



Adama Science & Technology University School of Electrical Engineering & Computing

Computer Science and Engineering Department

FSE project for 3rd year 1st semester

Yaleself supermarket management system

Section – 9

| No. | Name | ID |
|-----|----------------|--------------|
| 1. | Getaw Habtamu | UGR/17157/11 |
| 2. | Amin Kedir | UGR/16782/11 |
| 3. | Teklu Abebe | UGR/17704/11 |
| 4. | Serkalem Cheru | UGR/17642/11 |

Submitted to:
Instructor Nemerra

Acronym

SMMS ----- Supermarket management system

YSMMS ----- Yaleself supermarket management system

DB ----- Database

Table of Contents

| | |
|--|-----------|
| Chapter 1..... | 4 |
| 1.1. Introduction..... | 6 |
| 1.2. Statement of the problem | 6 |
| 1.3. Purpose of the project..... | 6 |
| 1.4. Objective..... | 6 |
| 1.4.1. General objective..... | 6 |
| 1.5. Feasibility study | 6 |
| 1.5.1. Economic feasibility | 7 |
| 1.5.2. Technical feasibility..... | 7 |
| 1.5.3. Operational feasibility..... | 7 |
| 1.6. Scope and limitation..... | 7 |
| 1.6.1. Scope of the study | 7 |
| 1.6.2. Limitation of the project | 8 |
| 1.7. Significance of the project | 8 |
| 1.8. Methodology | 8 |
| 1.9. Development tools | 9 |
| 1.10 Constraint and assumptions | 10 |
| 1.10. Test plan..... | 11 |
| 1.12 Overview of the project phase..... | 11 |
| 1.13 project plan..... | 11 |
| Chapter 2..... | 12 |
| 2. Description of existing system..... | 13 |
| 2.1. Major function of existing system..... | 13 |
| 2.2. Users of current system | 13 |
| 2.3. problem of current system | 13 |
| Chapter 3..... | 14 |
| 3. Proposed System..... | 14 |
| 3.1. Overview..... | 14 |
| 3.2. Functional requirement..... | 14 |
| 3.3. Non-functional requirement..... | 14 |
| 3.4. System model..... | 16 |
| 3.4.1. Scenario..... | 16 |
| 3.4.2. Use case model | 23 |
| 3.5. Object model..... | 26 |
| 3.5.1. Class diagram | 26 |

| | |
|--|-----------|
| 3.6. Dynamic model..... | 28 |
| 3.6.1. Sequence diagram..... | 28 |
| 3.6.2. Activity diagram | 40 |
| 3.6.3 State chart diagram..... | 54 |
| Chapter 4..... | 61 |
| 4. System design..... | 67 |
| 4.1. Overview of system design | 67 |
| 4.1.1. Purpose of the system design..... | 67 |
| 4.1.2 Design goal..... | 67 |
| 4.2. Proposed system architecture | 69 |
| 4.2.1. System process..... | 70 |
| 4.2.2. Subsystem decomposition..... | 72 |
| 4.2.3. Hardware/ software mapping..... | 73 |
| 4.2.4. Persistent data management..... | 74 |
| 4.2.5. Component diagram | 77 |
| 4.2.6 Deployment diagram | 78 |
| 4.2.7. Database design..... | 79 |
| 4.2.8. Access control | 80 |
| 4.2.9 User interface design | 81 |
| Appendix: software constraints | 82 |

Chapter One

Introduction

With the rapid development of modern science and technology, computer technology has penetrated into all fields and becomes the necessary tools for various industries, especially the Internet technology promotion and the establishment of the information highway. It makes the IT industry increasingly shows its unique advantages in the market competition. Into the digital age, there is huge data information waiting for processing and transmission, which makes the further development and use of the database is particularly urgent. As some small and medium-sized supermarkets in the domestic market, they are falling behind the large and medium-sized supermarkets during the information, but for these enterprises' resource management, information storage and processing also shows the urgent need. To adapt to market competition, it requires efficient handling and management methods, so it is indispensable that accelerate the process of the computerization of supermarket. Small and medium size supermarkets has an important position in Chinese economic development. At present, our country has a lot of small and medium-sized supermarkets, regions are widely distributed, and the information level of small and medium-sized supermarkets is still very low. With the development of the technology, the computer operation and management is increasingly simplified, computer knowledge is increasingly popularization. At the same time, the fast changing of market economy and intense competition, it is must become inevitable trend that the supermarket industry uses computer to manage inventory, sales, and many other links. As a small and medium size supermarkets, their main business is selling products. But at present a lot of questions exist in the system operation, all sales orders are artificial fill, time-consuming and prone to errors; inventory is artificial bookkeeping and unable to keep track of the most accurate inventory situation; For sales data, it often spends a lot of time and energy to calculate the sales performance of each business membership and each good sale. In order to solve the above problems and improve the economic benefit, we will prepare for computerized management for the supermarket.

General Situation of Small and Medium-sized Supermarket

With the increasingly fierce competition, how to reduce the cost has become the supermarket's vital problem. For ordinary supermarkets, involving the management of the sources of raw materials, sales and inventory, and the good or bad management is very important to the durability of the supermarket. Generally speaking, the user's demand for procurement, sales and inventory system is universal. Supermarket management system used in the supermarket's procurement, sales and warehouse department, controlling and tracking the whole business of supermarket procurement, sales and warehouse effectively. Using the supermarket's procurement, sales and inventory management system can effectively reduce the blind procurement, reduce the costs of procurement, reasonable control inventory, increase market sensitivity, and enhance the market competitiveness of the supermarkets. Compared with the domestic and foreign large-scale supermarket, small and medium size supermarkets have obvious difference in the operation and management. And in order to design the management software which conforms to the small and medium-sized supermarket, we will understand some characteristic of small and medium sized supermarket management. The function of management system which Small and medium-sized supermarkets need is not as comprehensive as big supermarkets and chain supermarkets, it puts forward higher requirements on concise and practical. Features of small and medium-sized supermarkets:

- i. The size is relatively small, and may not have their own warehouse, inventory backlog is less;
- ii. The employee is less and a person might have multiple positions, and has low quality skills. So, this requires a system to have perfect function of help and check, but also requires the system easy to operate, concise and clear;
- iii. The intervals of clearing inventory are uncertain, may be a long time hasn't inventory, may be at any time.

1.2 Statement of the problem

How can we solve supermarket should facilitate to the customers?

1.3 Purpose of the project:

The main purpose of this project is to solve supermarket management system in Ethiopia.

1.4 Objective

The objectives are: -

1. To study the function of supermarket management system.
2. To explore the challenges being faced by manual system.
3. To make software fast in processing with good user interface.
4. To insure accurate statistics of the product item.
5. For easy record of goods in store and proper identification.

1.4.1 General objective

The **general objective** of this project is to modernize the supermarket management system that is currently user in most companies by making the inventory controlled over on online. And our project also simplifies the work of customers by showing them available goods and they can also order so that the workers can bring all the stuffs they order.

1.5 Feasibility Study

Feasibility study is essential to evaluate the cost and benefits of the new system. On the basis of the feasibility study, decision will be taken on whether to proceed or to cancel the project. It includes the following.

1.5.1 Economic feasibility

For any system if the expected benefits equal or exceed the expected costs, the system can be judged to be economically feasible. In economic feasibility, cost benefit analysis is done in which expected costs and benefits are evaluated. Economic analysis is used for evaluating the effectiveness of the proposed system.

| | |
|---|---|
| <ul style="list-style-type: none"> - On average a person can waste 40 minutes a week in supermarket. - If a person buy 100 birr product $100 * 100$ people = 10000 birr | <ul style="list-style-type: none"> - When using our system this will decrease by half. So, supermarket can increase their daily users by 2x. - If a person buys 100 birr product $100 * 200$ people = 20000 birr - Our product saves 10000 birr per day. |
|---|---|

1.5.2 Technical feasibility

The entire group members expect the system to be technically feasible. The platform is going to be developed by following the Object-Oriented System Development technique. The team has the ability to develop this system without any difficulty since the team has studied the required methodologies and tools. So, the system will be technically feasible. The project members also have knowledge about MySQL to design the database of the system. Besides, the group members have enough capability to develop the project. So, the system is technically feasible. In general, using the different techniques, we can perform the work up to the desire of our customers.

1.5.3 Operational feasibility

We have performed a series of steps to solve business problems and user requirements. The system has definitely a positive impact for both the users and the supermarket owner. We follow each step how the existing systems are working and our technical skills transform the existing system into a better and improved one. The new/proposed system will be compatible with almost any personal computers.

1.6 Scope and Limitations

1.6.1 Scope

This research work covers stock control, management and tends to correct anomalies in supermarket business. Its analysis opening of new stocks, stock updates and ability to view exiting one. It provides quick way operation by capturing the manual processes and automating them. This project helpful to computerize item transaction, sales activity record keeping which is a very huge task and maintain the stock.

Features:

- Users can purchase, rate and give a comment about supermarket.
- The platform lets users make a payment through mobile banking.
- The platform provides a purchasing history for the user.
- User's information will be stored in a secure way.

1.6.2 Limitation

Due to shortage of time and basic factor like covid19 and many more this research has been limited to certain area in supermarket management and we couldn't gather information from supermarkets in person.

1.7 Significance of the project

The significance of our project is to manage supermarket efficiently by reducing unwanted costs like over buying of products and the customer may order products that are out of stock at that time.

1.8. Methodology**1.8.1 Data Source**

We gathered information from Ethiopian online supermarket systems like safe-way supermarket.

1.8.2 Fact finding methodologies

Observation: For last months we have seen many websites and the systems work in our country. We have visited a website called safe-way supermarket system.

Interview: Some of our group members have faced this problem and some other people faced this problem before.

1.8.3 System Development Methodologies

In the system development methodology of a project, we have used the component-based approach in which all system processes are placed into separate components so that all of the data and functions inside each component are semantically related. Because of this principle, it is often said that components are modular and cohesive.

1.9 Development Tools

Hardware and Software tools

1.9.1 Software tools

Client side

| Tools | Use |
|------------|-----------------------------------|
| Java swing | GUI toolkit for java |
| Java | language used for system back end |

Server-side tools

| Tools | Use |
|-----------|--------------------------------|
| JDBC | Database connectivity for java |
| MySQL 5.6 | For Back-End database server |

Other tools

| Tools | Use |
|--------------------|---|
| MS- Word 2019 | For documentation |
| LibreOffice Writer | For documentation |
| IntelliJ | For developing the software or writing the code |
| NetBeans | For developing the software or writing the code |
| Chrome | For browsing document in an output form |
| Telegram | To communicate |

1.9.2 Hardware tools

- **Personal Computer or laptop:** almost all tasks of the project are performed using computer.
- **Flash:** required for data transfer.
- **Note book and pen:** to collect data and to take notes from discussions.

1.10 Constraints and Assumptions

1.10.1 Constraints

- If internet connection is not available, the user might not get what he/she wants.
- If the users shouldn't have personal computer, they might not access our software.
- No integration with banks because of banks in Ethiopia doesn't has a public API.
- need approval of admin or privileged stuff for the payment

1.10.2 Assumptions

- Customer will save his/her time and money.
- Customer will be secure and be safe from robbery and other similar activities.
- The app serves from small sized supermarket to medium sized supermarket.
- Customer can order goods at home by using our system.

Table access levels

| Access level | Description |
|-------------------|---|
| customer | It is assumed to get detailed information, order goods, make payment. |
| Supermarket Owner | It is assumed that it has the access to control all stuffs, assign privilege for each stuff. |
| Admin | It is the main component of the system and it has control over all supermarket and the number of customers using the software to reach out the supermarket. |
| cashier | It is assumed to do the bills and receipts. |

1.11 Testing Procedure

1.11.1 Unit Testing

In this phase we try to eliminate faults in procedure and function point of view by using black box and white box testing.

White box testing

We will use this approach to know the internal working style of the system, test that all internal operations are performed according to specifications and all internal components have been exercised and the logical path of the system are correct.

Black box testing

In this technique, we will test to see if the function of the system to fully operational or error free. This includes, testing the interface of the system rather than the logical structure of the system.

1.11.2 Integration testing

In this phase we have focused on how individual software module are combined and tested as a group. It occurs after unit testing.

Tasks that we have performed under this phase:

- Prepare integration test plans
- Identify integration test objective
- Identify integration test acceptance criteria
- Perform the test according to our plan
- Document the test results

1.12 Overview of the project phase

A system development methodology refers to the framework that is used to plan and control the process of the developing the system. we can use different methodologies to develop our system. but we choose waterfall method because:

- it is simple to develop and easy to understand
 - Our system's requirement is known
-

1.13 Project Plan

Team Composition Table

| supermarket management system portal | | | | |
|---|----------------|---------------|--------------------------|--|
| No | Name | ID | Email | Role |
| 1 | Teklu Abebe | UGR /17704/11 | Tekluabebe0962@gmail.com | Requirement gathering, Designing, analysis, implementation and testing |
| 2 | Amin kendir | UGR /16782/11 | Aminkendir41@gmail.com | Coordinating, Requirement gathering, Designing, analysis, implementation and testing |
| 3 | Getaw Habtamu | UGR/17157/11 | Getawh@gmail.com | Coordinating, analysis, implementation and testing |
| 4 | Serkalem cheru | UGR /17642/11 | Tsehayserki@gmail.com | Coordinating, Requirement gathering, analysis, implementation and testing |

Chapter Two

Description of the Existing System

2.1 Major function of the current system

The current system operates manual supermarket management system, from stocks, products, ordering and purchases etc. recorded in a book. This is faced with errors, incompleteness, and insufficient data for analysis. Information regarding stocks, products, sales and purchases are not properly organized and managed. From the wholesalers to retailer bills, tickets, receipts of products are recorded in a book but further operations are not being properly handled. As a result, it is difficult in processing, updating and managing.

Way of payment

The payment is using mobile banking system.

2.2 User of the Current System

1. Admin
 - Adds/registers customers and manage them.
2. customer
 - Chooses the goods from listed options according to customer's want, efficiency and accommodations.
3. Supermarket owner
 - Provides privilege for the employees and admins.
4. Cashier
 - Calculates and prints bills.

2.3 Problem of the Current System

Labor-Intensive: A manual Super Market management systems is that they can be highly labor-intensive to operate. They require continuous monitoring to ensure that each transaction is accounted for and that products are maintained at the appropriate stocking levels. It is also more difficult to share inventory information throughout the business, because the lack of computerization makes accessing inventory records a more difficult process.

Human Error: A manual Supermarket management system relies heavily on the actions of people, which increases the possibility of human error. People might forget to record a transaction or simply miscount the number of goods. This results in needless additional orders that increase the company's inventory carrying costs and use up precious storage space. Inaccurate physical counts could also result in not ordering enough of a product, meaning the business could run out of a crucial item at the wrong time.

Time Wasting: A manual Supermarket management system has a huge tendency of time wasting as the sales manager could have a lot to tackle while many customers seek attention and this is really affecting the business.

2.4 Software Process Model

For our project, we used waterfall model.

2.4.1 The merit of the model in our project

This process model is easy to understand and implement. so, our project will not be complex.

CHAPTER 3

Proposed System

3.1 Overview

On today's world the usage of computerized system is increasing rapidly and the manual system is becoming rarer and rarer. in addition to that manual systems are becoming more time killing tasks. Therefore, we must go side to side with the technologies being created currently. And apart this the cost of places or spaces is becoming more expensive. So, we cannot store lots of products at a time. Therefore, we must think how we can manage our store efficiently to have balanced items.

The system we are going to develop simplifies the problems listed above.

3.2 Requirements of the proposed system

The functional requirement describes the core functionality of the system. This includes the data and functional process requirements.

3.2.1 Functional requirements

Process requirements describe what the application must do. Process requirements relate the entities and attributes from the data requirements to the users' needs. State the functional requirements in a manner that enables the reader to see broad concepts decomposed into layers of increasing detail.

- ✓ Keeping records of registration of customers.
- ✓ Keeping the records of products.
- ✓ Keeping the daily sell information.
- ✓ Storing the feedback given by the customers.
- ✓ Keeping details about the product it is delivered or not etc.
- ✓ Storing the items selected by the customer in the temporary storage /wish list.

3.2.2 Non-Functional Requirement

The non-functional requirements are a requirement which defines how the system supposed to do. They are not essential for the system but give support and quality to the system. They are more critical than individual requirement. So, the specify the following major emergent properties of supermarket management system:

✓ **Usability**

The system should be easy to learn and understandable for the user. Since the system will deliver to all low level and medium level supermarket, it must be easy to use and learn. In addition, it will have a user manual that tells the user how to use the system and we will give an enough training for the employees concerning on how to operate the system.

✓ **Reliability**

The system should be highly reliable. Since the system will be best in handling errors and display an appropriate error message in order to minimize system failures. as well as if error happen in the system the system can restart it.

✓ **Performance**

The system should need small space and respond within a short period of time. It depends on the performance of the hardware environment and employee who work at that supermarket.

✓ **Accessibility**

The system should be accessible at any time since it is needed in every time.

✓ **Maintainability**

The system should be maintainable. Because the interaction between subsystems will be loosely couple and the interaction between classes and operations will be highly cohered, changes made on our system such as adding other functionality shouldn't affect the existing functionality of the system.

✓ **Documentation**

At the completion of the project, every activity of the entire development, design and other process will be documented for future reference. There will also be a documentation of implementation for maintenance during application failure. Furthermore, this will help for further maintenance and reusability of our system.

✓ **Security**

These files which are going to be registered in the system have to be secured and must be kept in a secured manner. To satisfy these, the system will provide authorization level according to their managing level and restrict unauthorized access to these files.

3.3 System Model

3.3.1 Scenarios

Scenario 1

Name of scenario: LOGIN

Participating instance actors: all user

Entry condition:

- Internet connection has to be available.
- The user has a valid username/password.

Flow of events:

1. Users open the system.
2. Login form will be displayed.
3. Users fill username and password.
4. User clicks **LOGIN**.
5. Homepage will be displayed.

Exceptional flow: If the user enters invalid email and password, the system displays a message to enter correct email and password.

scenario 2

Name of scenario: REGISTER EMPLOYEE

Participating actors: Owner

Entry condition:

- ✓ Internet connection should be available.
- ✓ The Owner must have a valid pre-registered username and password when the software is developed.

Flow of event:

1. Owner Login in to the system.
2. Owner clicks **REGISTER EMPLOYEE** button
3. Owner fills required and optional fields
4. Owner clicks **REGISTER** button
5. Success message will be displayed

Exceptional flow: if the Owner enters wrong data, the system displays a message to enter correct data.

Scenario 3

Name of scenario: ASSIGN ROLE

Participating actors: Owner

Entry condition:

- ✓ Internet connection should be available.
- ✓ The Owner must have a valid pre-registered username and password when the software is developed.

Flow of event:

1. Owner Login in to the system.
2. Owner clicks **ASSIGN ROLE** button
3. Owner chooses Employee to assign role
4. Owner chooses the role to be assigned
5. Owner clicks **ASSIGN ROLE** button
6. Success message will be displayed

Scenario 4

Name of scenario: ADD PRODUCT

Participating actors: Owner

Entry condition:

- ✓ Internet connection should be available.
- ✓ The Owner must have a valid pre-registered username and password when the software is developed.

Flow of event:

1. Owner Login in to the system.
2. Owner clicks **ADD PRODUCT** button
3. Owner chooses products to be bought
4. Owner clicks **ORDER**
5. Success message will be displayed
6. The system sends email for product supplier to bring products

Scenario 5

Name of scenario: REMOVE PRODUCT

Participating actors: Owner

Entry condition:

- ✓ Internet connection should be available.
- ✓ The Owner must have a valid pre-registered username and password when the software is developed.

Flow of event:

1. Owner Login in to the system.
2. Owner clicks **REMOVE PRODUCT** button
3. Owner chooses products to be removed
4. Owner clicks **REMOVE**
5. Success message will be displayed

Scenario 6

Name of scenario: REGISTER CUSTOMER

Participating actors: Admin

Entry condition:

- ✓ Internet connection should be available.
- ✓ The admin must have a valid username/password and must login with its own user name and password which recognize by the system

Flow of event:

1. Admin Login in to the system.
2. Admin clicks **REGISTER CUSTOMER** button
3. Admin fills required and optional fields
4. Admin clicks **REGISTER** button
5. Success message will be displayed

Exceptional flow: if the Admin enters wrong data, the system displays a message to enter correct data.

Scenario 7

Name of scenario: REMOVE CUSTOMER

Participating actors: Admin

Entry condition:

- ✓ Internet connection should be available.
- ✓ The admin must have a valid username/password and must login with its own user name and password which recognize by the system

Flow of event:

1. Admin Login in to the system.
2. Admin clicks **REMOVE CUSTOMER** button
3. Admin chooses account to be deleted
4. Admin clicks **REMOVE** button
5. Success message will be displayed

Scenario 8

Name of scenario: GIVE COMMENT

Participating instance actors: Customers

Entry condition:

- Internet connection has to be available.
- The user has a valid username/password and must login with its own user name and password.

Flow of events:

1. Users Login in to the system.
2. Customer clicks **VIEW PRODUCT** button.
3. The system displays available goods in the system.
4. The user can see detail about goods by click **DETAIL** button.
5. Customer clicks **GIVE COMMENT** button.
6. Comment box will be displayed.
7. Customer gives a comment and click **COMMENT** button.

Exceptional flow: If the user enters wrong comment, click **EDIT** to reenter the comment.

Scenario 9

Name of scenario: BUY PRODUCT

Participating instance actors: customer

Entry condition:

- Internet connection has to be available.
- The user has a valid username/password and must login with its own user name and password.

Flow of events:

1. Customers Login in to the system.
2. Customer clicks **VIEW PRODUCT** button.
3. The system displays available goods in the system.
4. Customer clicks **ADD TO WISHLIST** button
5. customer chooses type and amount of product they want to buy.
6. Customer clicks **BUY** button.
7. The system asks you to transfer money to supermarket's bank account
8. After depositing money Customer clicks **REQUEST APPROVAL** button.
9. Admin confirms the customer's deposit based on its account number and approves the payment
10. Success message will be displayed.

Exceptional flow: if the customer doesn't deposit the money, 'deposit failed' message will be displayed.

Scenario 9

Name of scenario: EDIT PROFILE

Participating actors: All users

Entry condition:

- ✓ Internet connection should be available.

Flow of events:

1. user Login in to the system.
2. User's detail displayed at this page with **EDIT PROFILE** option.
3. If user wants to edit his/her information, then click **EDIT PROFILE** button.
4. Edit incorrect information and click **SUBMIT** button
5. Success message will be displayed

Exceptional flow: if the user enters wrong data, the system displays a message to enter correct data.

Scenario 10

Name of scenario: VIEW PRODUCT

Participating instance actors: all users

Entry condition:

- Internet connection has to be available.
- The user has a valid username/password and must login with its own user name and password.

Flow of events:

1. Users Login in to the system.
2. User clicks **VIEW PRODUCT** button.
3. The system displays available goods in the inventory.

Exceptional flow: If the user enters invalid email and password, the system displays a message to enter correct email and password.

Scenario 11

Name of scenario: VIEW COMMENT

Participating instance actors: all actors

Entry condition:

- Internet connection has to be available.
- The user has a valid username/password and must login with its own user name and password.

Flow of events:

1. Users Login in to the system.
2. User clicks **VIEW PRODUCTS** button
3. The system displays available goods in the system.
4. The user can see detail about goods by clicking **DETAIL** button.
5. The actor can look the given comment at the bottom of the page.

Exceptional flow: If the user gave wrong comment, admin can delete by clicking **DELETE** button.

Scenario 12

Name of scenario: CREATE BILL

Participating instance actors: cashier

Entry condition:

- Internet connection has to be available.
- The user has a valid username/password and must login with its own user name and password.

Flow of events:

6. Users Login in to the system.
7. User clicks **CREATE BILL** button.
8. The system displays recent transactions.
9. Cashier selects transaction and clicks **PRINT BILL**.

Exceptional flow: If the user enters invalid email and password, the system displays a message to enter correct email and password.

3.4 Use Case Model

A use case model is a useful tool for requirements elicitation. It provides graphical representation of the software systems requirements. The key elements in a use case model are actors (external entities), and the use cases themselves. Because use case models are simple both in concept and appearance, it is relatively easy to discuss the correctness of a use case model with anyone.

The UML represents a deliberate attempt to standardize the modeling notation used in software engineering, particularly object-oriented development.

Generally, as it describes the goals of users, interaction between actors and the required behavior of the system and the behavior of the system in satisfying these goals.

A use-case model consists of a number of model elements. The most important model elements are: use cases, actors and the relationships between them.

3.4.1 Actor Identification

An Actor is a being which accomplishes use cases upon a system. It is just an entity, meaning it can be a human or may be other system that plays an external role in the system.

Name: **CUSTOMER**

Description of the name: customer is a human being may be individual or group.

Role on the system: login, update his profile, view product, purchase product, give comment, view comment.

Name: **ADMIN**

Description of the name: admin is an individual human being

Role on the system: Login, register customer, remove customer, update profile.

Name: **OWNER**

Description of the name: supermarket owner is an individual human being

Role on the system: Login, register employees, assign role, add products, remove products, update profile.

Name: **CASHIER**

Description of the name: cashier is an individual human being.

