Taibah University
College of Computer Science and Engineering

CS284 Software Modeling and Analysis

Course Instructor: Marwah Alaofi

Requirement Analysis Document for

Supermarket Management System

by

Ameera Alahmadi	3752744
Omamah Alrehaili	3752216
Raneem Alnuman	3750258
Sana Qarah	3752533
Maryam Alradaddi	3752699

Contents

1.Introduction 2
2. Current System 2
3. Proposed System 2
3.1 Requirements elicitation 3
3.2 Functional requirements 6
3.3 Nonfunctional requirements 7
3.4 System Models 8
3.4.1 Functional model 8
3.4.1 Use case descriptions 9
3.4.2 Object model 17
3.4.3 Dynamic model 18
4. Conclusion 21

1.Introduction

This project deals with the analysis and modeling of a Supermarket management system; a system that handles the activities done to manage a supermarket, it is designed to reduce the effort required to do management tasks and make the existing system faster and more reliable. The main goal is to study and analyze the client needs to arrive at a definition of the problem and to derive the system requirements.

2. Current System

The client does not currently have a system that covers management tasks, instead the sales are processed using a cash register device that only takes manual input and doesn’t store the sale record after the invoice has been printed, thus stock status can only be estimated by calculating the money in the cash register and checking the products shelves at the end of each week, most of the details and important records are maintained using paper, tasks are handled by many employees yet productivity is low.

3. Proposed System

The proposed system shall minimize the effort and resources used in the management system. It provides some activities and functions which shows the process between companies or suppliers, the supermarket store and repository, customers and employees, it improves communication between employees and provides the store owner with data that can be used for analysis to increase the sales profit.

3.1 Requirements elicitation

We started the requirements elicitation process by conducting an interview with a potential user of the system; a manager at a small supermarket branch.

Unfortunately, the interview didn't help us fully understand the procedures that go into managing a supermarket, but it was a good starting point in which we gathered information about the current system. to fully understand what goes into managing a supermarket we had to research, after researching and gathering information we found out that there is a big room for improvement specially in the way the supermarket manages stock, and how they handle sales records and other important data.

After the interview, research and brainstorming sessions we ended up with a lot of information about the topic, to make sure that the system we are designing is achievable and within our client needs we used an affinity map (*figure 3.1*) to cut down and organize the information we collected, this has helped us prioritize and understand the boundaries and functions of the system and it made the prototyping process easier.

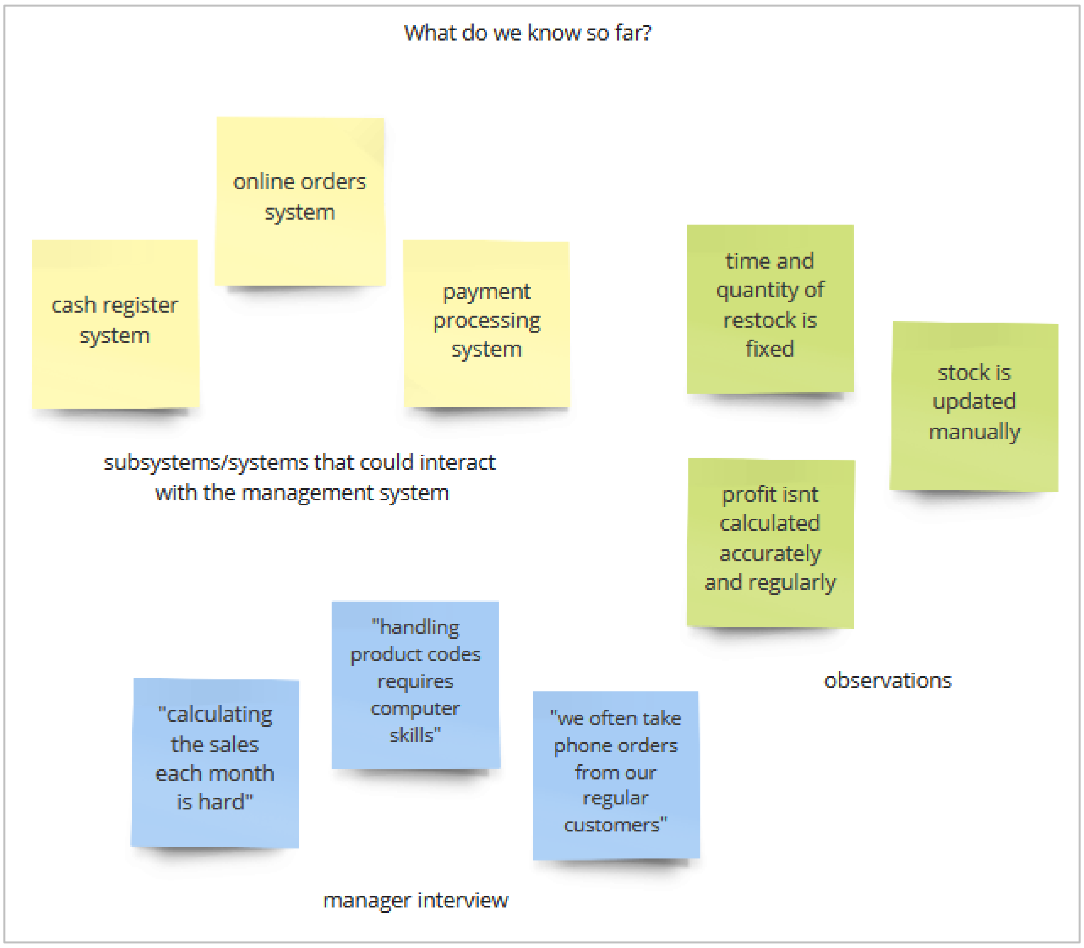
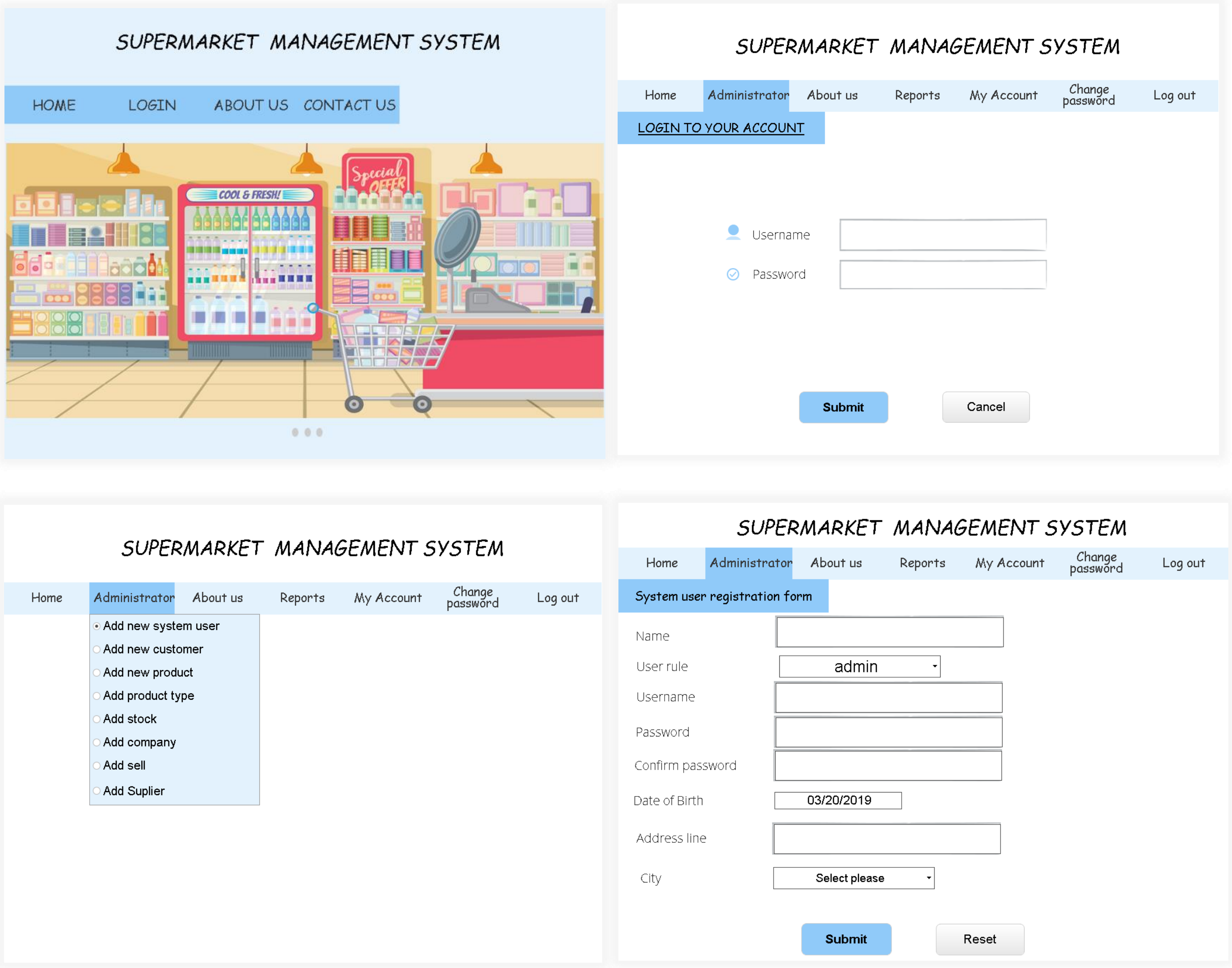

