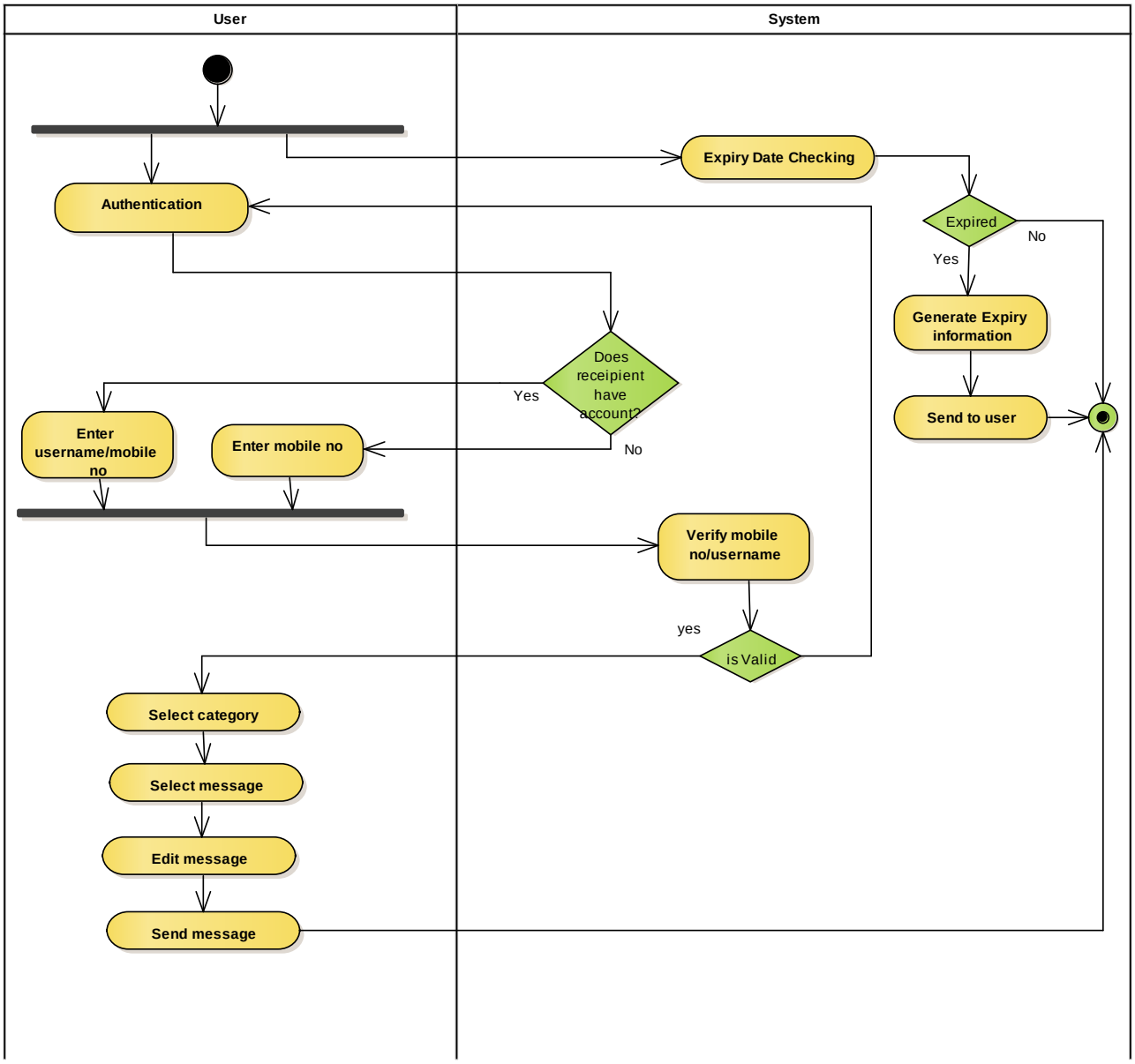


Figure 33: notification

5 DATA MODELING OF GMS

5.1 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.

5.1.1 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.

5.1.1.1 NOUN IDENTIFICATION

We identified all the nouns whether they are in problem space or in solution space from our story

Table 1: Noun Indentification

Serial no	Noun	Problem/solution space	Attributes
1	Authentication	p	
2	System	p	
3	User	s	6-13,17
4	Administrator	s	6-13,17
5	shopkeeper	s	6-13,15-17
6	Full name	s	
7	Permanent address	s	
8	Current address	s	
9	Designation	s	

10	NID	s	
11	User name	s	
12	Mobile number	s	
13	Password	s	
14	Account	s	11,13,17
15	Salary	s	
16	Commission	s	
17	Id	s	
18	characters	p	
19	underscore	p	
20	alphanumeric	p	
21	field	p	
22	input	p	
23	validation	p	
24	creation	p	
25	message	s	
26	information	p	
27	Log in	p	
28	Sign up	p	
29	Log out	p	
30	recovery	p	
31	retry	p	
32	Approval	p	
33	shop	p	
34	Management	p	
35	HR Management	p	
36	Product	s	17, 37-43
37	Name	s	
38	quantity	s	
39	price	s	
40	Delivery date	s	
41	Expiry date	s	
42	Company name	s	
43	Status	s	
44	inventory	s	45

45	Inventory level	s	
46	Contact	s	37,48-49
47	item	p	
48	Mobile number	s	
49	status	s	
50	Customer	s	37,48-49
51	Supplier	s	42,48-49
52	Client	s	17,37,81-83
53	Maintenance	s	17,37,43
54	Details	p	
55	Cost	s	
56	Expenditure	s	17,37,55,49
57	Electric bill	p	
58	Repairing bill	p	
59	Elements	p	
60	Furniture	p	
61	Bulb	p	
62	Accounting	p	
63	Agreement	s	17,37,68-74
64	Loan	s	17,37,68-74
65	Late payment	s	17,37,68-74
66	money	p	
67	Transaction	s	17,37,38,68,76-79,56
68	Type	s	17,37,68-74
69	Payee	s	
70	Receiver	s	
71	Amount	s	
72	Occurrence date	s	
73	Return date	s	
74	Description	s	
75	Due	s	17,37,68-74
76	Transaction time	s	
77	Transaction date	s	
78	amount	s	

79	Unit cost	s	
80	receipt	p	
81	Mobile number	s	
82	address	s	
83	email	s	
84	purpose	p	
85	Inventory	s	17,86
86	Inventory level	s	
87	Occasions	p	
88	Purposes	p	
89	Employee	p	
90	Modification	p	
91	Financial report	s	92-96
92	cash	s	
93	Payable	s	
94	Receivable	s	
95	profit	s	
96	Capital	s	
97	assets	p	
98	notification	s	25,99-104
99	Notification id	s	
100	sender	s	
101	receiver	s	
102	time	s	
103	type	s	
104	description	s	
105	week	p	
106	month	p	
107	calculation	p	
108	list	p	

5.1.1.2 Potential Data objects:

- Administrator-6-13,17
- Shopkeeper-6-13,15-17

- User-6-13,17
- Customer-37,48-49
- Product-17, 37-43
- Account-11,13,17
- Loan- 17,37,68-74
- Late payment-17,37,68-74
- Supplier-42,48-49
- Maintenance -17,37,43
- Management cost -17,37,43
- Transaction-17,37,38,68,76-79,56
- Due-17,37,68-74
- Notification-25,99-104
- Report-92-96
- Inventory -45
- client-17,37,81-83
- Agreement-17,37,68-74
- contact-37,48-49

5.1.1.3 Analysis for finalizing Data Objects

- Both administrator and shopkeeper have some common attributes. So their common attributes can be stored as **User**.
- Maintenance elements and management cost have some same attributes so an entity called "**Expenditure**" can store all information regarding these two entities.
- Loan, due and late payment involve with transaction. Information about those can be stored as **Agreements**.
- Details of shopkeeper is stored in **Shopkeeper** for updating information and tracking his activities.
- **Administrator** information must be saved for finding and verifying account.
- Transaction information are stored in **Transaction** and needs to be stored for profit calculation and generating a report.
- Report is generated through **Transaction** and must be kept for the

records of previous transaction.

- **Product** holds information on products. It will help to modify or update any information about that particular product.
- Client and user have **Contact** which needs to be stored.
- User and client receives **Notification**. System generates notification. So messages needs to be saved and all information must be stored.
- **Account** keeps information of user account.
- Information about supplier and customer is stored in **Client** as notifications will be send to them for many purposes.

