INSTITUTE OF INFORMATION TECHNOLOGY
NOAKHALI SCIENCE AND TECHNOLOGY
UNIVERSITY

Software Requirements Specification

for

NOURISHUNITY

A restaurant's social impact

Prepared By

Nowshin Fabiha Ibnat	BKH2125031F
Md Imteaz Ahmed	ASH2125035M
Dipraj Dhar	MUH2125004M

19 February, 2024

Table of Contents

1.Introduction

The introduction of software requirement specification provides an in-depth document that describes what the software does, how will it be performed and includes information about all the functional and non-functional requirement. In today's world, where millions go to bed hungry every night, and food waste continues to mount, addressing both issues simultaneously is not just a moral imperative but also an economic and environmental necessity. The Food Rescue and Redistribution Initiative is a serious project aimed at tackling these intertwined challenges head-on by rescuing surplus food from various sources and redistributing it to those in need. The detailed requirements and about the system (NourishUnity) are provided in this document.

1.1 Purpose

The purpose of the Food Rescue and Redistribution Initiative is to combat hunger by redistributing surplus food to those in need while simultaneously reducing food waste and promoting sustainability. Another purpose of the Food Rescue and Redistribution Initiative is to build partnerships and community resilience by collaborating with local organizations, businesses, and volunteers to address food insecurity and waste collectively.

1.2 Project Scope

The main scope of this project named NourishUnity is to develop a web-based application. As more than 90% people use smartphone or pc and use internet, they can easily can use this software by smartphone or PC. For this reason, we are targeting to implement our system for web application.

NourishUnity aims to address food waste, promote sustainability, and alleviate hunger by facilitating the rescue and distribution of food to those in need. The platform also seeks to engage volunteers, and provide culinary training to empower individuals and communities.

1.3 References

Software Requirements, Third Edition (Karl Wiegers and Joy Beatty)

1.4 Overview

The Food Rescue and Redistribution Initiative aims to tackle the pressing issues of food waste and hunger by rescuing surplus food from various sources and redistributing it to those in need. The project's scope includes identifying sources of surplus food, establishing logistics for transportation and storage, developing a distribution network to reach food-insecure populations, building partnerships with local organizations and volunteers, ensuring food safety and quality, engaging the community through education and outreach, and monitoring and evaluating the initiative's impact. By addressing these key aspects, the initiative strives to create a sustainable solution that not only alleviates hunger but also reduces food waste and fosters community resilience.

2. Stakeholders and their Characteristics

There are many users in this project. Among them three types are main. These are Users, Chef and Restaurant. Others are Volunteer, Local Organization, Administrator, System. Administer will be the main author, who have the main access and control of this site.

2.1. Users

This actor represents individuals who interact with the system to access various features such as online food ordering, volunteer engagement, etc. Users could include customers.

2.2. Chef

Chefs are individuals responsible for conducting culinary training sessions, providing structured interviews and feedback sessions, and possibly contributing to food preparation.

2.3. Restaurant

These characteristics highlight the role of restaurants, contributing to its objectives of promoting food rescue, community engagement, and sustainable practices in the food industry.

2.4. Volunteer

Volunteers are individuals who offer their services or time to assist with activities such as organizing community events, conducting culinary training, managing food rescue and distribution, and engaging other volunteers.

2.5. Local Organization

These actors represent organizations or entities that collaborate with NourishUnity to facilitate food rescue and distribution efforts within the community. They may also engage in partnerships for various initiatives and contribute to the platform's mission.

2.6. Administrator

Administrators have privileged access to system management functionalities, including system administration, collaboration with secondary stakeholders, and potentially overseeing the overall operation and effectiveness of the platform.

2.7. System

This actor represents the NourishUnity platform itself, which orchestrates the interaction between various stakeholders, processes user inputs, manages data, and ensures the smooth functioning of the system.

3. Design and Implementation Contraints:

In order to ensure the project’s success, we used design and implementation limitations.

3.1. User Interface Technology

UI stands for User Interface. It is the point of contact between humans and computers. It is also the way through which a user interacts with an application or a website.

3.1.1. Language

HTML

HTML is the code that is used to structure a web page and its content. Precisely, the coding that organizes a web page's content is called HTML. With the help of HTML, you can tell a web page whether it should be recognized as a paragraph, list, heading, link, image, multimedia player, form, or any other of the many other components that are now supported, or even a new element that you design.

javaScript

JavaScript is a versatile programming language commonly used for building interactive and dynamic web application. It is widely supported by modern web browsers and has extensive frameworks and libraries such as Node.js, React, Angular, which can fecilitate the development of both frontend and backend components of the platform.

CSS

CSS is a stylesheet language used to describe the presentation of a document written in HTML or XML. CSS specifies how items should be shown in various media, including speech, paper, screens and other media. One of the fundamental languages of the open web, CSS is defined by the W#C specification and is supported by all major browsers.

Bootstap (Front_end framework)

Bootsrap is a free and open source front-end web framework for designing websites and web applications. It includes optional JavaScript extensions along with HTML and CSS-based design templates for navigation, buttons, forms, and other interface elements. It only addresses front-end development, unlike many web frameworks. Along with CSS, Bootstrap would be utilized to create the application's styling.

PHP

PHP is a server-side scripting language specifically designed for web development. It powers many popular content management system and e-commerce platforms, making it a viable option for building web applications like NourishUnity.

3.1.2. Code Editor

A code editor, such as Visual Studio code, Atom or Subline Text, would be used write and edit the code for web application.

3.2. Server-Side Technology

When an application is used, server-side development refers to the processes that happen in the background. Databases, scripting, website architecture, backend logic, APIs and servers are the main topics covered.