What do we know so far?
online orders system
cash register system
payment processing system
subsystems/systems that could interact with the management system
time and quantity of restock is fixed
stock is updated manually
profit isnt calculated accurately and regularly
observations
"handling product codes requires computer skills"
"calculating the sales each month is hard"
"we often take phone orders from our regular customers"
manager interview

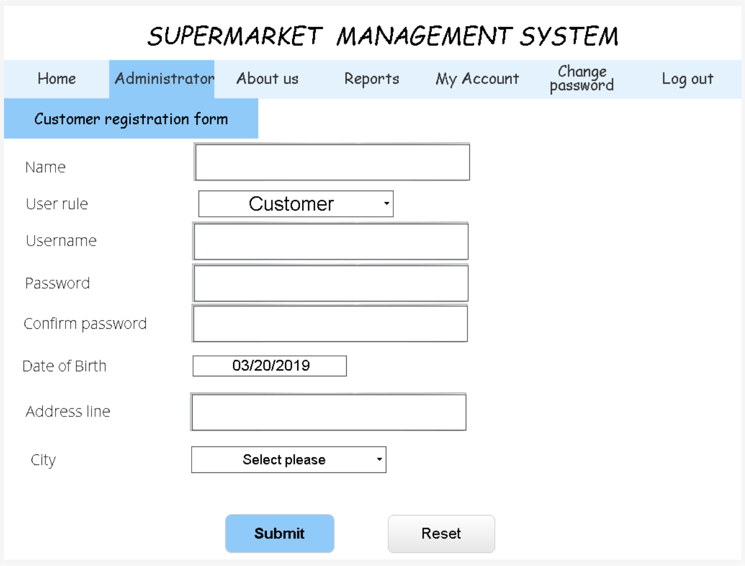
Figure 3.1 Affinty map of collected information

Now we have sufficient information about the system we started working on the prototype, during the prototyping process, our goal was to define the roles of the systems users and their operations, we designed three interfaces for three actors, administrator, manger, salesperson. Each of them accesses the system through a login page that protects the system from unauthorized access.



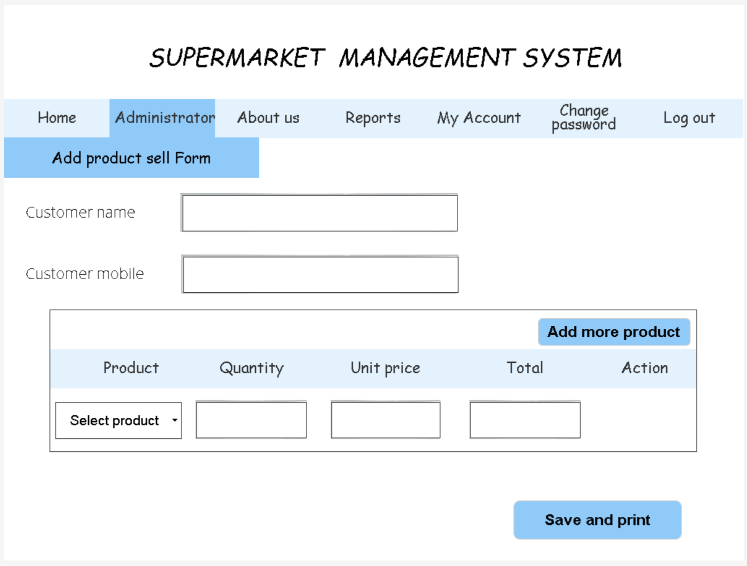
Name	
User rule	admin
Username	
Password	
Confirm password	
Date of Birth	03/20/2019
Address line	
City	Select please