Role on the system: Login, Update profile, Create bill

3.4.2 Use Case Diagram

A use case diagram illustrates a set of use cases for a system, the actors of these use cases, the relations among the actors and these use cases, and the relations among the use cases.

The UML notation for a use case diagram is in which

- ✓ An Oval represents **use case**.
- ✓ Stick figure represents **actor**.
- ✓ Line between an actor and a use case represents that the actor initiates and/or participates in the processor simply it **shows connection**.

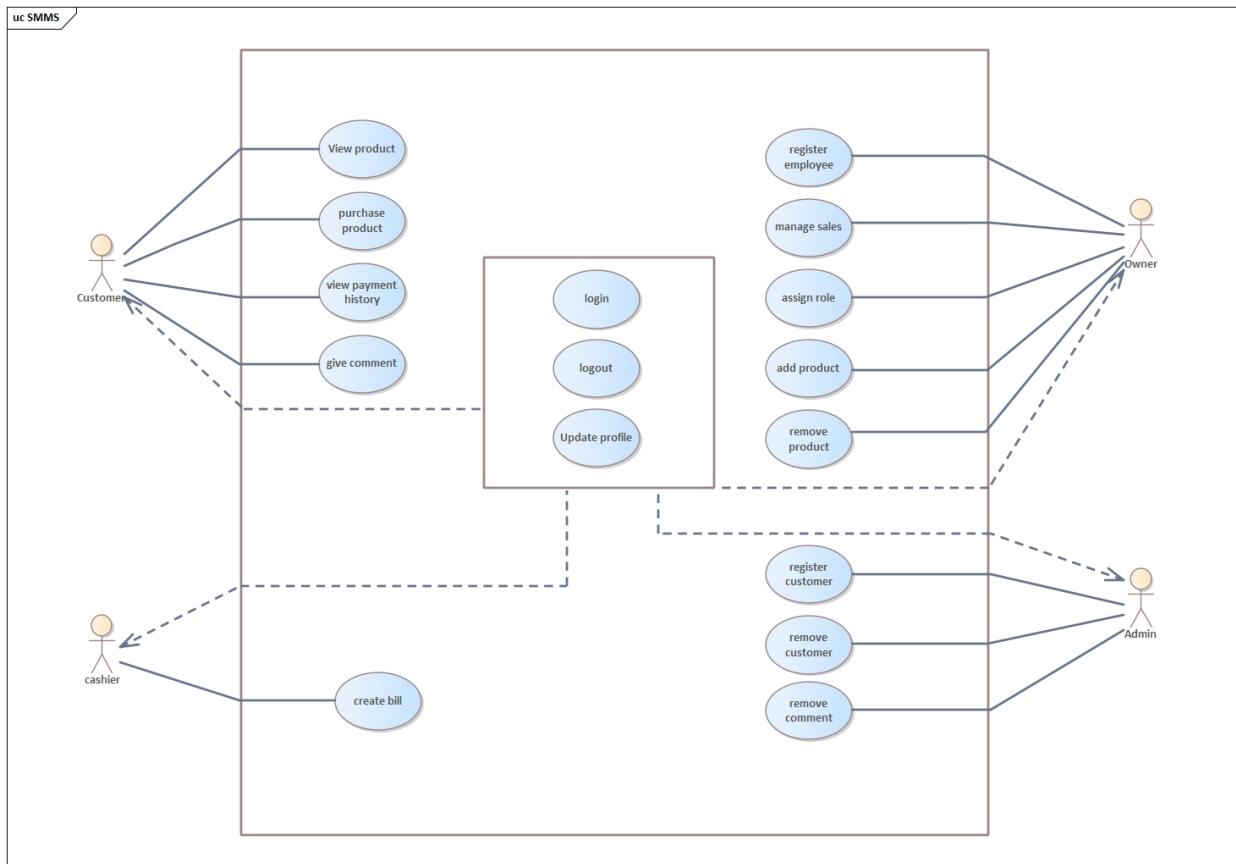


Figure 3.4.2.1 use case diagram of supermarket management system

3.7 Object Model

3.7.1 Class Diagram

The Class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling from the structure of the application, and for detailed modeling translating the models into programming code. It is a static model, describing what exists and what attributes and behavior it has rather than how something is done.

The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

Generally, it shows the classes of the system interaction which are typically used to

- ✓ Explore domain concept
- ✓ Analyze requirement in the form of conceptual analysis model

In the class diagram, classes are represented with boxes that contain three compartments:

- The top compartment contains the name of the class. It is printed in bold and centered, and the first letter is capitalized.
- The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase.
- The bottom compartment contains the operations the class can execute. They are left-aligned and the first letter is lowercase.

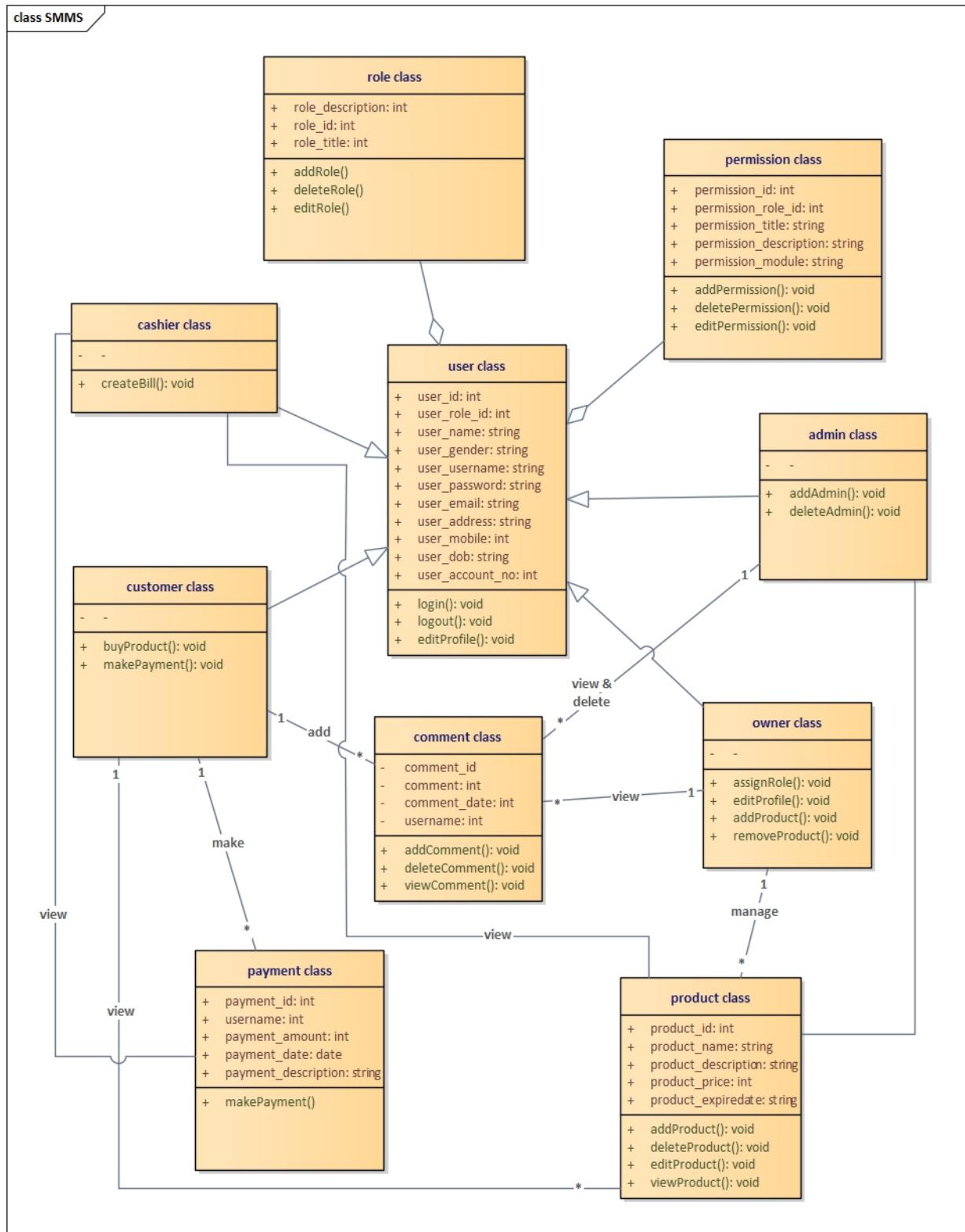


Figure 3.7.2.1 Class diagram of supermarket management system

3.8.1 Sequence Diagram

A Sequence diagram shows object interaction arranged in time sequence .it depicts the objects and classes involves in the scenario and the sequence of messages exchanged between the object needed to carry out the functionality of the scenario. Sequence diagram are typically associated with use case realization in the logical view of the system under development. Sequence is sometimes called event diagram or event scenario.

A sequence diagram shows, as parallel vertical lines, different processes or objects that simultaneously, and as horizontal arrows the message exchanges between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