5.1.1.4 Final Data Objects

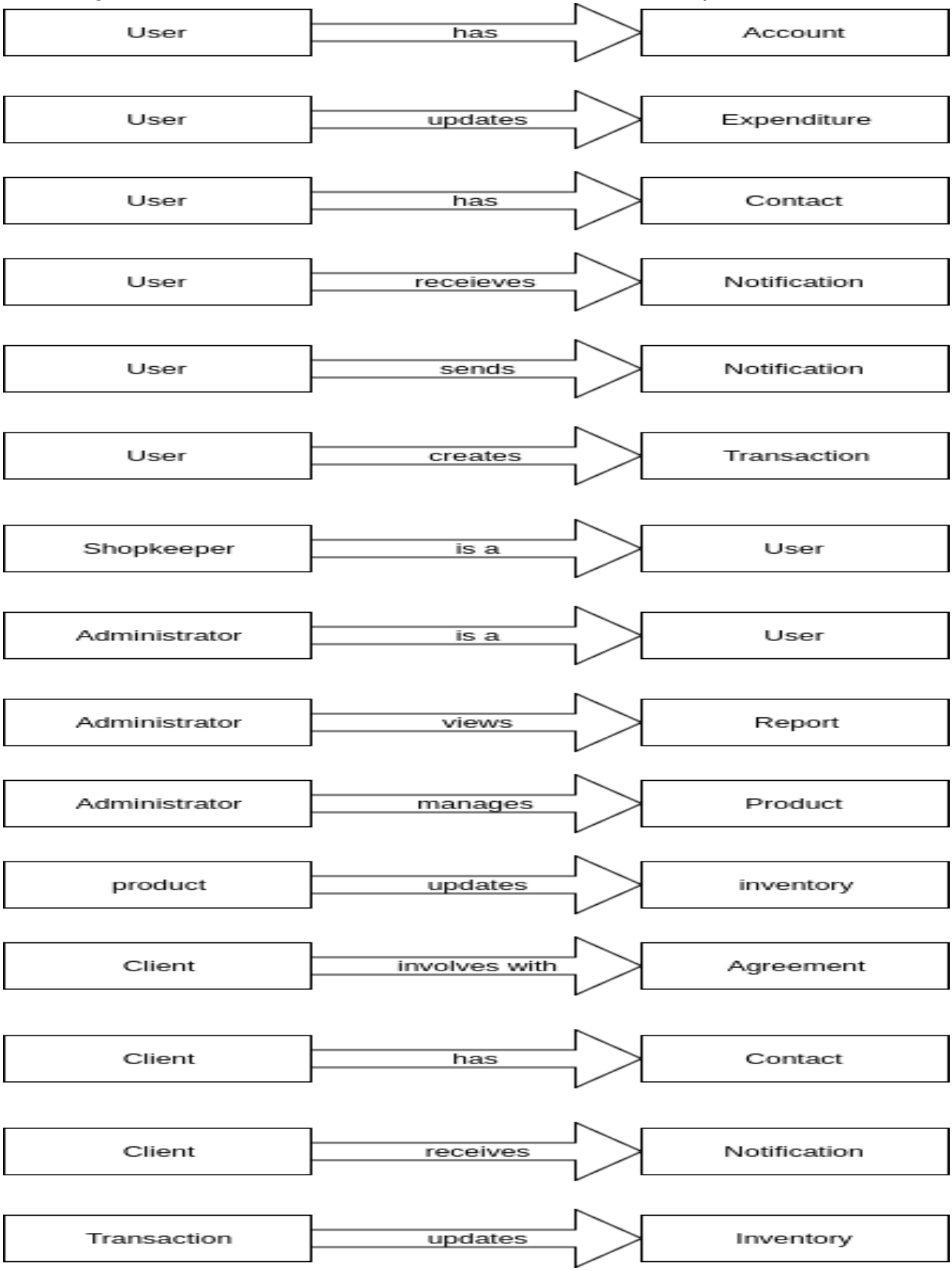
Table 2: Final data object

1	User: Full name, Permanent address, Current address Designation, NID, user Id
2	Administrator: Admin id, User id, Full name, Permanent address, Current address, Designation, NID
3	Shopkeeper: Shopkeeper id, User id, salary, commission, full name, permanent address, current address, designation, NID
4	Account: User name, User id, Password
5	Client: Client id, Name, Address, Type
6	Product: Product id, Admin id, Product name, Quantity, Price, Delivery date, Expiry date, Company Name, status
7	Expenditure: Type, Cost, Status, User id, EID
8	Transaction: Transaction id, User Id, Transaction date, Time, Unit cost, Total cost
9	Agreement: Transaction id, Type, Payee Id, Receiver Id, Amount, Status, Occurrence date, Return date, Description , AID
10	Report: Admin id, Report Id, type, Date, Cash, Account payable, Account Receivable, profit

11	Contact: Contact Name, Email, Mobile Number
12	Notification: Time, Notification id, Receiver User Id, Type, Description

5.2 DATA OBJECT RELATIONSHIPS

Data objects are connected to one another in different ways.



5.3 ENTITY RELATIONSHIP DIAGRAM

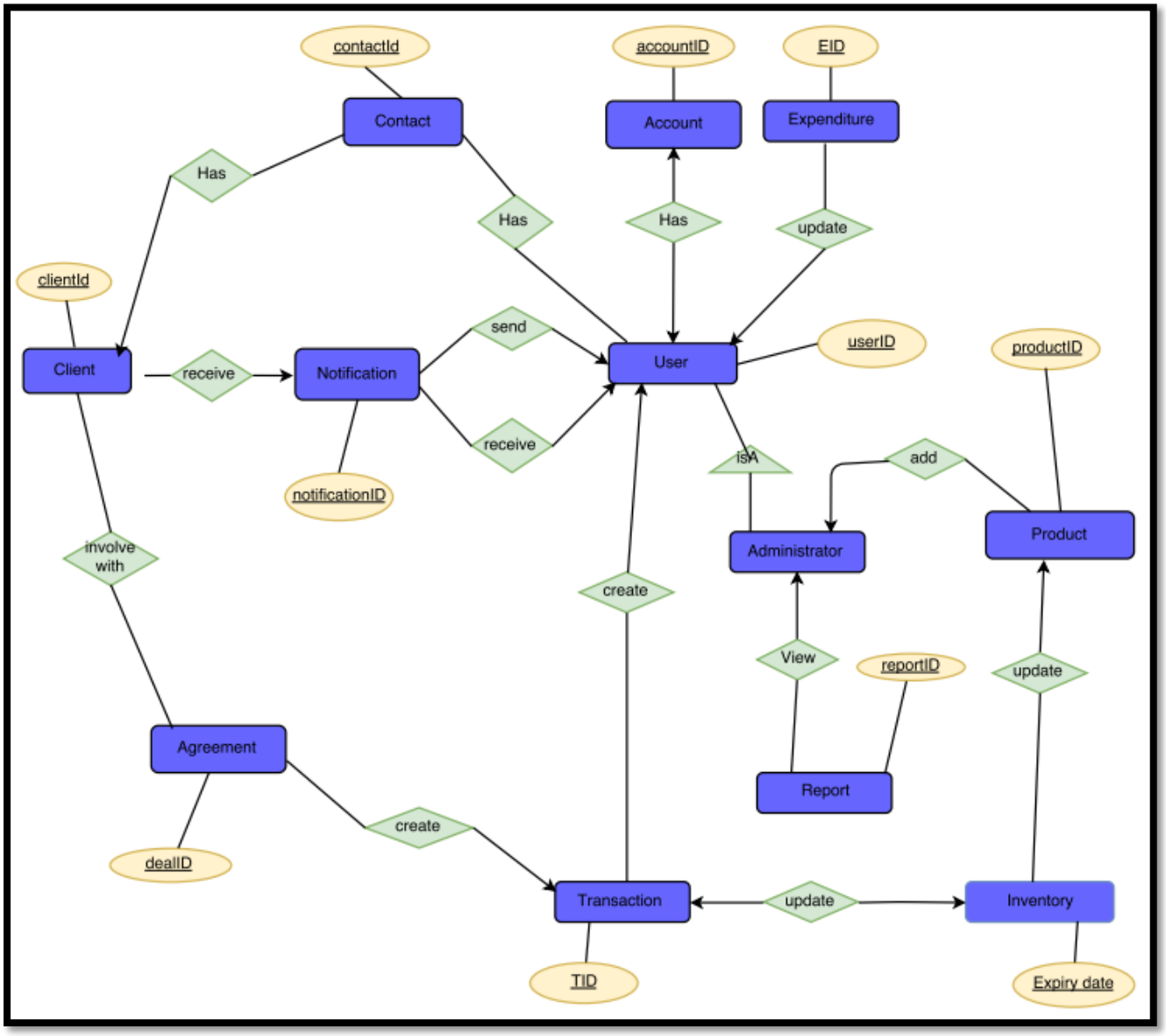


Figure 34: Entity Relationship Diagram

5.4 SCHEMA DIAGRAM

Table 3: Schema for user

User		
Attributes	Type	Size
User Name	Varchar2	80
Full name	Varchar2	80
Permanent address	Varchar2	80
Current address	Varchar2	80
Designation	Varchar2	80
NID	Varchar2	80
Mobile number	Varchar2	80
Password	Varchar2	80

Table 4: Account Schema diagram

Account		
Attribute	Type	Size
User id	Varchar2	30
User name	Varchar2	80
password	Varchar2	80

Table 5: Schema for Administrator

Administrator		
Attribute	Type	Size
UserId	Varchar2	80
Admin id	Varchar2	20
Full name	Varchar2	80
Permanent address	Varchar2	80