Prototype screenshots


SUPERMARKET MANAGEMENT SYSTEM

Customer registration form

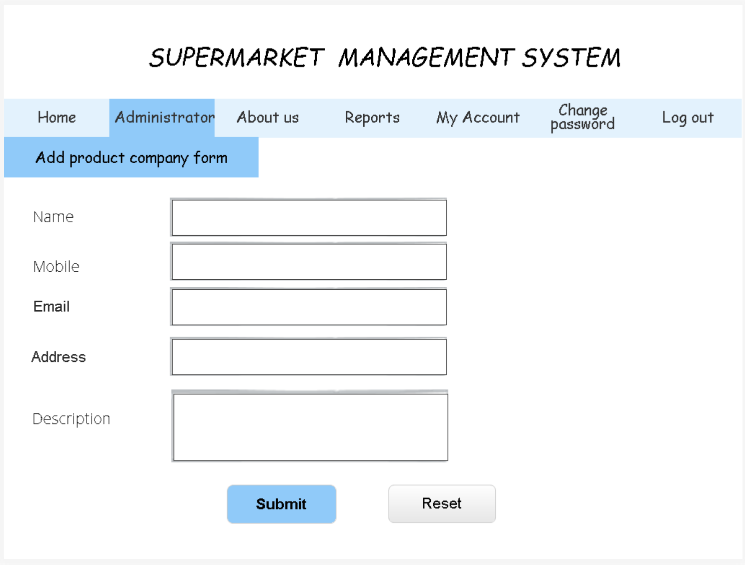
Name	
User rule	Customer
Username	
Password	
Confirm password	
Date of Birth	03/20/2019
Address line	
City	Select please


SUPERMARKET MANAGEMENT SYSTEM

Add product sell Form

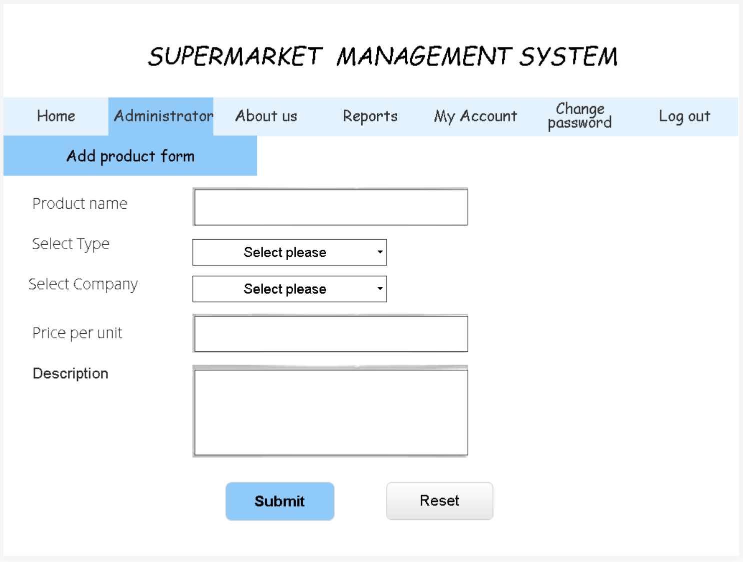
Customer name	
Customer mobile	

Product	Quantity	Unit price	Total	Action
Select product				


SUPERMARKET MANAGEMENT SYSTEM

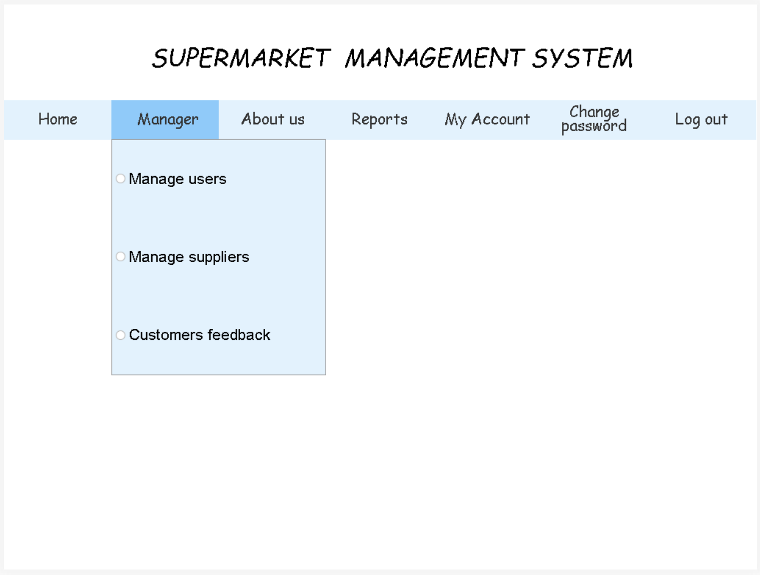
Add product company form

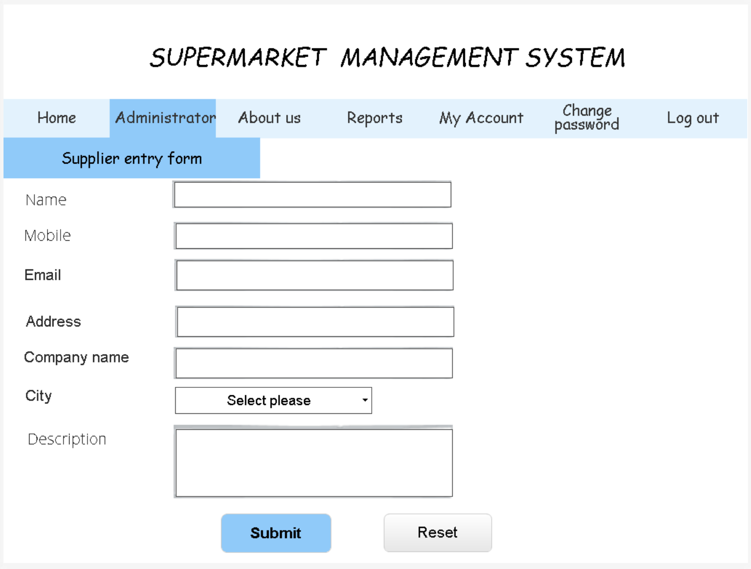
Name	
Mobile	
Email	
Address	
Description	


