

REPUBLIQUE DU CAMEROUN
PAIX – TRAVAIL – PATRIE

MINISTERE DE
L'ENSEIGNEMENT SUPERIEUR

FACULTE D'INGENIERIE ET DE
TECHNOLOGIE



REPUBLIC OF CAMEROON
PEACE – WORK – FATHERLAND

MINISTRY OF HIGHER
EDUCATION

FACULTY OF ENGINEERING
AND TECHNOLOGY

***** UNIVERSITY OF BUEA *****

DEPARTMENT OF COMPUTER ENGINEERING
COURSE TITLE: INTERNET PROGRAMMING AND MOBILE
PROGRAMING
COURSE CODE: CEF440

TASK
PRACTICAL DESIGN OF A FOOD WASTE
MANAGEMENT SYSTEM

COURSE INSTRUCTOR: Dr KAMENI VALERY

Group 12 Members:

S\N	NAMES	MATRICULE
I.	AYAMBA BLESS BISSONG	FE20A018
II	NGONG MARINUS NGONG	FE20A077
III	NDIBONG ENOW EBAI-TABOT	FE20A072
IV	MBI ENOW LEONARD APELGRYN	FE20A063
V	AMINJA SIMPLICE AJONG	FE20A009

Table of content

1	INTRODUCTION AND TERMINOLOGIES.....
	• Problem Statement.....
	• Purpose.....
	• Scope.....
	• Objective.....
2	OVERALL DESCRIPTION
	• Product Perspective.....
	• Product Functions.....
	• User Characteristics.....
	• Constraints.....
	• Assumptions and Dependencies.....
3	DESIGN AND IMPLEMENTATION
	• Class diagram of the system.....
	• Sequence diagram of the system.....
	• Use case diagram of the system.....
	• Activity diagram of the system.....
4	IMPORTANCE OF VARIOUS DIAGRAMS
5	SOFTWARE TOOLS
	• Advantages and Disadvantages.....
6	CONCLUSION

1. INTRODUCTION

Food waste is a major issue that has significant economic, environmental, and social impacts. In order to address this issue, many organizations are turning to food wastage management system software to help them track and reduce food waste. This report will provide an overview of the system tools and design for such a software. By implementing a food wastage management system software, organizations can gain greater insight into their food waste and take steps to reduce it, resulting in cost savings and a more sustainable operation.

I. PROBLEM STATEMENT

Food waste is a major issue with significant impacts. One challenge in addressing food waste is the lack of accurate data on the types and amounts of food being wasted. A food wastage management system software can help organizations collect and analyze data on food waste, providing them with the insights they need to reduce waste and improve their operations.

II. PURPOSE

Is to help organizations collect and analyze data on food waste in order to identify key waste reduction opportunities and take effective action to reduce waste. By implementing a food wastage management system software, organizations can gain greater insight into their food waste and take steps to reduce it, resulting in cost savings and a more sustainable operation.

III. SCOPE

- System Design and Implementing a food wastage management system software

IV. **OBJECTIVES**

- To implement the design of a food wastage management system software that meets the data, functional, non-functional, and system requirements

2. OVERALL DESCRIPTION

I. Product Perspective:

The proposed system is a web-based application designed to reduce or stop food wastage in a community. It would allow market sellers, restaurant owners, charity organizations, and orphanages to register and upload their surplus food stocks at a cheaper price. Users would be able to register, view the different food options, and purchase food from the system at all times.

II. Product Functions:

- Permit market sellers to register and upload their food stocks into the system at a cheaper price.
- Permit restaurant owners to register their restaurants and display the different meals prepared in the system along with their prices.
- Permit users to register, view the different food uploaded by restaurants and market sellers, and purchase food from the system at all times.
- Permit charity organizations and orphanages to register into the system and purchase food from restaurants and market sellers.
- Integrate a payment system using an API from a payment provider.

III. User Characteristics:

- The users of this system would include market sellers, restaurant owners, charity organizations, orphanages, and individual users.

IV. Constraints:

Some potential constraints for this project could include:

- Limited resources for development and maintenance of the system.
- Ensuring that all parties involved are able to use the system effectively.
- Ensuring that the payment system is secure and reliable.

V. Assumptions:

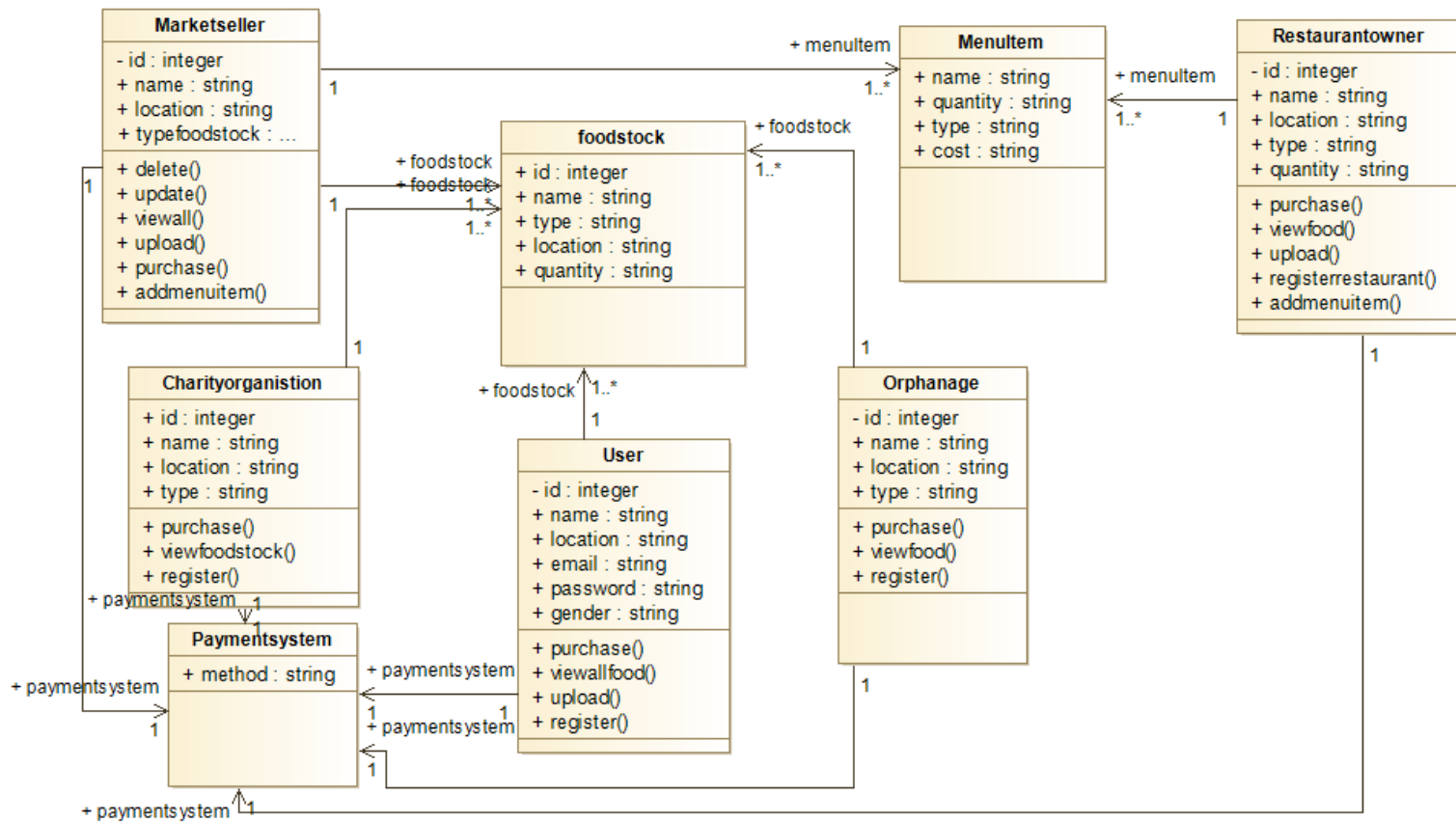
- It is assumed that market sellers, restaurant owners, charity organizations, and orphanages in the community would be willing to participate in the system and upload their surplus food stocks at a cheaper price.
- It is assumed that users would be interested in using the system to purchase food at a cheaper price.
- It is assumed that a payment provider would be available to integrate with the system for processing payments.