3.2.1. Database Server

We will use MySQL database server to store all of the information of this system. The reason behind to choose the database server are given below:

- Security
- Performance diagnostics
- Reporting and data mining
- Fault tolerance

4. Requirement Specification:

Requirement specification is the process of identifying and documenting the specific requirements that the NourishUnity app needs to meet. It is an important step in the software development process as it helps to ensure that the final product meets the needs of the users and stakeholders. Here are some examples of requirements that might be included in the requirement specification for this project:

4.1 Functional requirements

Functional requirements are the specific functionality that the NourishUnity app needs to provide to meet the needs of the users and stakeholders. Here are some examples of functional requirements that might be included in this project:

4.1.1. Registration

FR-1	**Registration**
Description	**To allow registration of restaurant owner,volunteer,chef**
Stackholders	Customers ,Volunteers, Chef, Restaurant Staff
Priority	**High**

4.1.2. Log In

FR-2	**Login**
Description	**Who are involved in the system management**
Stackholders	Customers,Volunteers, Chef, Restaurant Staff

Priority	**High**

4.1.3. Food Rescue and Redistribution

FR-3	**Food Rescue and Redistribution**
Description	**Rescue food from restaurant and distributing to the authorized person**
Stackholders	**Volunteers, Authorities,Chef, Restaurant Staff ,Local Organization**
Priority	**High**

4.1.4. Request System Administration

FR-4	**Request System Administration**
Description	**Request for registration,logIn,receiving food ,donating food.**
Stackholders	**Authorities**
Priority	**Medium**

4.1.5. Accept Culinary Training

FR-5	**Accept Culinary Training**
Description	**Accepting culinary training request of user or staff.**

Stackholders	**Customer, Volunteer,Chefs, Restaurant Staff**
Priority	**High**

4.1.6. Delete Culinary Training

FR-6	**Delete Culinary Training**
Description	**Deleting culinary training request**
Stackholders	**Authorities**
Priority	**Low**

4.1.7. Collaboration with Local Organizations.

FR-7	**Collaboration with Local Organization**
Description	**Collaboration with local organization,restaurant owner who are interested to this system.**
Stackholders	**Authorities and Organization**
Priority	**High**

4.1.8. Structured Interviews and Feedback Session.

FR-8	**Structured Interviews and Feedback**

Description	**Getting feedback from user and updating according to the response.**
Stackholders	**Customers, Volunteers,Chefs, Restaurants Staff**
Priority	**High**

4.1.9. Create Blog

FR-9	**Create Blog**
Description	**Creating blog for creating awerness.**
Stackholders	**Administrator**
Priority	**High**

4.1.10. Delete Blog

FR-10	**Delete Blog**
Description	**Deleting unnessesary blog**
Stackholders	**Administrator**
Priority	**Medium**

4.1.11. Create Comment

FR-11	**Create Comment**
Description	**Creating comment**
Stackholders	**Volunteers,Chefs,Customer, Authorities, Restaurant staff**
Priority	**Medium**

4.1.12. Delete Comment

FR-12	**Delete Comment**
Description	**Deleting comment**
Stackholders	**System**
Priority	**Low**

4.1.13. Add Donate Receiver

FR-13	**Add Donate Receiver**
Description	**Adding those who are interested to donate**
Stackholders	**System.**
Priority	**High**

4.1.14 Delete Donate Receiver

FR-14	**Delete Donate Receiver**
Description	**Deleting donate receiver who are not interested**
Stackholders	**System**
Priority	**High**

4.1.15 Donate

FR-15	**Donate**
Description	**Donating food to the authorized person**
Stackholders	**Organization,Customer, Volunteer, Restaurant**
Priority	**High**

4.1.16. Make Payment

FR-16	**Make Payment**
Description	**To ensure the payment transaction**
Stackholders	**Customer,Organizer**

Priority	**High**

4.1.17 System Management

FR-17	**System Management**
Description	**Managing the whole system**
Stackholders	**System, Administrator**
Priority	**High**

4.2 Non-Functional Requirements

These requirements specify the quality characteristics that the app needs to possess such as performance, security, usability,rating and views etc.

4.2.1 Rating and Views

NFR-1	**Rating and Views**
Description	**Rating to bring update for the system**
Stackholders	**Customers, Organization,User**
Priority	**Medium**

4.2.2 Perfomance

NFR-2	**Performance**
Description	**How efficiently the system works.**
Stackholders	**User, Volunteers, Restaurant staff**
Priority	**High**

4.2.3 Security

NFR-3	**Secuirity**
Description	**Securing the payment system and other transaction.**
Stackholders	**System, Administrator**
Priority	**High**

4.2.4 Usability

NFR-4	**Usability**
Description	**Making the interface user friendly**
Stackholders	**User ,Customer,Volunteer, Restaurant staff**

Priority	**High**

4.3 Security Requirements

These requirements specify the measures that need to be taken to protect the app and its users' data, such as data encryption, user authentication, and access controls.

4.3.1 User Authentication

SR-1	**User Authentication**
Description	**User authentication to access the system.**
Stackholders	**System, Administrator**
Priority	**High**

4.3.2 Encryption Of Sensitive Data

SR-2	**Encryption Of Sensitive Data**
Description	**Encryption of personal data which are related to finance.**
Stackholders	**System, Administrator**
Priority	**High**

4.3.3 Access Control

SR-3	**Access Control**

Description	**Data transmission between the user's device and the server should be secured using HTTPS.**
Stackholders	**System, Administrator**
Priority	**High**

4.3.4 Regular Software Updates

SR-4	**Regular Software Updates**
Description	**According to the user feedback ,bringing updates to the system**
Stackholders	**System, Administrator**
Priority	**Medium**

4.4 Perfomance Requirements

These requirements specify the desired level of responsiveness and speed of the app, as well as the number of concurrent users it should be able to handle.

4.4.1 Load Time

PR-1	**Load times**
Description	**The application should have a fast load time,ideally under a few seconds.**
Stackholders	**User,Customer, Administrator**
Priority	**High**

4.4.2 Responsiveness

PR-2	**Responsiveness**
Description	**The application should be responsiveness to user interaction and provide instant feedback for actions taken.**
Stackholders	**Customer, Administrator**
Priority	**High**

4.4.3 Concurrent User

PR-3	**Concurrent User**
Description	**The application should be able to handle a high number of concurrent users without any significant performance degradation.**
Stackholders	**Customer**
Priority	**High**

4.4.4 Optimization

PR-4	**Optimization**
Description	**The application should be optimized for performance, including techniques like caching, magnification and code splitting.**
Stackholders	**Administrator**
Priority	**High**

4.5 Technical Requirements

These requirements specify the technical constraints and limitations that the app needs to meet, such as the need to work offline, or to work on different devices.