SUPERMARKET MANAGEMENT SYSTEM

Add product form

Product name	
Select Type	Select please
Select Company	Select please
Price per unit	
Description	


SUPERMARKET MANAGEMENT SYSTEM


SUPERMARKET MANAGEMENT SYSTEM

Supplier entry form

Name	
Mobile	
Email	
Address	
Company name	
City	Select please
Description	

Prototype screenshots

3.2 Functional requirements

1. ***The Administrator*** shall be able to create and manage (delete, edit) accounts for the different users of the system and grant account permissions based on the employee's required tasks.

2. ***The Administrator*** shall be able to print/export report documents (sales, suppliers, products) generated from the data stored in the system.

3. ***The Manager*** shall be able to register new suppliers into the system and have the ability to edit/remove them.

4. ***The Manager*** shall be able to add new products to the system.

5. ***The Manager*** shall be able to notify suppliers of stock status, and send restock requests.

6. ***The Manager*** shall be able to update stock when new shipments arrive by uploading shipment information.

7. ***The Salesperson*** shall be able to process purchases into the system, this includes new in-store purchases, and delivery orders.

8. ***The Salesperson*** shall be able to print an invoice for each purchase.

9. ***The Salesperson*** shall be able to add new customers to the system.

10. When a product is purchased, and an invoice has been printed stock should be automatically updated.

3.3 Nonfunctional requirements

- Usability: The system shall be easy to use and doesn’t require a lot of training when the supermarket hires new staff.

- Availability: The system should have the ability to operate no less than 18 hours a day.

- Supportability: The system should be developed in a way that it can be easily expanded to cover more than one branch in the future without modifications to the existing system.

- Compatibility: the system should be compatible with the other subsystems/devices (cash register device / payment processing system).

3.4 System Models

3.4.1 Functional model

A functional model is a structured representation of the functions, activities or processes within the modeled system.

Use case diagram is used to present the functional model, it shows the functional requirement for each user in the system as use cases and provide the relationships between actors, use cases and actors with use cases.

The actors in this system are admin, manager and sales person and the diagram show the user goals for each actor and the relationships between them.