1. login

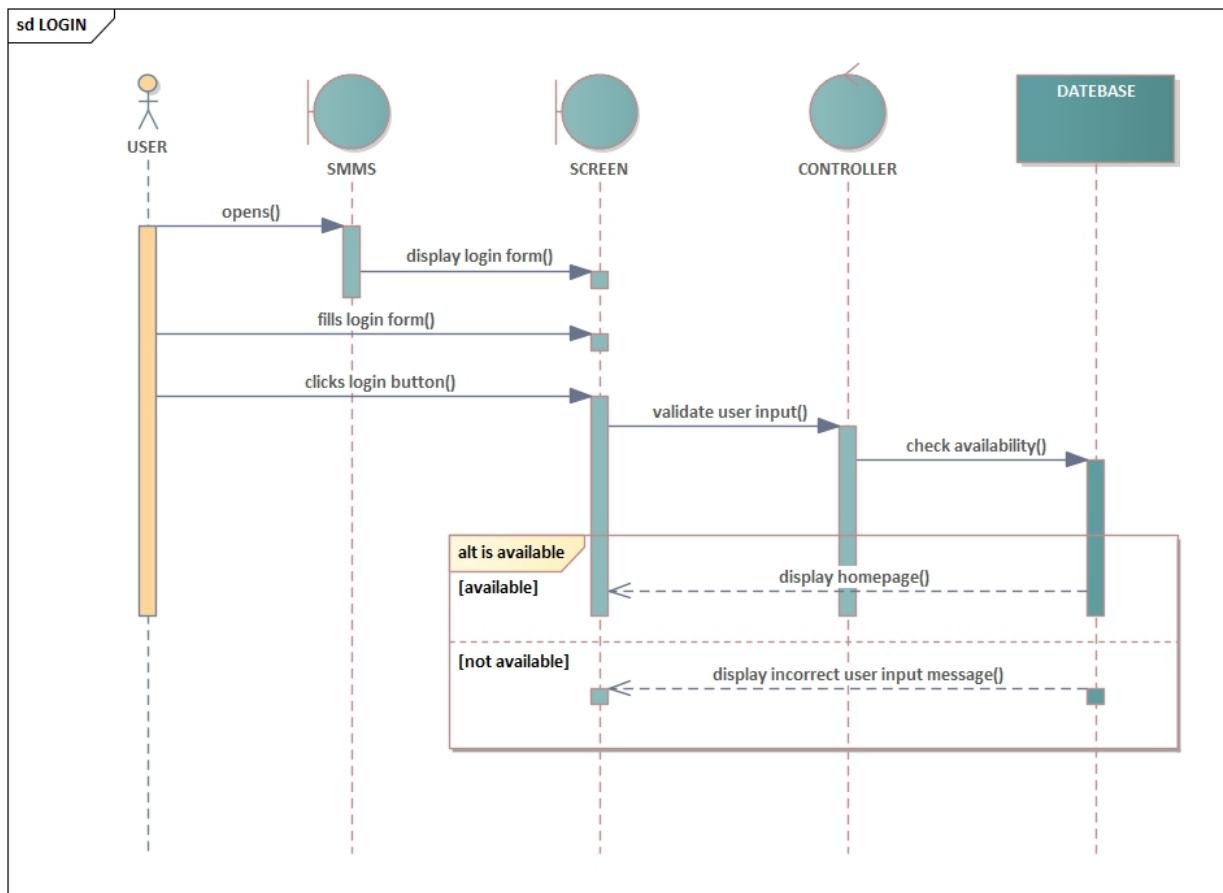


Figure 3.8.1.1 sequence diagram of LOGIN

2. Register employee

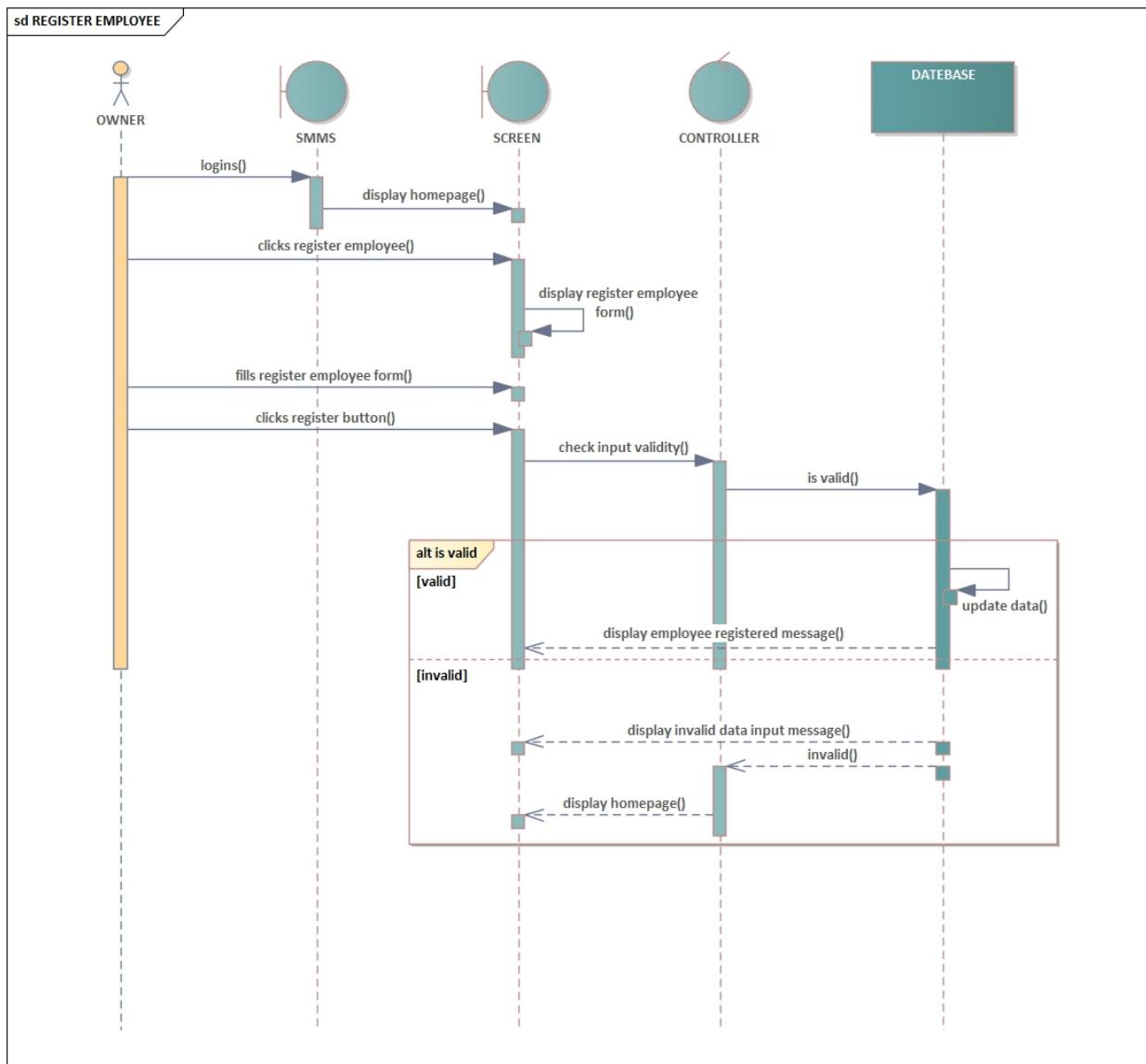


Figure 3.8.1.2 sequence diagram for register employee

3. Assign role

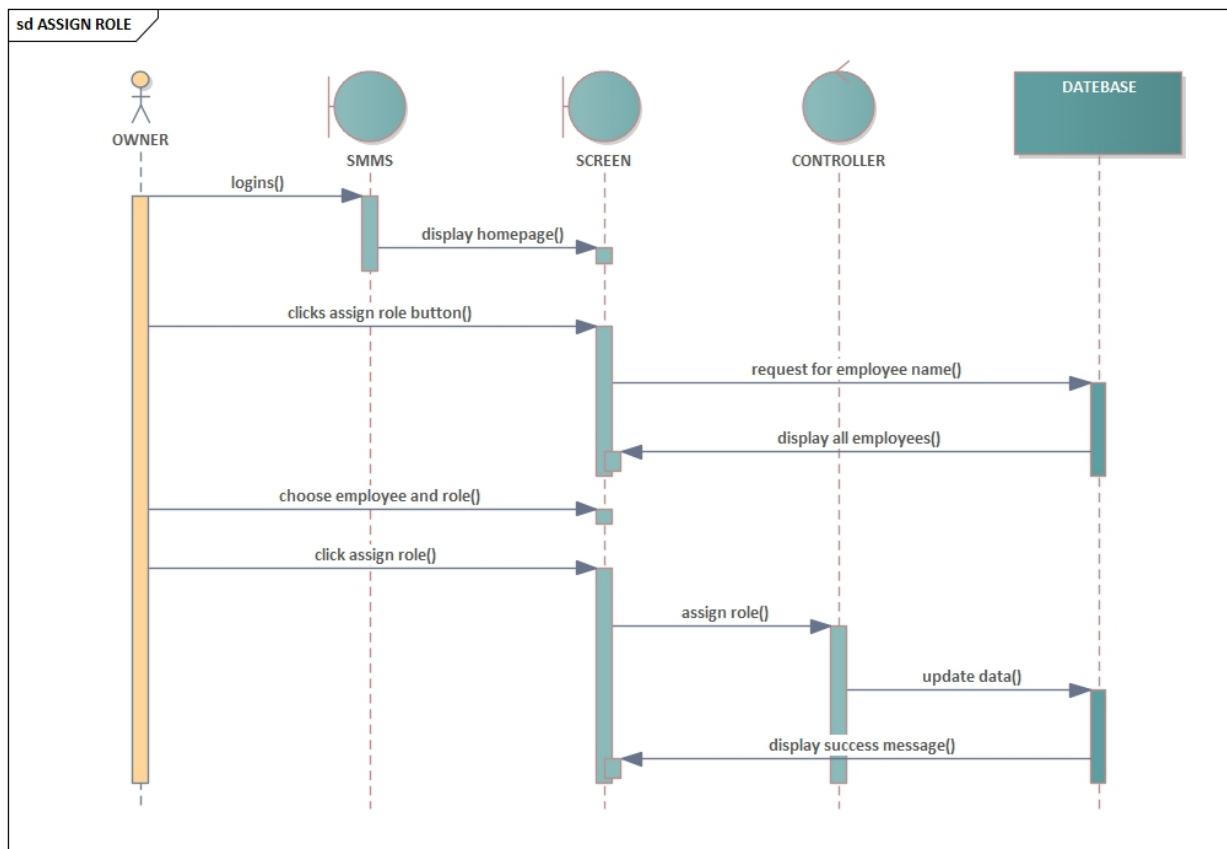


Figure 3.8.1.3 sequence diagram of assign role

4. add product

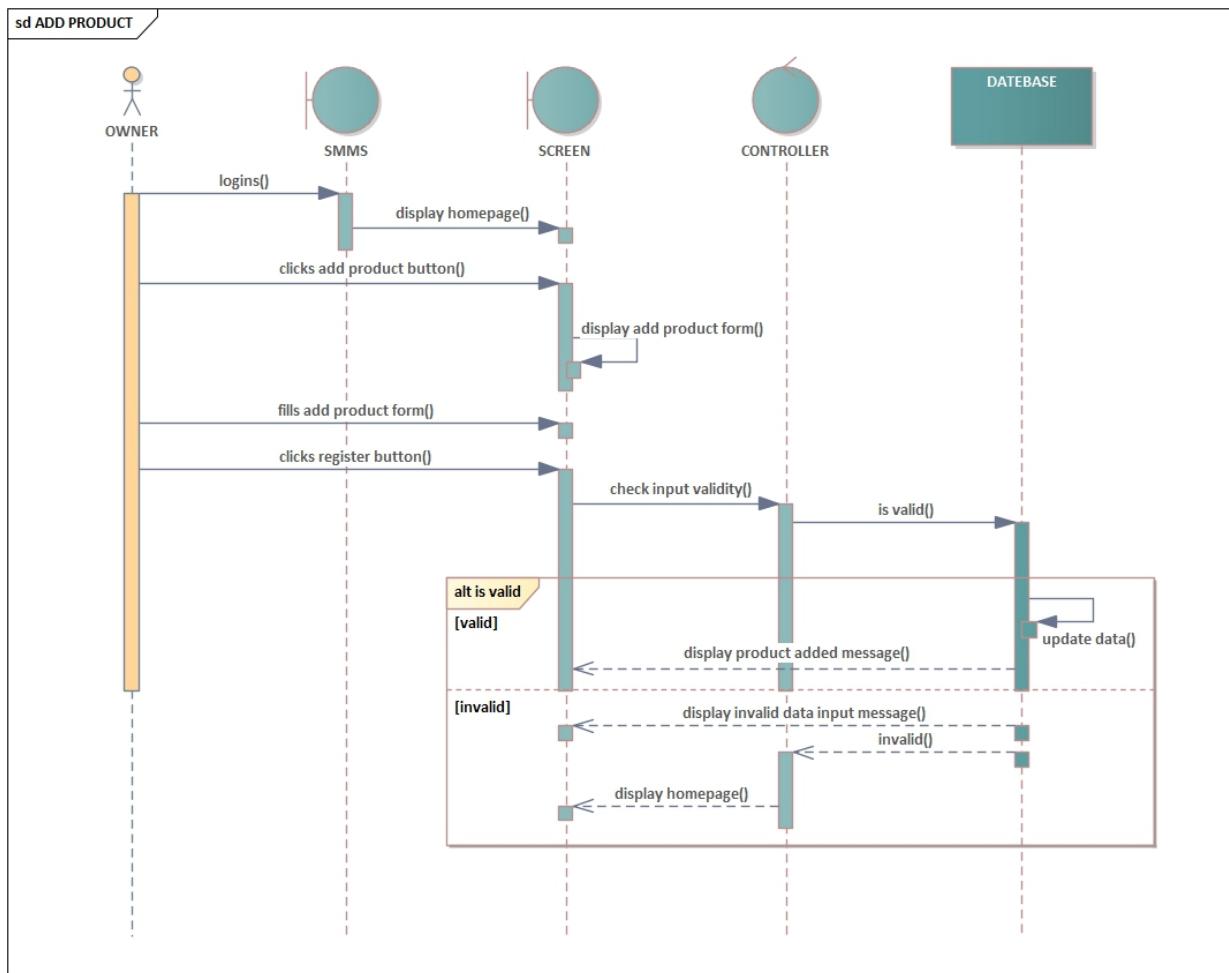


Figure 3.8.1.4 sequence diagram of add product

5. Remove product

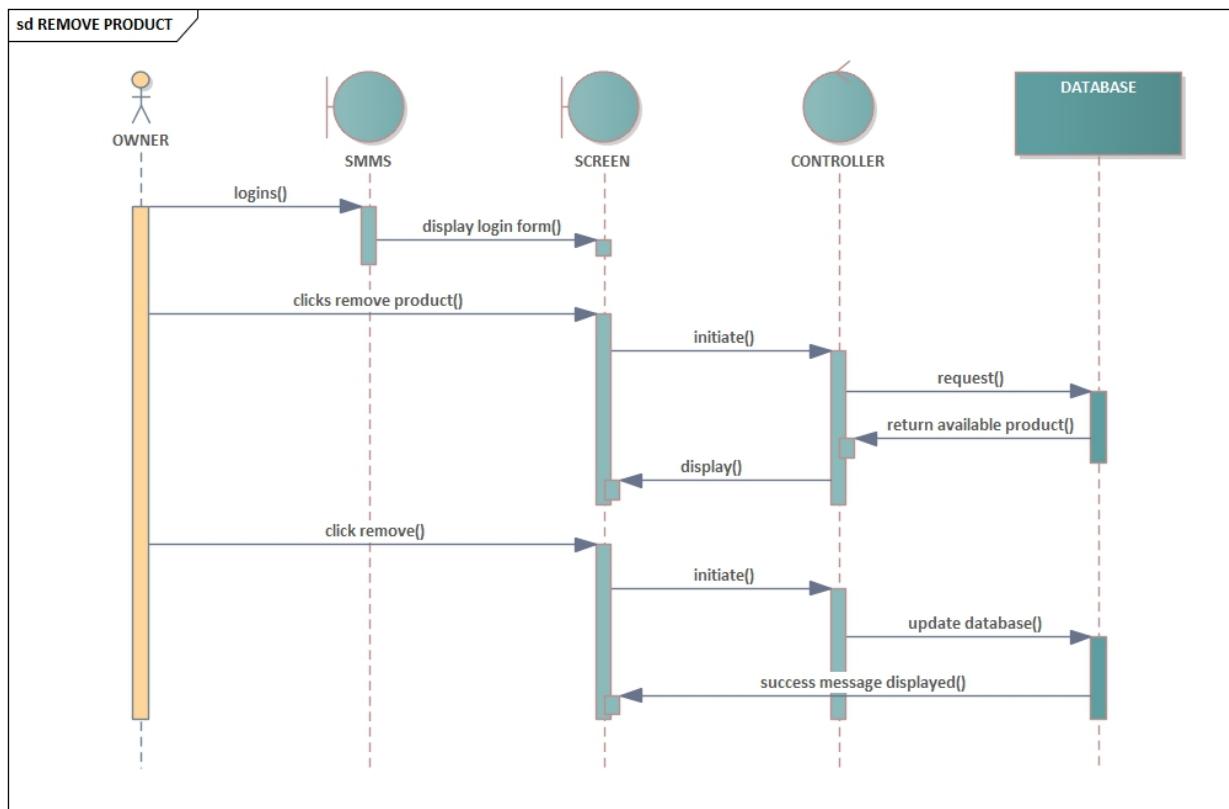


Figure 3.8.1.5 sequence diagram of remove product

6. Register customer

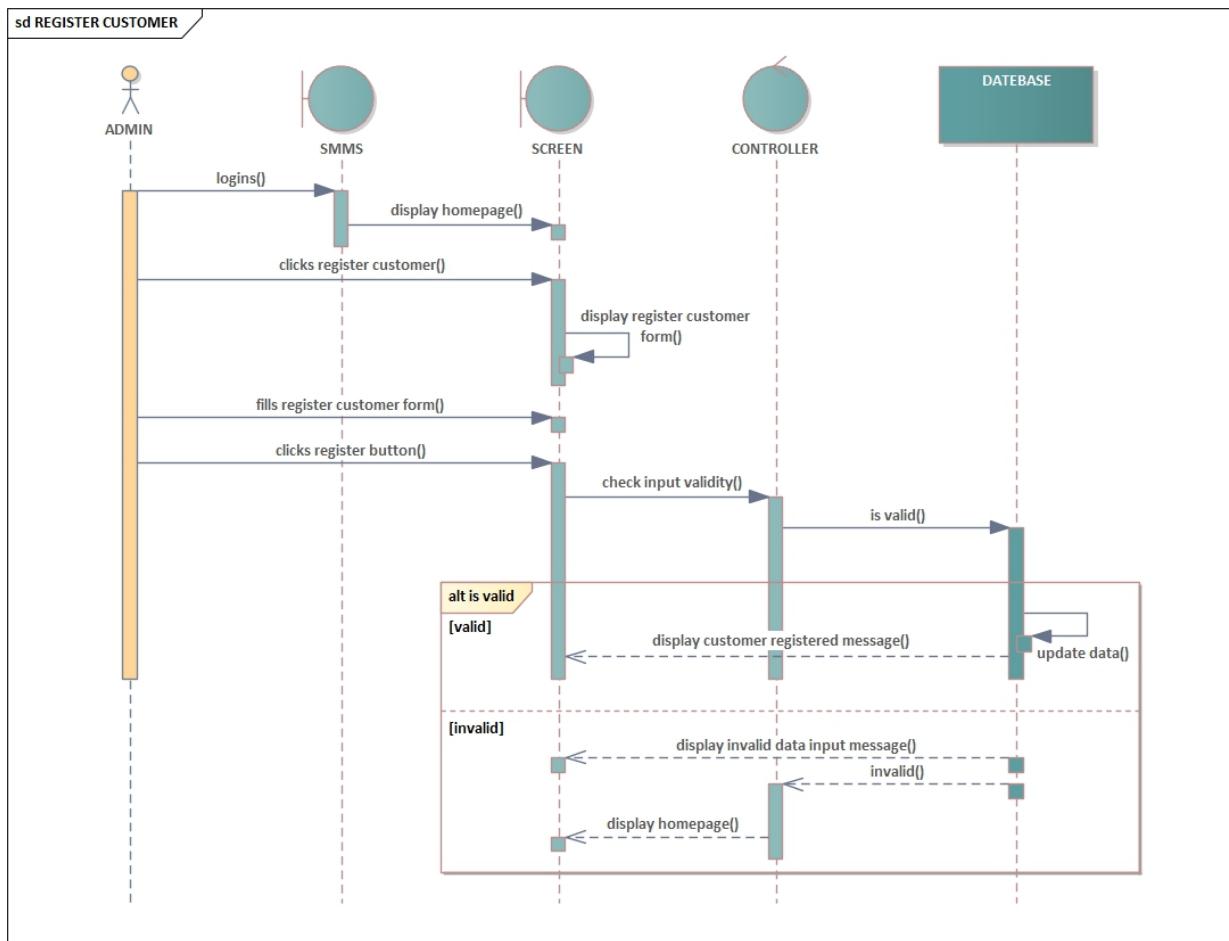


Figure 3.8.1.6 sequence diagram of register customer

7. Remove customer

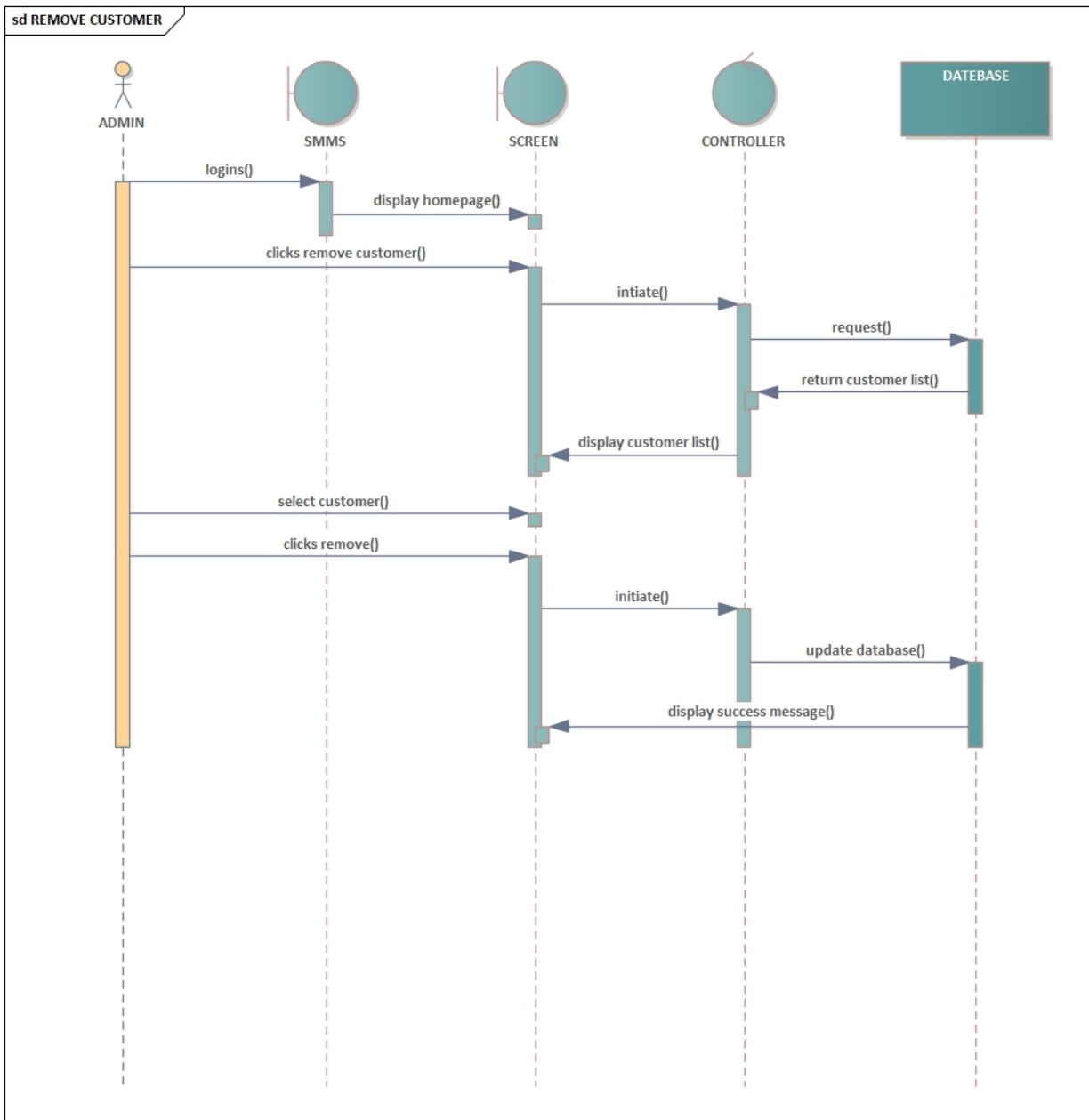


Figure 3.8.1.7 sequence diagram of remove customer

8. Give comment

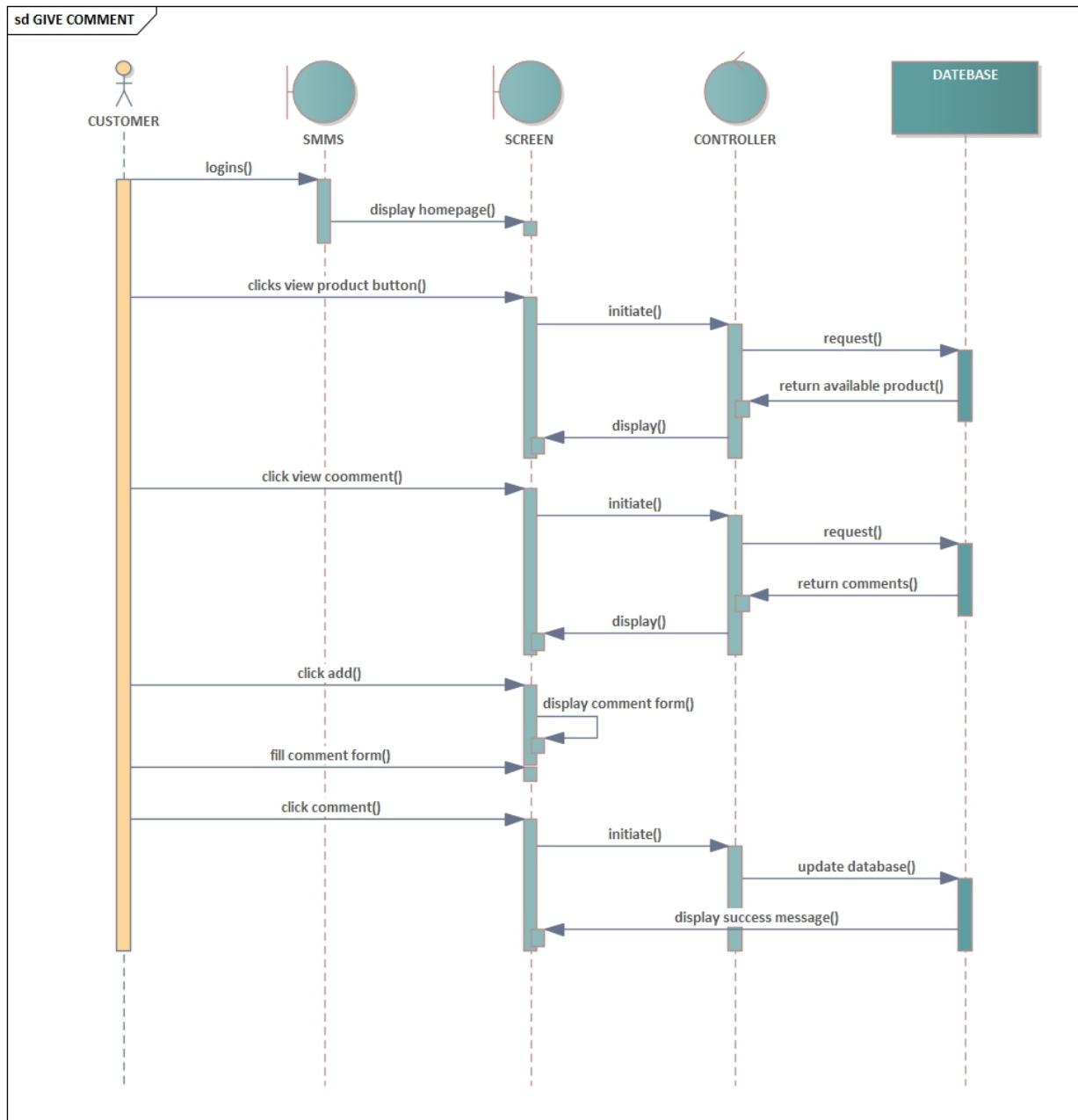


Figure 3.8.1.8 sequence diagram of give comment

9. Buy product

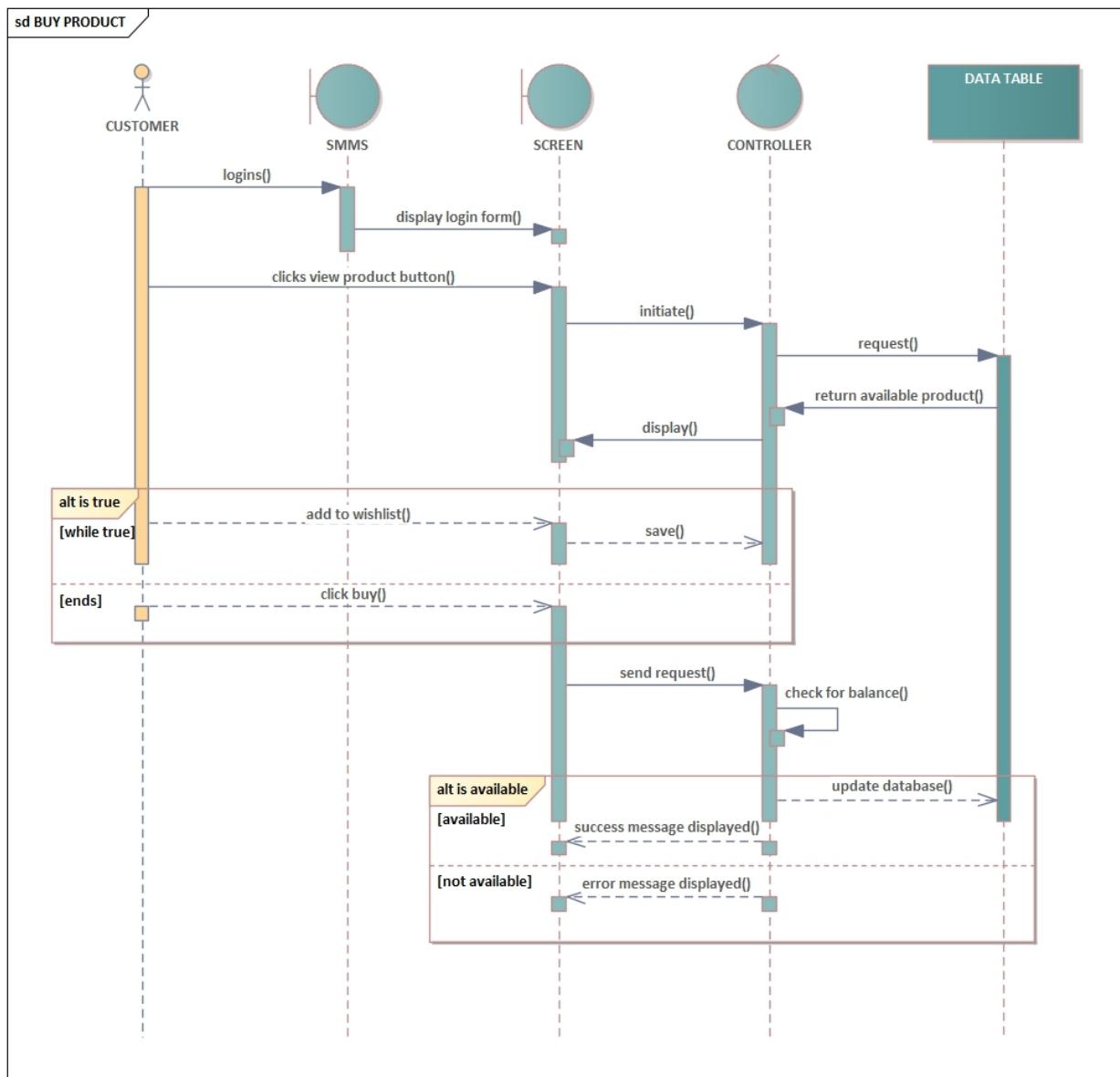


Figure 3.8.1.9 sequence diagram of buy product

10. Edit profile

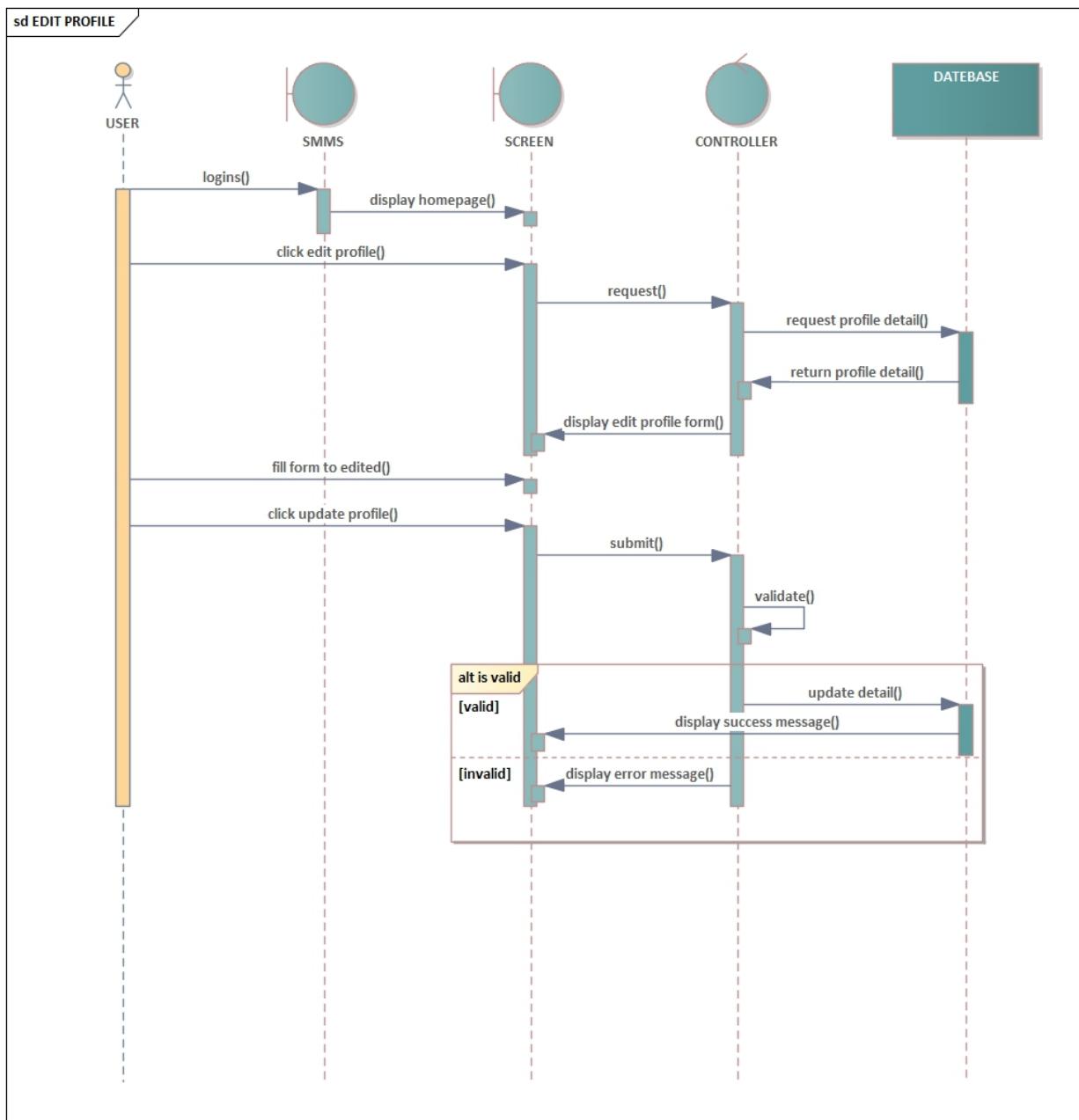


Figure 3.8.1.10 sequence diagram of edit profile

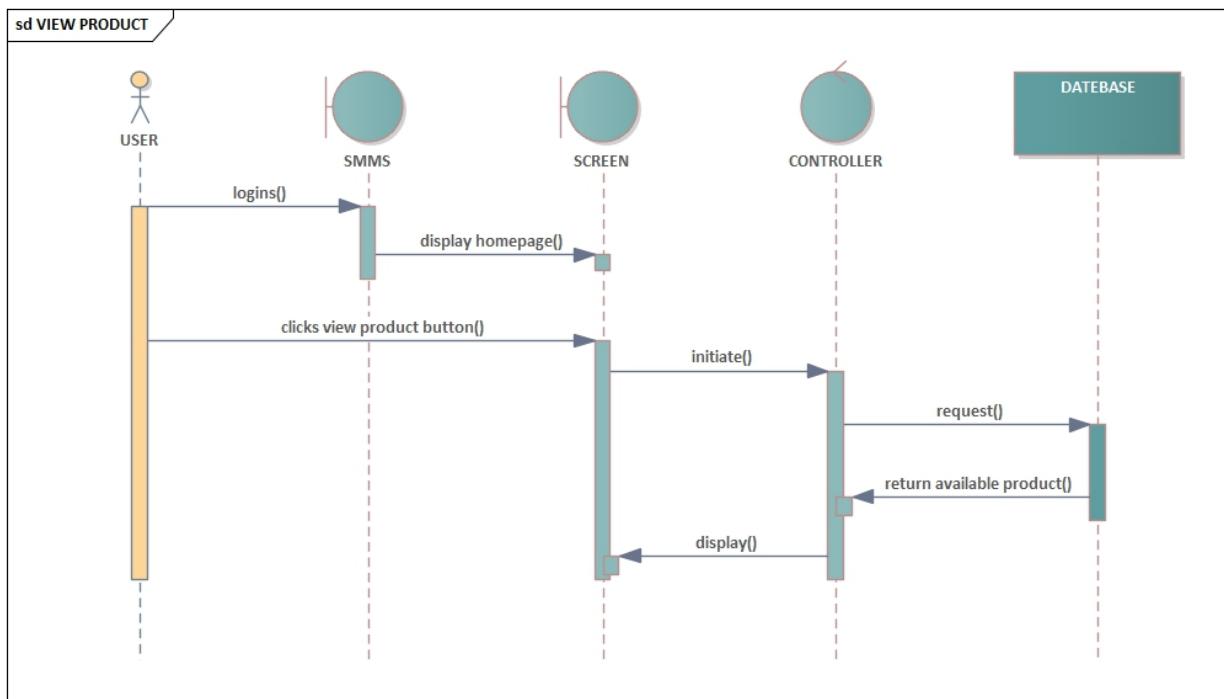
11. View product

Figure 3.8.1.11 sequence diagram of view product

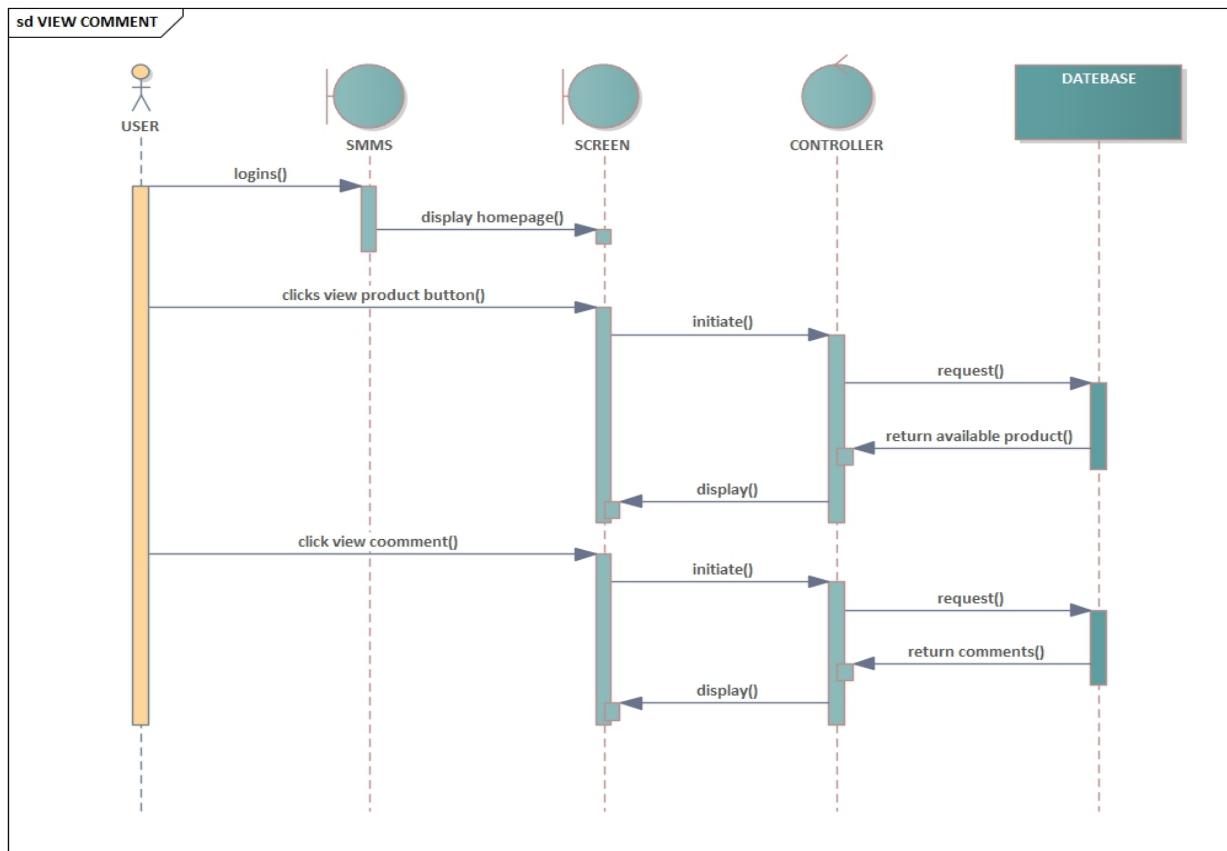
12. View comment

Figure 3.8.1.12 sequence diagram of view comment

13. Create bill

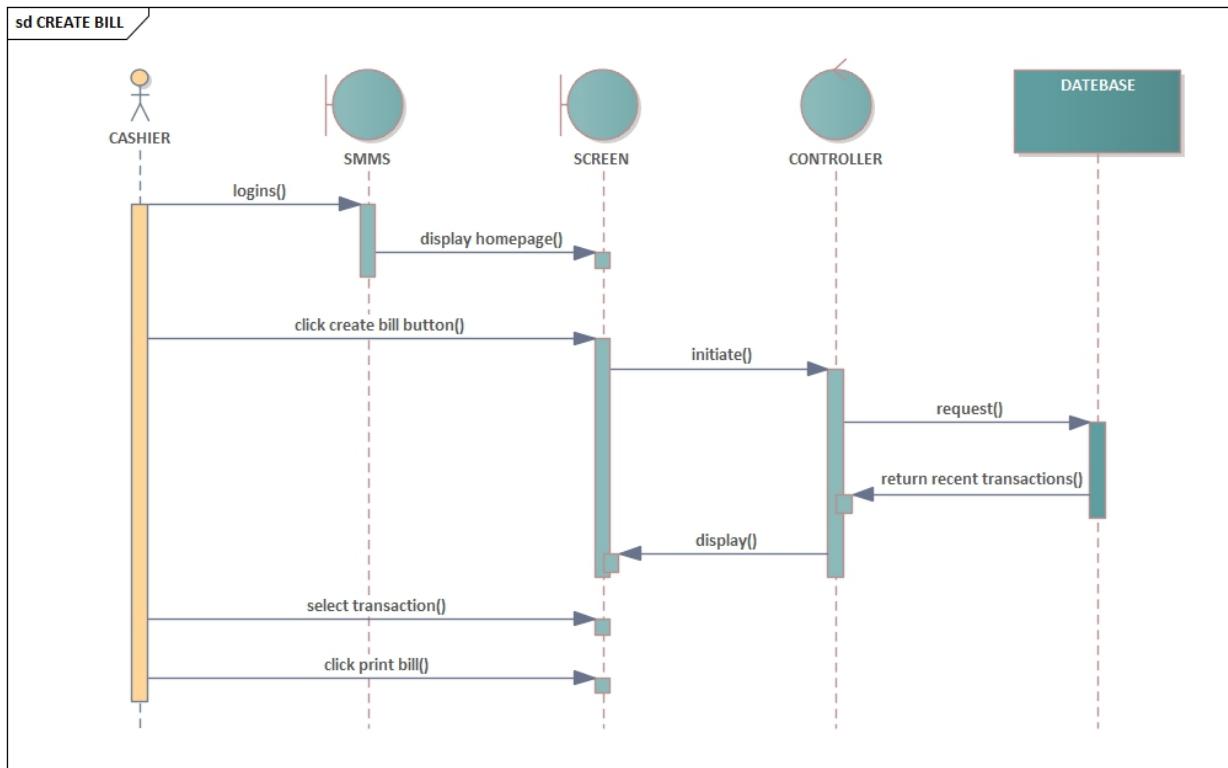


Figure 3.8.1.13 sequence diagram of create bill

3.8.2. Activity diagram

An activity diagram is used to understand the flow of work that an object or component performs. It can also be used to visualize the interaction between different use cases. One of the strengths of activity diagrams is the representation of concurrent activities. Some of the activity diagrams of our system are listed below.