4.5.1 Web and Software Platform

TR-1	**Software Platform**
Description	**The application will be a web-based platform that can be accessed through a browser on desktop and mobile devices**
Stackholders	**Customer, Administrator**
Priority	**High**

4.5.2 Database

TR-2	**Database**
Description	**A database will be used to store information about restaurants, menu items, prices, and user ratings.**
Stackholders	**System,Customer**
Priority	**High**

4.5.3 User Authentication

TR-3	**User Authentication**
Description	**The application will have user authentication features to allow users to log in and provide ratings**
Stackholders	**Administrator,Customer**
Priority	**High**

4.5.4 Search Functionality

TR-4	**Search Functionality**
Description	**A search bar will be provided to allow users to search for restaurants based on location, cuisine, or restaurant name.**
Stackholders	**Customer, Administrator , Organization, Volunteer**
Priority	**High**

4.5.5 Map Integration

TR-5	**Map Integration**
Description	**The application will integrate with a mapping API to display the location of each restaurant on a map.**
Stackholders	**Customer, Administrator, System**
Priority	**High**

4.5.6 Rating System

TR-6	**Rating Sytem**
Description	**A five-star rating system will be implemented for users to rate their experience at each restaurant.**
Stackholders	**Customer, Administrator**
Priority	**Medium**

4.5.7 Security

TR-7	**Secuirity**
Description	**The application will have a responsive design to ensure a good user experience on different devices.**
Stackholders	**Customer, Administrator**
Priority	**High**

5. Requirement Engineering Process

Requirements Engineering (RE) determines software requirements according to customer requirements or needs. Requirements engineering process includes –

♣ Requirements elicitation

♣ Needs modeling

♣ Requirements analysis

♣ Requirements assurance & validation

♣ Requirements management.

- **Requirements Elicitation Techniques**

This involves gathering requirements from stakeholders, including users, restaurant owners, and other relevant parties, to determine the needs and goals of the project. This can be done through interviews, surveys, and other methods.

- **Requirements Analysis**

This involves reviewing the requirements gathered from elicitation and determining the feasibility of the project, including any technical or legal constraints.

- **Requirements Specification**

This involves documenting the requirements in a clear and concise manner, including any functional and non-functional requirements. This documentation serves as the basis for the design and development of the project.

- **Requirements Validation**:

This involves verifying that the requirements meet the needs of the stakeholders and are complete and accurate. This can be done through testing and review by stakeholders.

- **Requirements Management**:

This involves tracking and managing changes to the requirements throughout the project lifecycle to ensure that they are up to date and accurate.

- **Requirements Verification**:

This involves testing the project to ensure that it meets the requirements and provides the desired functionality and user experience.

5.1 Requirements elicitation Technique

Requirements elicitation in NourishUnity project involves gathering information about the needs and goals of the stakeholders to determine what the project should deliver. The following are some of the methods that can be used for requirements elicitation in this project:

- **User Surveys**:

Conducting surveys with potential users to understand their needs and preferences, including what they would like to see in a food tracker application.

- **User Interviews**:

Conducting in-depth interviews with users to understand their specific needs and pain points when it comes to rescue food.

5.1.1 Hold Elicitation

Interviews Stakeholder interviews are a vital part of the requirement engineering process for Fast Service. One-on-one interviews with customers and service providers are an efficient way to gather requirements and understand their specific needs. This information can then be used to resolve conflicts more effectively during group workshops. By conducting individual interviews beforehand, the time spent in group workshops is reduced and allows for a focus on resolving any remaining conflicts. We mainly perform our interview based on some specific criteria.

- Short description about easy food tracking
- Inform about use of personal information
- Dealing with customer
- Review System
- Payment System

5.1.2 Distribute Questionnaires

The questionnaire is an effective tool for gathering information on styles, changes in attitudes and preferences, and user satisfaction. To minimize fatigue or frustration for the respondent, our questions were kept concise and grouped together based on topics. This allowed the respondent to focus on specific areas and provided a clear rationale for each question. The main advantage of using this survey approach was the ability to collect responses in a standard manner, allowing for the consolidation of information from a large number of people. We use two separate set of questionnaires for this process-

- For Customer
- For Restaurants Owner

5.1.3 Competitor Analysis

Analyzing competitor applications to understand what features and functionality are popular and in demand in the market.

5.2 Requirements analysis

Requirements analysis in NourishUnity project involves reviewing and evaluating the requirements gathered from the requirements elicitation process to determine the feasibility of the project. The following are some of the steps involved in the requirements analysis phase:

- **Requirements Prioritization**:

Determining which requirements are most important and should be given priority in the development of the application.

- **Requirements Validation**:

Verifying that the requirements are complete, accurate, and consistent with the goals of the project.

- **Requirements Traceability**:

Mapping the requirements to specific features and functionality in the application, to ensure that all requirements are met.

- **Feasibility Analysis:**

Evaluating the technical and economic feasibility of the project, including any limitations or constraints that need to be taken into account.

5.3 Requirements specification

Requirements specification in the Nourishunity project involves documenting the requirements in a clear and concise manner, so that they can be used as a basis for the design and development of the application. The following are some of the steps involved in the requirements specification phase:

- **Requirements Documentation**:

Writing detailed and unambiguous documentation of the requirements, including functional and non-functional requirements.

- **Requirements Organization**:

Organizing the requirements in a clear and logical manner, with a focus on usability and maintainability.

- **Requirements Traceability**:

Ensuring that each requirement is traced back to its source, so that changes can be tracked and evaluated.

- **Requirements Communication**:

Communicating the requirements to stakeholders and other relevant parties, to ensure that everyone is on the same page and that the requirements are clearly understood.

5.4 Requirements validation

Requirements validation in the NourishUnity project involves checking that the requirements have been correctly implemented in the application and that they meet the needs and goals of the stakeholders. The following are some of the methods that can be used for requirements validation in this project:

- **Prototype Testing**:

Creating a working prototype of the application and testing it with users to validate the requirements and identify any issues or areas for improvement.

System Interface Analysis

The initial step in System Interface Analysis is to determine the systems with which the new system will need to communicate. This can involve various types of systems, such as servers on the internet, software on the same host, hardware devices, or other systems with different functions. To ensure effective communication between the new system and other systems, it is crucial to accurately identify the systems that will be involved and to understand their functions and requirements.

- **Requirements Traceability**:

Using the traceability information to verify that all requirements have been implemented in the application.

- **Stakeholder Review:**

Engaging stakeholders to review the application and provide feedback on whether the requirements have been met.

Review the Requirements

The process of negative peer review, particularly the thorough evaluation method, is a hallmark of the highest-quality software development processes. Our team of reviewers, representing diverse perspectives, thoroughly evaluated written requirements, analysis models, and information related to disability to ensure that the software meets the highest standards. This type of rigorous review helps to identify any weaknesses or areas for improvement in the software development process.

Test the Requirements

We test constitute an alternative view of the requirements. We also conduct writing tests about how to tell if the expected functionality was correctly implemented. Derive tests from the user requirements to document the expected behavior of the product under specified conditions.

- **Automated Testing**:

Using automated testing tools to verify that the requirements have been implemented correctly and that the application behaves as expected.