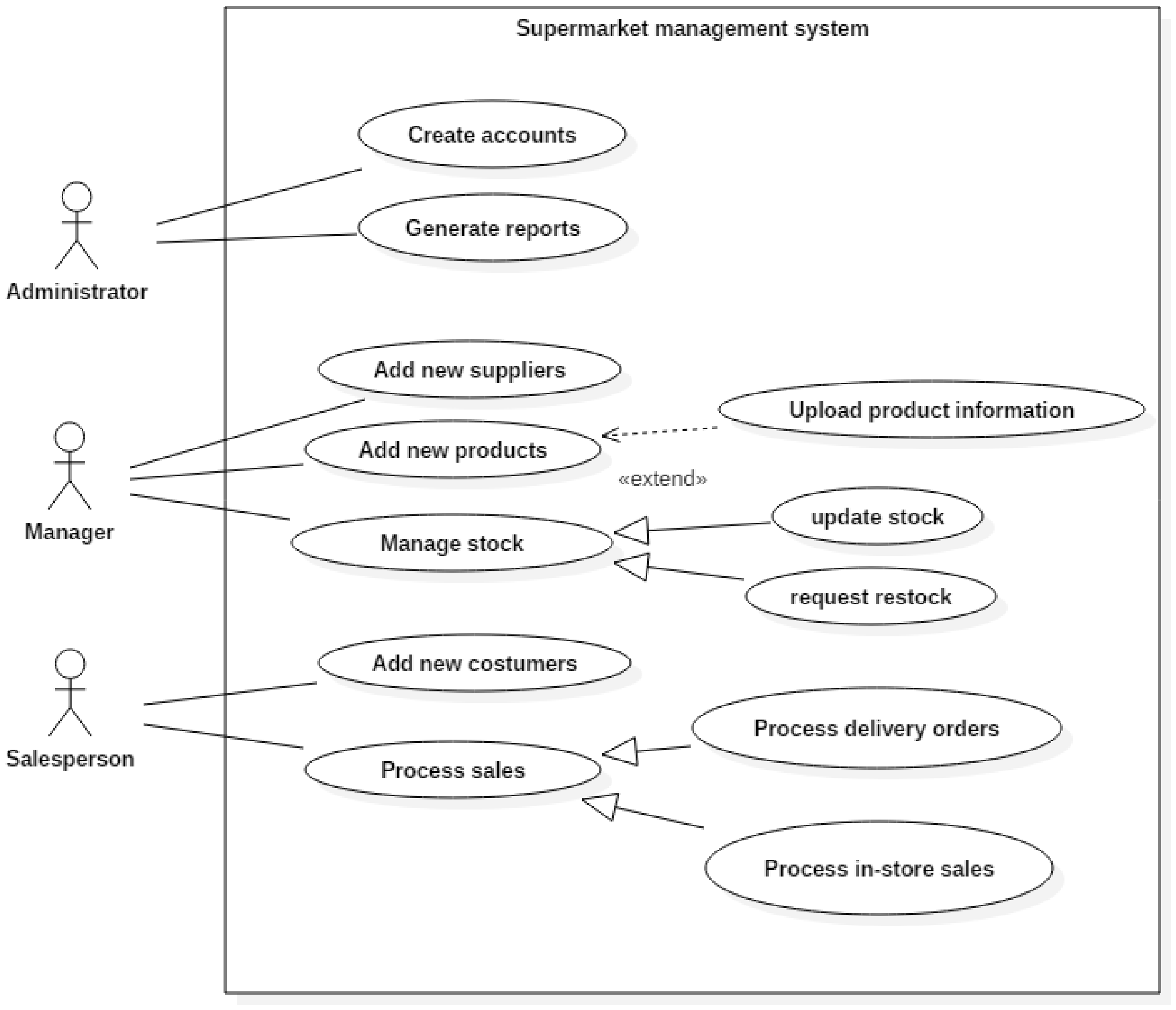


Figure 3.4.1.1 Use case diagram

3.4.1 Use case descriptions

UC_01	Create accounts
Actors	Administrator
Description	This use case allows the administrator to create system accounts for the users (employees), a user must have a unique username and password, users also have different permissions granted by the administrator based on their occupation.
Precondition	The administrator has an account, and have the new user(employee) information
Main Flow	1. The admin enters his username and password to log in to the system 2. The system verifies the admin login details 3. The system displays the admin control page 4. The admin chooses the operation "Add new user" 5. The admin enters user details (username, full name, email, occupation, phone number…) 6. The admin selects the permissions this user account can access 7. The system generates a password for the user 8. The system creates the user account and saves it to the database 9. The system sends the login information to the user' email
Post condition	the user account is created with the login information sent to his email
Alternative flow	2a. Invalid username or password 1. the system displays an "invalid username or password" message 2. The system asks for username and password 3. Use case resumes at main flow step 1

UC_02	Generate Reports
Actors	Administrator
Description	This use case allows the admin to generate report documents from the different types of data stored in the system' database.
Precondition	- the administrator is logged in - the database is not empty
Main Flow	1. Admin goes to the reports page 2. Admin chooses data to be included in the reports (purchase indent, sales report…) 3. Admin selects time period of data 4. Admin selects the operations to be included on the report (charts, calculations…) 5. Admin submits information 6. System generate the report document 7. System prompts the admin to choose between export or print 8. Admin enter choice 9. System comply
Postcondition	Report is either printed or exported
Alternative Flow	3a. there's no data for selected time period 1. The system prompts the admin to select a broader/different time period 2. Use case continue main flow step 4

UC_03	Add new supplier
Actors	manger
Description	Suppliers provide the supermarket with products to sell, this use case allows the manger to register new suppliers into the system.
Precondition	The manger is logged in and has supplier information ready to be entered
Main Flow	1. Manger selects to add a new supplier 2. System displays supplier entry form 3. Manger fills in supplier information (id, name, contact info…) 4. Manger submit entry. 5. System saves and update database.
Postcondition	New supplier is added, or existing supplier is updated
Alternative Flow	3a supplier already stored in database 1. System detects that supplier already stored in database 2. System display stored supplier information 3. The system prompts the manger to choose to override or create new supplier 4. Use case resumes at main flow step 4

UC_04	Add new product
Actors	manger
Description	This use case allows the manger to add new products to the system with specific type, company, supplier, dimensions and colors if available.
Precondition	Manger is logged in and has product information ready to be registered
Main Flow	1. Manger selects add product page 2. The system displays product entry form 3. [extension point: upload products info (UC_05)] 4. Manger fills in entry form 5. Manger submits entry 6. The system saves and update database
Postcondition	New product is added to the system
Alternative Flow	4a product is already stored in the database 1. System detects that product is already stored in database 2. System display stored product information 3. The system prompts the manger to choose to override or stock update existing product 4. Use case resumes at main flow step 5

UC_05	Upload product info
Actors	manger
Description	The manger can request suppliers to provide a product information documents after every shipment containing new products to the supermarket, this use case allows the manger to directly upload the multiple products information from the document to the system without having to manually fill out an entry form for each product.
Extends Use Cases	Add new product (UC_04)
Precondition	The manger is logged in and has a product information document to upload
Main Flow	1. The manger selects to upload products information 2. The system prompts the manger to upload the file containing the products info 3. The manger uploads the file 4. The system reads the file and sort the products info 5. The system updates the database with the new products info
Postcondition	New products are added to the database
Alternative Flow	4a. wrong format or missing fields 1. System detects that a file format is wrong or has missing data fields 2. System show error message 3. System prompts the manger to complete data field manually 4. Use case resumes at main flow step 5

UC_06	Mange stock
Actors	manger
Description	This use case allows the manger to handle the stock related operations such as updating the stock when new shipments arrive and request restock from supplier
Precondition	The manger is logged in and wants to manage stock
Main Flow	1. The manger chooses the stock control page 2. the system displays the stock control page 3. the system notifies the manger with low stock products 4. the system prompts the manger to choose to update stock or request restock 5. the manger selects operation 6. the system opens operation page
Postcondition	The manger completed a stock management operation

UC_07	Request restock
Actors	manger
Description	This use case allows the manger to request restock from supplier
Precondition	The manger has checked stock status
Main Flow	1. [Manage stock (UC_06)] 2. the manger selects a supplier to order restock shipment 3. the system retrieves supplier restock records from database 4. the stock management system analyzes stock record of products and calculates sufficient restock quantity 5. the system displays restock form to manger 6. the system prompts the manger to edit or confirm restock form 7. the manger submits entry 8. the system generates restock invoice from the form and send it to the supplier
Postcondition	The manger completed a restock request
Alternative Flow	4a manger wishes to fill restock form manually 1. the system displays empty restock form 2. the manger enters products id's 3. the manger enters products quantities 4. use case continue at main flow step 6

UC_08	Update stock
Actors	manger
Description	This use case allows the manger to update the stock after receiving new shipments
Precondition	The manger is logged in and wants to update stock
Main Flow	1. [Manage stock (UC_06)] 2. The manger selects to update stock 3. The system prompts the manger to upload shipment invoice 4. The manger uploads invoice 5. The manger confirms entry 6. The system updates the stock database
Postcondition	The manger completed a stock management operation
Alternative Flow	3a manger wishes to fill in restocked products manually 1. the system displays list of products 2. the manger selects products 3. the manger enters products quantities 4. use case continue at main flow step 5

UC_09	Add new customer
Actors	Salesperson
Description	This use case allows the salesperson to add new customers to the system.
Precondition	Salesperson has customers information ready to be registered
Main Flow	1. The salesperson login to the system 2. The salesperson selects "add new customer" 3. The system displays customer entry form 4. The salesperson fills in customer entry form 5. The salesperson submits customer information 6. The system saves the information to the database
Postcondition	A new customer is created
Alternative Flow	4a. customer already exists 1. The system detects customer is already registered in database. 2. The system prompts the salesperson to modify existing customer or create new customer. 3. Use case resumes at step 4.