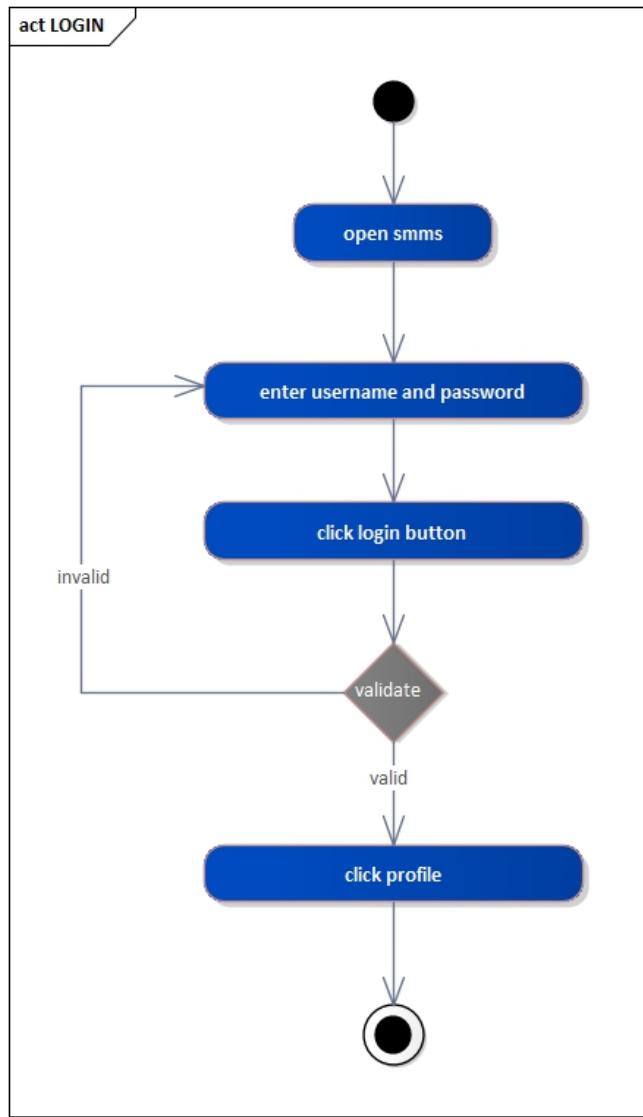
1. Login

Figure 3.8.2.1 activity diagram of login

2. Register employee

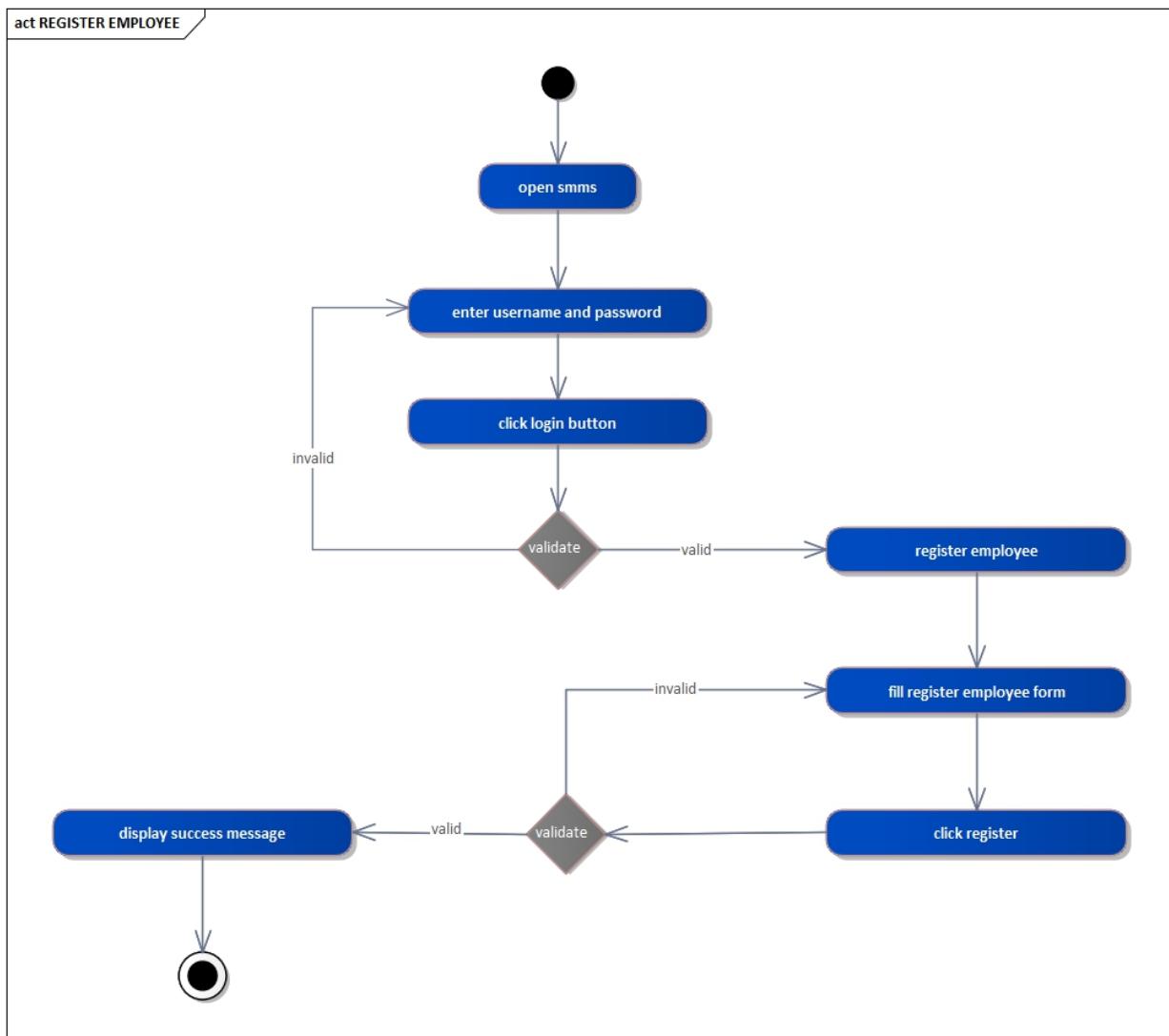


Figure 3.8.2.2 activity diagram of register employee

3. Assign role

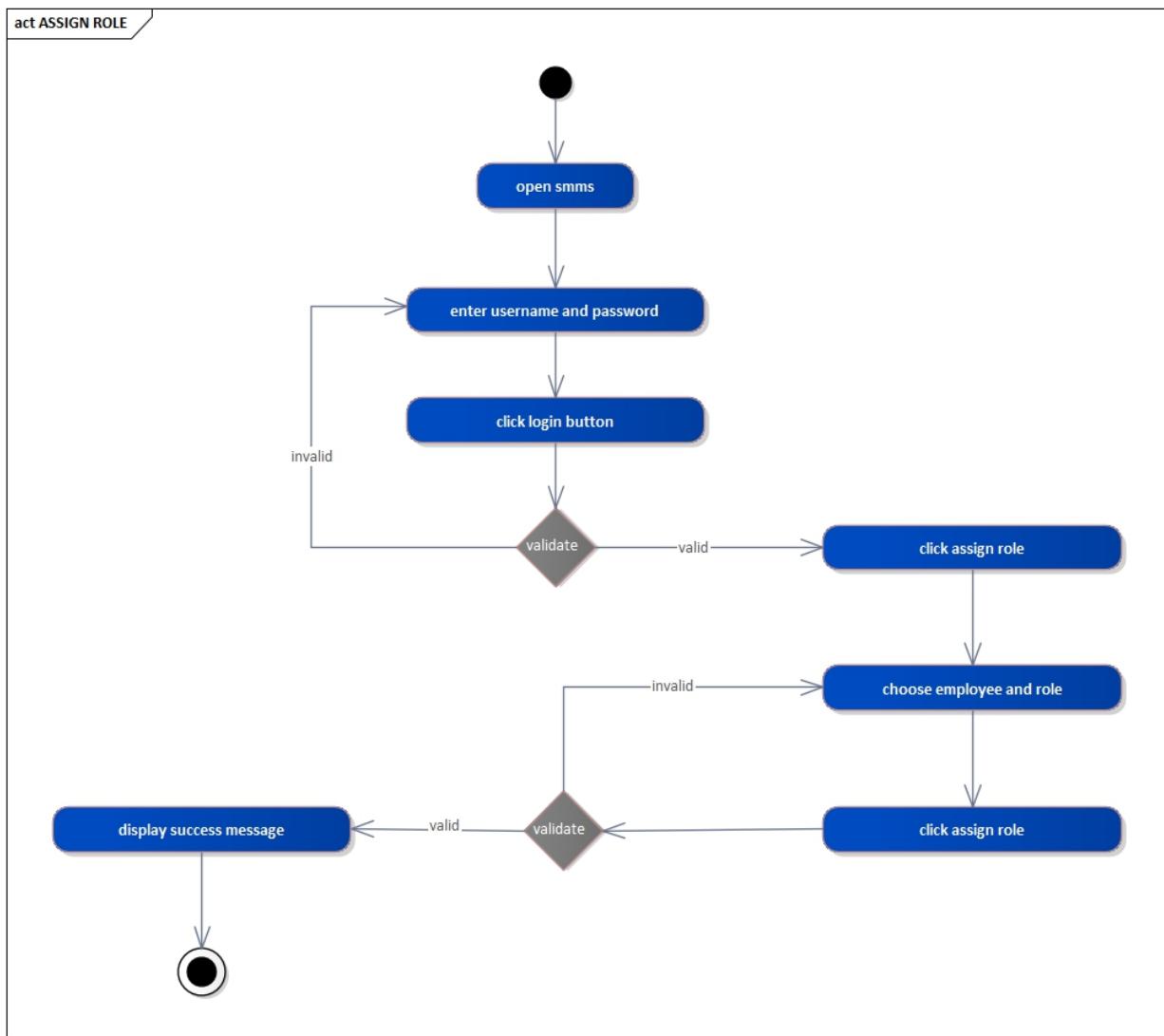


Figure 3.8.2.3 activity diagram of assign role

4. Add product

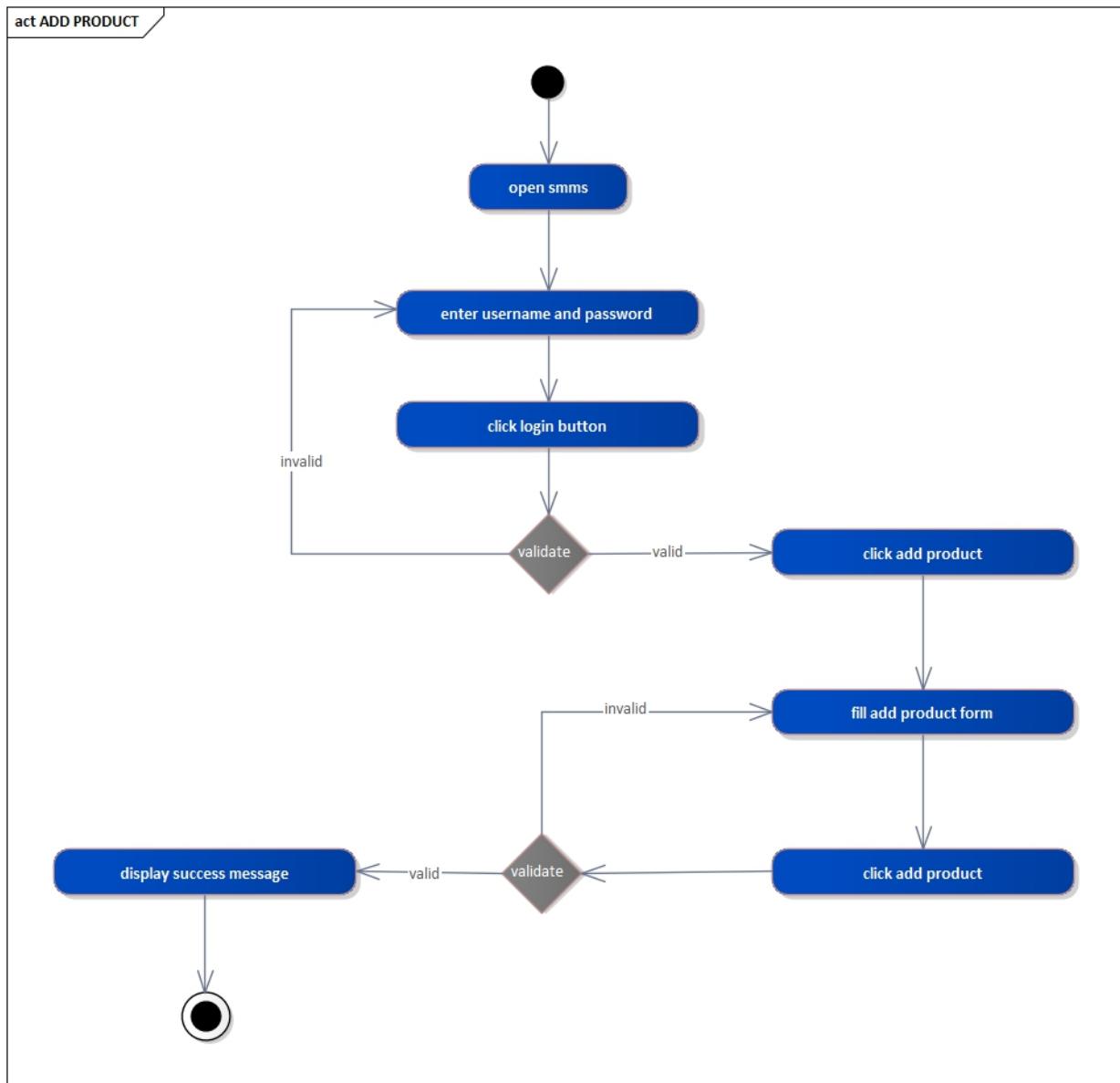


Figure 3.8.2.4 activity diagram of add product

5. Remove product

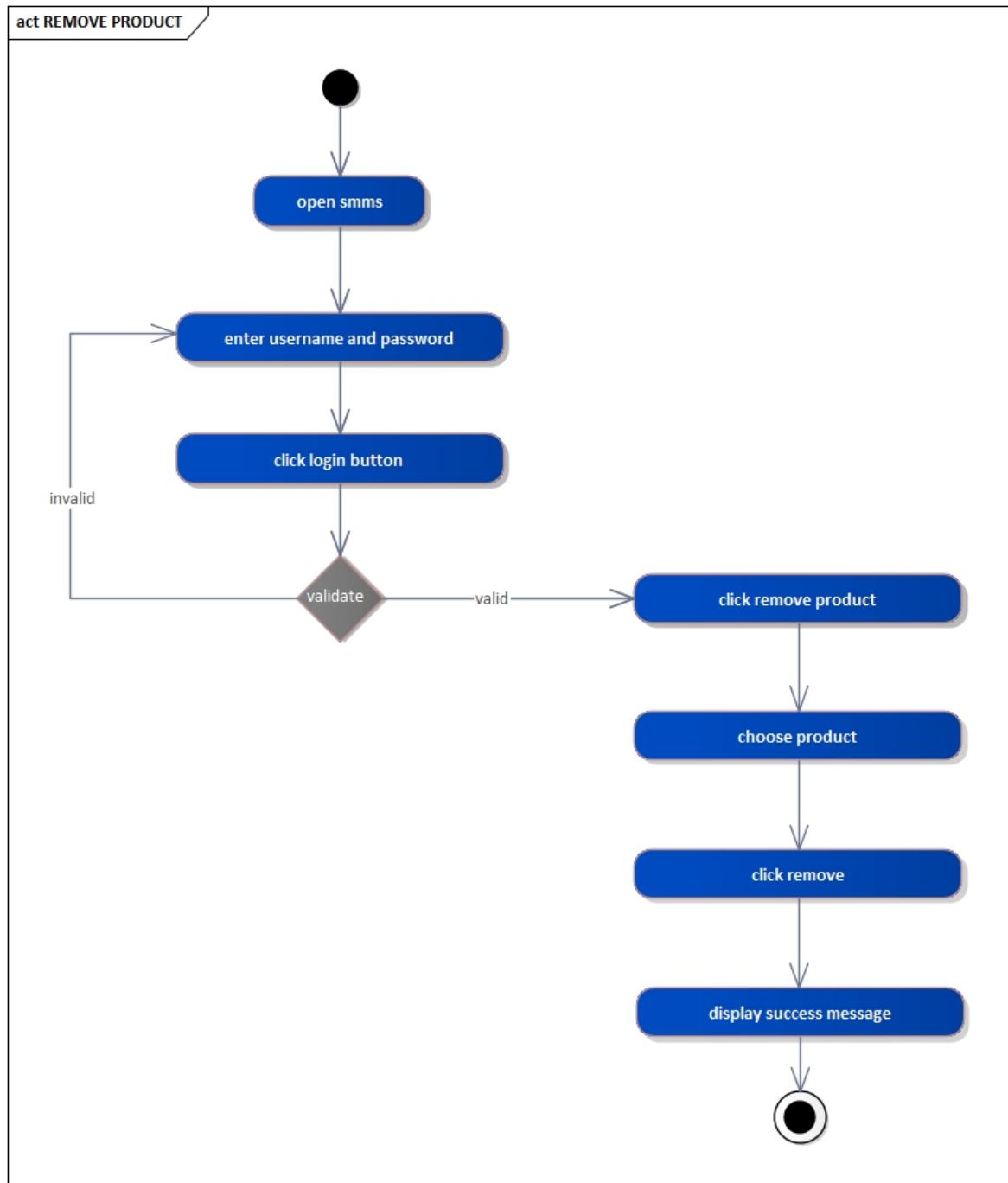


Figure 3.8.2.5 activity diagram of remove product

6. Register customer

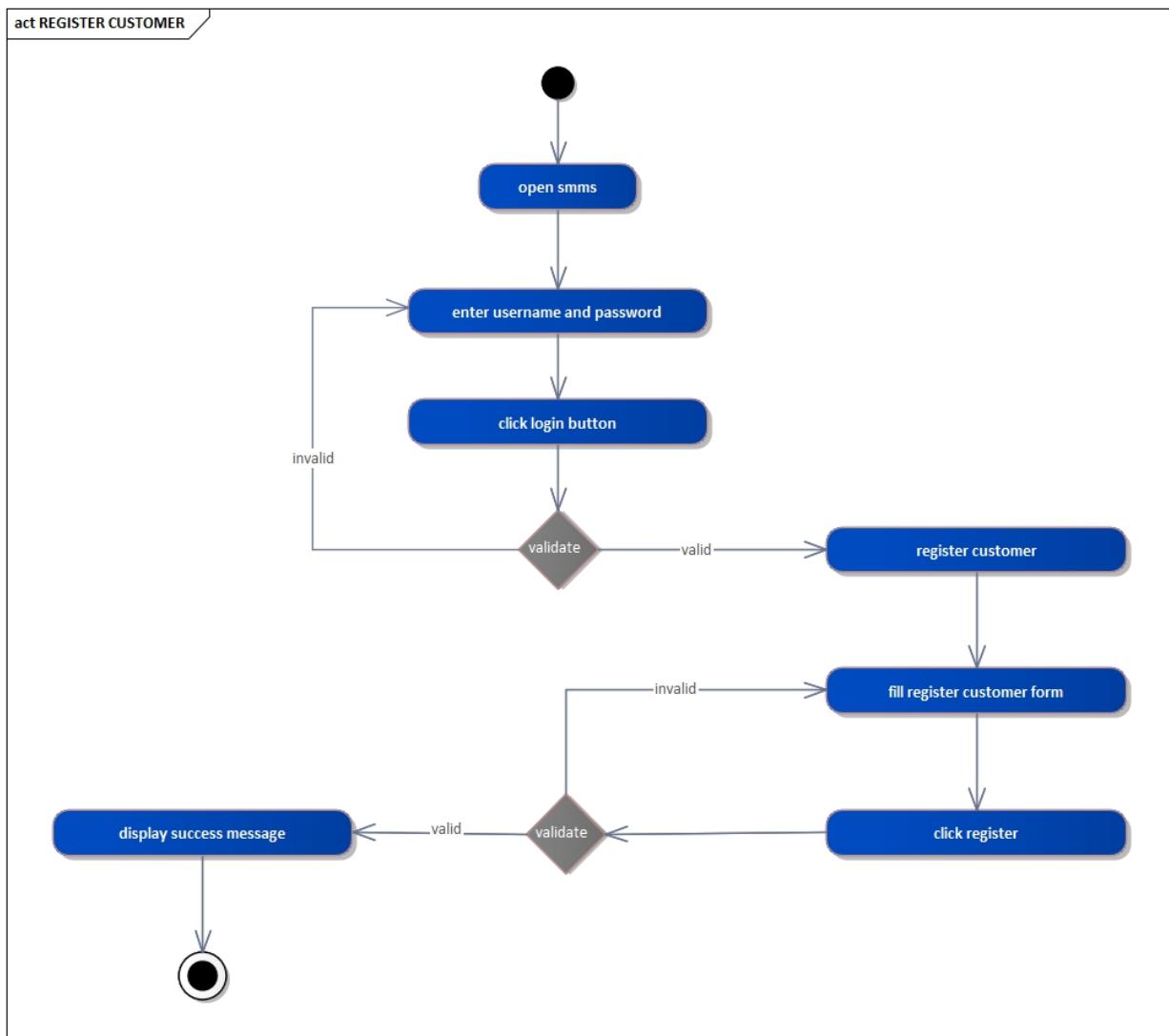


Figure 3.8.2.6 activity diagram of register customer

7. Remove customer

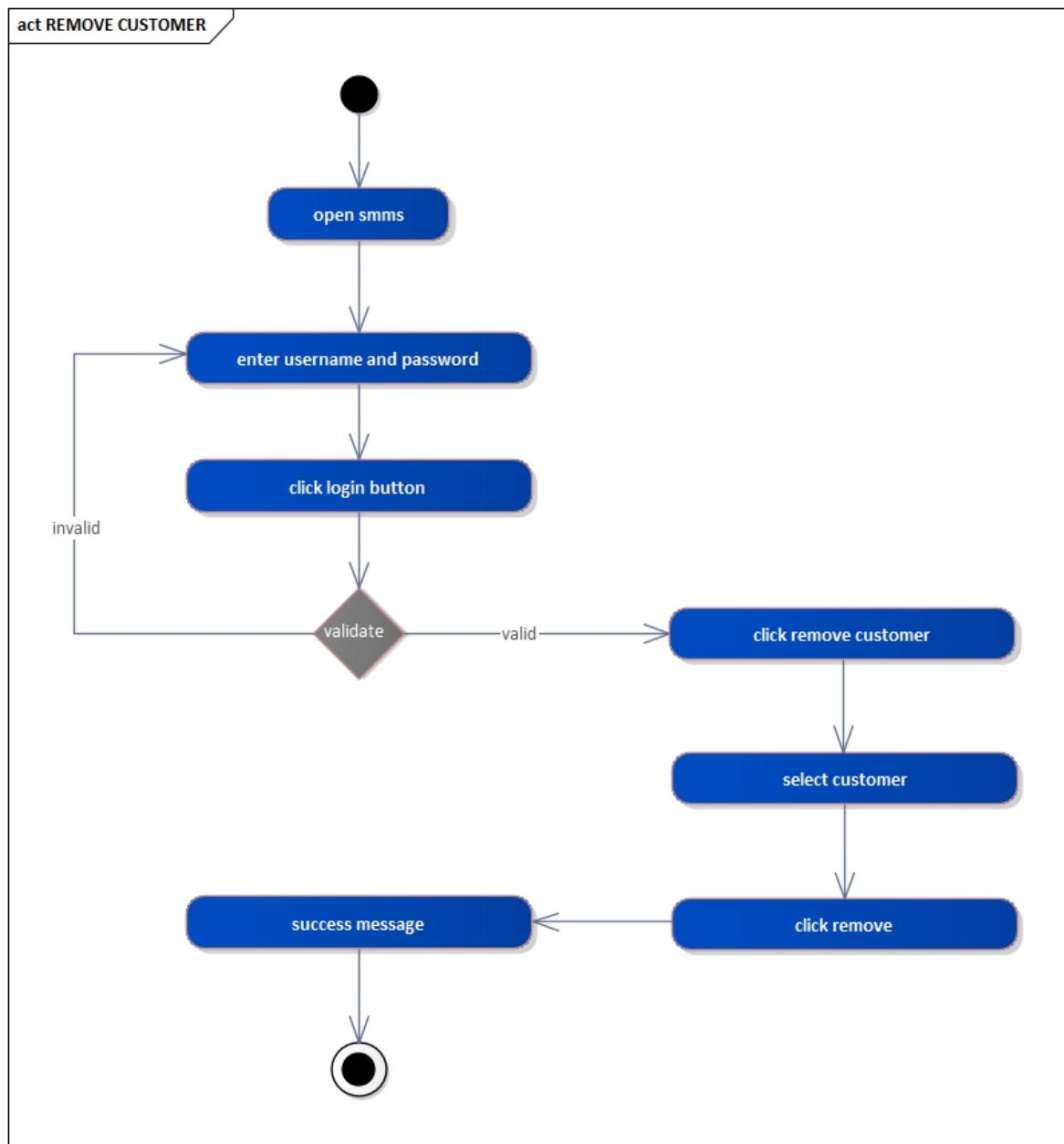


Figure 3.8.2.7 activity diagram of remove customer

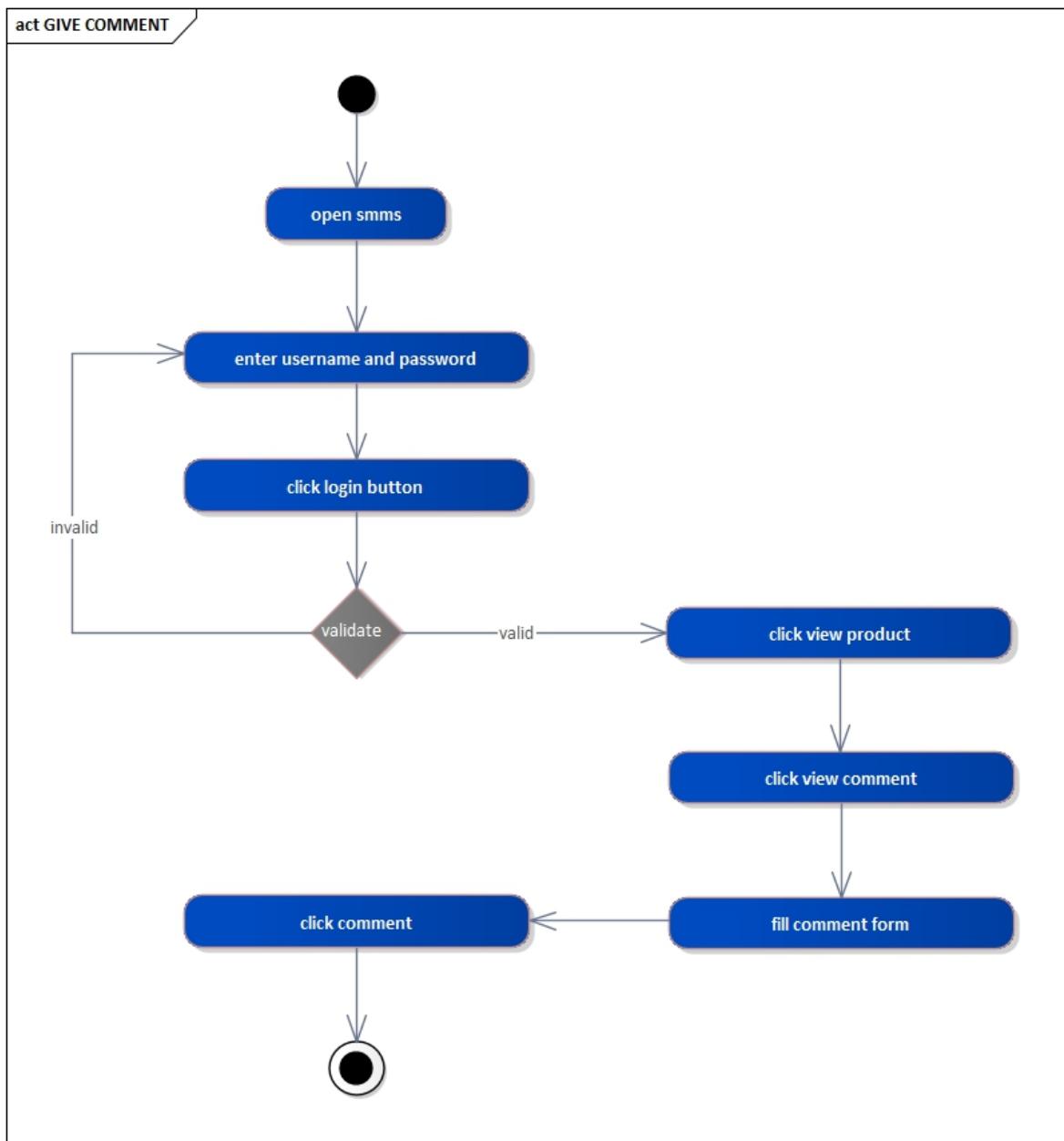