Current address	Varchar2	80
Designation	Varchar2	80
NID	Varchar2	80

Table 6: Schema for Product

Product		
Attribute	Type	Size
Product key	Varchar2	30
Product id	Varchar2	80
Admin id	Number	10
Product name	Number	20
Quantity	Varchar2	80
Price	Varchar2	80
Delivery date	Varchar2	80
Expiry date	Varchar2	80
Company name	Varchar2	80
status	Varchar2	80

Table 7: Schema for Transaction

Transaction		
Attribute	Type	Size
Transaction id	Varchar2	30
User id	Varchar2	20
Transaction date	Varchar2	80
Time	Varchar2	20
Unit cost	Number	20
Total cost	Number	20

Table 8: Schema for Client

Client		
Attribute	Type	Size
Client id	Varchar2	80
Name	Varchar2	80
Address	Varchar2	20
Type	Varchar2	20

Table 9: Schema for Agreement

Agreement		
Attribute	Type	Size
AID	Varchar2	30
Transaction type	Varchar2	20
Transaction id	Varchar2	80
Payee id	Varchar2	20
Receiver id	Varchar2	20
Amount	Number	20
Status	Varchar2	20
Occurrence date	Varchar2	20
Return date	Varchar2	20
Description	Varchar2	20

Table 10: Schema for Expenditure

Expenditure		
Attribute	Type	Size
EID	Varchar2	80
User id	Varchar2	20
Type	Varchar2	10
Cost	Number	10
status	Varchar2	20

Table 11: Schema User contact

Client Notification		
Attribute	Type	Size
Receiver User id	Varchar2	80
Receiver User id	Varchar2	80
Notification id	Varchar2	80
Time	Varchar2	80
Type	Varchar2	10
Description	Varchar2	100

Table 12: Financial report

Financial report		
Attribute	Type	Size
Cash	Varchar2	30
Report id	Varchar2	20
Admin id	Varchar2	80
Type	Varchar2	20
Date	Varchar2	20
Account payable	Number	20
Account receivable	Number	20
Profit	Profit	20

Table 13: Schema for Contact

Contact		
Attribute	Type	Size
User id	Varchar2	30
Name	Varchar2	20
Mobile number	Varchar2	80
Type	Varchar2	20

Table 14: Schema for Notification

Notification		
Attribute	Type	Size
User id	Varchar2	30
Notification id	Varchar2	20
Time	Varchar2	80
Type	Varchar2	20
Description	Varchar2	20

6 CLASS-BASED MODELING FOR GMS

This Chapter is intended to describe class based modeling of "**Grocery 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 CLASSIFICATION

To identify the potential classes, we have first selected the nouns from the solution space of the story. These were then characterized in seven general classification. 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 15: General Classification

No	Noun	General Classification
1	Authentication	3,5
2	User	4,5,7
3	Administrator	4,5,7

4	Shopkeeper	4,5,7
5	Account	2,4,7
6	User Name	
7	User Id	
8	Password	
9	Contact	2,7
10	Full name	
11	Current address	
12	Permanent address	
13	NID	
14	Designation	
15	Product	2,7,5
16	Product Id	
17	Product name	
18	Inventory	6,2
19	Quantity	
20	Expiry date	
21	Delivery date	
22	Company name	
23	Price	
24	Client	1,5,7
25	Client Id	
26	Client type	
27	Client name	
28	Dealings	3,6,7
29	Customer	1,5,7
30	Client address	
31	Late payment	3
32	Paid amount	
33	Due	3
34	Occurrence date	

35	Issue date	
36	Payee	
37	Receiver	
38	Notification	3,7
39	Supplier	1,5,7
40	Message	2
41	Status	
42	Mobile number	2
43	Receipt	2,6,7
44	Item	2,7
45	Amount	
46	Total price	
47	Transaction	3,4,7
48	Type	
49	Loan	3,7
50	Shop name	
51	Financial report	2,7
52	Cash	
53	Payable	
54	Receivable	
55	Asset	
56	profit	
57	system	2,4,7
58	database	2,4,7
59	interface	2,4,7
60	expenditure	2,7

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.

- Retain information
- Needed services
- Multiple attributes
- Common attributes
- Common operations
- Essential requirements

no	noun	Selection criteria
1	Authentication	3
2	User	1-5
3	Administrator	1-5
4	Shopkeeper	1-5
5	Account	1-5
6	Contact	3-5
7	Product	1,3,4,5
8	Client	3,4,5,6
9	Deal	3,4,5
10	Late payment	3,4,5
11	Customer	3,4
12	Due	3,4,5
13	Supplier	3,4
14	Notification	3,4,5

15	Receipt	3,4,5
16	Item	3,4,5
17	Transaction	1,3,4,5
18	Loan	3,4,5
19	Financial report	1,6
20	Database	1
21	System	1
22	Interface	1
23	expenditure	3,4,5

6.4 ASSOCIATE NOUN AND VERB IDENTIFICATION

We now identify the nouns and verbs associated with the potential classes to better find out the attributes and methods of each class.

Table 16: Associate noun and identification

No	Potential class	Noun	verb
1	User	Full name, permanent address, current address, NID, designation, contact, account	Updating user information(full name, permanent address, current address, NID, designation, contact), log out, creating notification, record transaction, recording late payment, recording due, updating price
2	Administrator	Full name, permanent address, current address, NID, designation, contact, account	Approving account, managing inventory, adding employee, firing employee, removing employee, updating employee information, recording expenditure
3	Account	User name, password, user Id, contact	Updating contact, generating user Id, matching password, changing password,
4	Authentication	User, database,	Log in, sign up, recovering

		interface	account, updating user, finding user
5	Product	Product Id, name, quantity, expiry date, delivery date, price, company name	Updating products, generating product Id
6	Employee	Name, address, contact, NID, employee ID	Updating employee information, generating employee Id
7	Late Payment	Customer, return date, amount, time, status	Updating late payment information, increasing amount, decreasing amount, updating financial report, generating Id
8	Customer	Id, name, address, contact	Updating customer information
9	Due	Due date, pay date, supplier, amount	Updating due information, increasing amount, decreasing amount, updating financial report
10	Supplier	Company name, id, contact	Updating supplier information, generating id
11	Notification	Id, time, account, message, contact	Generating id, sending message, receiving message, storing, removing
12	contact	Mobile number, status, name	Checking repeat, updating information
13	item	Unit price, quantity, total price, product name	Calculating total price, setting item information
14	Receipt	Item, amount	Removing item, adding item, calculating total amount, printing, updating some information about item
15	Transaction	Id, paid amount, transaction type	Determine due, determine late payment, updating cash, updating financial report, generating Id, processing due, processing late payment
16	Loan	Id, shop name, quantity, item	Generating id, updating loan information

17	Financial Report	Cash, payable, receivable, date	Showing report, updating information, calculating profit
18	System	Database, interface	Creating database, deleting database, accessing database, disconnecting database, creating temporary files, authentication, showing interface, showing files, locking system
19	Database	Connection, table name, status	Insert, update, delete, checking connection, search
20	Validation	Regex full name, regex NID, regex user name, regex mobile number, regex password	Verifying user name, password, full name, mobile number, NID
21	Management cost	name, amount	Updating information about expenditure, updating financial report
22	Maintenance element	Name, status	Updating information about expenditure,updating financial report

6.5 ATTRIBUTE SELECTION

After identifying the classes, we have specified their attributes and methods.

Table 17: Attribute Selection

No	Name	Attributes
1	System	Database Interface
2	Interface	User
3	Authentication	user Database AuthenticationInterface
4	User	fullName permanentAddress currentAddress designation NID Contact

		Account userId
5	Account	userName password userId cantact accountId
6	Shopkeeper	salary commission isPaid
7	AdministrativeUser	fullName permanentAddress currentAddress designation NID Contact Account userId
8	Product	productId productName price quantity expiryDate deliveryDate companyName totalPrice
9	Agreements	type payee receiver amount occuranceDate returnDate id,quantity Product Client description
10	Notification	notificationId time Account message Contact
11	Contact	name mobileNumber status
12	Receipt	amount

		Product
13	Transaction	TID amount paidAmount receipt transactionType userId
14	FinancialReport	cashId amount payableAmount receivableAmount
15	Expenditure	EID type amount status
16	Client	CID name Contact address type payable receivable
17	Database	connection isConnected tableName

6.6 METHOD IDENTIFICATION

After identifying the classes, we have specified their methods.