5.5 Requirements management

Requirements management in NourishUnity project involves the processes of planning, tracking, and controlling changes to the requirements throughout the project lifecycle. The following are some of the key aspects of requirements management in this project:

- **Requirements Planning:** Defining the approach and resources needed for requirements management, and establishing the processes and tools to be used.
- **Requirements Tracking:** Monitoring and controlling changes to the requirements throughout the project, to ensure that they are correctly implemented and that the application meets the needs of the stakeholders.
- **Requirements Change Control:** Managing the process of making changes to the requirements, including evaluating the impact of changes, communicating changes to stakeholders, and updating the requirements documentation accordingly.

6. Scenario Based Modelling

6.1. Use Case Diagram

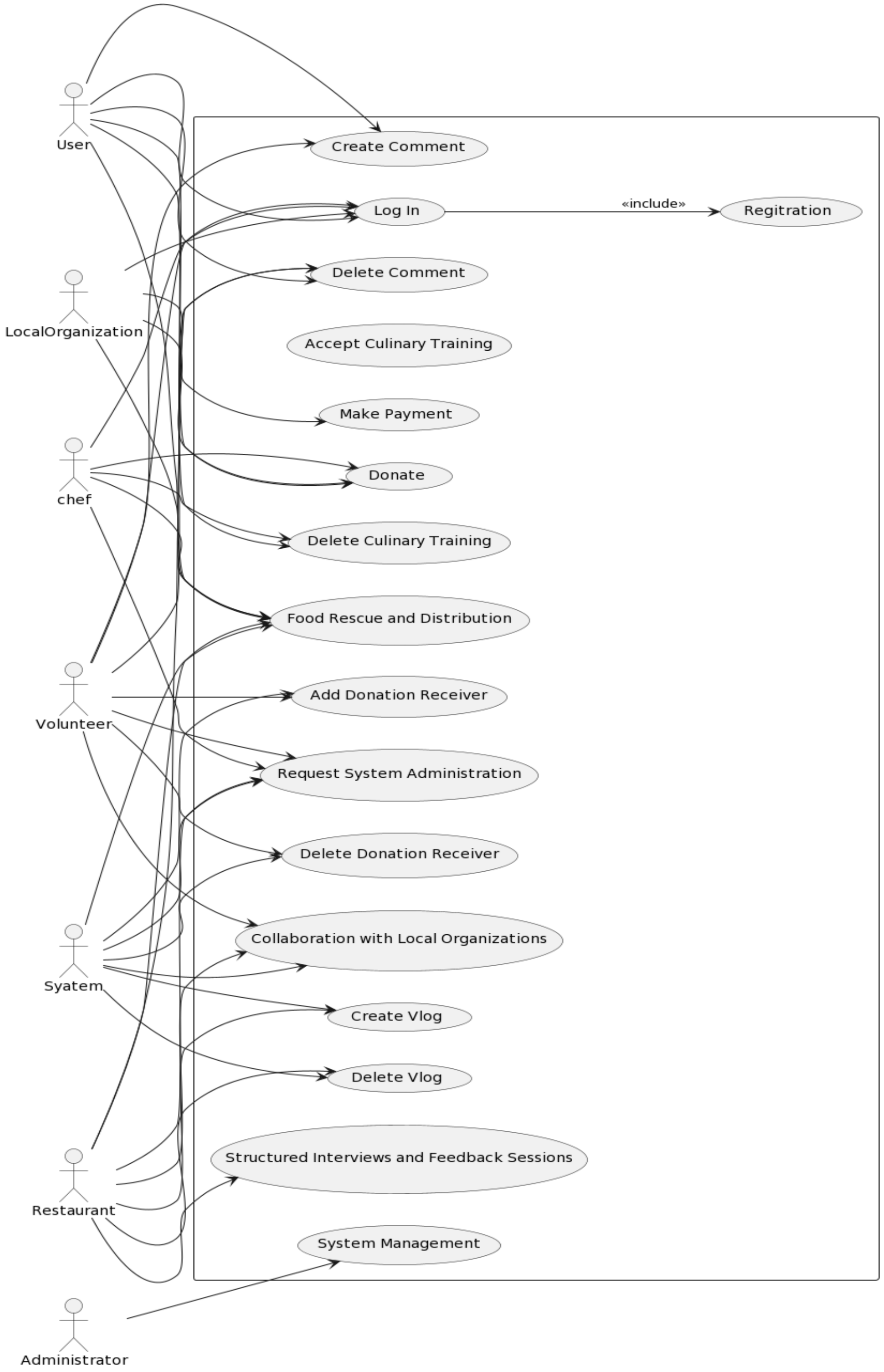

Figure 1 use case Diagram

6.2. Use Case Description

User Case	**Registration**
Goal	To allow registration

<table>
<tr><td><b>Preconditions</b></td><td>Enter into home page</td></tr>
<tr><td><b>Success End Condition</b></td><td>Registration Successful</td></tr>
<tr><td><b>Failed End Condition</b></td><td>Registration unuccessful</td></tr>
<tr><td><b>Primary Actors:</b>
<b>Secondary Actors:</b></td><td>Users</td></tr>
<tr><td><b>Trigger</b></td><td>Registration Button</td></tr>
<tr><td><b>Main Success Flows</b></td><td><table>
<tr><td>Step</td><td>Action</td></tr>
<tr><td>1</td><td>Enter registration details</td></tr>
<tr><td>2</td><td>Create user account</td></tr>
<tr><td>3</td><td>Redirect to log in</td></tr>
<tr><td>4</td><td>Successful</td></tr>
</table></td></tr>
<tr><td><b>Alternative Flows</b></td><td>Unsuccessful</td></tr>
<tr><td><b>Quality Requirements</b></td><td>N/A</td></tr>
</table>

<table>
<tr><td><b>User Case</b></td><td><b>Log In</b></td></tr>
<tr><td><b>Goal</b></td><td>Allow user to log in</td></tr>
<tr><td><b>Preconditions</b></td><td>User should be registered</td></tr>
<tr><td><b>Success End Condition</b></td><td>Log In successful</td></tr>
<tr><td><b>Failed End Condition</b></td><td>Log in failed</td></tr>
<tr><td><b>Primary Actors:</b>
<b>Secondary Actors:</b></td><td>User, chef, volunteers
System</td></tr>
<tr><td><b>Trigger</b></td><td>Log in Button</td></tr>
<tr><td><b>Main Success Flows</b></td><td><table>
<tr><td>Step</td><td>Action</td></tr>
<tr><td>1</td><td>Enter User name and password</td></tr>
<tr><td>2</td><td>Authenticate User</td></tr>
<tr><td>3</td><td>Redirect to dashboard</td></tr>
<tr><td>4</td><td>Log In successful</td></tr>
</table></td></tr>
<tr><td><b>Alternative Flows</b></td><td><table>
<tr><td>1</td><td>Show error message</td></tr>
<tr><td></td><td>Redirect to dashboard</td></tr>
</table></td></tr>
<tr><td><b>Quality Requirements</b></td><td>N/A</td></tr>
</table>