8. Give comment

Figure 3.8.2.8 activity diagram of give comment

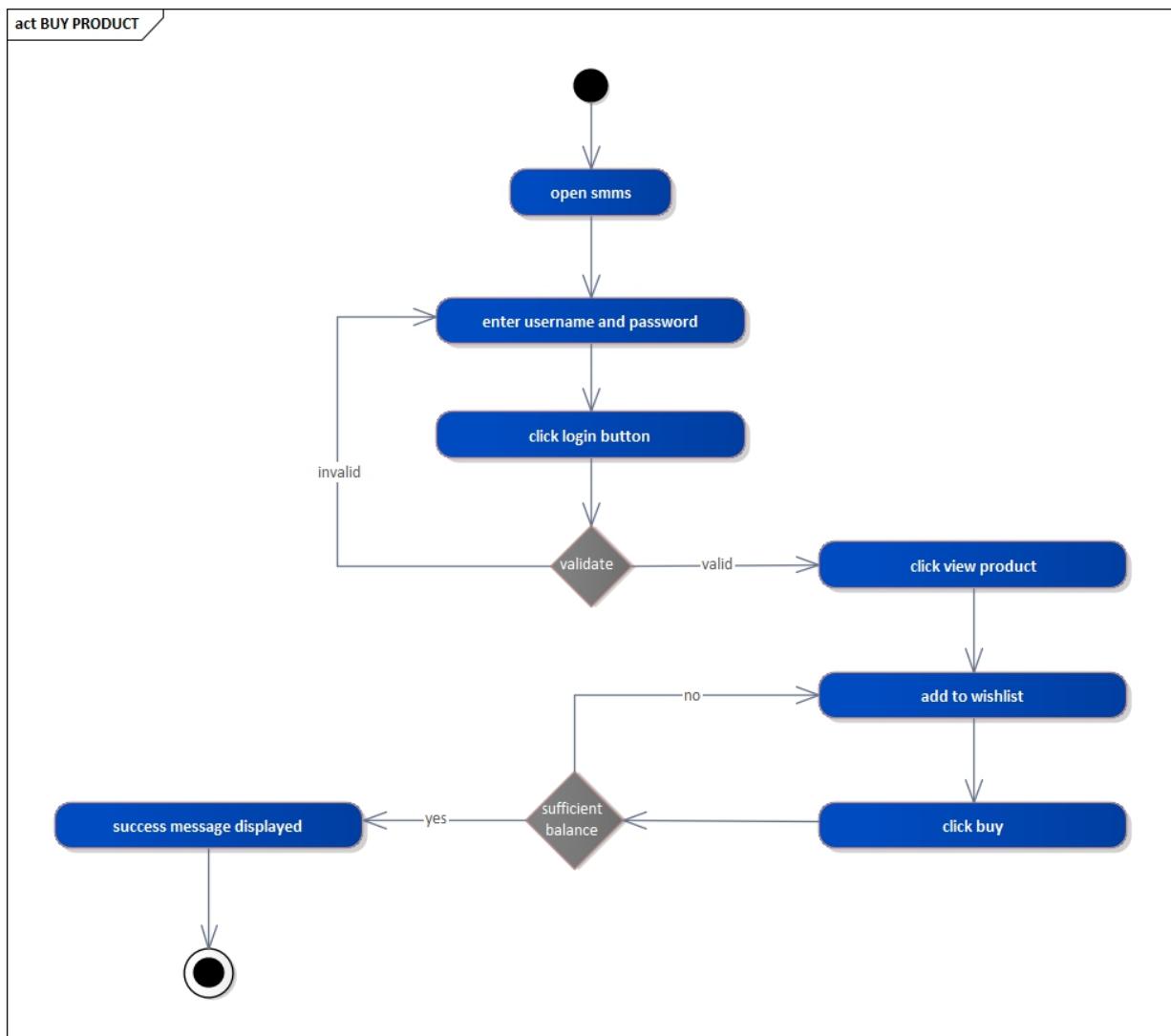
9. Buy product

Figure 3.8.2.9 activity diagram of buy product

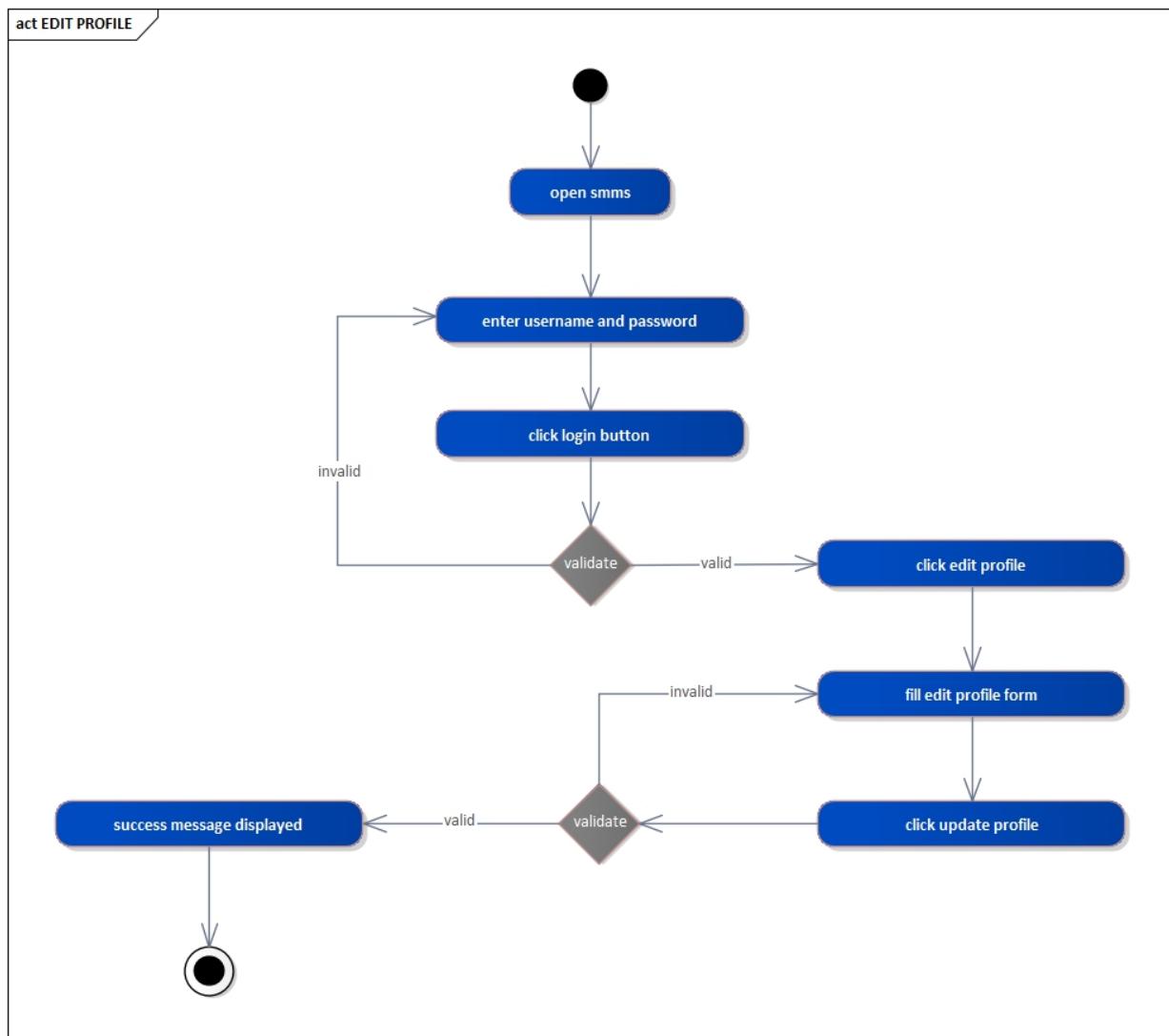
10. Edit profile

Figure 3.8.2.10 activity diagram of edit profile

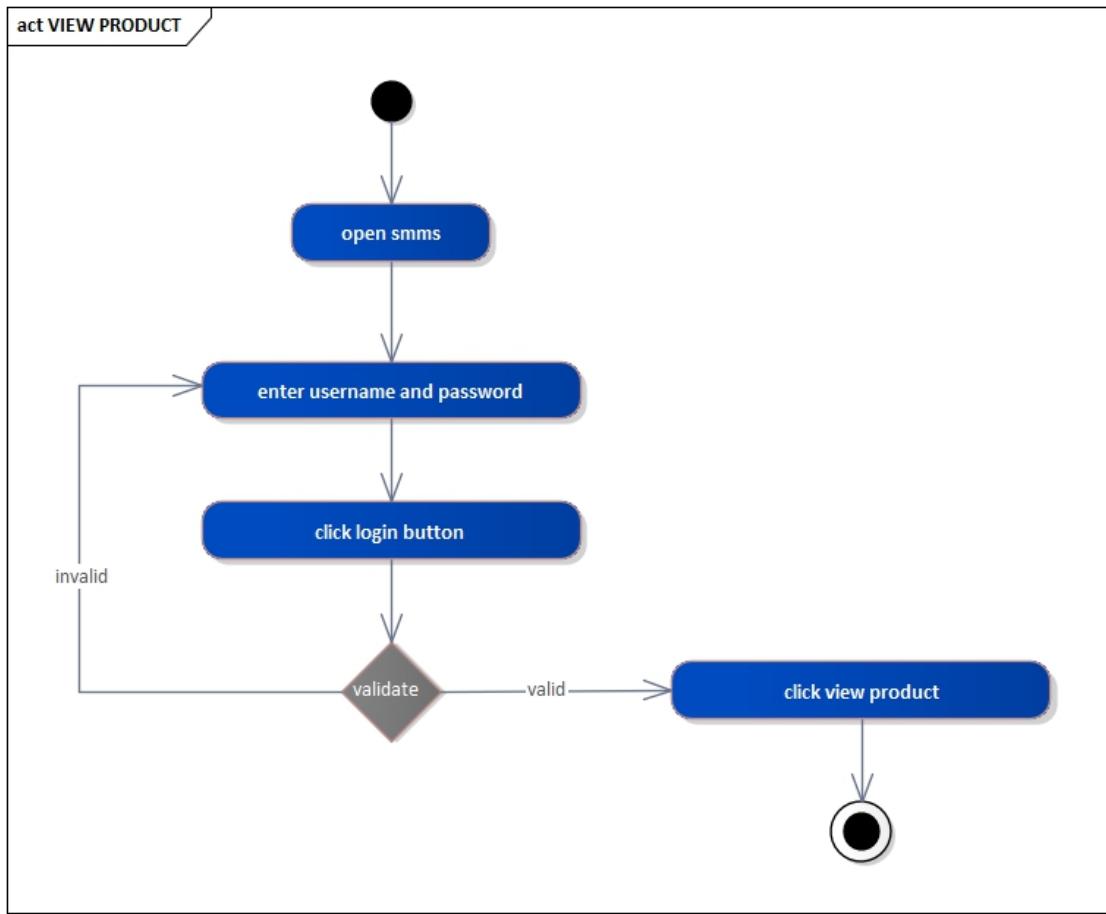
11. View product

Figure 3.8.2.11 activity diagram of view product

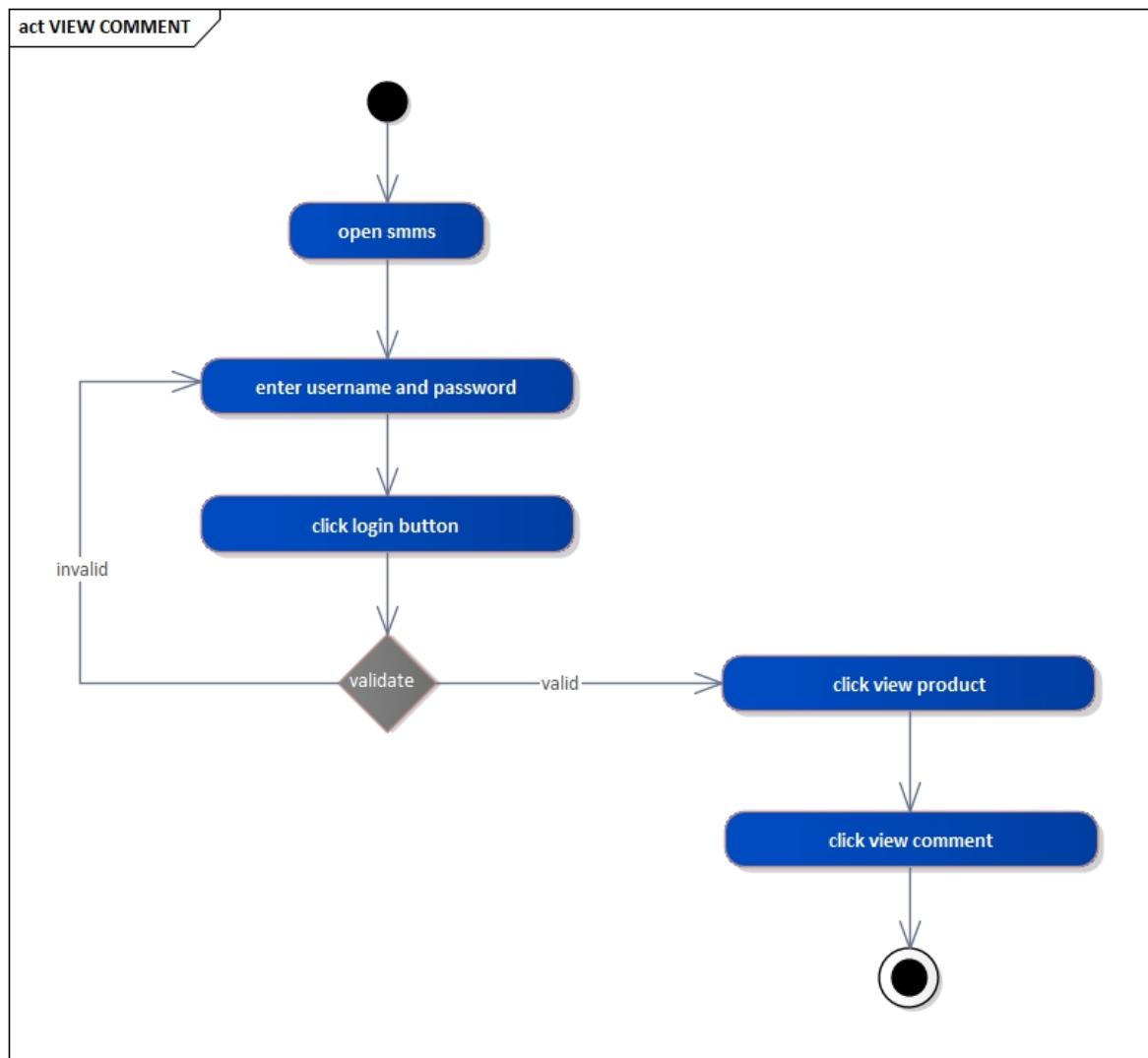
12. View comment

Figure 3.8.2.12 activity diagram of view comment

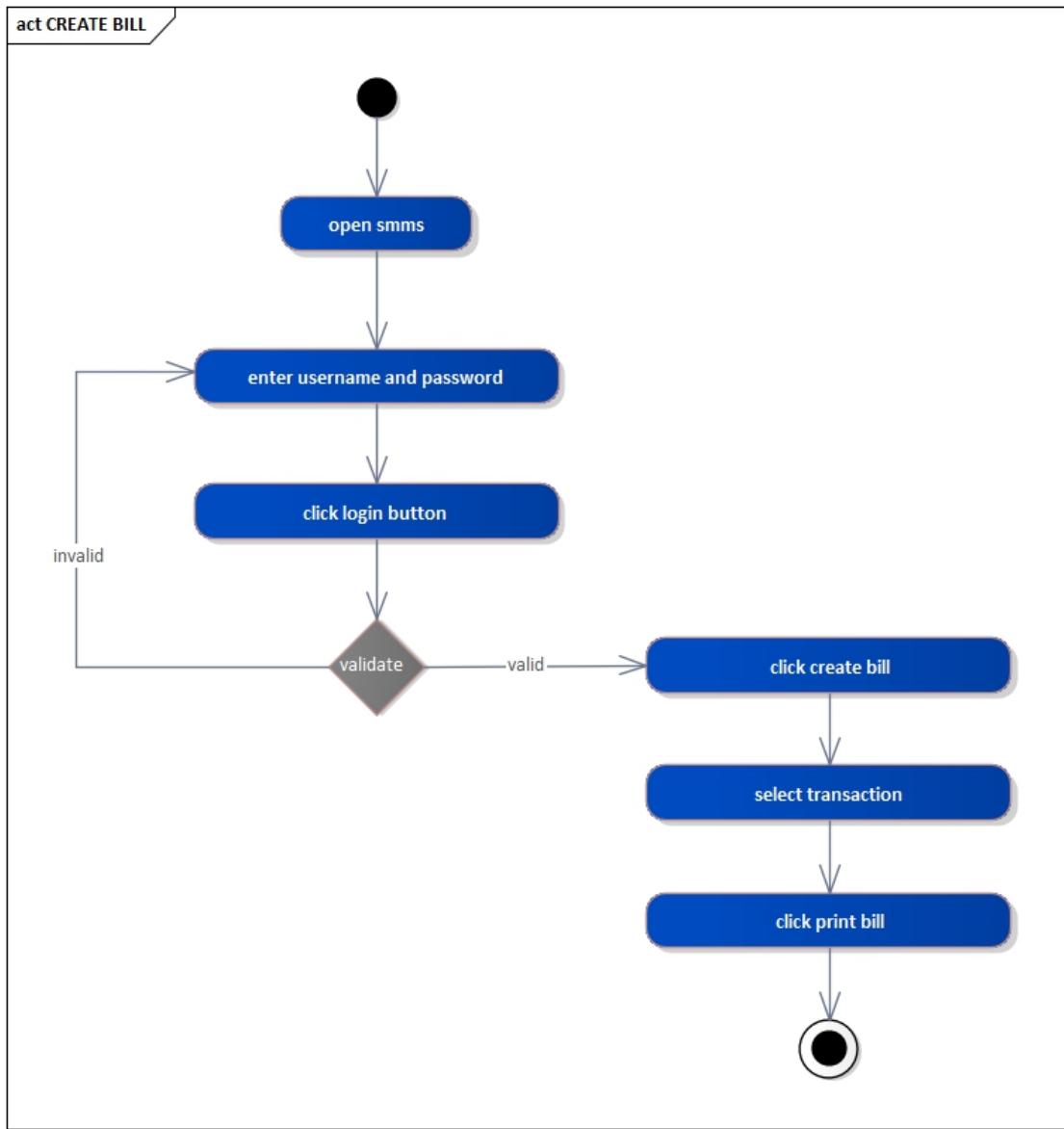
13. Create bill

Figure 3.8.2.13 activity diagram of create bill

3.8.3 State chart diagram

It is used to describe the externally visible behavior of a system or of an individual object. Some of the state chart diagrams of our system are described below.