Table 18: Method Indentification

No	Name	Methods
1	System	• connectDatabase() • createInterface() • checkEvents() • deleteData() • authenticate() • showInterface() • saveFiles() • disconnectDatabase() • lockSystem() • createNotification() • showFinancilaReport()
2	Interface	• showAuthenticationAction()

		• showMenu() • getInput() • getAction() • authenticate()
3	Authentication	• setUser() • validateInput() • login() • signUp() • findUser() • logOut() • getUser() • pause() • recoverAccount()
4	User	• validateNid() • validateFullName() • verifyUser() • setCurrentAddress() • setContact() • setAccount() • getFullName() • setFullName() • getPermanentAddress() • getCurrentAddress() • getDesignation() • getNID() • getContact() • getAccount() • createNotification() • recordTransaction() • recordDealingInformation() • updateInventory() • recordExpenditure()
5	Shopkeeper	• calculateSalary() • getSalary() • setSalary() • getCommission() • setCommission()
6	AdministrativeUser	• approveAccount() • addShopkeeper() • removeShopkeeper() • updateShopkeeperInformation() • includeProduct() • exludeProduct() • withdraw() • viewReport() • provideSalary()
7	Product	• calculatePrice()

		• getProductID() • getProductName() • getPrice() • getQuantity() • getExpiryDate() • getDeliveryDate() • getCompanyName() • getTotalPrice() • setProductID() • setProductName() • setPrice() • setQuantity() • setExpiryDate() • setDeliveryDate() • setCompanyName() • setTotalPrice() • searchProduct() • addProduct() • removeProduct()
8	Agreements	• generateID() • increaseAmount() • decreaseAmount() • calculateAmount() • getType() • getPayee() • getReceiver()
9	Client	• calculatePayable() • calculateReceivavle() • generateId()
10	Notification	• sendMessage() • receiveMessage() • storeMessage() • removeMessage()
12	Contact	• checkDuplicity()
13	Receipt	• addProduct() • removeProduct() • calculateAmount() • printReceipt()
14	Transaction	• generateId() • calculatePrice() • createExpenditure() • paidAmount() • determineDealingAmount() • updateCash() • updateFinancialReport() • processDealingInformation() • createReceipt()

		• updateInventory()
15	FinancialReport	• showReport() • calculateProfit() • calculateMonthlyProfit()
16	Expenditure	• updateFinancialReport() • generteId()
17	Database	• insert() • update() • delete() • search()

6.7 FINALIZING CLASSES

To identify final classes we need to first check that if there can be any hierarchies or merges. These are given below

1. 'User' has common attributes and methods with 'AdministrativeUser' and 'Shopkeeper' so we merge them as 'User' and add the common and unique attributes and methods to 'User'.

2.'MaintenanceElements' and 'ManagementCost' can be merged as 'Expenditure'. The attributes for this class are:

- EID
- expenditureName
- cost
- Status

3. 'LatePayment', 'Due' and 'Loan' can be merged as 'Deal' as these classes have same attributes. Attributes of this class are

- payee
- receiver
- amount
- occurrenceDate

- returnDate
- id
- quantity
- Product
- type

4.'Customer' and 'Supplier' can be merged as 'Client' as both of them are external entity and have some same attributes. Both entities can be distinguished by an attribute called 'type'. Attributes are given below

- id
- name
- contact
- address
- type
- payable
- Receivable

5. 'System' class will be used for showing user interface, creating database, managing temporary files.

6. 'Database' class is for saving, searching, updating and removing information into the system.

7. 'ProductCatagory' and 'Product' has same attributes. So we can merge these two classes as 'Product'.

- productid
- ProductName
- Price
- Quantity
- expiryDate
- DeliveryDate
- CompanyName

- TotalPrice

8. 'User', 'Customer' and 'Supplier' all of them have contact. So a class 'Contact' is introduced for holding information regarding contact. Attribute are

- name
- mobileNumber
- Status

6.8 CLASS CARDS

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

Table 19: Class Card for System Class

System	
Attribute	Method
• Database • Interface	• connectDatabase() • accessDatabase() • showInterface() • createInterface() • deleteData() • saveFiles() • pause() • createNotification() • disconnectDatabase()
Responsibilities	Collaboration
• Handling database • Creating interface • Authentication	Authentication Interface Database User

Table 20: Class card for Interface class

Interface	
Attribute	Method
• User	• showAuthenticationMenu() • authenticate() • showMenu() • getInput() • getAction() • authenticate()
Responsibilities	Collaboration
• Creating interface • Authentication	Authentication Database User

Table 21: Class card for Authentication

Authentication	
Attribute	Method
• User • Database	• signUp() • validateInput() • setUser() • login() • findUser() • getUser()
Responsibilities	Collaboration
• getting input for signup/login • finding user from database	User Database

Table 22: Class card for User

User	
Attribute	Method
• fullName • permanentAddress • currentAddress • designation • NID • Contact • userID	• validateInput() • verifyUser() • recoverAccount() • createNotification() • recordTransaction() • recordDealingInformation() • updateInventory() • recordExpenditure() • updateExpenditure() • logout() • getAccount() • setCurrentAddress() • setContact() • setAccount() • getFullName()

	• setFullName() • getPermanentAddress() • getCurrentAddress() • getDesignation() • getNID() • getContact()
Responsibilities	Collaboration
• verifying user data • creating account • involving in transaction	Authentication System Database Transaction Contact Agreement Notification Product

Table 23: Class card for AdministrativeUser

AdministrativeUser	
Attribute	Method
• fullName • permanentAddress • currentAddress • designation • NID • Contact • Account • userID	• approveAccount() • addShopkeeper() • removeShopkeeper() • updateShopkeeperInformation() • includeProduct() • exludeProduct() • withdraw() • viewReport() • provideSalary()
Responsibilities	Collaboration
• HR management • Approving Account • Managing inventory • Recovering expenditure	Authentication Product Expenditure FinancialReport

Table 24: Class card for Product

Product	
Attribute	Method
• productid • productName • price • quantity • expiryDate • deliveryDate • companyName • totalPrice	• generateProductId() • searchProduct() • addProduct() • removeProduct() • calculatePrice() • getProductID() • getProductName() • getPrice() • getQuantity() • getExpiryDate() • getDeliveryDate() • getCompanyName() • getTotalPrice() • setProductID() • setProductName() • setPrice() • setQuantity() • setExpiryDate() • setDeliveryDate() • setCompanyName() • setTotalPrice()
Responsibilities	Collaboration
• Calculating total involve amount in a transaction	

Table 25: Class card for Agreement

Agreement	
Attribute	Method
• type • payee • receiver • amount • occurrenceDate • returnDate • id	• generateID() • createAgreement() • search() • increaseAmount() • decreaseAmount() • calculateAmount() • getType()

• quantity • Product • Client • description	• getPayee() • getReceiver() • takeInput() • getAmount() • getReturnDate() • getOccurrenceDate() • getId() • getQuantity() • setAmount() • setReturnDate()
Responsibilities	Collaboration
• managing information regarding deal • calculating amount	Client FinancialReport

Table 26: Class card for Client

Client	
Attribute	Method
• clientID • name • contact • address • type • payable • receivable	• generateId() • calculatePayable() • calculateReceivable() • getId() • getName() • setName() • getContact() • setContact() • getAddress() • setAddress() • createClient()
Responsibilities	Collaboration
• work as structure of customer and supplier	Contact

Table 27: Class Card for Notification

Notification	
Attribute	Method
• notificationId • Time • Message • Contact	• sendMessage() • receiveMessage() • storeMessage() • removeMessage()
Responsibilities	Collaboration
• managing notification system • storing message	Client User

Table 28: Class Card for Contact

Contact	
Attribute	Method
• contactId • name • mobileNumber • status • email	• checkDuplicacy() • generateContactId() • getStatus() • setStatus() • getMobileNumber() • setMobileNumber()
Responsibilities	Collaboration
• sending message for password recovery • storing contact with unique name	

Table 29: Class card for Receipt

Receipt	
Attribute	Method
• amount • Product	• addProduct() • removeProduct() • calculateAmount() • printReceipt()

	• getProduct() • setAmount()
Responsibilities	Collaboration
• calculating amount of each item • containing information of sold products	Product

Table 30: Class card for Transaction

Transaction	
Attribute	Method
• TId • Amount • paidAmount • receipt • transactionType • userId	• generateId() • createReceipt() • calculatePrice() • createTransaction() • paidAmount() • determineDealingAmount() • updateCash() • updateFinancialReport() • processDealingInformation() • updateInventory()
Responsibilities	Collaboration
• updating cash • recording transaction information	Financial report Agreement Product

Table 31: Class card for FinancialReport

FinancialReport	
Attribute	Method
• cashID • amount • payableAmount • receivableAmount • date • profit	• showReport() • getCash() • getReceivableAmount() • getPayableAmount() • calculateProfit() • calculateMonthlyProfit() • getDate()
Responsibilities	Collaboration
• calculating profit • providing report on daily transaction	AdministrativeUser Client

Table 32: Class card for Expenditure

Expenditure	
Attribute	Method
• EID • type • amount • status	• generateEID() • getType() • getAmount() • getStatus() • setType() • setAmount() • setStatus()
Responsibilities	Collaboration
• generate EID • Update cash	Financial report Database

Table 33: Class card for Database

Database	
Attribute	Method
• isConnected • tableName	• insert() • update() • delete() • search()
Responsibilities	Collaboration
• Insert element • Update element • Delete element • Search element	

7 BEHAVIORAL MODELING OF GMS

7.1 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.