<table>
<tr><td><b>User Case</b></td><td><b>Food Rescue and Distribution</b></td></tr>
<tr><td><b>Goal</b></td><td>To confirm food distribution</td></tr>
<tr><td><b>Preconditions</b></td><td></td></tr>
<tr><td><b>Success End Condition</b></td><td></td></tr>
<tr><td><b>Failed End Condition</b></td><td>N/A</td></tr>
<tr><td><b>Primary Actors:</b>
<b>Secondary Actors:</b></td><td>Volunteer,system
Chef</td></tr>
<tr><td><b>Trigger</b></td><td></td></tr>
<tr><td><b>Main Success Flows</b></td><td>
<table>
<tr><td>Step</td><td>Action</td></tr>
<tr><td>1</td><td>Restaurant available</td></tr>
<tr><td>2</td><td>Notify restaurant</td></tr>
<tr><td>3</td><td>Restaurant confirm surplus food</td></tr>
<tr><td>4</td><td>Food rescue and distribution confirmed</td></tr>
</table>
</td></tr>
<tr><td><b>Alternative Flows</b></td><td>Food rescue and distribution failed</td></tr>
<tr><td><b>Quality Requirements</b></td><td></td></tr>
</table>

<table>
<tr><td><b>User Case</b></td><td><b>Culinary Training</b></td></tr>
<tr><td><b>Goal</b></td><td>Conduct culinary training</td></tr>
<tr><td><b>Preconditions</b></td><td>Log in as chef</td></tr>
<tr><td><b>Post Condition</b></td><td>N/A</td></tr>
<tr><td><b>Success End Condition</b></td><td></td></tr>
<tr><td><b>Failed End Condition</b></td><td>N/A</td></tr>
<tr><td><b>Primary Actors:</b>
<b>Secondary Actors:</b></td><td>Chef,system
N/A</td></tr>
<tr><td><b>Trigger</b></td><td></td></tr>
<tr><td><b>Main Success Flows</b></td><td><table>
<tr><td>Step</td><td>Action</td></tr>
<tr><td>1</td><td>Request culinary training</td></tr>
<tr><td>2</td><td>Trainer available</td></tr>
<tr><td>3</td><td>Notify</td></tr>
<tr><td>4</td><td>Culinary training successful</td></tr>
</table></td></tr>
<tr><td><b>Alternative Flows</b></td><td>Culinary training failed</td></tr>
<tr><td><b>Quality Requirements</b></td><td></td></tr>
</table>

<table>
<tr><td><b>User Case</b></td><td><b>Collaboration with Local Organizations</b></td></tr>
<tr><td><b>Goal</b></td><td>To collaborate with organization</td></tr>
<tr><td><b>Preconditions</b></td><td>Local organization must be in the system</td></tr>
<tr><td><b>Post Condition</b></td><td>N/A</td></tr>
<tr><td><b>Success End Condition</b></td><td>Collaboration successful</td></tr>
<tr><td><b>Failed End Condition</b></td><td>N/A</td></tr>
<tr><td><b>Primary Actors:</b>
<b>Secondary Actors:</b></td><td>Restaurant ,local organization ,system
Administrator</td></tr>
<tr><td><b>Trigger</b></td><td>Call /meet</td></tr>
<tr><td><b>Main Success Flows</b></td><td><table>
<tr><td>Step</td><td>Action</td></tr>
<tr><td>1</td><td>Request collaboration</td></tr>
<tr><td>2</td><td>Available local organization</td></tr>
<tr><td>3</td><td>Notify</td></tr>
<tr><td>4</td><td>Collaboration successful</td></tr>
</table></td></tr>
<tr><td><b>Alternative Flows</b></td><td>Collaboration unsuccessful</td></tr>
<tr><td><b>Quality Requirements</b></td><td>1. System should provide a call option
2. System should provide a google meet</td></tr>
</table>

User Case	**Structured Interviews and Feedback Sessions**
Goal	To understand the stakeholders demand
Preconditions	
Post Condition	Update project
Success End Condition	Session completed successfully
Failed End Condition	N/A
Primary Actors: **Secondary Actors:**	Volunteers ,chef ,system Administrator
Trigger	Feedback option
Main Success Flows	Step – Action 1 – Request 2 – Staff available 3 – Notify 4 – Successful
Alternative Flows	No staff available
Quality Requirements	1. System should provide clear and easy to use option

<table>
<tr><td><b>User Case</b></td><td><b>Update profile</b></td></tr>
<tr><td><b>Goal</b></td><td>To allow the restaurant owner to update their existing profile information.</td></tr>
<tr><td><b>Preconditions</b></td><td>Restaurant Owner has already created a profile and is logged into the web application.</td></tr>
<tr><td><b>Post Condition</b></td><td>N/A</td></tr>
<tr><td><b>Success End Condition</b></td><td>Restaurant's profile information is updated and saved in the android application.</td></tr>
<tr><td><b>Failed End Condition</b></td><td>N/A</td></tr>
<tr><td><b>Primary Actors:</b>
<b>Secondary Actors:</b></td><td>Restaurant Owner
N/A</td></tr>
<tr><td><b>Trigger</b></td><td>Owner selects "Update Profile" option from the application.</td></tr>
<tr><td><b>Main Success Flows</b></td><td>
<table>
<tr><td>Step</td><td>Action</td></tr>
<tr><td>1</td><td>Restaurant Owner is taking to profile editing page.</td></tr>
<tr><td>2</td><td>Restaurant Owner clicks on update option.</td></tr>
<tr><td>3</td><td>Then, can update any of their existing name, location, menu list and can upload new photos.</td></tr>
<tr><td>4</td><td>Restaurant Owner confirms and saves their update information.</td></tr>
</table>
</td></tr>
<tr><td><b>Alternative Flows</b></td><td>N/A</td></tr>
<tr><td><b>Quality Requirements</b></td><td>1. System should provide clear and easy-to-use options for updating profile information. 2. Profile forms should be concise and easy to understand. 3. The system should store and track customer profile information accurately. 4. The system should have a way to confirm changes with the customer.</td></tr>
</table>