UC_10	Process sale
Actors	Salesperson
Description	This use case allows the salesperson to process customers purchases from scanning the product to printing sales invoices.
Precondition	The customer has a cart of products and is ready to checkout.
Main Flow	1. The salesperson registers a new sale for the customer. 2. The system creates a new sale record. 3. The salesperson scans the product bar code. 4. The system retrieves product information (name, type, price). 5. The system adds the product to the sale record. 6. The Salesperson repeats steps 3 to 5 until all products are added to the sale record. 7. The salesperson closes the sale record. 8. The system calculates total price. 9. The system displays total price to salesperson
Postcondition	The customers' purchase is processed.
Alternative Flow	3a system doesn’t recognize product bar code 1. The salesperson enters product id 2. Use case continue at main flow step 4

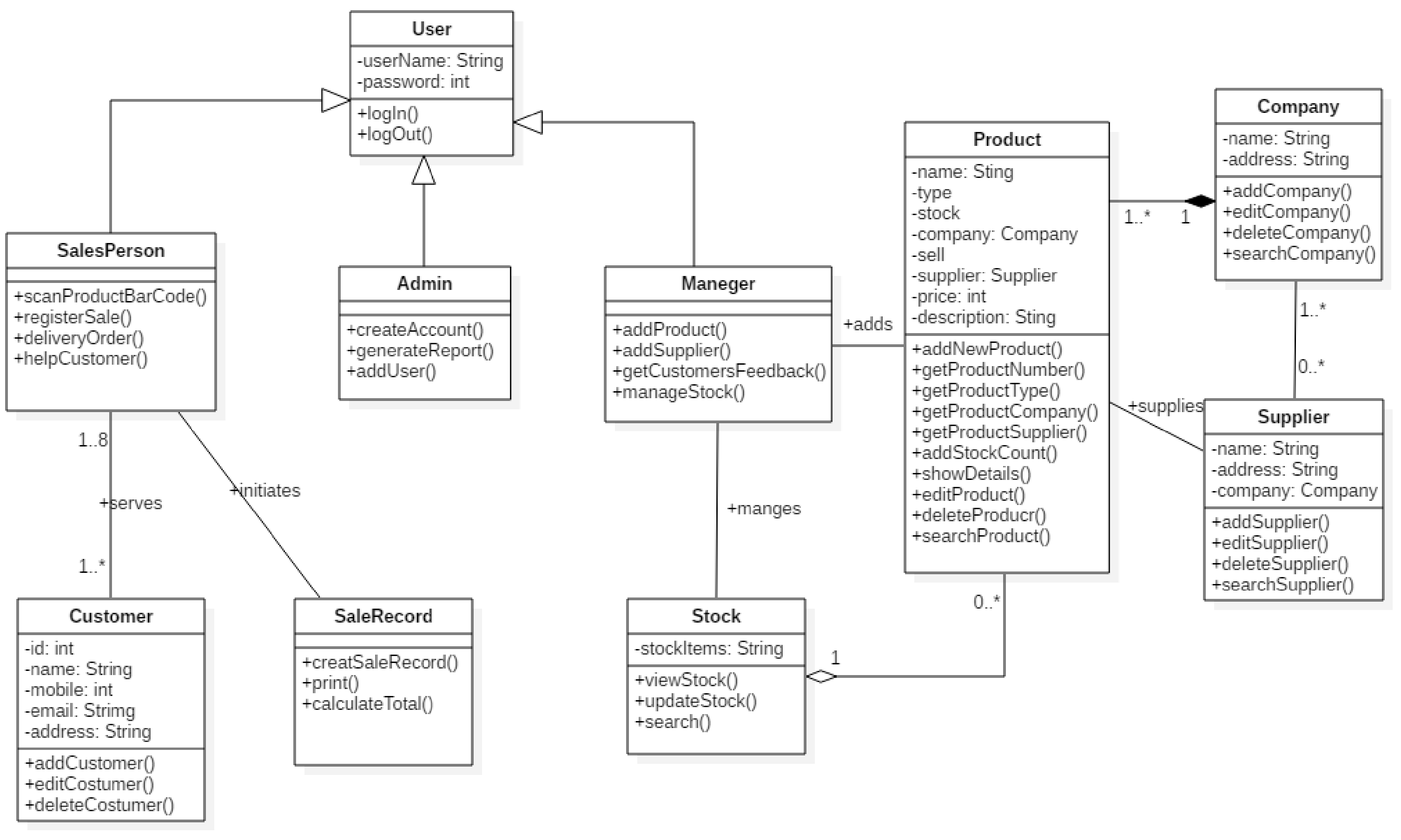
UC_11	Process in-store sales
Actors	Salesperson
Description	This use case allows the salesperson to process the customers sale after they finish shopping in the supermarket and are ready to check-out at the counter.
Precondition	- The customer has their goods at the counter - The salesperson is logged in and has the cash register open.
Main Flow	1. The salesperson Process a sale (use case: UC_545) 2. The salesperson displays total price to the customer. 3. The salesperson process customer' payment. 4. The system print invoice for the customer.
Postcondition	The customers' in-store purchase is registered into the system and an invoice has been printed.
Alternative Flow	3a. fail in processing payment 1. Payment system handles issue 2. The salesperson retries processing the payment and main flow resume at step 3.

UC_12	Process delivery orders
Actors	Primary actor: salesperson. Secondary actor: delivery agent.
Description	This use case allows the salesperson to process customers delivery orders.
Precondition	The customer has ordered products for delivery from the supermarket
Main Flow	1. The system retrieves customer details 2. The salesperson takes the customer order 3. The salesperson Process a sale (use case: UC_545). 4. The system adds customers address to the invoice 5. The system print invoice 6. The package is handed to the delivery agent.
Postcondition	The customer order is being handled by the delivery agent.
Alternative Flow	1a. customer details are not registered 1. The sales person triggers Add new customer use case (UC_333) 2. Main flow resumes at step 2
Comments	Delivery orders are taken by phone or by external system (website, app)

3.4.2 Object model

Class diagram is used to present the object model, it shows the overall structure of the system by providing the fundamental classes and objects within the system including the attributes and operations of each class and the relationships needed between them.

In this system we have User class which can be Sales Person class, Admin class or manager class. There's also Product class which is owned and created by the Company class and provided by Supplier. Also the System has Costumer class Stock class and Sale Record Class.