Table 34: Event Indentification

Serial no	Events	Primary object	Collaborator	Invoked method
1.	Provide input for account creation	Administrative User	Authentication	signUp()
2.	Provide input for account creation	shopkeeper	Authentication	signUp()
3.	administrative account will be confirmed	Administrative User		addAccount()
4.	Shopkeeper account will be confirmed	AdministrativeUser		addAccount()
5.	Provide input for login	User	Authentication	login()
6.	Checks valid input	System	User	Login(), signUp(), verifyName(), verifyPassword(),findUser()
7.	Verify password	Authentication	User	verifyPassword()
8.	Verify user name	Authentication	User	verifyName()
9.	For invalid input error message is shown	System	Interface	showErrorMessage()
10.	Account will be locked for a certain period for consecutively three wrong input	System		lockAccount()
11.	Want to recover account	User	Authentication, Notification	recoverAccount(),recover()
12.	Send forgotten password through his contact	System	Notification,User	sendMessage()

13.	Want to log out	User	Authentication	logout()
14.	Remove shopkeeper	AdministrativeUser	Shopkeeper, Database	removeEmployee(), remove()
15.	Update shopkeeper information	AdministrativeUser	Shopkeeper, Database	update()
16.	Pays shopkeeper salary	AdministrativeUser	User	provideSalary()
17.	include product	AdministrativeUser	Product, Database	update-Inventory(), insert()
18.	ID will be auto-generated	Product, User, Client, Deal, Expenditure, Transaction		generateId()
19.	Exclude product	AdministrativeUser	Product, Database	update-Inventory(), remove()
20.	Update inventory	AdministrativeUser	Database	update()
21.	Update price for any products	Adminstrative User	Product, Database	update-Inventory()
22.	Record expenditure information	User	Expenditure, Database	record-Expenditure(), insert()
23.	Calculates profit	Financial report	Database	calculate-Profit()
24.	Withdraw money	AdministrativeUser	Database	withdraw()
25.	See the financial report	AdministrativeUser	Interface,Database	veiwReport()
26.	User information will be stored	System	Database	insert()
27.	Add products to receipt	User	Transaction,Receipt	createReceipt(), addProductToReceipt()
28.	Remove product from receipt	User	Transaction,Receipt	createReceipt, removeProductFromReceipt()

29.	Calculates total price	System	Receipt	calculateAmount()
30.	Create late payments	User	Agreement, Client	createAgreement()
31.	Record transaction information	User	Transaction, Database	createTransaction()
32.	Create due	User	Agreement, Client	createAgreement()
33.	Create Loan	User	Agreement, Client	createAgreement()
34.	Print receipt	Transaction	Receipt	createReceipt, print
35.	Record dealing information	User	Agreement	createAgreement()
36.	Show interface	System	Interface	showInterface()
37.	Update late payments	User	Agreement	Update()
38.	Update loan	User	Agreement	Update()
39.	Update Due	User	Agreement	Update()
40.	Updates cash	Transaction	Database	updateCash()
41.	Create notification	System, User	Notification, Client	createNotification()
42.	Sends notification	User	Notification	sendMessage()
43.	Receive notification	System	User	receiveMessage()
44.	Removes notification	System	Database	Remove()
45.	Checks for unsaved files	System		saveFiles()
46.	Ask confirmation before taking action on unsaved file(s)	System		saveFiles()
47.	Power off the system	System		saveFiles(), diconnectDatabase()

7.1.1 EVENTS AFTER ANALYSIS

Table 35: Merged Events

Serial No	Events	Primary Object	Collaborator	Invoked Methods
1.	Logs into account by authentication	User	Authentication	logIn() validateInput()
2.	Manages products	User	Product Transaction	updateInventory() updateCash()
3.	Creates transaction	User	Transaction	recordTransaction()
4.	Sends and receives notification	User	Notification	sendMessage() receiveMessage() createNotification()
5.	Records expenditure	User	-	recordExpenditure() updateExpenditure()
6.	Manages agreements	User	Transaction Agreement	createAgreement() search() recordDealingInformation()
7.	Logs out	User	Authentication	logOut()
8.	Recovers accounts	User	Authentication	accountRecovery()
9.	Chooses type of operation	administrative User	-	getType()
10.	Approves shopkeeper account	administrative User	-	approvesAccount()
11.	Withdraws money from the shop	administrative User	-	withdraw()
12.	Can see report	administrative User	-	viewReport()
13.	Manages shopkeeper	adminstrativeUser	-	addShopkeeper() removeShopkeeper() updateShopkeeper()
14.	Includes products	administrative User	-	includeProduct()
15.	Excludes	administrative	-	excludeProduct()

	products	User		
16.	Handles user sign up	Authentication	-	signUp() findUser() validateInput()
17.	adds user to database	Authentication	Database	setUser()
18.	Handles log in	Authentication	-	logIn() findUser()
19.	Gets the user	Authentication	-	getUser()
20.	Helps user to recover account	Authentication	-	recoverAccount()
21.	Connects database	System	Database	connectDatabase()
22.	Accesses database	System	Database	accessDatabase()
23.	Creates temporary files	System	-	createTemp-oraryFiles()
24.	Shows interface	System	-	showInterface()
25.	Deletes data	System	Database	deleteData()
26.	Is shut down by itself	System		shutDown() saveFiles() disconnectDatabase()
27.	Creates receipt	Transaction	-	createReceipt()
28.	Calculates total price	Transaction	-	calculateTotalPrice() getPaidAmount
29.	Updates cash	Transaction	-	updateCash() search() printReceipt()
30.	Creates agreements	Transaction	-	createAgreements() updateCash() printReceipt()
31.	Stores information	Agreement	-	getter() setter()
32.	Updates amount	Agreement	-	increaseAmount() decreaseAmount()
33.	Stores client	Client	-	getter()

	information			setter()
34.	Stores product information	Product	-	getter() setter()
35.	Stores receipt information	Receipt	-	getter() setter()
36.	Stores expenditure information	Expenditure	-	getter() setter()
37.	Stores report information	FinancialReport	-	getInformation() setInformation()
38.	Calculates profit	FinancialReport	-	calculateProfit()
39.	Updates information	Database	-	update ()
40.	Inserts information	Database	-	insert ()
41.	search information	Database	-	search()
42.	deletes information	Database	-	delete()
43.	Shows menu	Interface	-	showMenu()
44.	Shows authentication menu	Interface	-	showAuthenticationMenu()
45.	Authenticate user	Interface	Authentication	authenticate()