User Case	Create Blog

<table>
<tr><td><b>Goal</b></td><td>To make awareness</td></tr>
<tr><td><b>Preconditions</b></td><td>Log in</td></tr>
<tr><td><b>Post Condition</b></td><td>N/A</td></tr>
<tr><td><b>Success End Condition</b></td><td>Successfully creating blog</td></tr>
<tr><td><b>Failed End Condition</b></td><td>N/A</td></tr>
<tr><td><b>Primary Actors:</b>
<b>Secondary Actors:</b></td><td>System
User</td></tr>
<tr><td><b>Trigger</b></td><td>Post</td></tr>
<tr><td><b>Main Success Flows</b></td><td><table>
<tr><td>Step</td><td>Action</td></tr>
<tr><td>1</td><td>Request</td></tr>
<tr><td>2</td><td>Enter blog detail</td></tr>
<tr><td>3</td><td>Saves blog data</td></tr>
<tr><td>4</td><td>Publish blog</td></tr>
</table></td></tr>
<tr><td><b>Alternative Flows</b></td><td>Failed to create blog</td></tr>
<tr><td><b>Quality Requirements</b></td><td>System should provide clear and easy to use option</td></tr>
</table>

<table>
<tr><td><b>User Case</b></td><td><b>Delete Blog</b></td></tr>
<tr><td><b>Goal</b></td><td>Deleting blog</td></tr>
<tr><td><b>Preconditions</b></td><td>Create blog</td></tr>
<tr><td><b>Post Condition</b></td><td>N/A</td></tr>
<tr><td><b>Success End Condition</b></td><td>Delete blog successfully</td></tr>
<tr><td><b>Failed End Condition</b></td><td>N/A</td></tr>
<tr><td><b>Primary Actors:</b>
<b>Secondary Actors:</b></td><td>System
N/A</td></tr>
<tr><td><b>Trigger</b></td><td>Delete</td></tr>
<tr><td><b>Main Success Flows</b></td><td><table>
<tr><td>Step</td><td>Action</td></tr>
<tr><td>1</td><td>Delete blog</td></tr>
<tr><td>2</td><td>Select blog to delete</td></tr>
<tr><td>3</td><td>Confirms blog deletion</td></tr>
<tr><td>4</td><td>Successful</td></tr>
</table></td></tr>
<tr><td><b>Alternative Flows</b></td><td>System cancels blog deletion</td></tr>
<tr><td><b>Quality Requirements</b></td><td>System should provide clear and easy to use option</td></tr>
</table>

User Case	**Create Comment**
Goal	To create comment
Preconditions	User has entered the club
Post Condition	N/A
Success End Condition	Comment created
Failed End Condition	Creation failed
Primary Actors: **Secondary Actors:**	User N/A
Trigger	comment button
Main Success Flows	Step – Action 1 – Create a comment 2 – Enter comment detail 3 – Validates comment details 4 – Comment is created
Alternative Flows	System cancels comment details
Quality Requirements	N/A

<table>
<tr><td><b>User Case</b></td><td><b>Delete Comment</b></td></tr>
<tr><td><b>Goal</b></td><td>To delete comment</td></tr>
<tr><td><b>Preconditions</b></td><td>Create a comment</td></tr>
<tr><td><b>Post Condition</b></td><td>N/A</td></tr>
<tr><td><b>Success End Condition</b></td><td>Deleting comment</td></tr>
<tr><td><b>Failed End Condition</b></td><td>N/A</td></tr>
<tr><td><b>Primary Actors:</b>
<b>Secondary Actors:</b></td><td>User
N/A</td></tr>
<tr><td><b>Trigger</b></td><td>Delete comment</td></tr>
<tr><td><b>Main Success Flows</b></td><td>
<table>
<tr><td>Step</td><td>Action</td></tr>
<tr><td>1</td><td>Delete a comment</td></tr>
<tr><td>2</td><td>Delect comment to delete</td></tr>
<tr><td>3</td><td>System confirms comment deletion</td></tr>
<tr><td>4</td><td>System notifies user that comment is deleted</td></tr>
</table>
</td></tr>
<tr><td><b>Alternative Flows</b></td><td>System cancels comment deletion</td></tr>
<tr><td><b>Quality Requirements</b></td><td>N/A</td></tr>
</table>

<table>
<tr><td><b>User Case</b></td><td><b>Add Donation Receiver</b></td></tr>
<tr><td><b>Goal</b></td><td>To add donation receiver</td></tr>
<tr><td><b>Preconditions</b></td><td>Log in</td></tr>
<tr><td><b>Post Condition</b></td><td>N/A</td></tr>
<tr><td><b>Success End Condition</b></td><td>Donation receiver successful</td></tr>
<tr><td><b>Failed End Condition</b></td><td>N/A</td></tr>
<tr><td><b>Primary Actors:</b>
<b>Secondary Actors:</b></td><td>User,system</td></tr>
<tr><td><b>Trigger</b></td><td>Add donation</td></tr>
<tr><td><b>Main Success Flows</b></td><td>
<table>
<tr><td>Step</td><td>Action</td></tr>
<tr><td>1</td><td>Add a donation receciver</td></tr>
<tr><td>2</td><td>Enter receiver detail</td></tr>
<tr><td>3</td><td>System validates receiver</td></tr>
<tr><td>4</td><td>System notifies user that receiver is added</td></tr>
</table>
</td></tr>
<tr><td><b>Alternative Flows</b></td><td>System cancels receiver addition</td></tr>
<tr><td><b>Quality Requirements</b></td><td>N/A</td></tr>
</table>

<table>
<tr><td><b>User Case</b></td><td><b>Delete Donation Receiver</b></td></tr>
<tr><td><b>Goal</b></td><td>To delete donation receiver</td></tr>
<tr><td><b>Preconditions</b></td><td>Create blog</td></tr>
<tr><td><b>Post Condition</b></td><td>N/A</td></tr>
<tr><td><b>Success End Condition</b></td><td>Deleting donation receiver</td></tr>
<tr><td><b>Failed End Condition</b></td><td>N/A</td></tr>
<tr><td><b>Primary Actors:</b>
<b>Secondary Actors:</b></td><td>User, system
N/A</td></tr>
<tr><td><b>Trigger</b></td><td></td></tr>
<tr><td><b>Main Success Flows</b></td><td><table>
<tr><td>Step</td><td>Action</td></tr>
<tr><td>1</td><td>Add a donation receiver</td></tr>
<tr><td>2</td><td>Enter receiver details</td></tr>
<tr><td>3</td><td>System validates receiver details</td></tr>
<tr><td>4</td><td>Successful</td></tr>
</table></td></tr>
<tr><td><b>Alternative Flows</b></td><td>System cancels receiver addition</td></tr>
<tr><td><b>Quality Requirements</b></td><td>N/A</td></tr>
</table>