1. Login

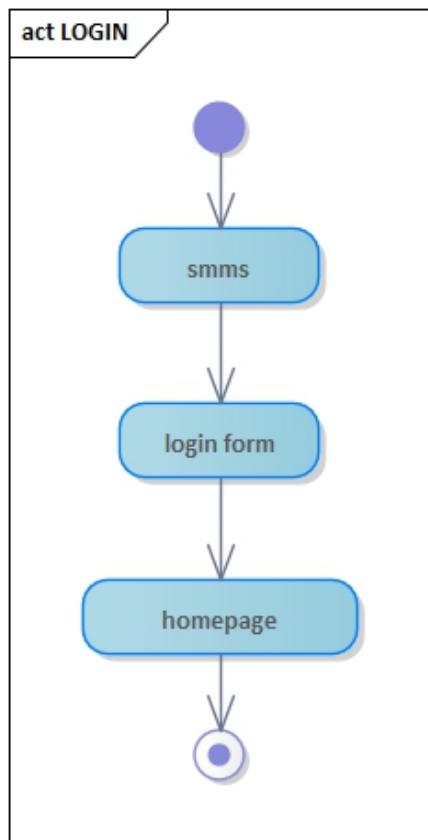


Figure 3.8.3.1 state diagram of login

2. Register employee

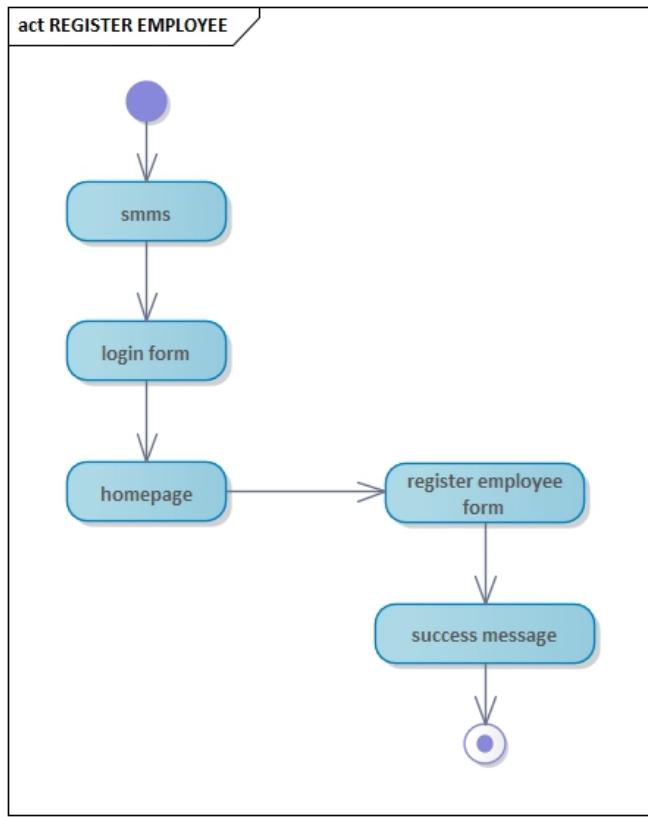


Figure 3.8.3.2 state diagram of register employee

3. Assign role

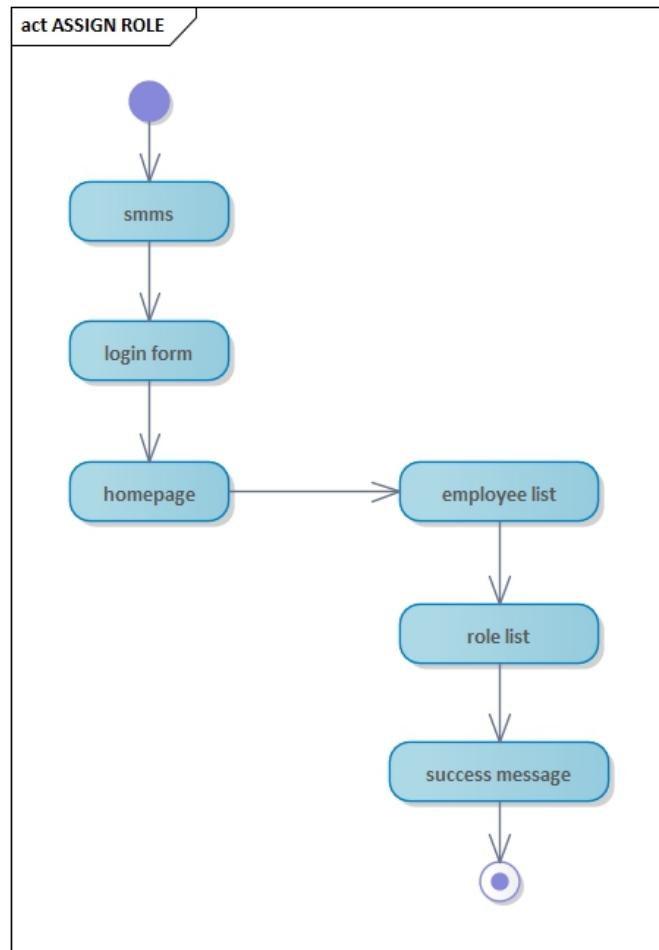


Figure 3.8.3.3 state diagram of assign role

4. Add product

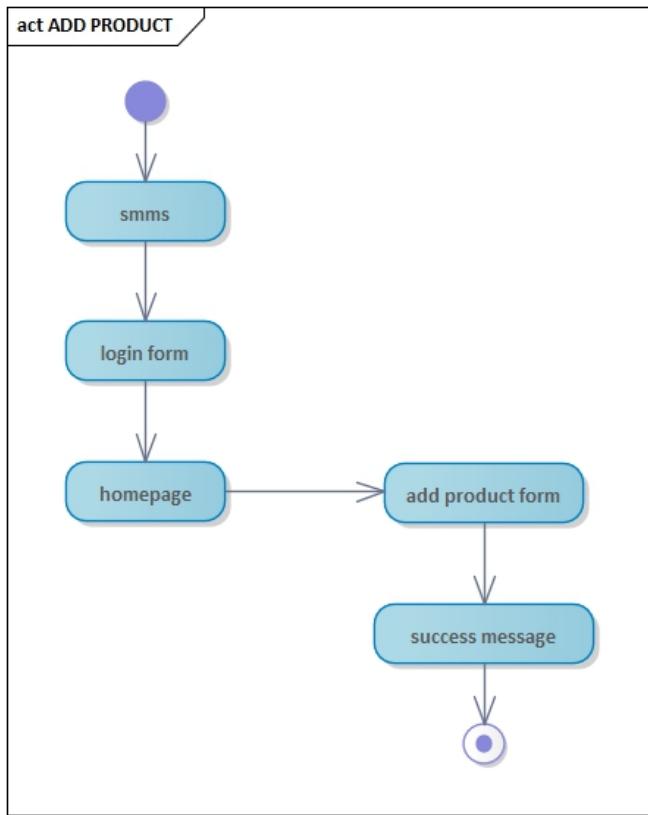


Figure 3.8.3.4 state diagram of add product

5. Remove product

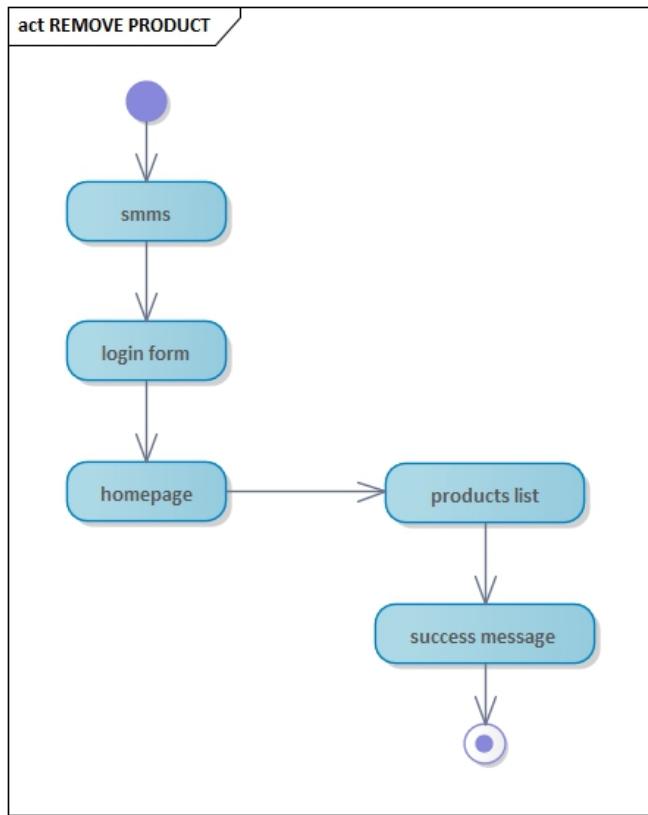


Figure 3.8.3.5 state diagram of remove product

6. Register customer

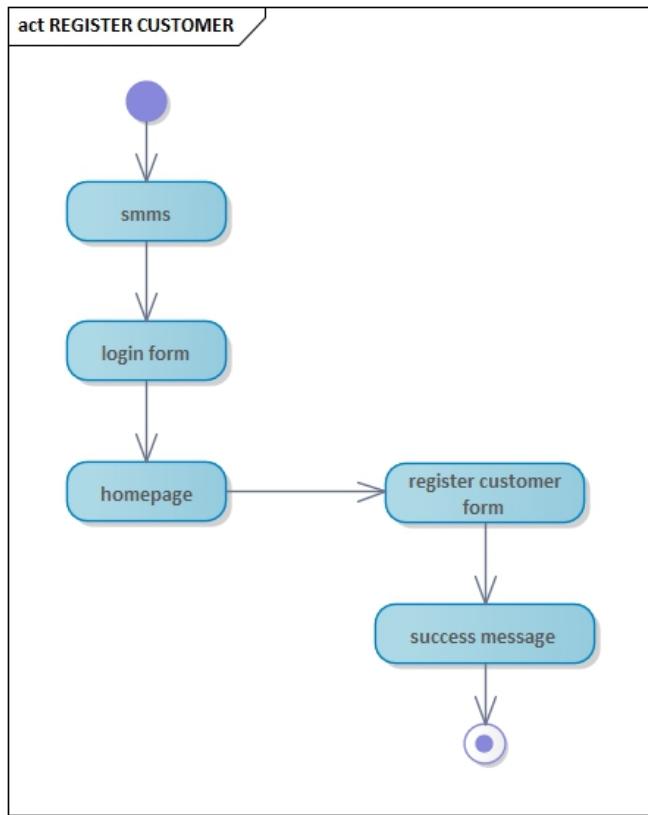


Figure 3.8.3.6 state diagram of register customer

7. Remove customer

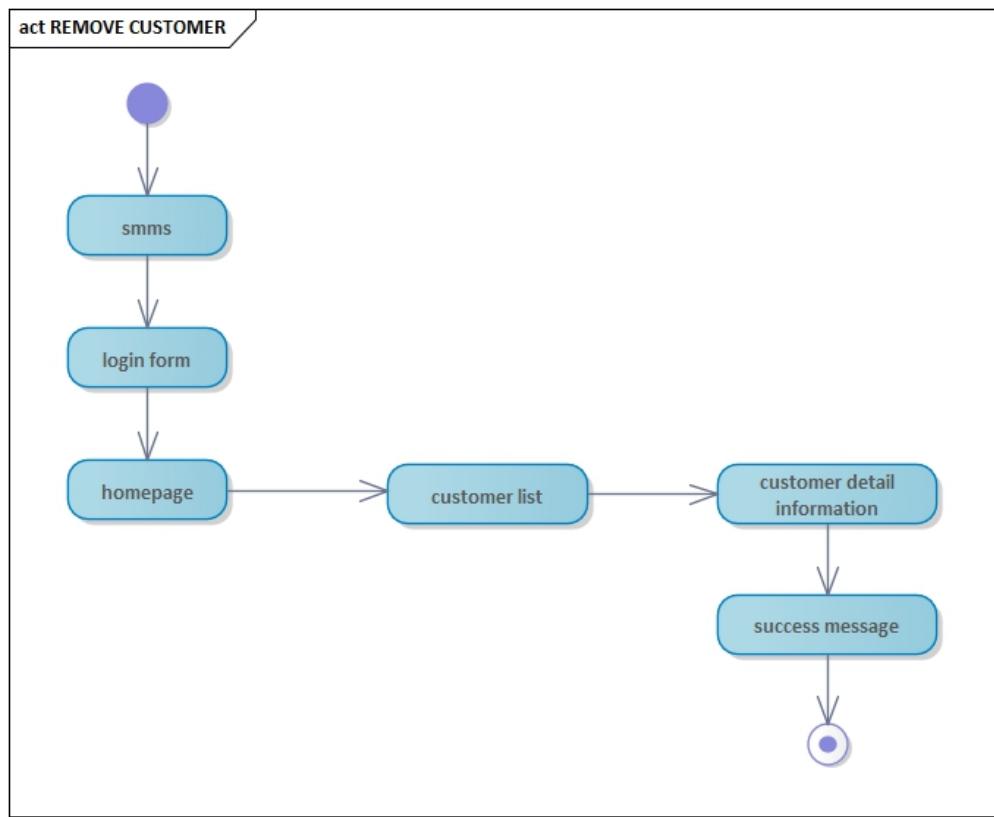


Figure 3.8.3.7 state diagram of remove customer

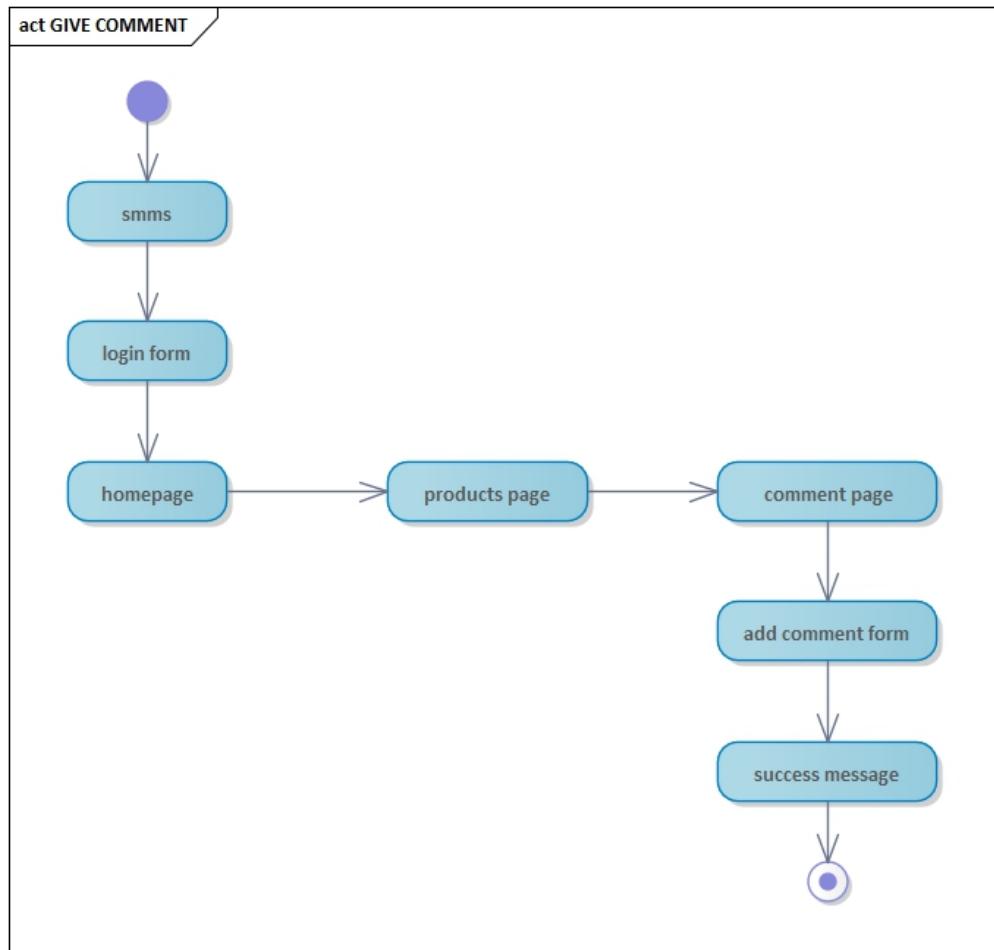
8. Give comment

Figure 3.8.3.8 state diagram of give comment

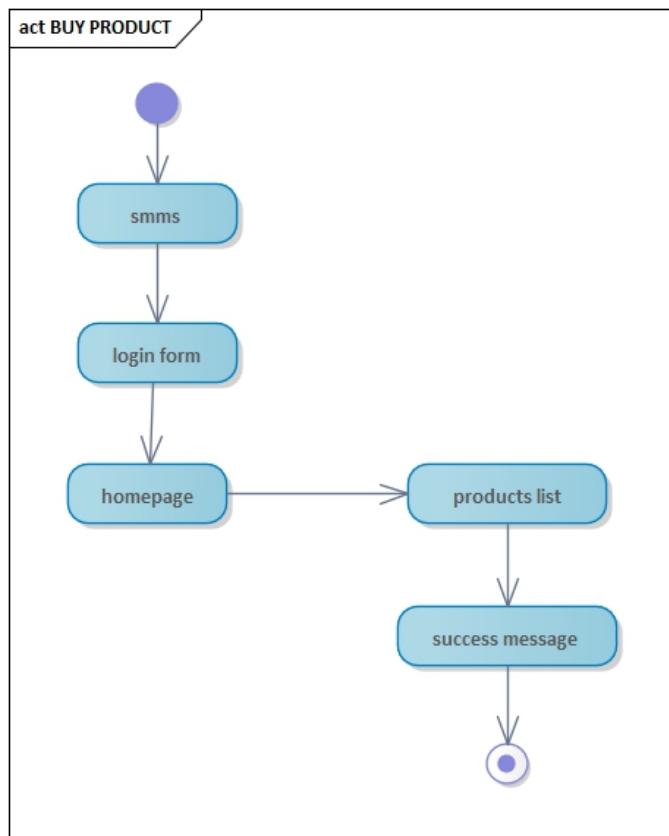
9. Buy product

Figure 3.8.3.9 state diagram of buy product

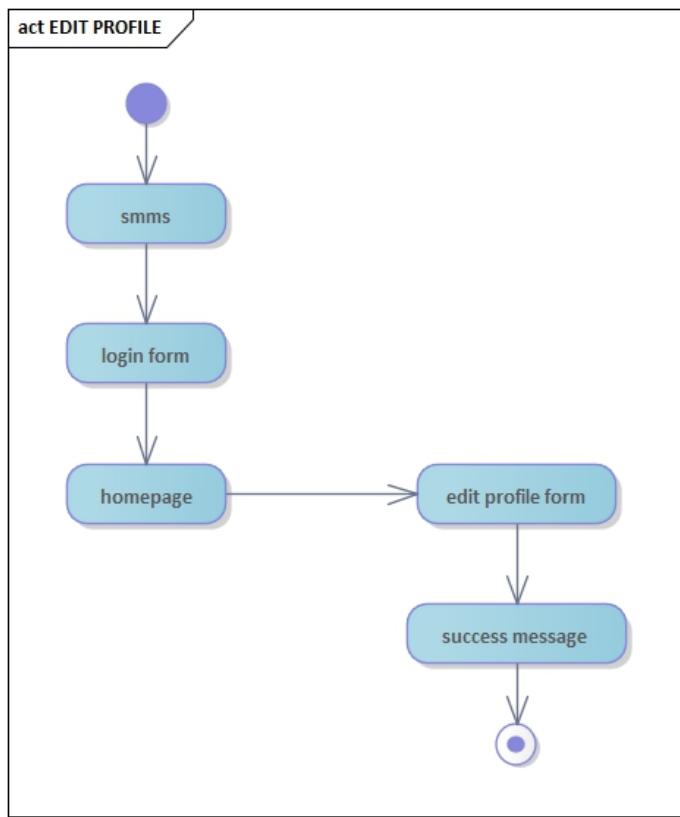
10. Edit profile

Figure 3.8.3.10 state diagram of edit profile

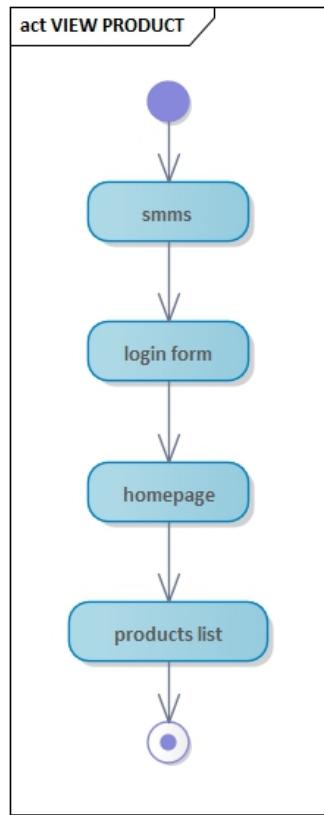
11. View product

Figure 3.8.3.11 state diagram of view product

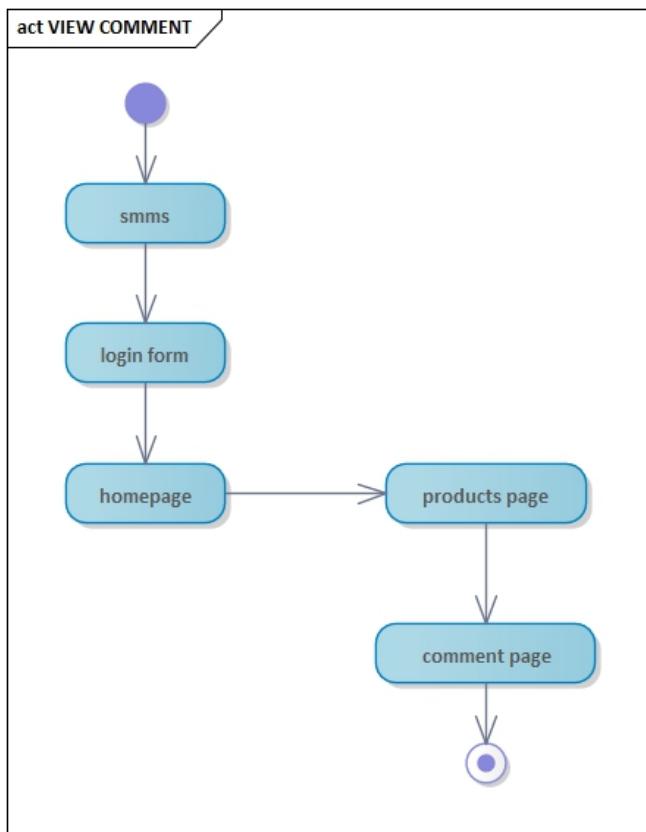
12. View comment

Figure 3.8.3.12 state diagram of view comment

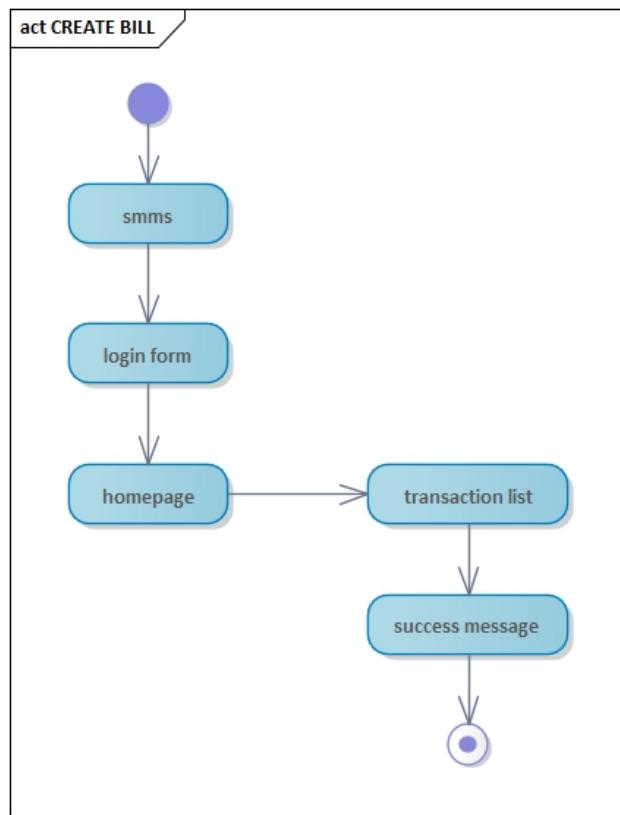
13. Create bill

Figure 3.8.3.13 state diagram of create bill

Chapter 4

System Design

4.1 Overview

This is the System Design document is for the Yaleself supermarket management system. The document includes the design goals, the proposed system Design and the object design.