7.1.2 STATE TRANSITION DIAGRAMS

Authentication

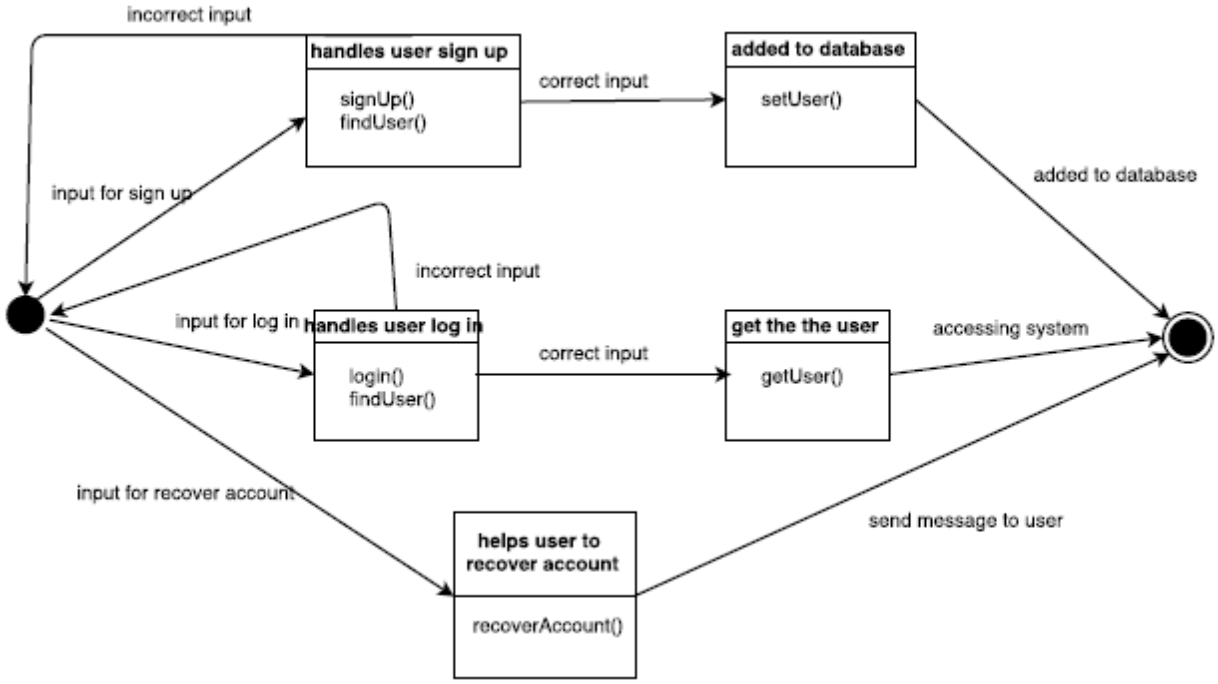


Figure 35: Authentication State Diagram

User

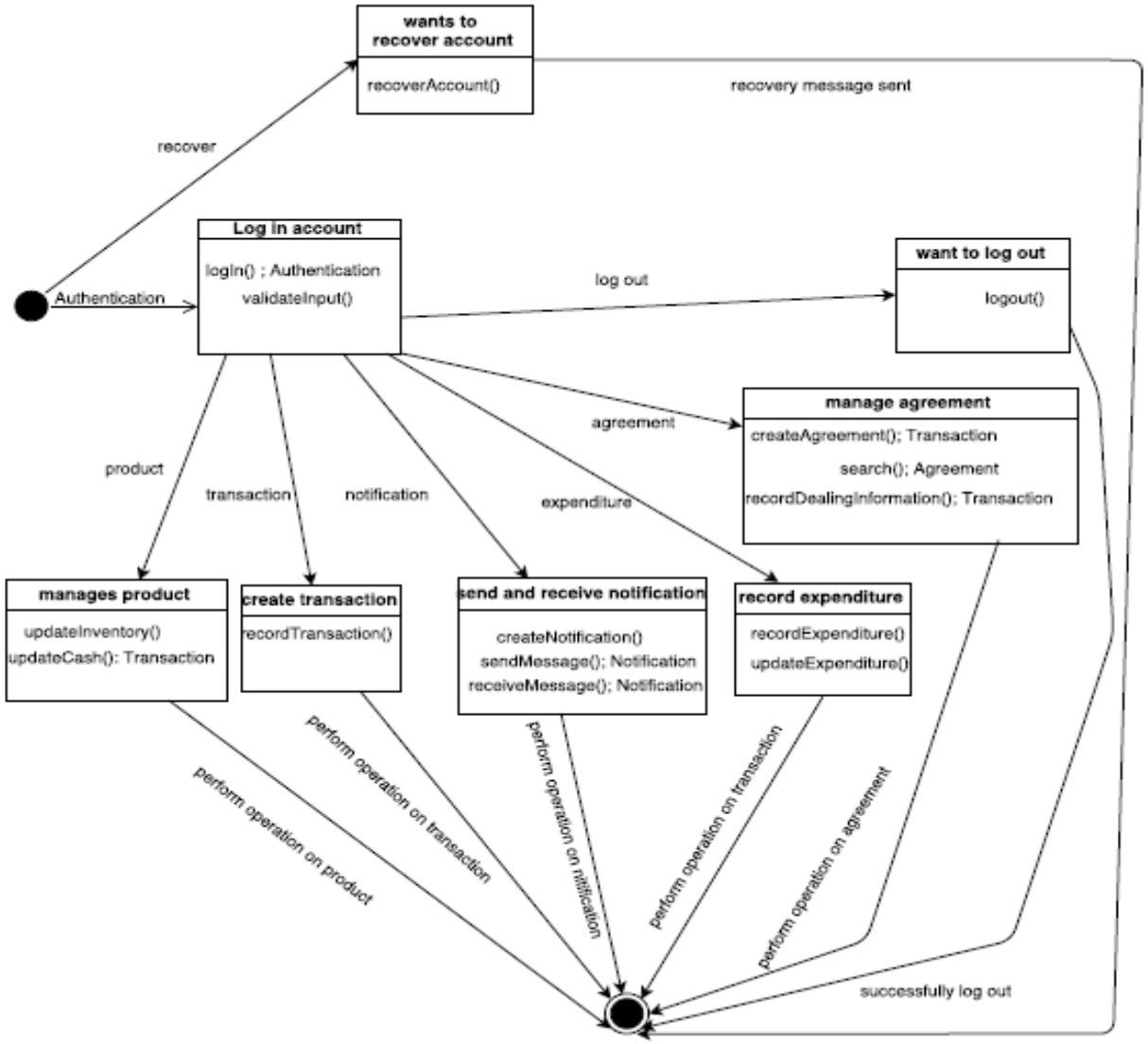


Figure 36: User State Diagram

Administrative User

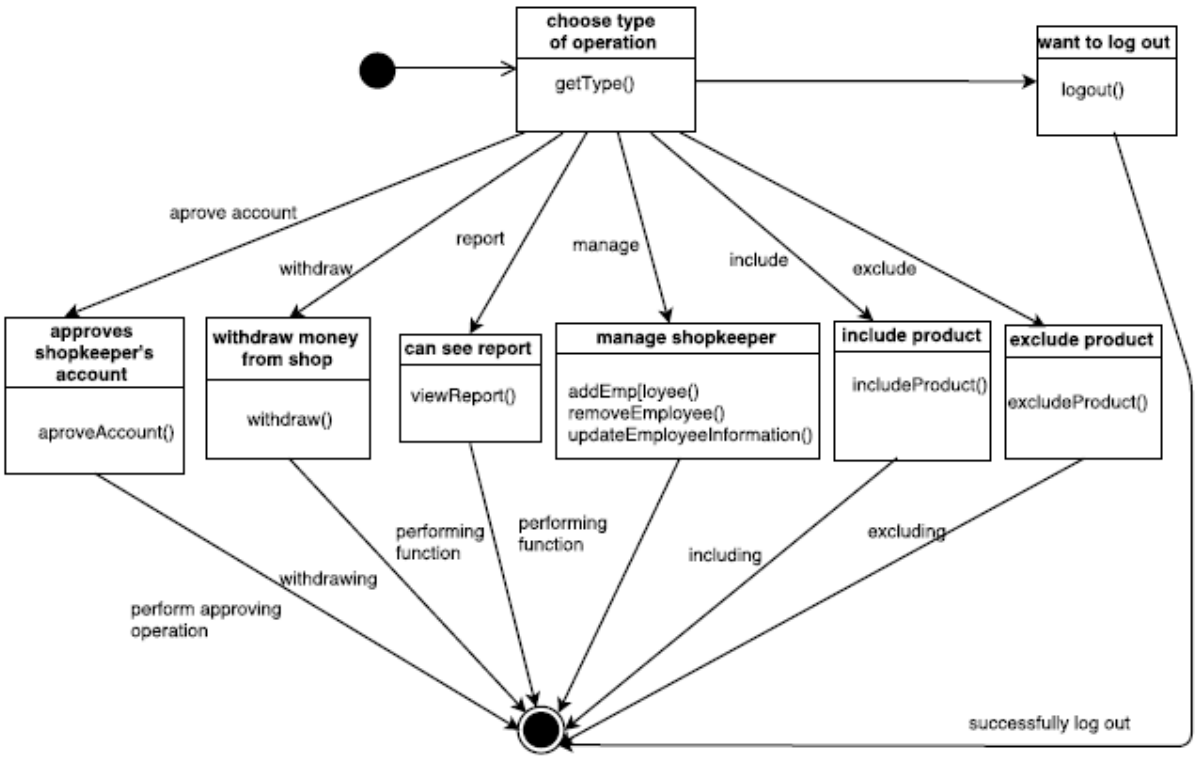


Figure 37: Administrator State Diagram

Transaction

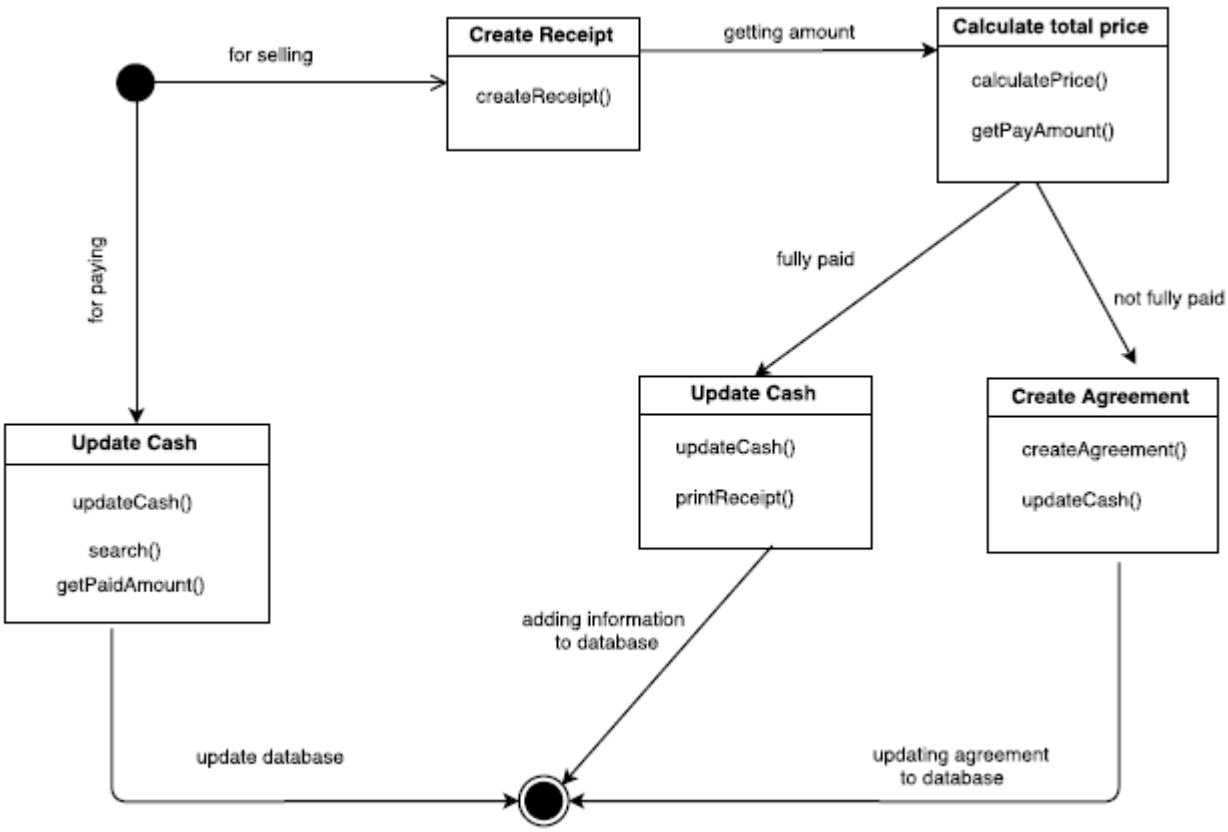


Figure 38: Transaction State Diagram