VI. Dependencies:

- The development of this system would depend on the availability of resources such as funding and personnel.
- The success of this system would depend on the participation of market sellers, restaurant owners, charity organizations, and orphanages in the community.
- The system would depend on a reliable internet connection for users to access and use it.

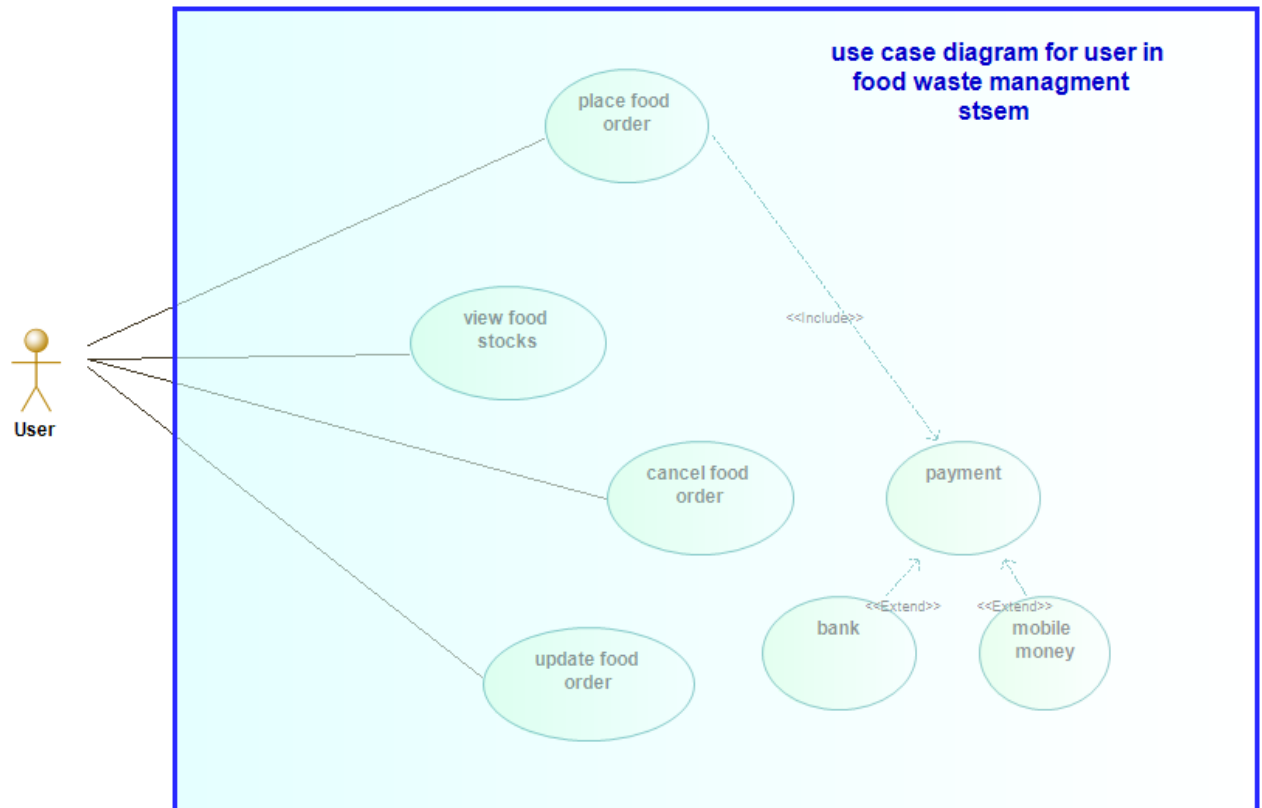
3. DESIGN AND IMPLEMENTATION

a. CLASS DIAGRAM OF THE SYSTEM

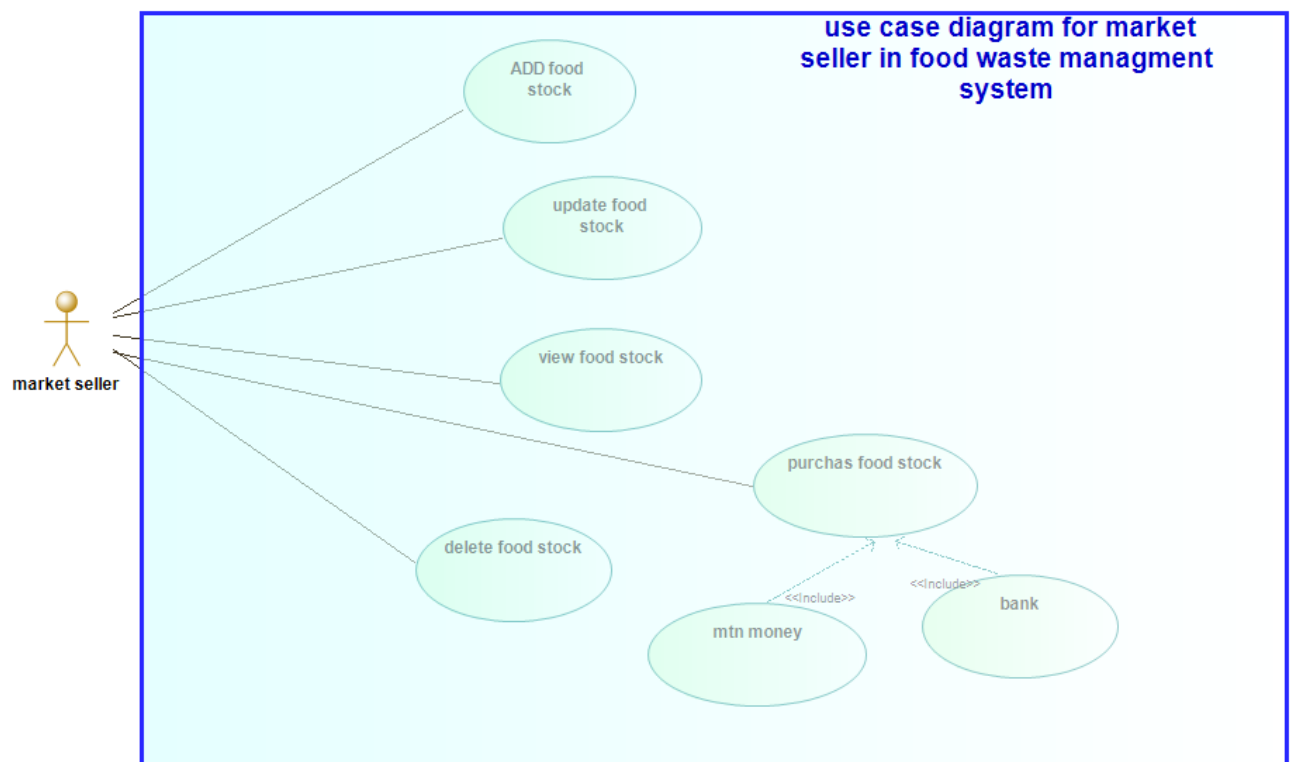


b. USE CASE DIAGRAMS

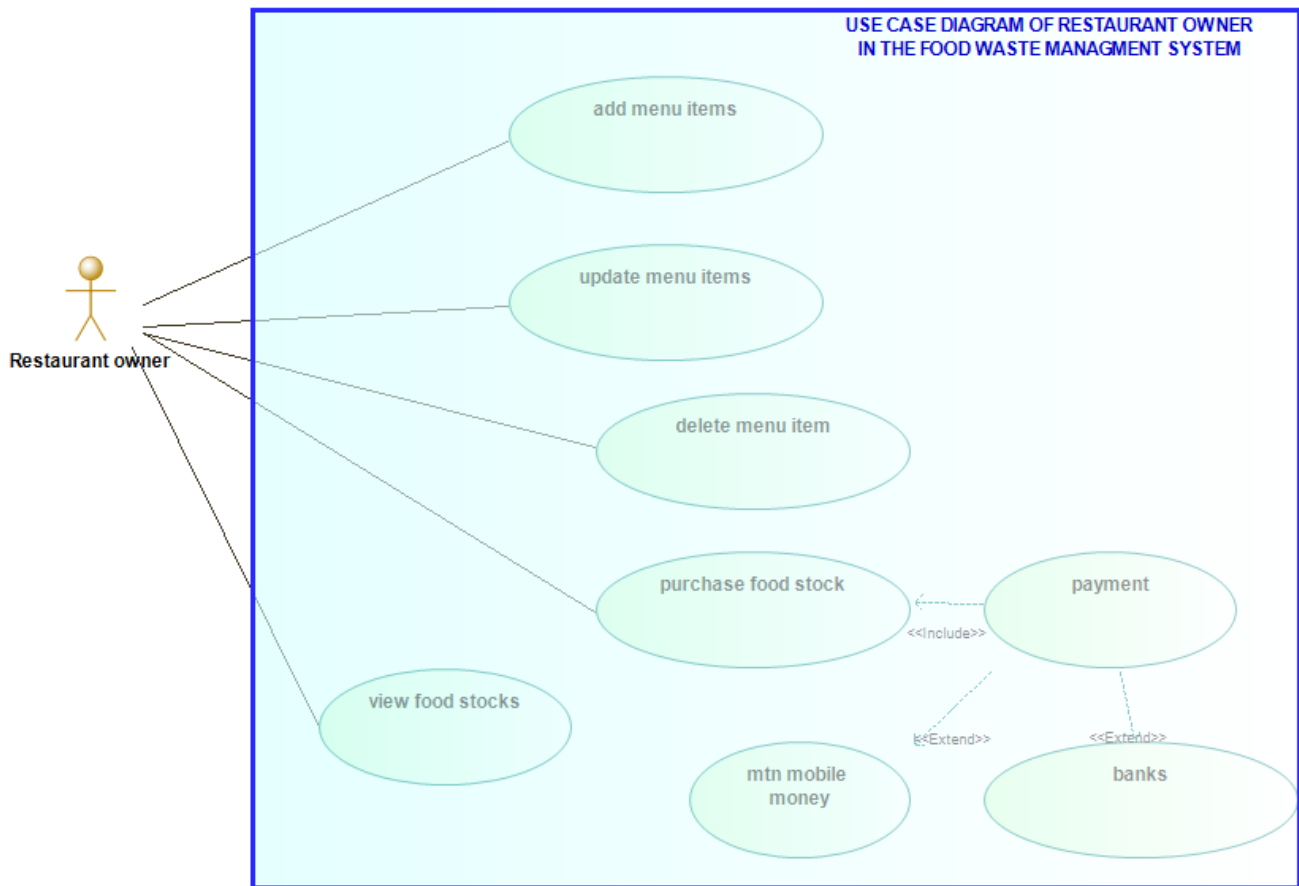