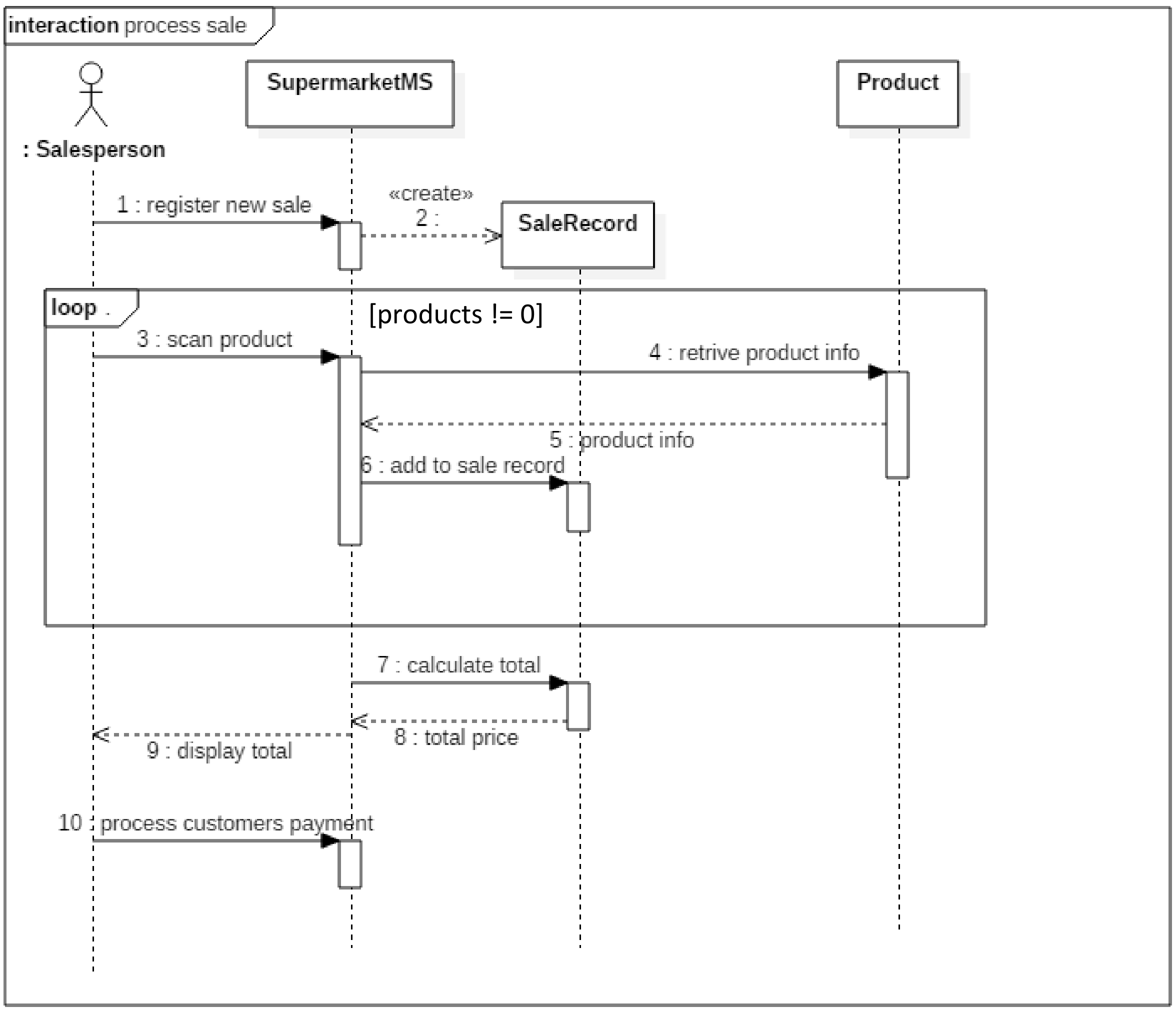


3.4.3 Dynamic model

Dynamic models are generally models that contain or depend upon an element of time, and represents the interactions between objects in the system over time.

Sequence diagram is used to present the dynamic model, it shows the interactions between objects within a use case with the actor and all the messages and fragments needed.

In this system process sale use case has been chosen in this sequence diagram which done by its actor the sales person with its participants, loop fragment is used to show the iteration within scanning and adding the product to the sales record.



this sequence diagram represents the request restock use case which done by its actor the manager with its main objects participants: system, database and supplier, and the messages needed for them to interact with each other.