Agreement

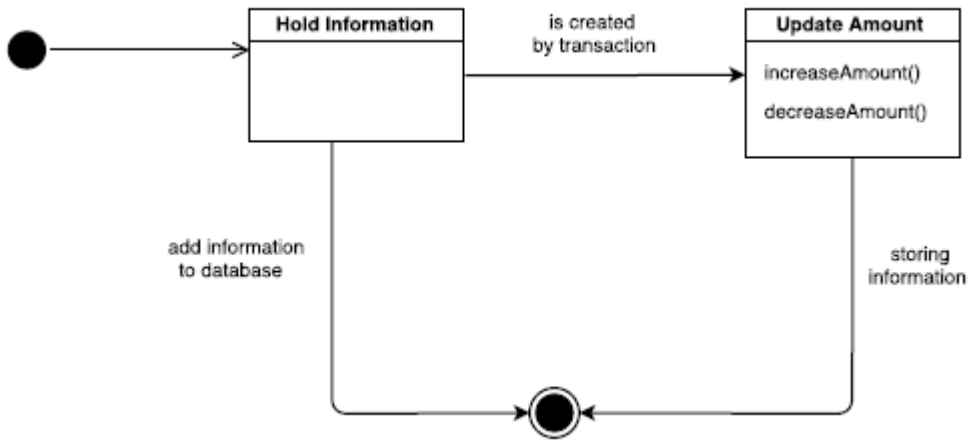


Figure 39: Agreement State Diagram

System

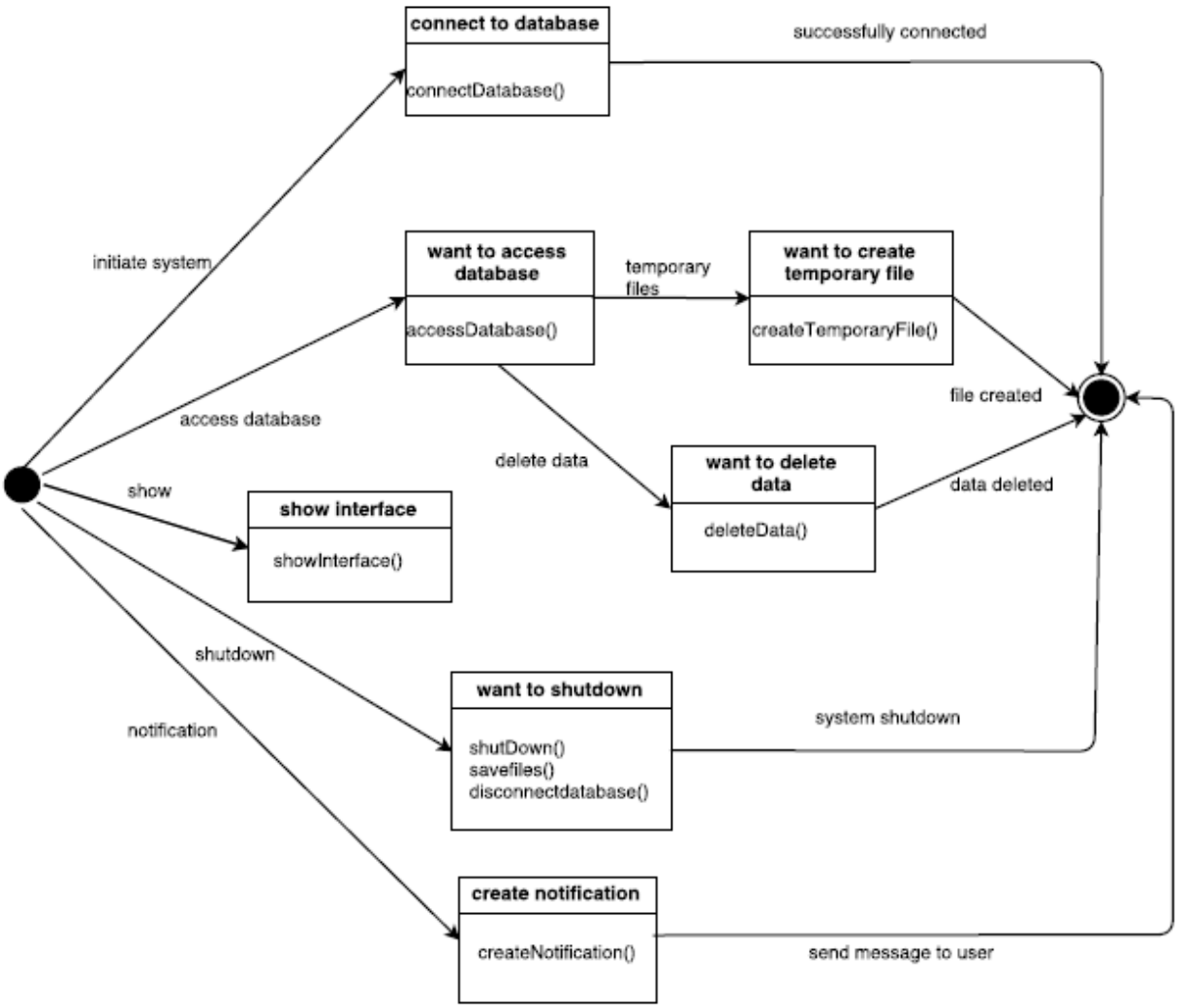


Figure 40: System State Diagram

Interface

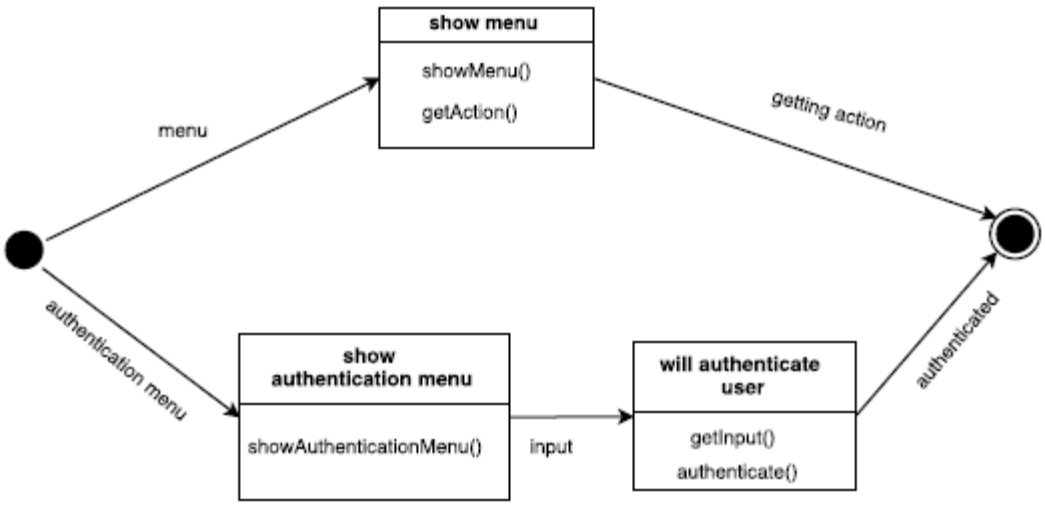


Figure 41: Interface State Diagram

Product

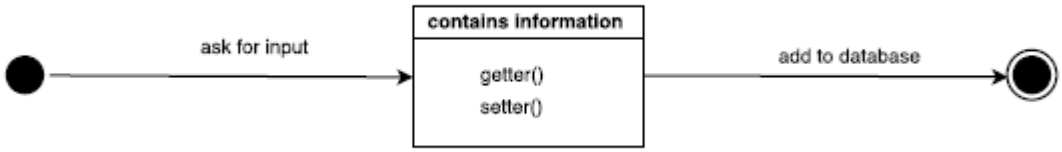


Figure 42: Level-0 GMS

Receipt

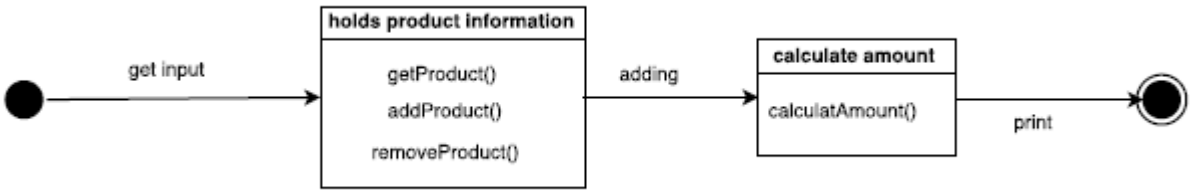