❖ User



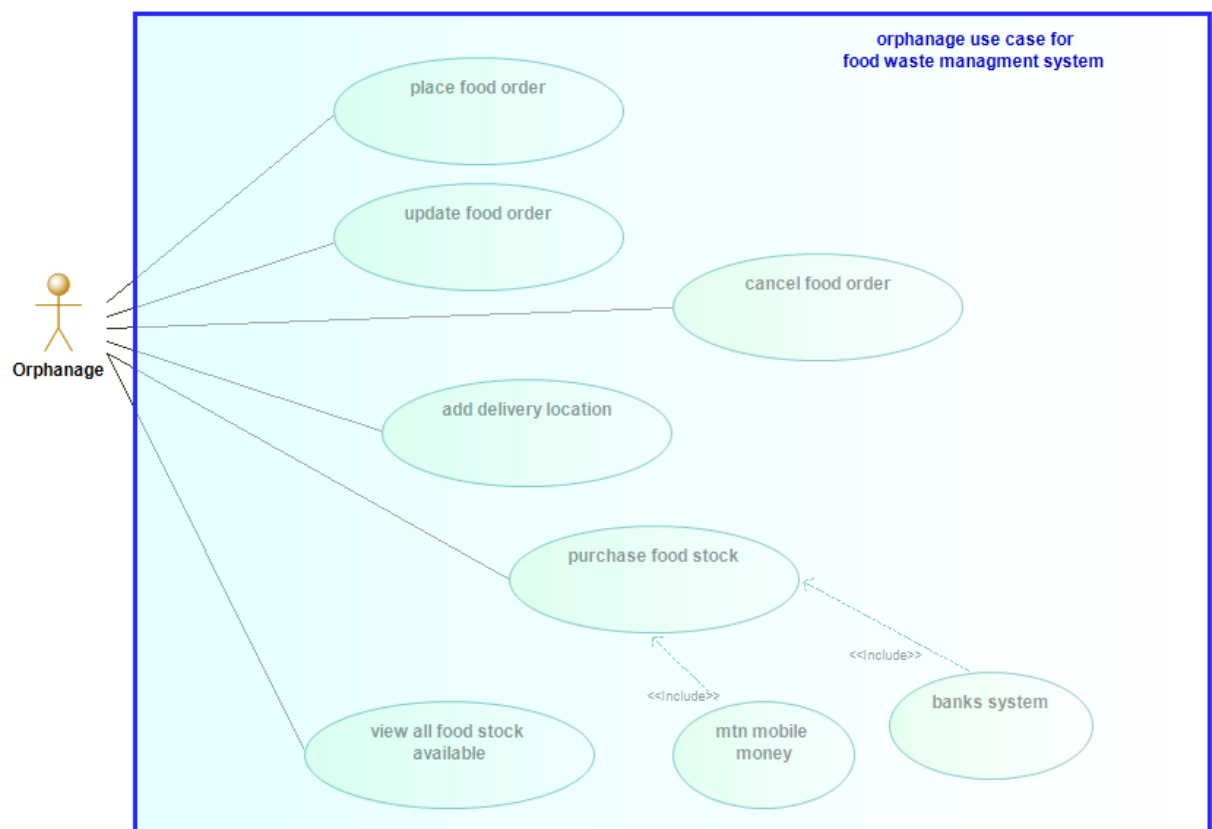
❖ Market Seller



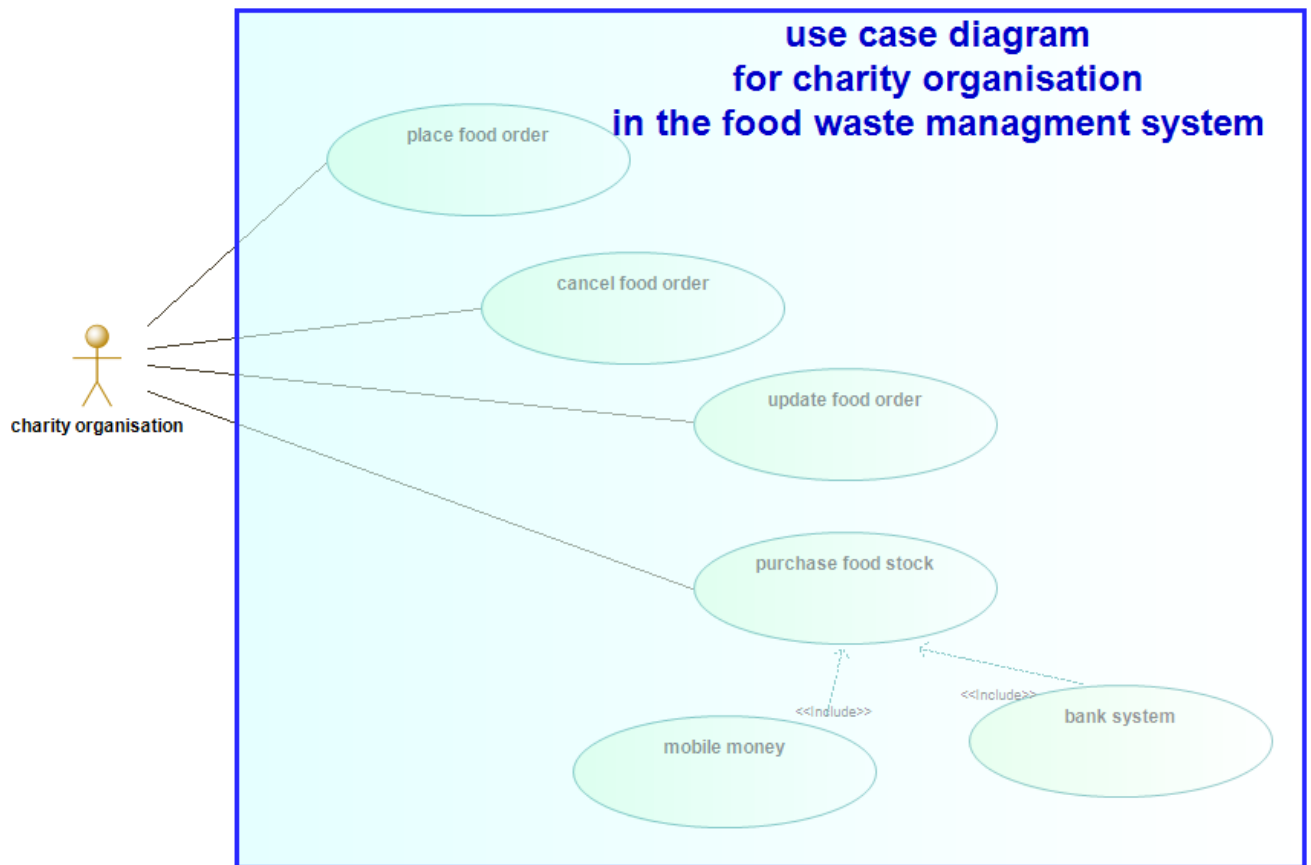
❖ Restaurant owner



❖ Orphanage

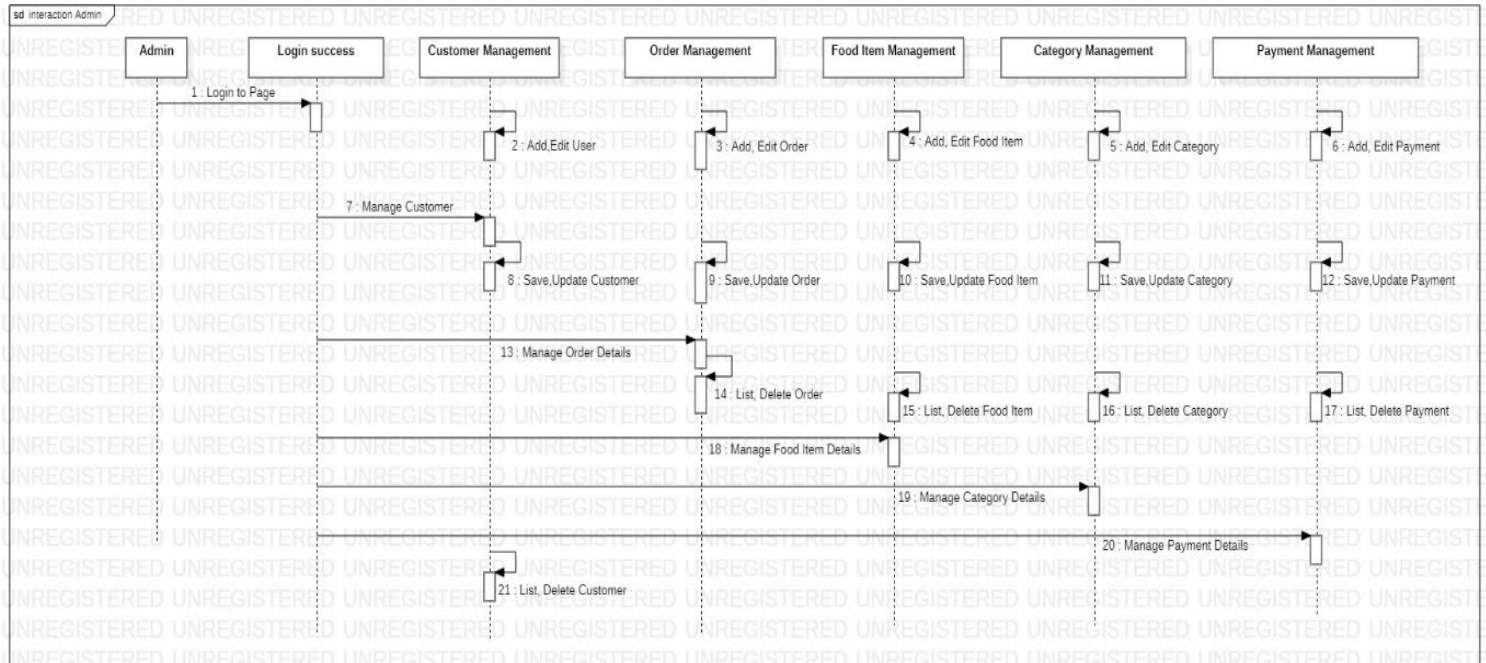