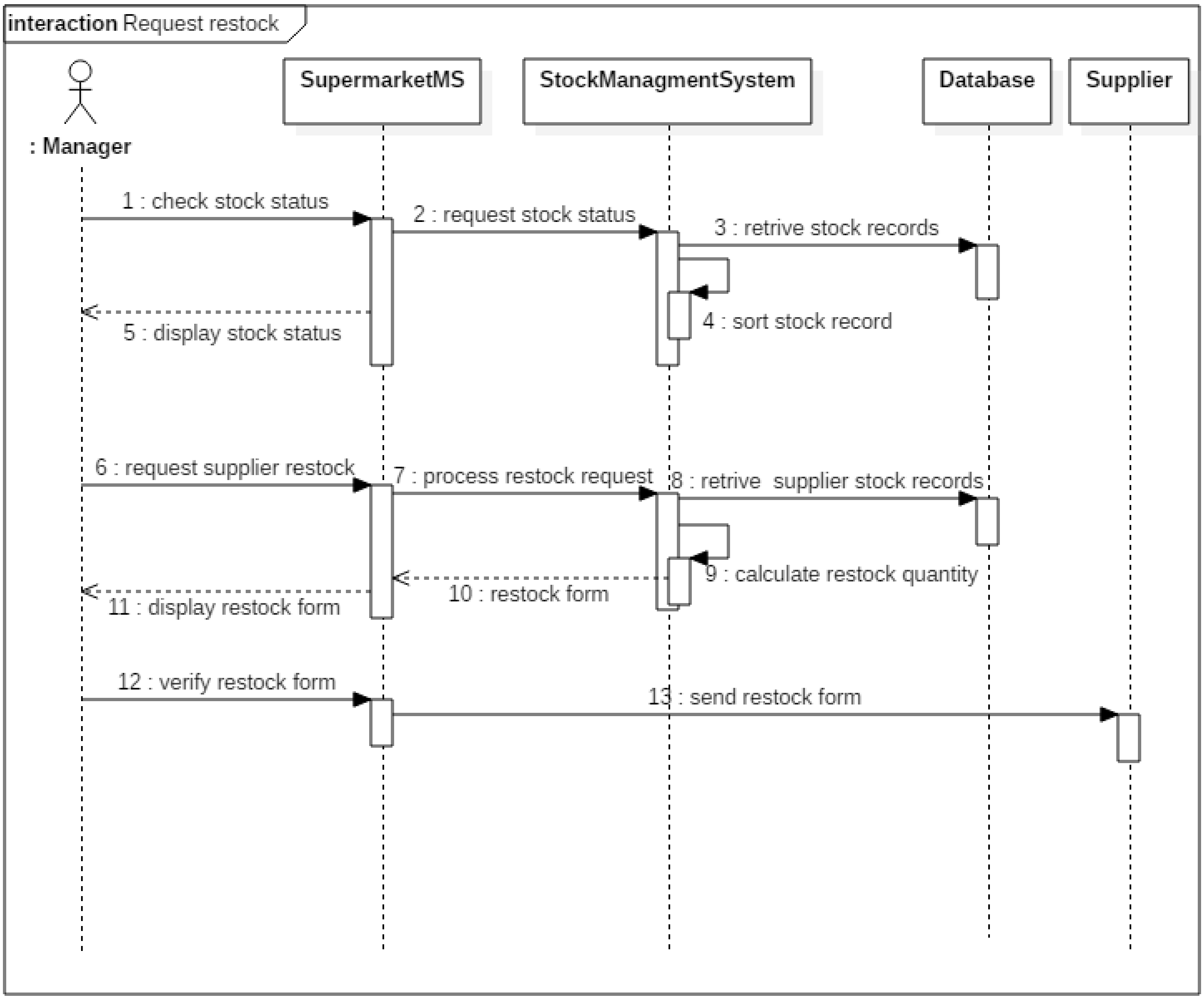


Figure 3.4.3.2 Sequence diagram for Request restock

State diagram is used to present the dynamic model, it shows all the possible states that an object can have in different use cases and the events that changes the state from one state to the other.

In this system this state diagram explains the states that a SaleRecord object go through while the salesperson process a customers' sale.

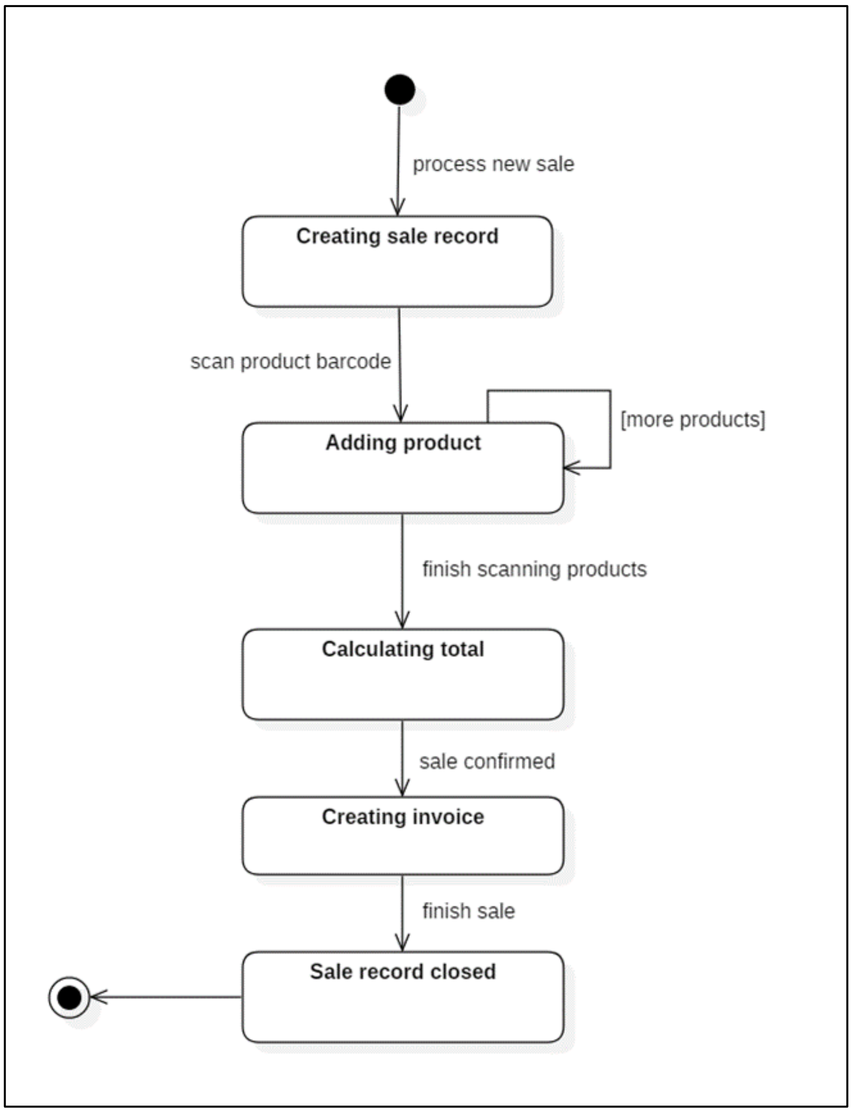


Figure 3.4.3.3 state chart diagram for SaleRecord

4. Conclusion

Supermarket Management System is a software developed and designed to reduce the time taken to handle the sales activity. It is designed to replace an existing manual system and will reduce manual work, calculations and will also provide periodic reports any time.

We learned a lot of new skills skimming the existing systems and building our project including the knowledge acquired from our courses content revision and from searching the Internet. We specifically learned how to make a prototype using its specified tools and how important it is in software engineering.

since working with a client is a new experience for us, we faced some challenges in communicating with them and deriving a satisfactory system solution for their problems, we also found it difficult to model the sequence diagram for the request restock use case since we don’t have a full view of how the operation is going to be implemented.

This project could be extended by including some additional users for the system with new functions, it also can be extended to fully support an online ordering system to serves the customers (besides in store) so they can buy products online at their homes and it will make the management easier and less reliant on staff.