Figure 43: Receipt State Diagram

Notification

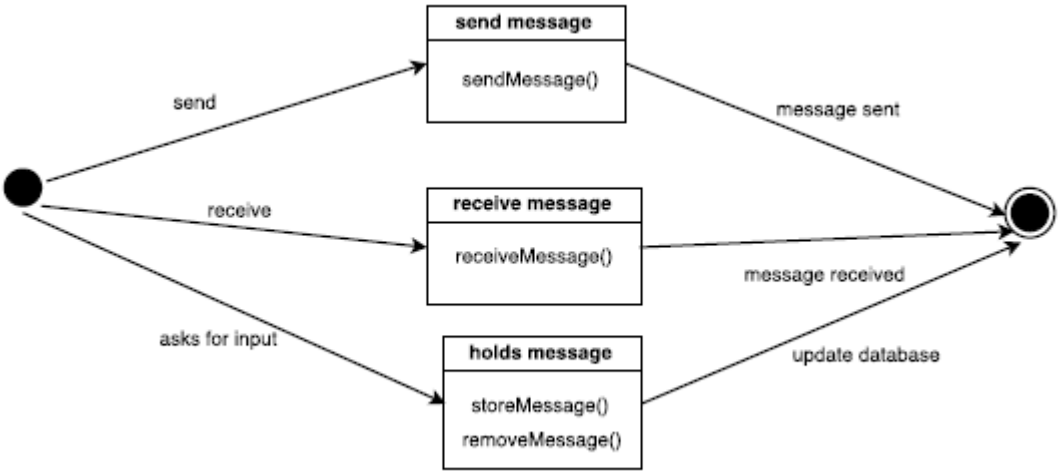


Figure 44: Notification State Diagram

Database

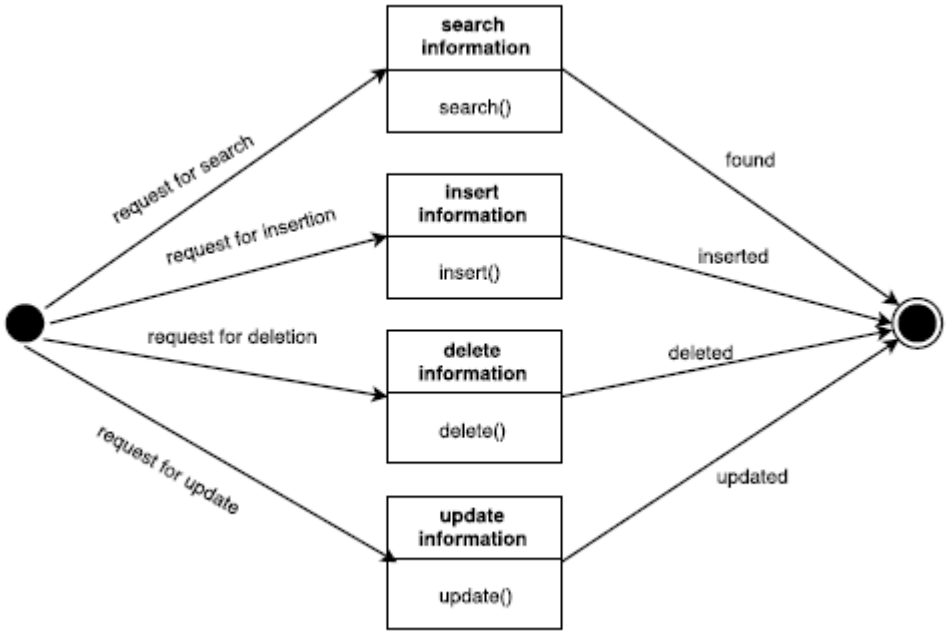


Figure 45: Database State Diagram

Financial Report

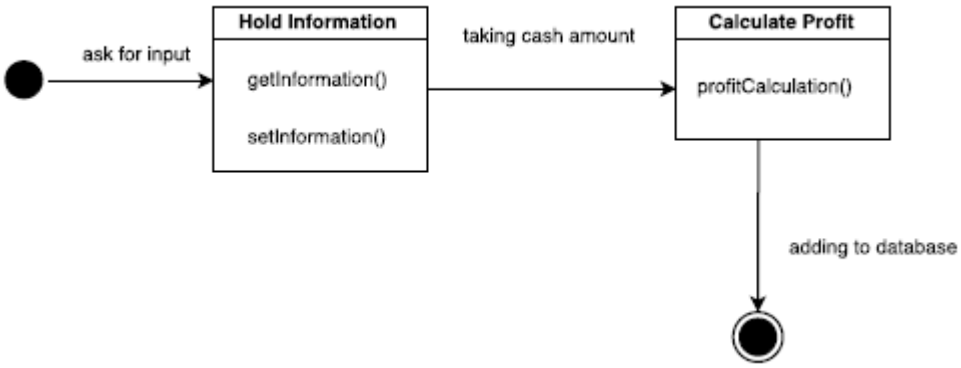


Figure 46: Financial Report State Diagram

7.2 SEQUENCE DIAGRAM

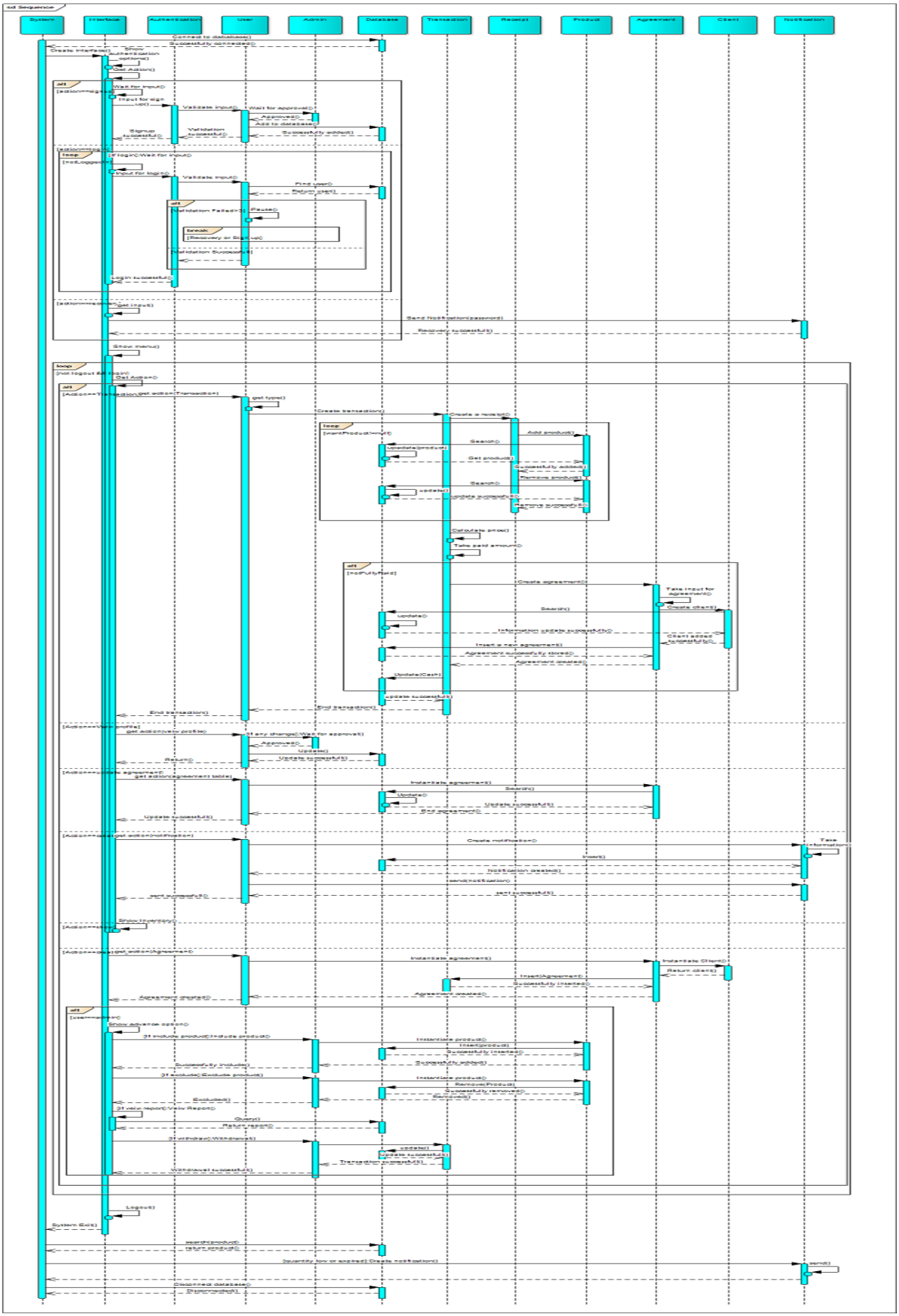


Figure 47: Sequence Diagram