❖ Charity Organisation

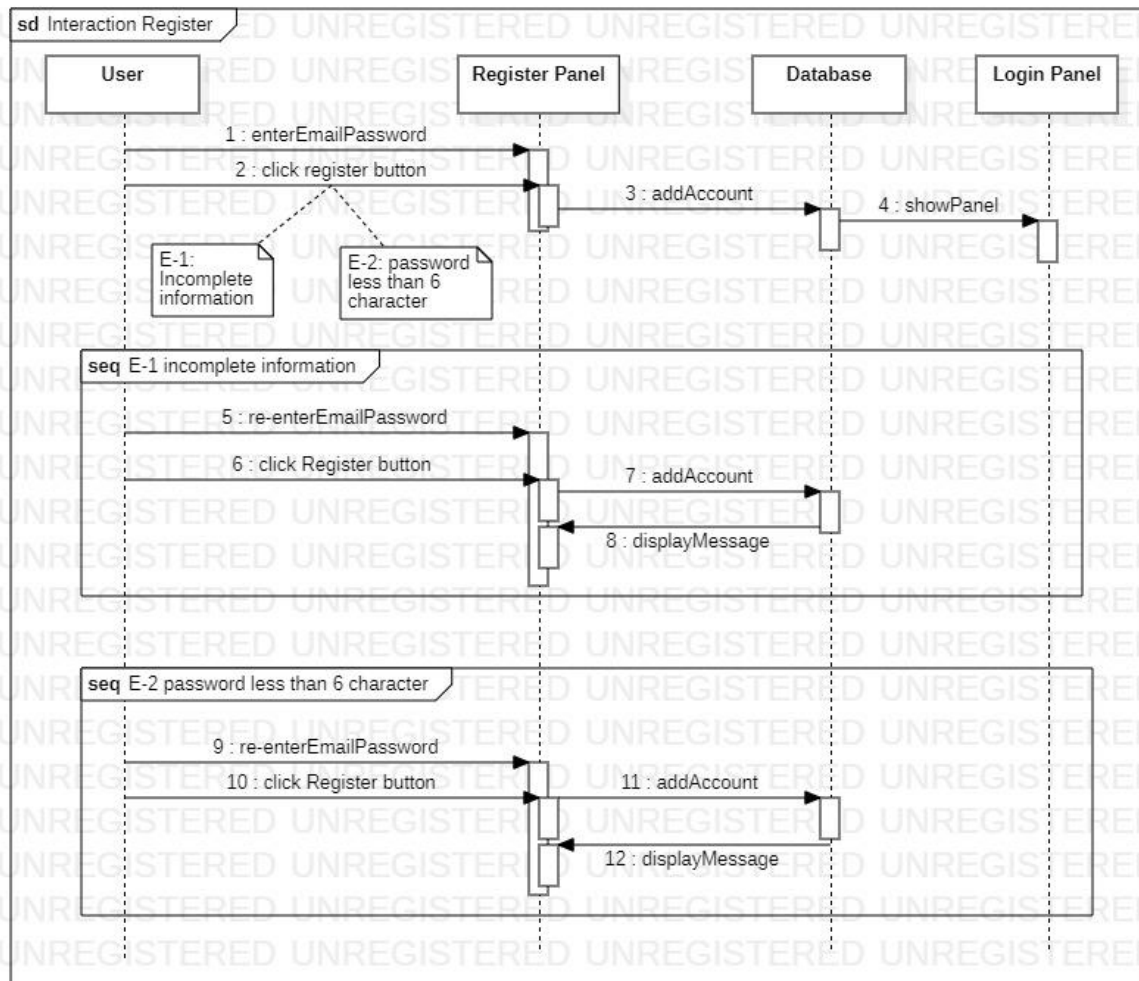


c. SEQUENCE DIAGRAMS

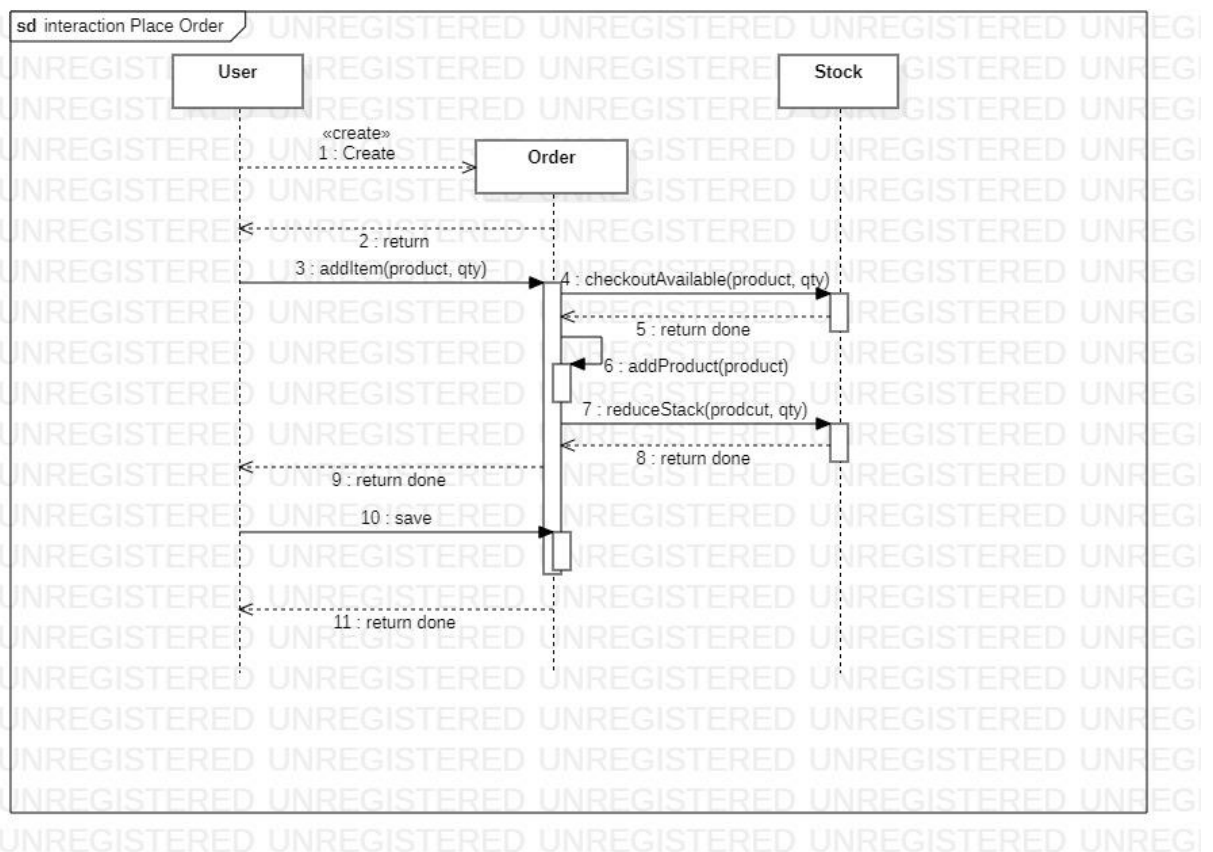
❖ Admin



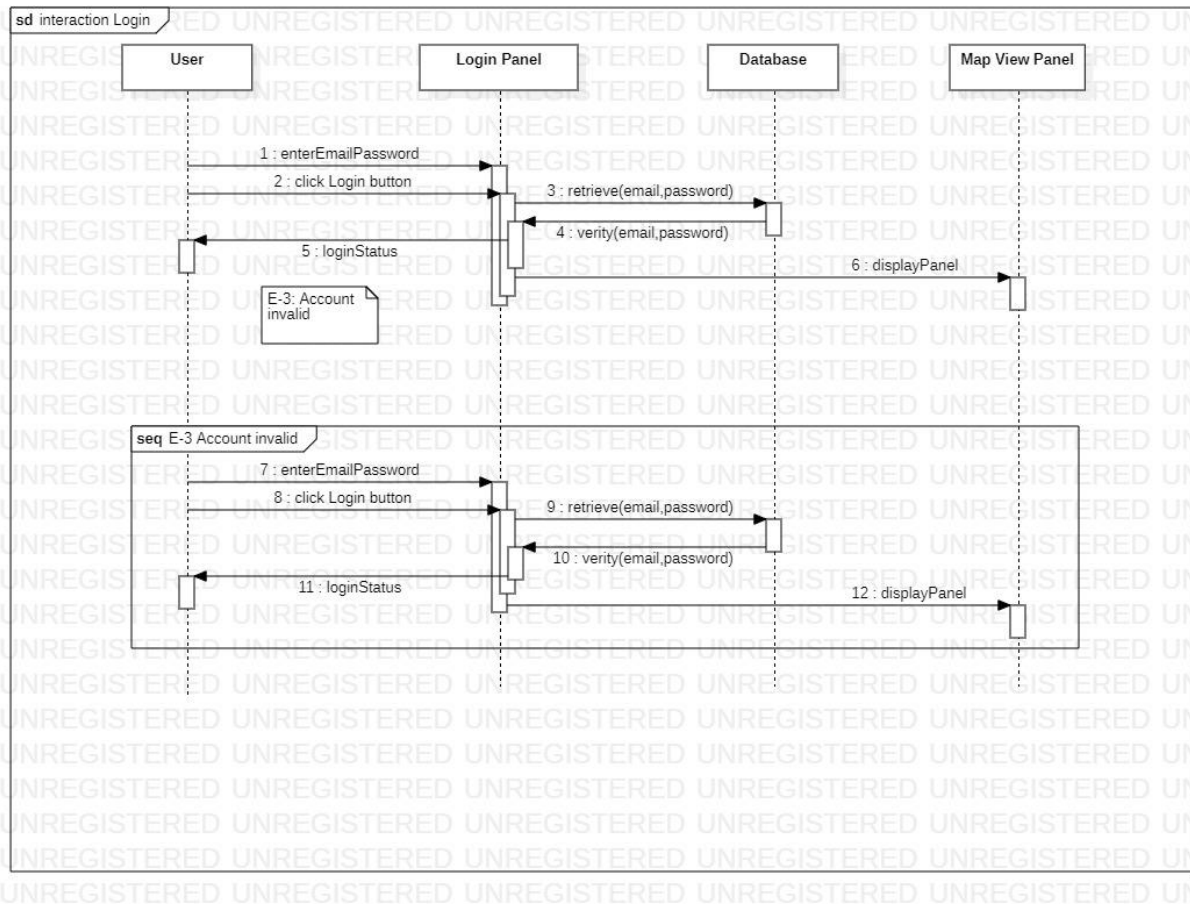
❖ Register:



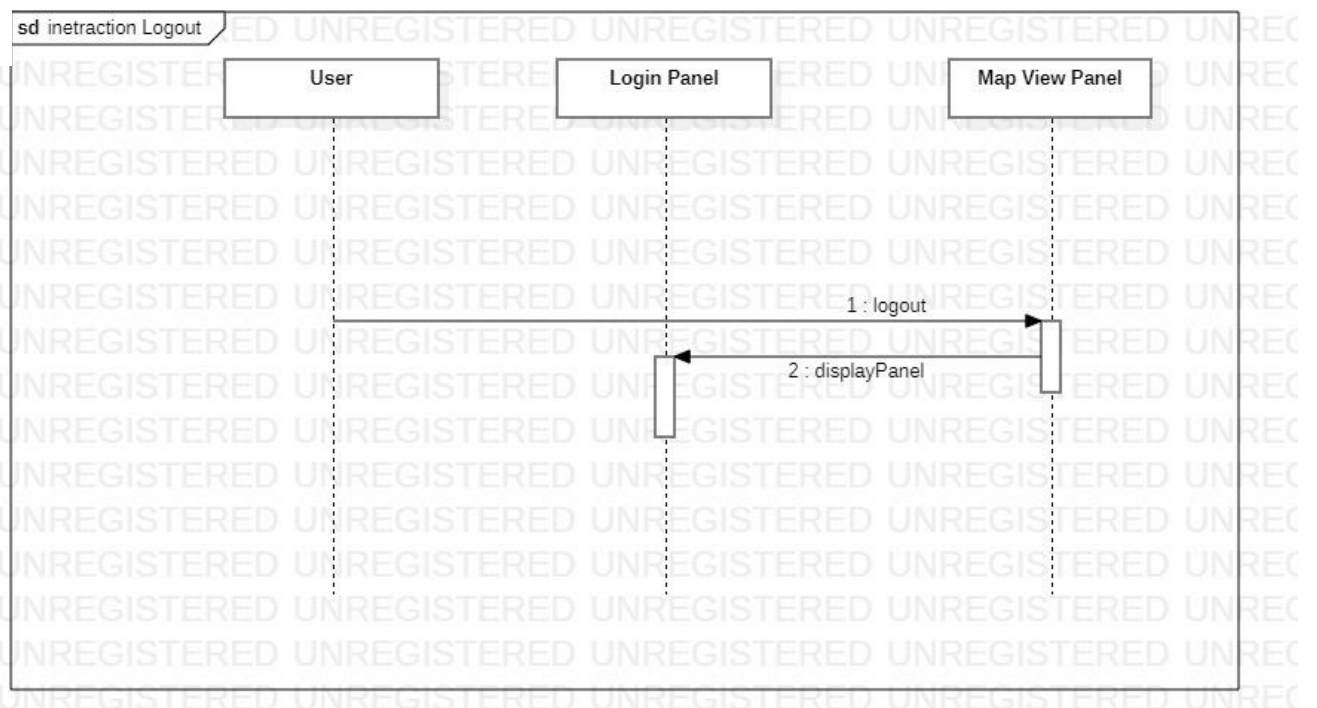
❖ Place Order:



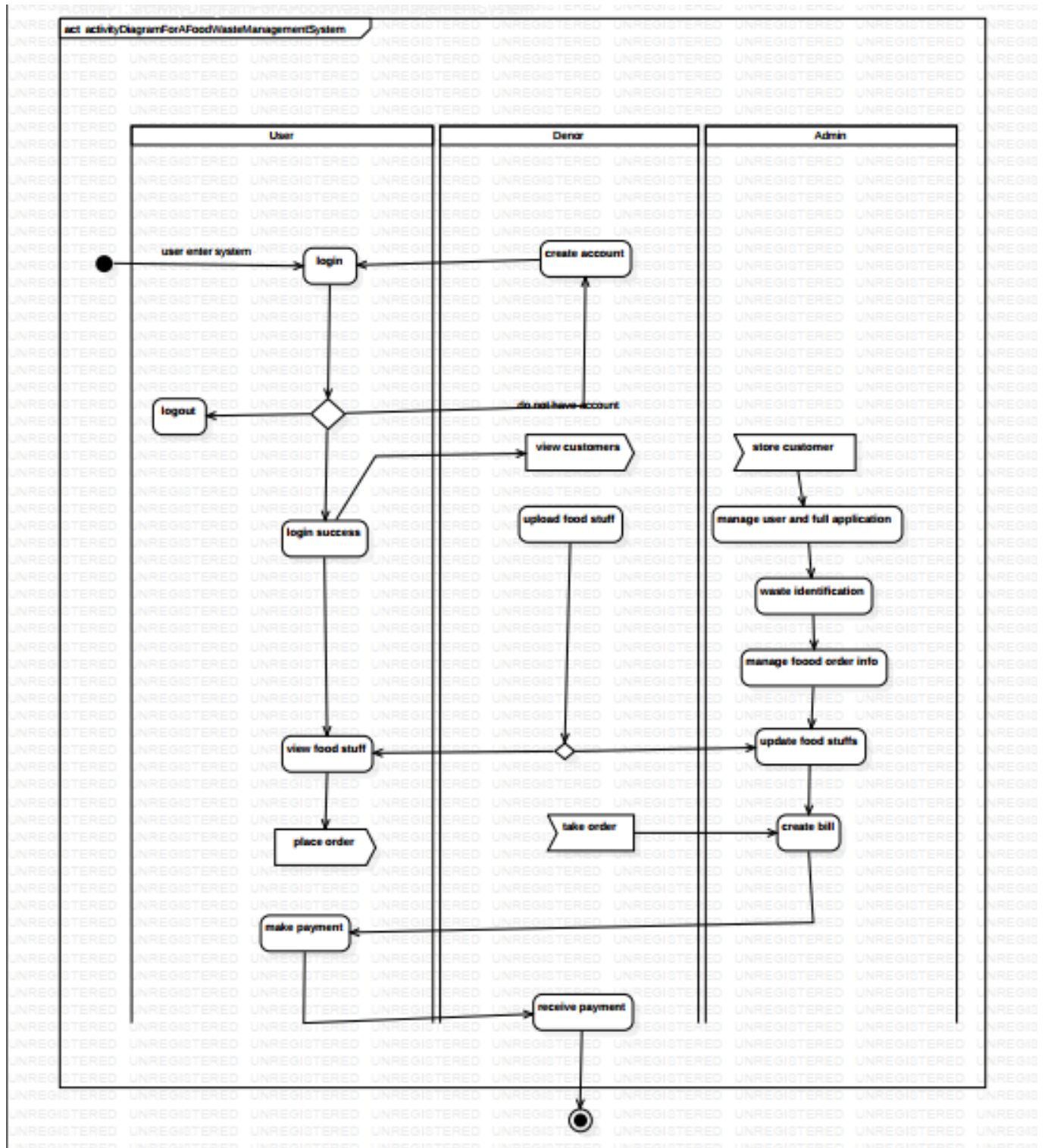
❖ Login:



❖ Logout:



d. ACTIVITY DIAGRAM



4. Importance of the different Diagrams

Class diagram, sequence diagram, use case diagram, and activity diagram are all important components of a software design process. Here's a brief overview of the importance of each diagram:

1. **Class diagram:** A class diagram is a visual representation of the classes, interfaces, and relationships in a system. It is used to model the static structure of a system and provides a clear picture of the system's objects and their relationships. Class diagrams are important for identifying the different components of a system, their attributes and methods, and how they relate to each other.
2. **Sequence diagram:** A sequence diagram is used to model the dynamic behavior of a system. It shows the interactions between objects in a system over time. Sequence diagrams are important for understanding the flow of messages and data between objects in a system, and for identifying potential issues or bottlenecks in the system's design.
3. **Use case diagram:** A use case diagram is used to model the interactions between users and a system. It shows the different use cases or scenarios that the system supports, and how users interact with the system to achieve their goals. Use case diagrams are important for understanding the requirements of a system, and for ensuring that the system meets the needs of its users.

4. **Activity diagram:** An activity diagram is used to model the workflow or business process of a system. It shows the different activities or steps in a process, and how they are connected. Activity diagrams are important for understanding the flow of work in a system, and for identifying potential issues or inefficiencies in the system's design.

Overall, these diagrams are all important tools for software designers to use in order to create a clear and comprehensive understanding of the system they are designing. By using these diagrams, designers can identify potential issues early on in the design process, and ensure that the final system meets the needs of its users.

5. Software Design Tools

Here are four tools commonly used for designing diagrams, along with their advantages and disadvantages:

1. Lucid chart:

Advantages:

- User-friendly interface with drag-and-drop functionality
- Cloud-based, so diagrams can be accessed from anywhere
- Collaboration features for working with others in real-time
- Integrates with various other tools, such as G Suite, Microsoft Office, and Atlassian

Disadvantages:

- Limited functionality in the free version
- May be slower to load larger diagrams
- Limited support for customization and advanced features compared to some other tools

2. Visual Paradigm:

Advantages:

- Comprehensive UML modeling tool with support for various types of diagrams
- Includes many advanced features, such as code generation and reverse engineering

- Offers a range of integrations and plugins for other development tools
- Supports collaboration and version control

Disadvantages:

- Can be overwhelming for beginners due to the large number of features
- Expensive compared to some other tools
- Some users may find the interface less intuitive than other tools

3. PlantUML:

Advantages:

- Uses plain text syntax to create diagrams, which can be faster and more efficient than using a graphical interface
- Supports a wide range of different diagram types, including UML, flowcharts, and network diagrams
- Free and open-source
- Can be used from the command line or integrated into other tools

Disadvantages:

- Limited support for customization and advanced features compared to some other tools
- Syntax can be more difficult to learn for beginners
- May not be suitable for more complex diagrams or large-scale projects

4. Microsoft Visio:

Advantages:

- Easy to use for those familiar with Microsoft Office products

- Offers a wide range of pre-built shapes and templates
- Integrates with other Microsoft products, such as SharePoint and Excel
- Comprehensive range of features for advanced users

Disadvantages:

- Expensive compared to some other tools, particularly for the full version
- Limited support for collaboration and version control
- May be less suitable for non-Windows users or those not already familiar with Microsoft products

6. StarUML:

Advantages:

- Free and open-source
- User-friendly interface with drag-and-drop functionality
- Supports a range of modelling languages, including UML and BPMN
- Offers code generation and reverse engineering
- Can be used on Windows, Mac, and Linux
- Supports collaboration through cloud-based sharing

Disadvantages:

- Limited support for customization and advanced features compared to some other tools
- May not be suitable for larger or more complex projects
- Limited documentation and community support compared to some other tools

7. Modelio:

Advantages:

- Comprehensive range of features, including support for various modelling languages, code generation, and reverse engineering
- Offers a range of plugins and extensions for customization and integration with other tools
- Supports collaboration and version control
- Can be used on Windows, Mac, and Linux
- Free and open-source

Disadvantages:

- May be overwhelming for beginners due to the large number of features
- Limited documentation and community support compared to some other tools
- Can be slower to load and use with larger diagrams
- Limited support for cloud-based collaboration

Overall, both StarUML and Modelio are powerful tools for designing diagrams and modelling software systems. StarUML is a good choice for simpler projects or those on a tight budget, while Modelio is a more comprehensive tool for larger or more complex projects.

CONCLUSION

This system's aims to reduce food waste and help feed undernourished communities by allowing market sellers, restaurant owners, charity organizations and orphanages to register and upload their food stocks at a cheaper price. Users would be able to register and purchase food from these parties at all times through the system. The system would be designed with a user-friendly interface and would use secure and reliable hardware, software and communication interfaces. By bringing together these different parties and providing a platform for them to sell their surplus food, the system has the potential to make a positive impact on both the environment and the community.