4.1.1 Purpose of the system

This document describes the design issues of the overall system. It provides the complete architectural overview of the proposed system. It is intended to capture and express the significant architectural decisions which have been made on the system.

4.1.2 Design Goal

The Design Goals specify the qualities of the system that should be achieved and addressed during the design of the system. The design goals for the system are grouped into four categories. These are:

1. Performance
2. Dependability
3. Maintenance
4. End user

1. Performance

The system should be fast to respond to users request. Such as viewing products, ordering products, and give comments. The system performs its task within a user acceptable time and space. This includes the following: -

- **Response time:** - depending on the strength of available network the system should be response in short period of time.
- **Storage space:** -to do work efficiently the processor to be more than 2GB RAM and HD storage to be more than 150MB.

2. Dependability

Our system includes the following dependability criteria's: -

- **Reliability:** YSMMS system should be reliable.
- **Fault Tolerance:** - YSMMS should be fault tolerance even if there is unexpected input.
- **Security:** -YSMMS should be secured and should not allow user to access unauthorized data.
- **Availability:** - YSMMS should be available 24/7.

3. Maintainability

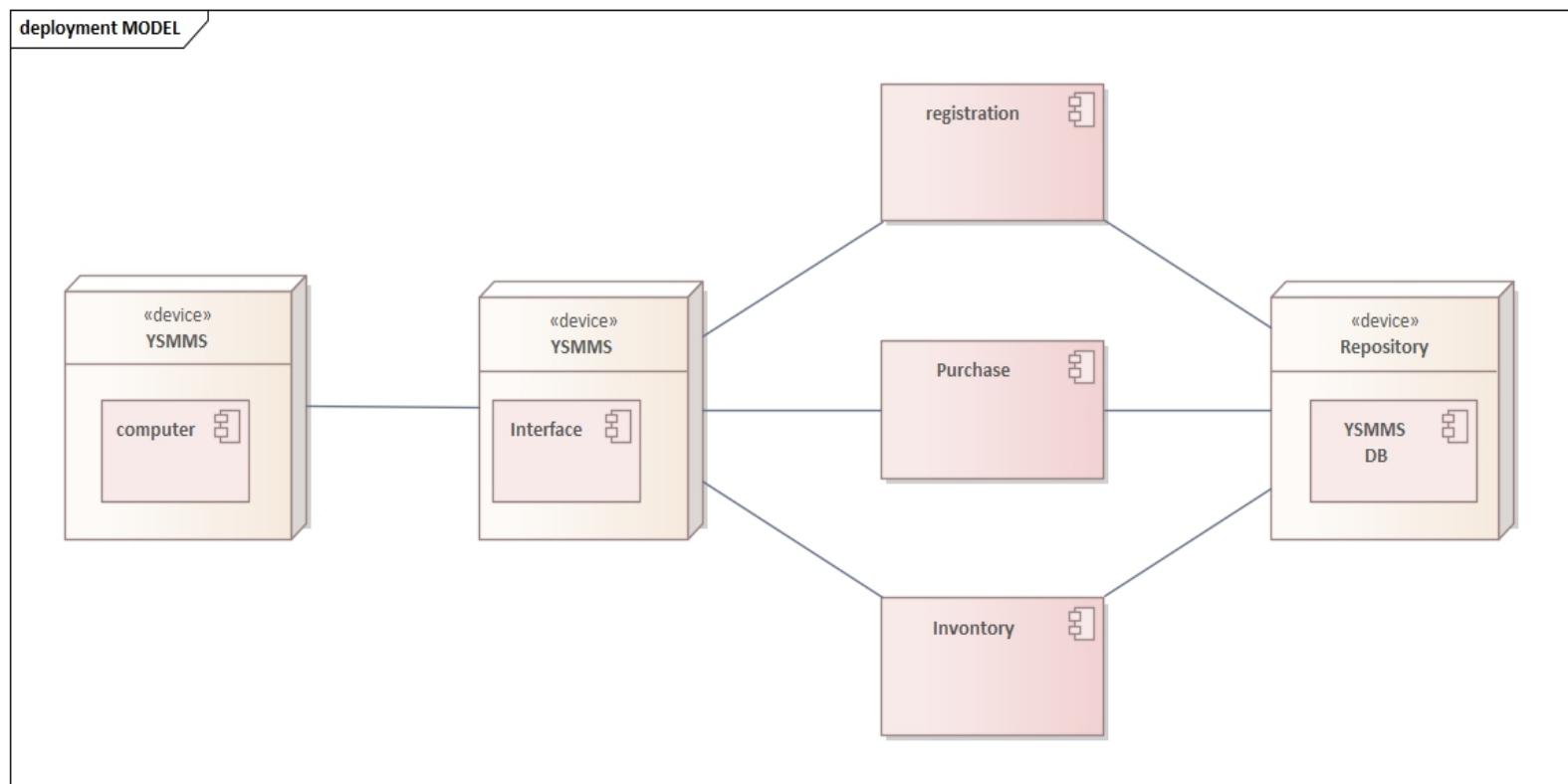
To be maintainable the system should meet the following maintenance criteria:-

- **Modifiability:** YSMMS system should be modifiable for further modification and enhancement of the system.
- **Portability:** - the system is developed to be viewed and retrieved from any web browser regardless of their version and platform it resides in it.
- **Extensibility:** - if it is needed to add new functionality to the system, this must be achieved by only making a separate page and integrate this page with the existing system.
- **Readability:** - our system code is readable and understandable by itself.

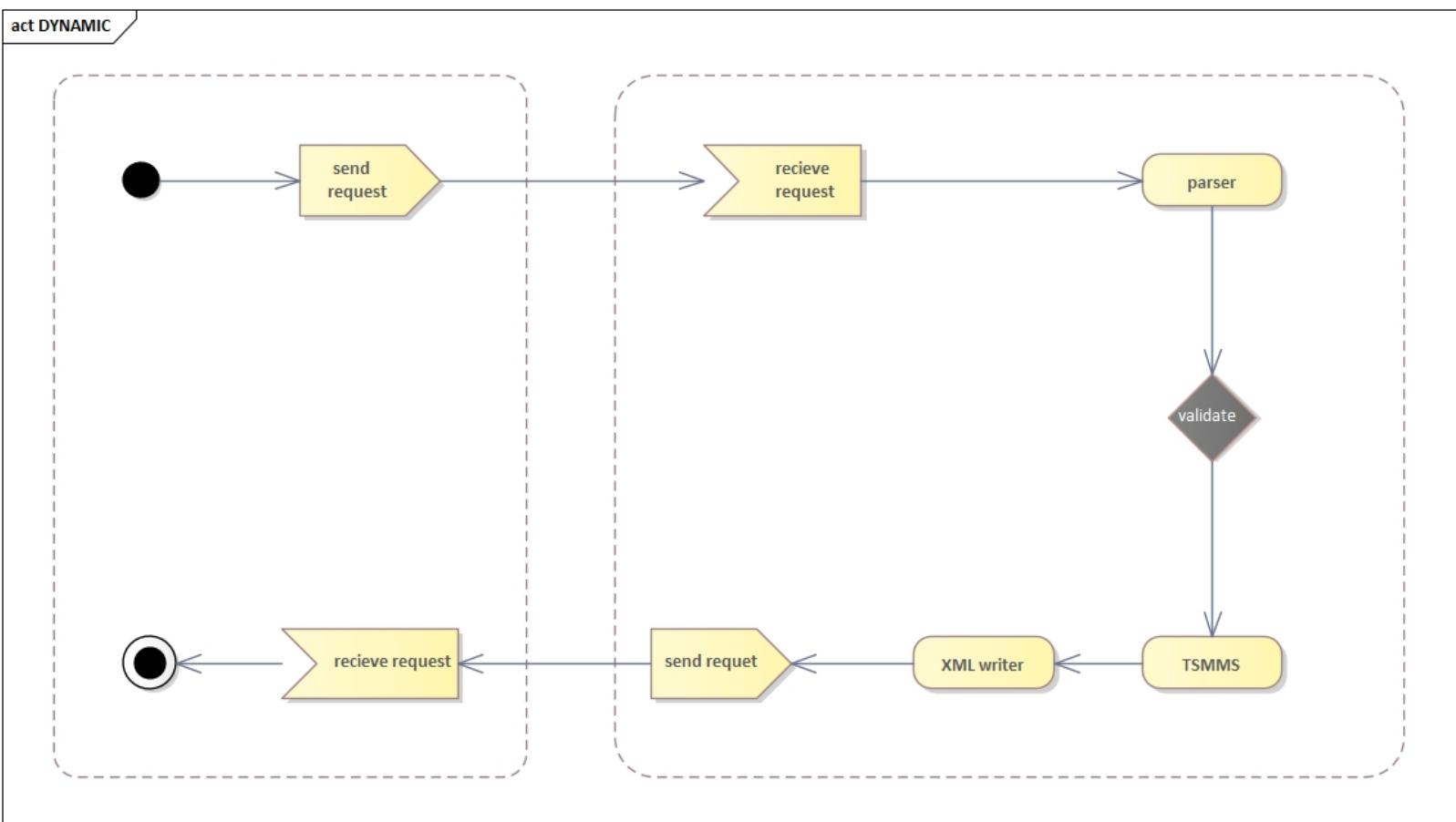
4. End User Criteria

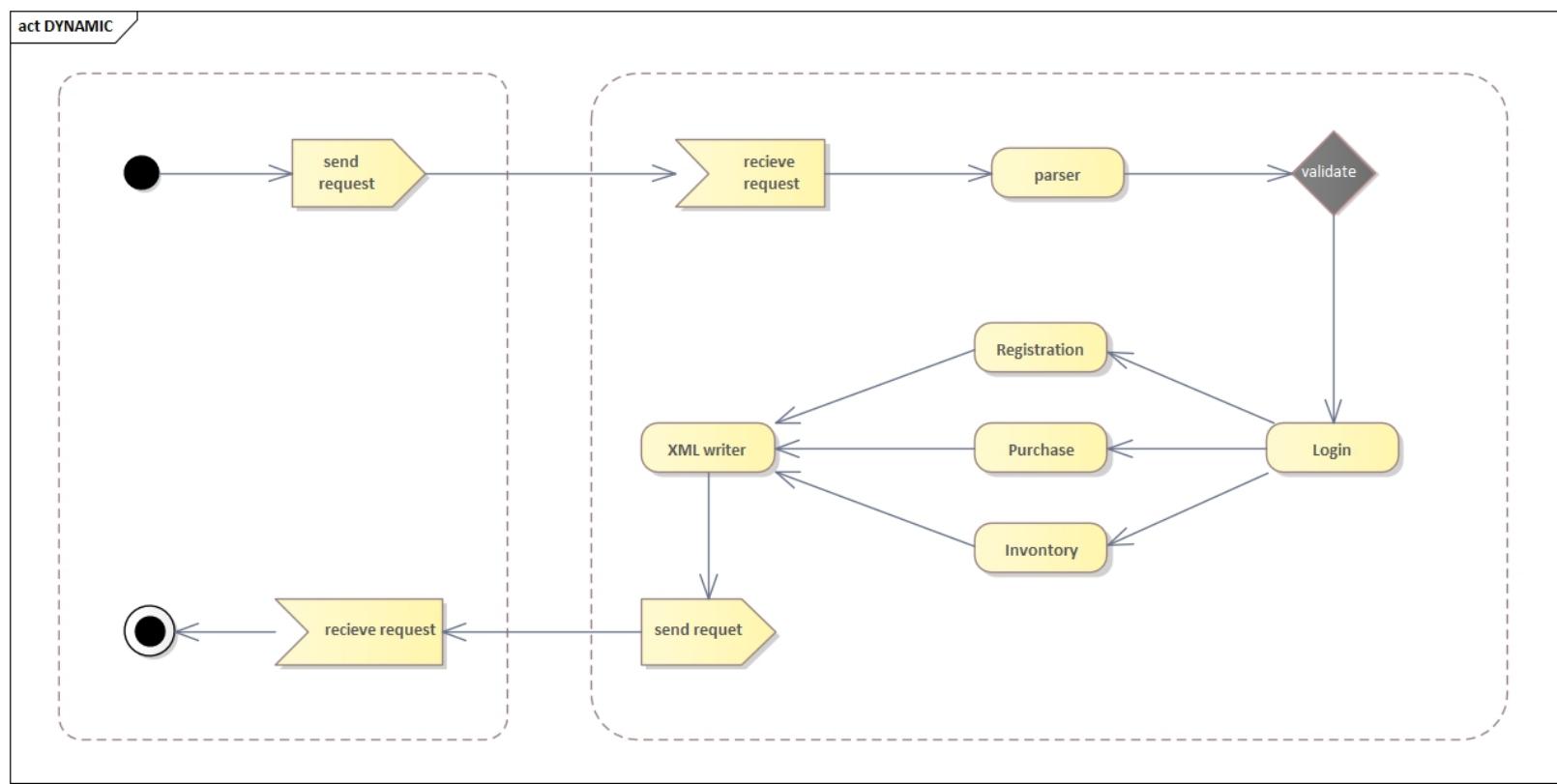
The system should have simple and understandable graphical user interface such as forms and buttons, which have descriptive names. It should give reliable response for each user comment. All the interfaces, forms and buttons are written or designed in a simple language or common language so that the user can access it without any difficult.

4.2. Proposed system architecture



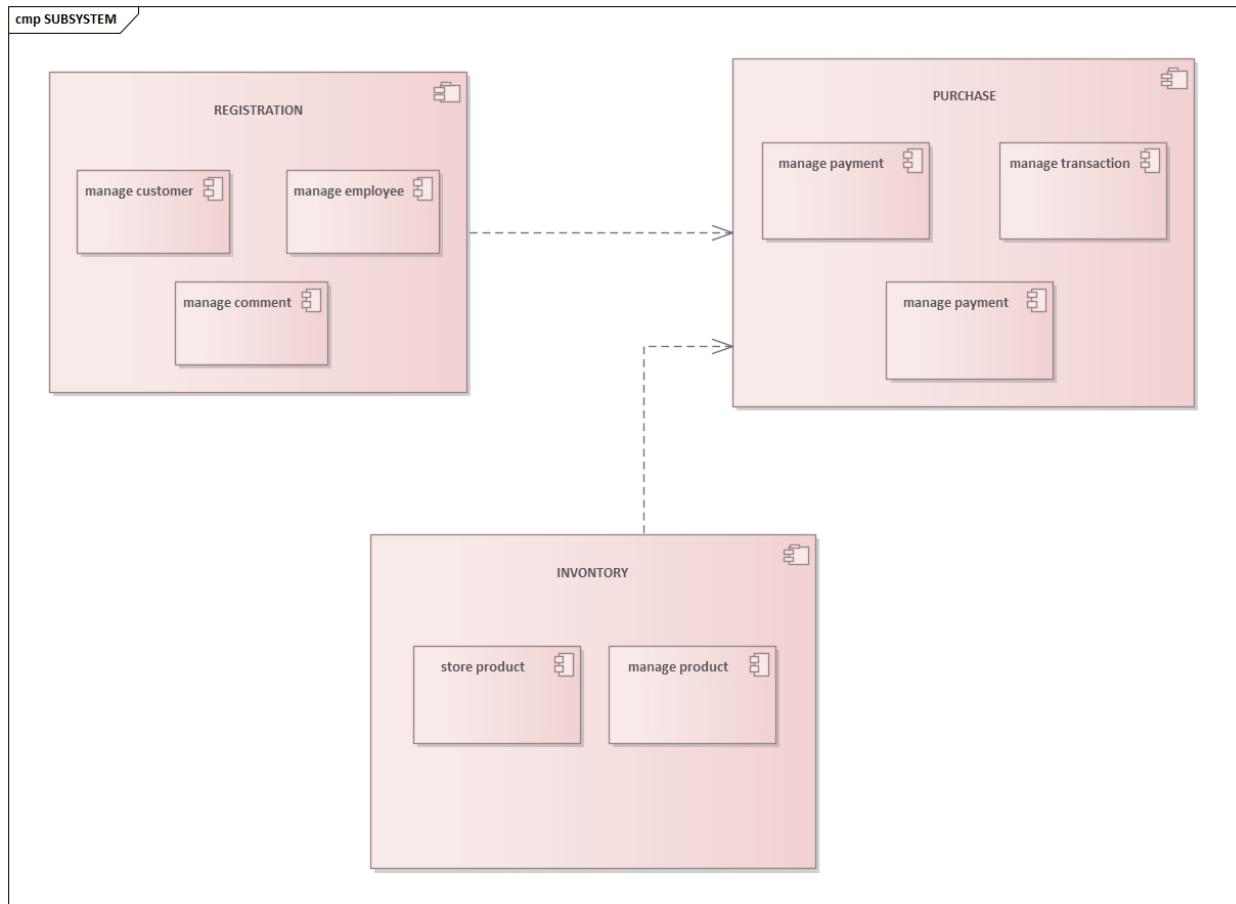
4.2.1 System process





4.2.2 Subsystem decomposition

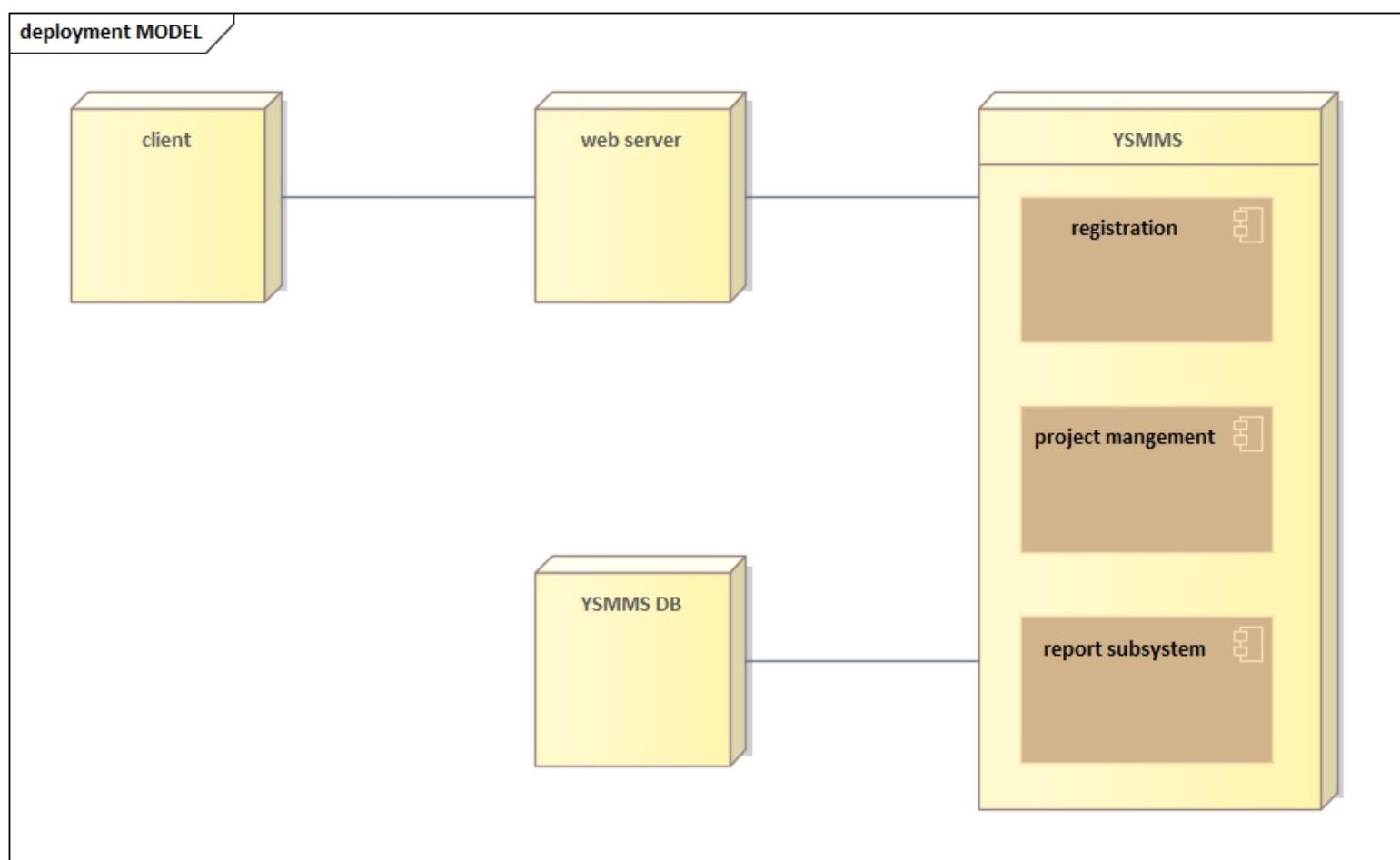
To reduce the complexity of the solution domain, we decompose a system into simpler parts, called subsystems. The main need of this portion is to design the external part of the system. In this project, there are three sub-system decompositions.



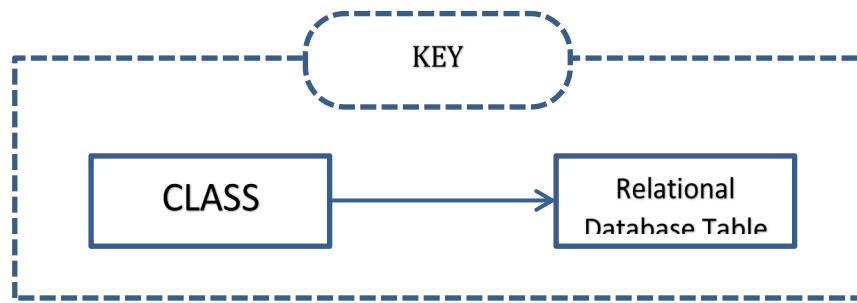
4.2.3 Hardware/ software mapping

When we say hardware/software mapping for the system, it describes how subsystems are assigned to hardware and off-the-shelf components. It also lists the issues introduced by multiple nodes and software reuse.

In this system design mainly there are three hardware components. The client side, server side and database side. When the team applies the system, necessary software will be loaded to each side hardware components. Network should be installed between each side. Then each sub system software will be assigned and configured to the mapped hardware. Then the local area network will be connected to the internet and the system will be functional. But now it is a design phase. The hardware software mapping of the system is described below with a simple diagram.



4.2.4. Persistent data management



The purpose of this section is to show the mapping of the objects/classes of the system, identified during the analysis stage, in to the corresponding relational database.