User Case	**Donate**
Goal	To donate
Preconditions	Donator must be in database
Post Condition	N/A
Success End Condition	Successful Donation
Failed End Condition	N/A
Primary Actors: **Secondary Actors:**	User System
Trigger	Donate button
Main Success Flows	<table><tr><td>Step</td><td>Action</td></tr><tr><td>1</td><td>Donation</td></tr><tr><td>2</td><td>Select donation receiver and enter donation details</td></tr><tr><td>3</td><td>System validates donation details</td></tr><tr><td>4</td><td>System notifies that donation is completed</td></tr></table>
Alternative Flows	System cacels donation
Quality Requirements	Ensuring the quality of the food

<table>
<tr><td><b>User Case</b></td><td><b>Make Payment</b></td></tr>
<tr><td><b>Goal</b></td><td>For successful payment</td></tr>
<tr><td><b>Preconditions</b></td><td>Authentication should occur first</td></tr>
<tr><td><b>Post Condition</b></td><td>N/A</td></tr>
<tr><td><b>Success End Condition</b></td><td>Making payment successfully</td></tr>
<tr><td><b>Failed End Condition</b></td><td>N/A</td></tr>
<tr><td><b>Primary Actors:</b>
<b>Secondary Actors:</b></td><td>User
Administrator</td></tr>
<tr><td><b>Trigger</b></td><td>Make payment</td></tr>
<tr><td><b>Main Success Flows</b></td><td>
<table>
<tr><td>Step</td><td>Action</td></tr>
<tr><td>1</td><td>Payment</td></tr>
<tr><td>2</td><td>System prompts user to enter payment details</td></tr>
<tr><td>3</td><td>System validates payment details</td></tr>
<tr><td>4</td><td>successful</td></tr>
</table>
</td></tr>
<tr><td><b>Alternative Flows</b></td><td>System cancels payment</td></tr>
<tr><td><b>Quality Requirements</b></td><td>The system should be consistent</td></tr>
</table>

User Case	**System Management**
Goal	for proper management of system
Preconditions	N/A
Post Condition	N/A
Success End Condition	Regular update
Failed End Condition	N/A
Primary Actors: **Secondary Actors:**	System N/A
Trigger	N/A
Main Success Flows	N/A
Alternative Flows	N/A
Quality Requirements	N/A