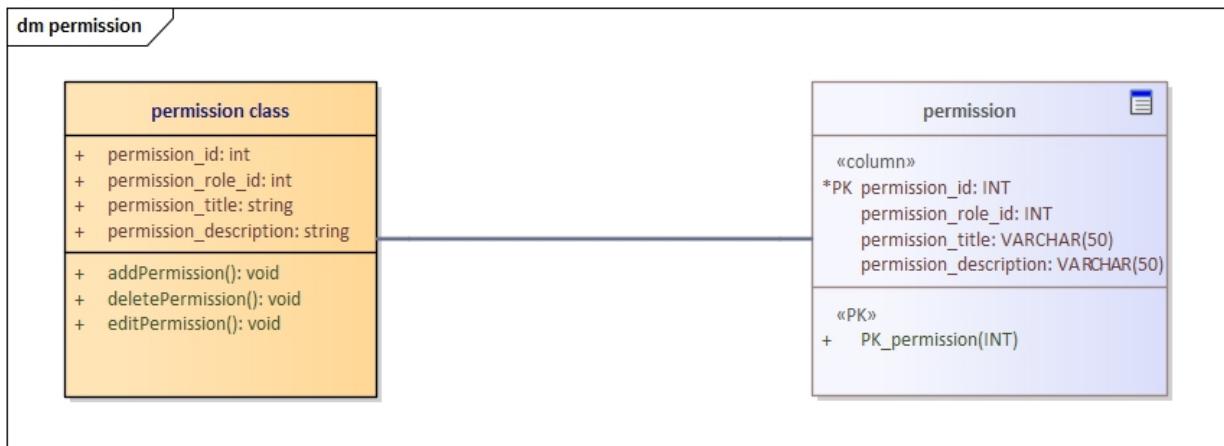


figure 4.2.4.1 database diagram for permission class

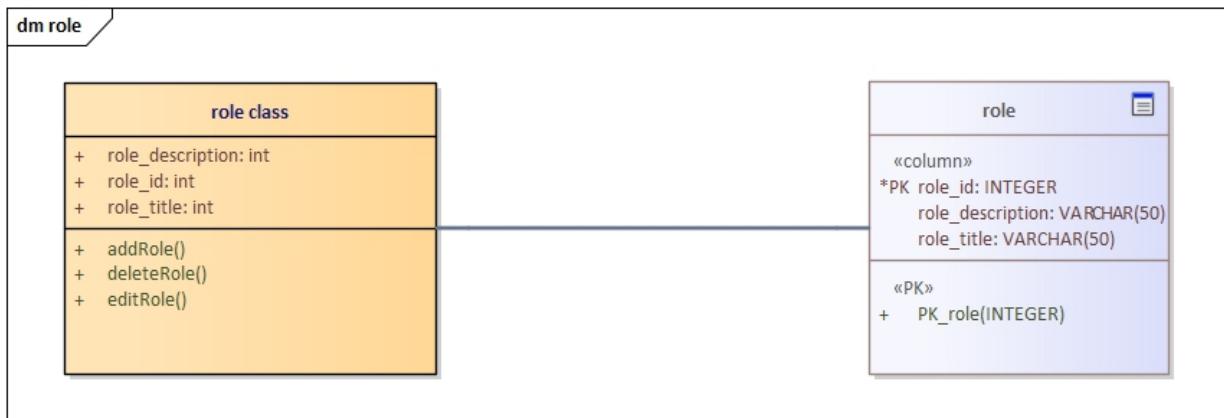


figure 4.2.4.2 database diagram for role class

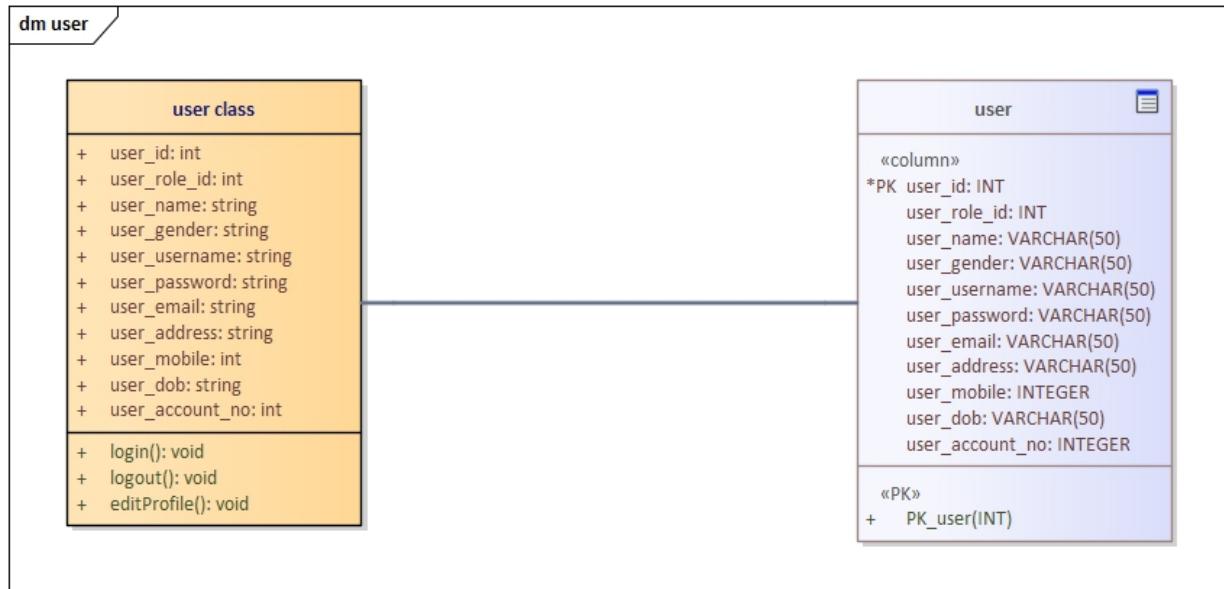


figure 4.2.4.3 database diagram for user class

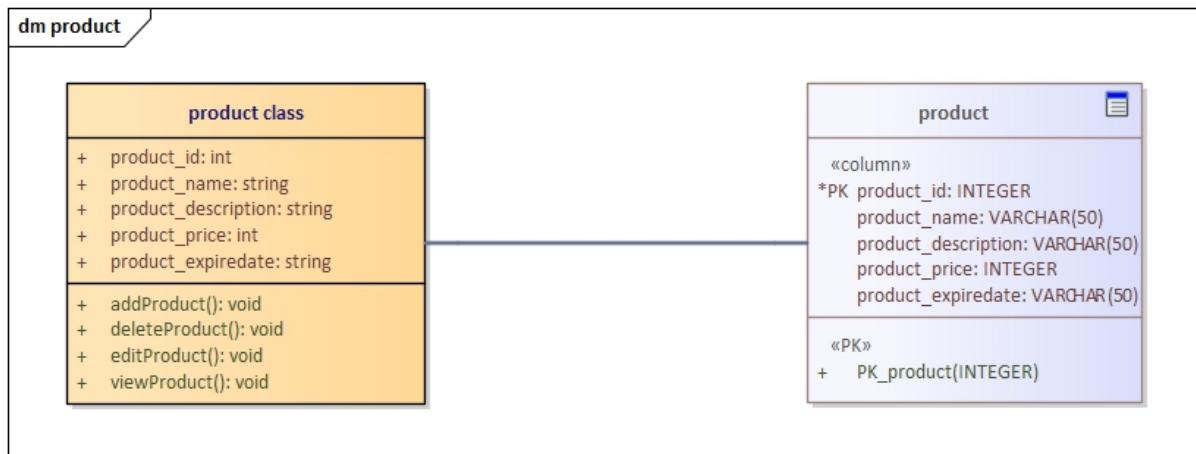


figure 4.2.4.4 database diagram for product class

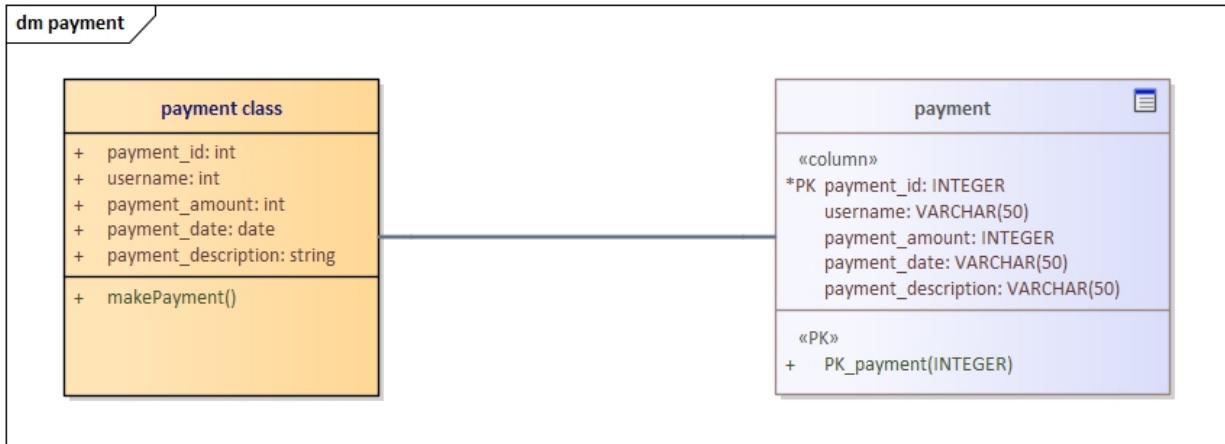


figure 4.2.4.5 database diagram for payment class

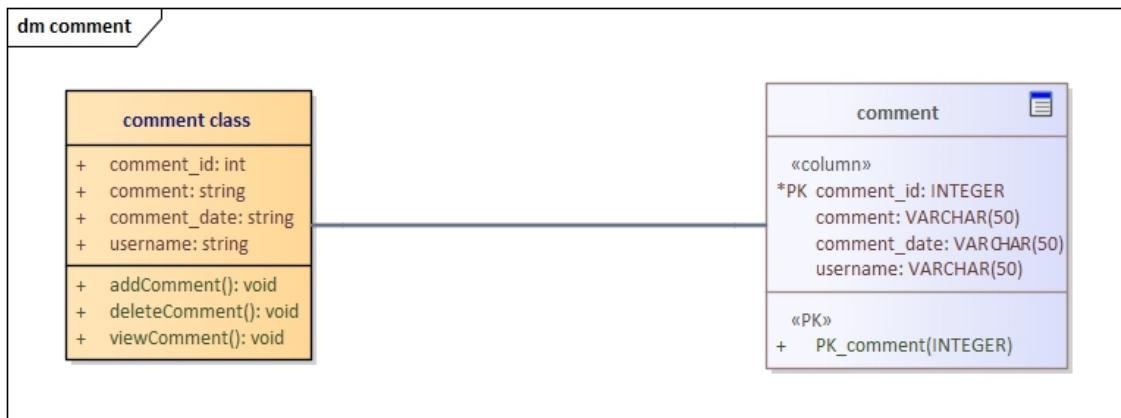
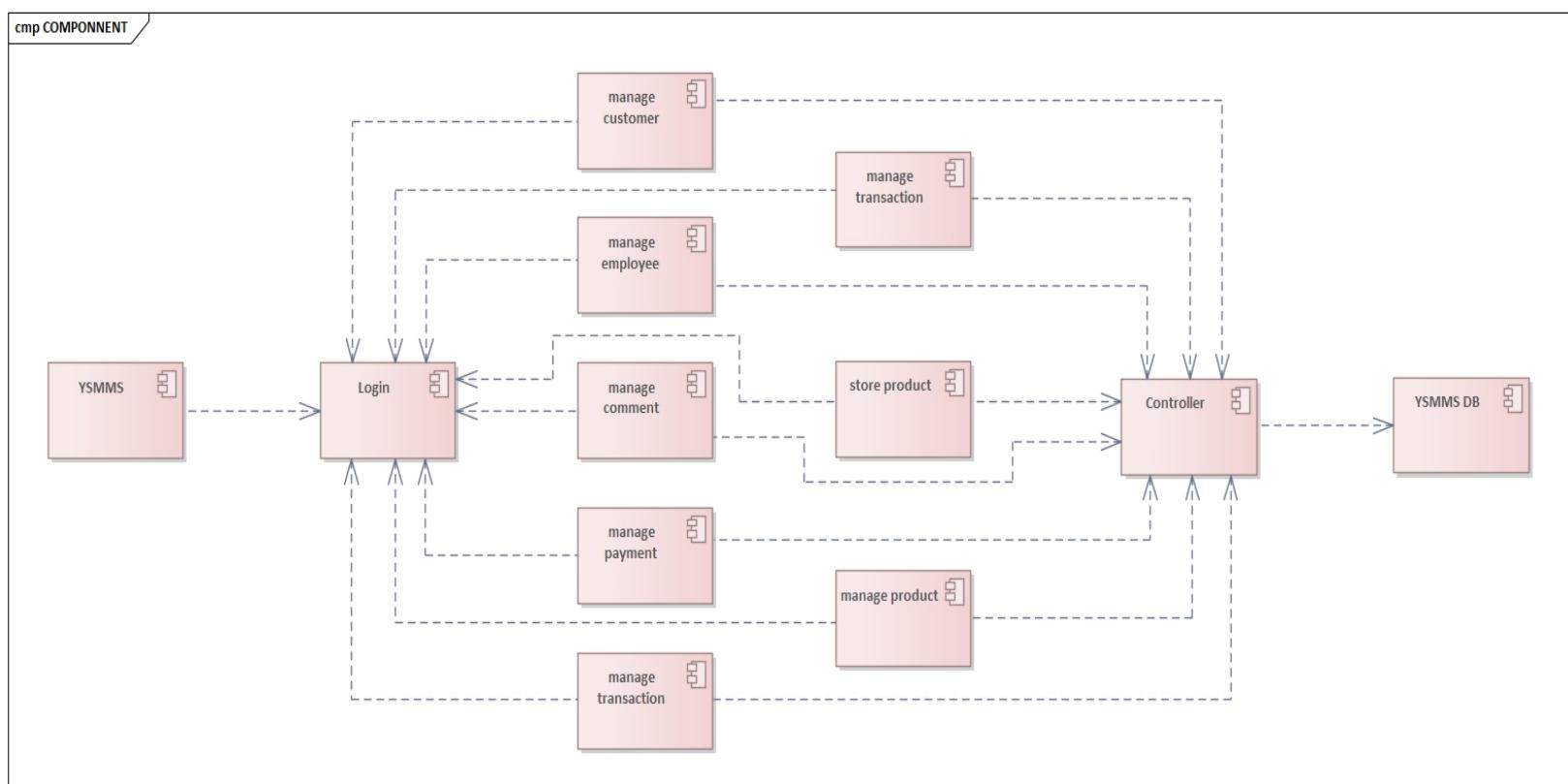


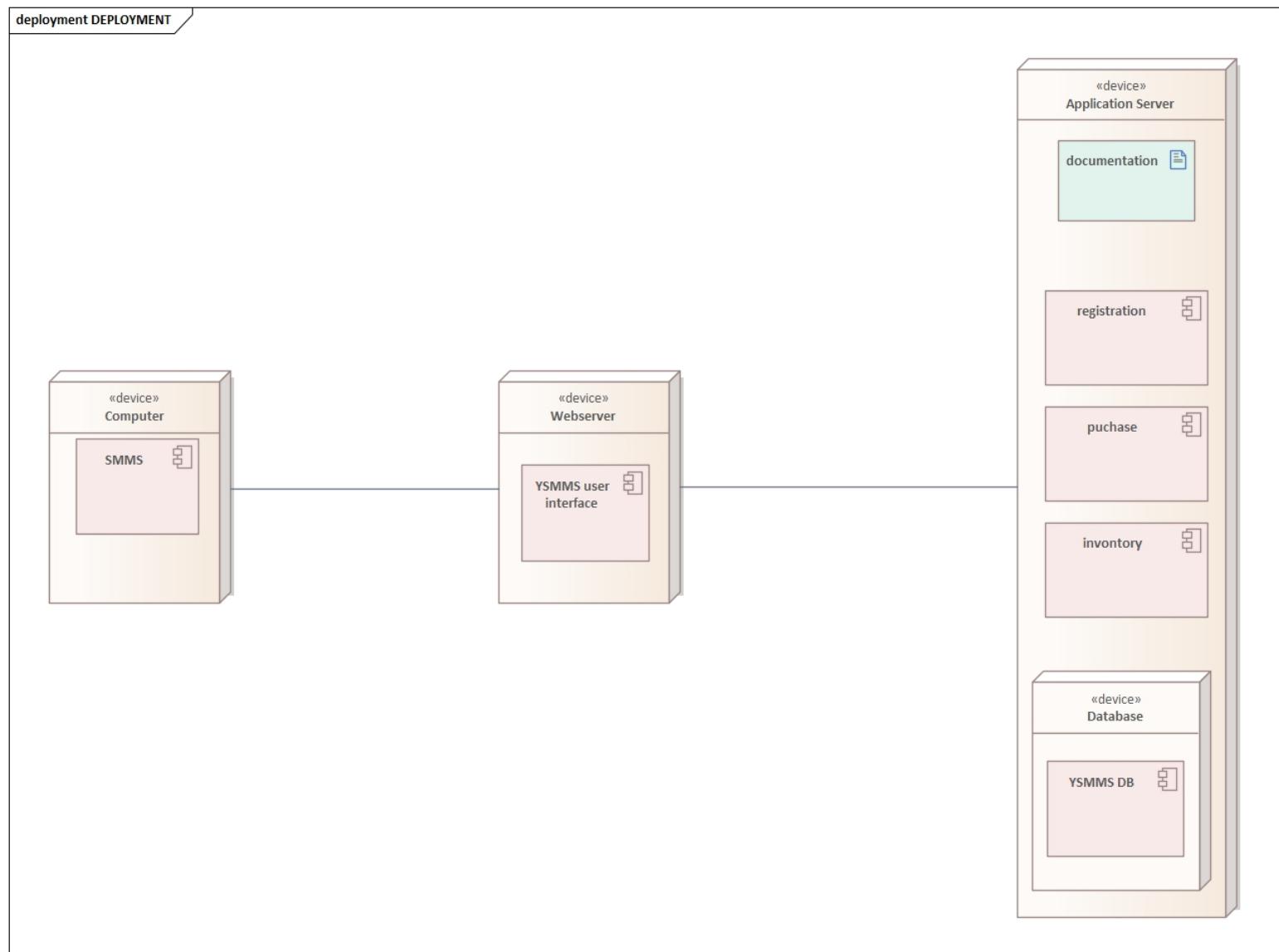
figure 4.2.4.6 database diagram for permission class

4.2.5. Component diagram

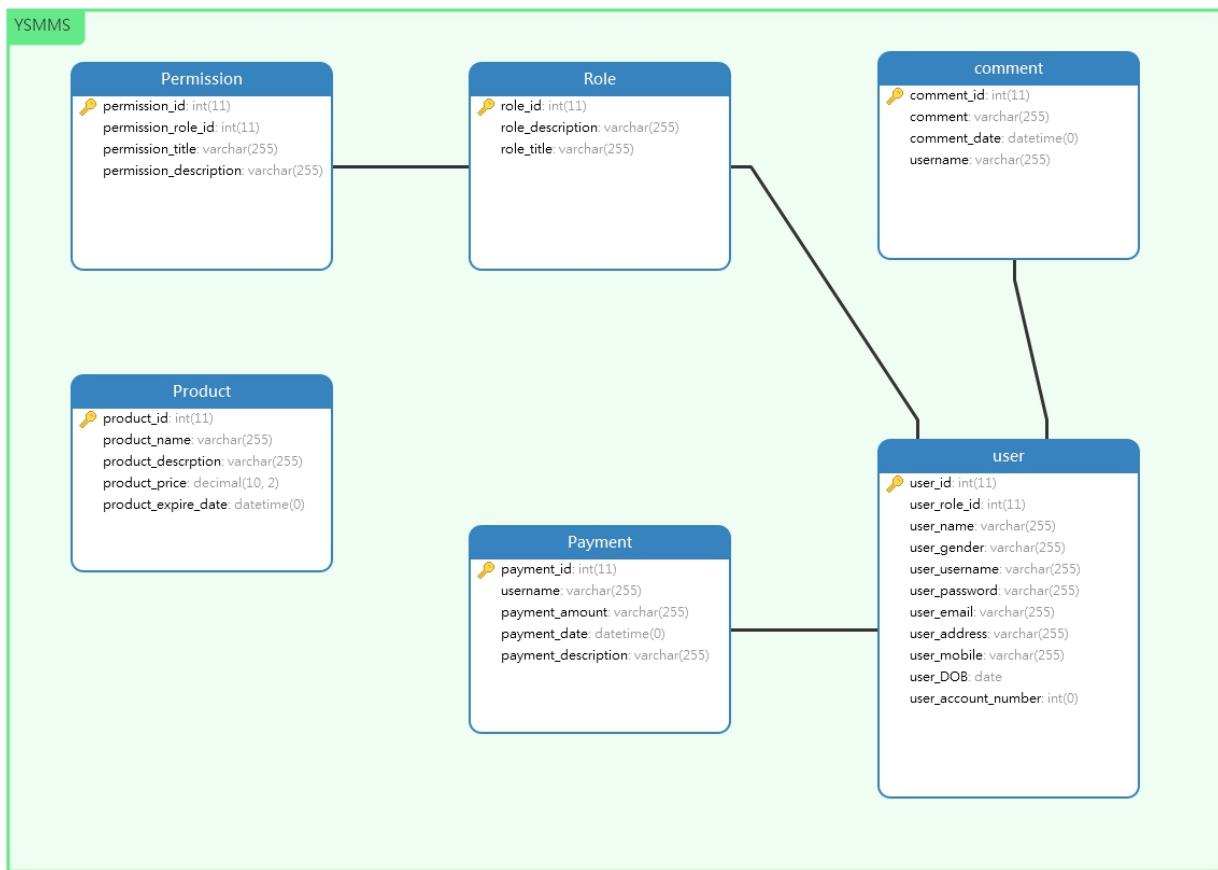
The following component diagram represents a group of graphs of components connected by dependency relationships and dependencies are shown as dashed arrows from the client component to the supplier component.



4.2.6 Deployment diagram



4.2.7 Database design



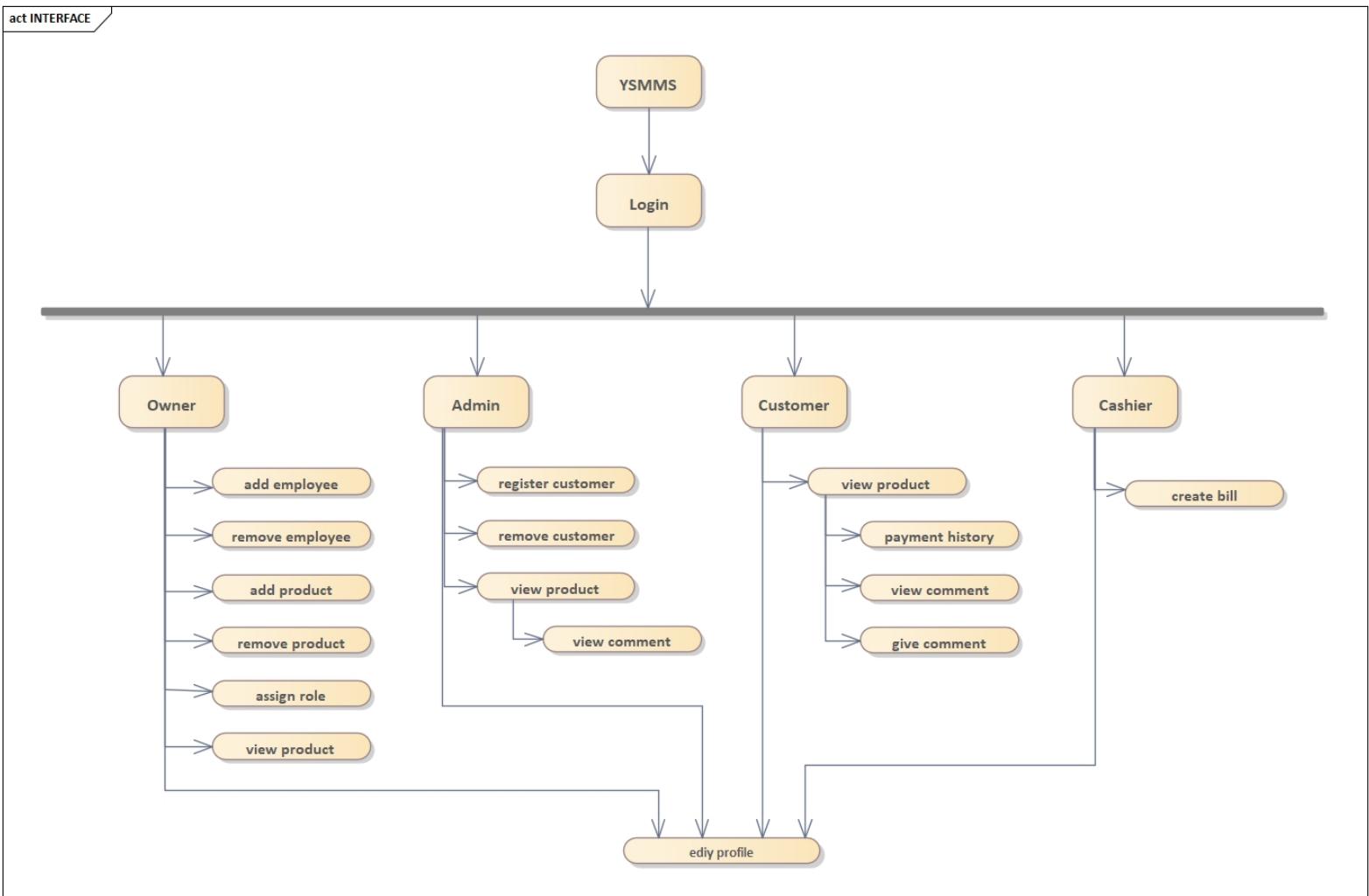
4.2.8 Access control

In multiuser systems, different actors have access to different functionality and data. We modeled these distinctions by associating different **use cases** to different actors. This is described by the following access matrix.

| | Role | Permission | Product | Comment | Payment | User |
|----------|---|--|---|------------------------------------|----------------|---|
| Owner | Assign role() Add role() Delete role() Edit role() | Add permission() Edit permission() Delete permission() | View product() Add product() Remove product() | View comment() | | Add product() Remove product() Edit profile() |
| Admin | | | View product() | View comment() delete comment() | | Add customer() Remove customer() Edit profile() |
| Cashier | | | | | Create bill() | Edit profile() |
| Customer | | | View product() buy product() | View comment() add comment() | Make payment() | Edit profile() |

4.2.9 User interface design

Navigational paths



Software constraints

For **Owner**, enter

Username: **owner**

Password: **owner**

For **admin**, enter

Username: **admin**

Password: **admin**

For **customer**, enter

Username: **customer**

Password: **customer**

We worked on the login and view product section only, because of time limitation.

on view product add to Wishlist feature is working.