6.3. Sequence Diagram

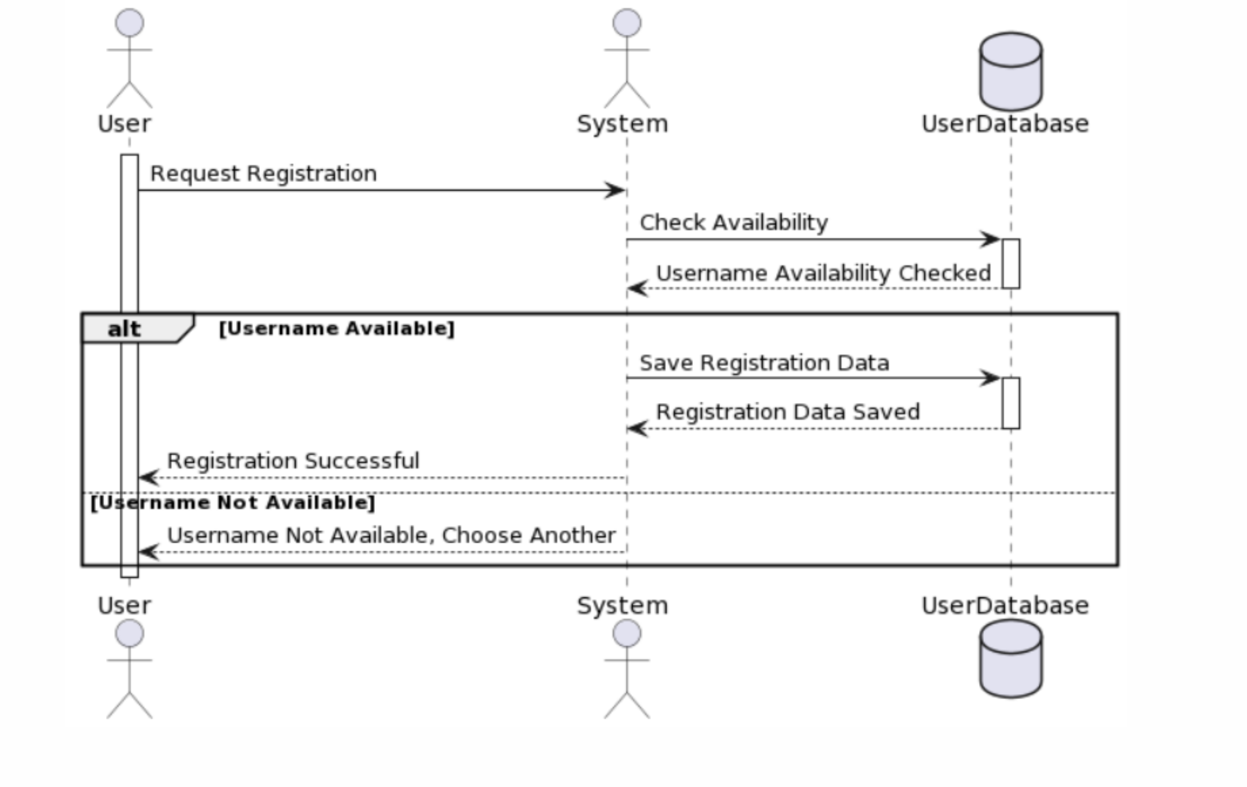

Figure 2 Registration

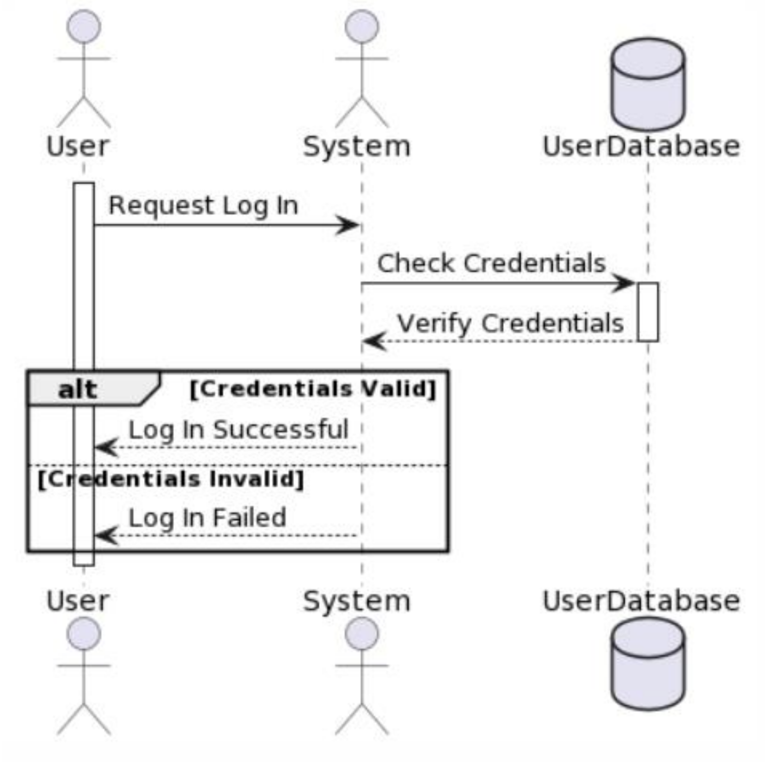

Figure 3 Log in

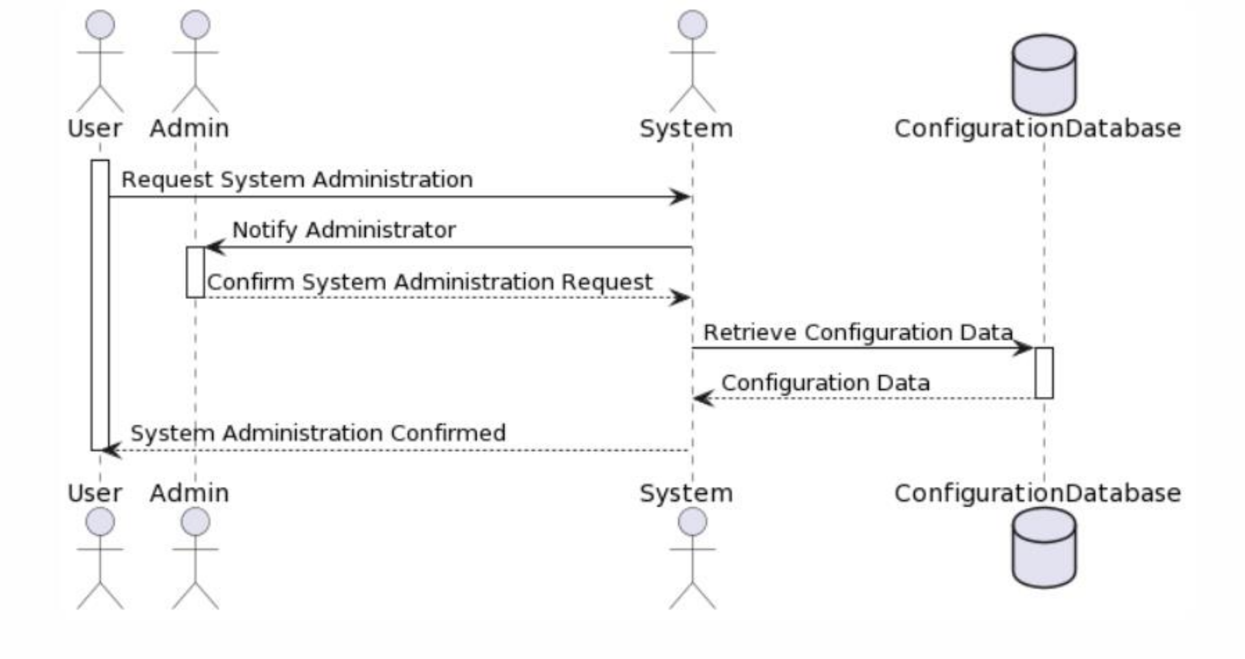
User
Admin
System
ConfigurationDatabase
Request System Administration
Notify Administrator
Confirm System Administration Request
Retrieve Configuration Data
Configuration Data
System Administration Confirmed
User
Admin
System
ConfigurationDatabase

Figure 4 Administer System

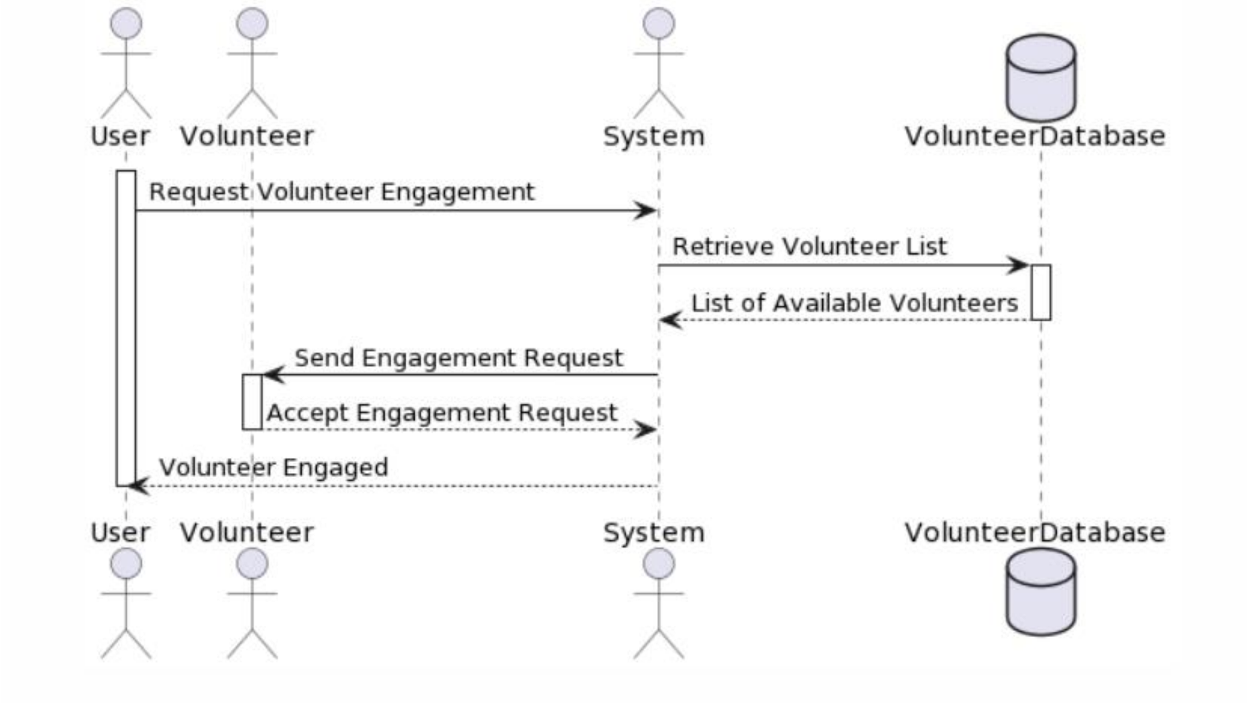

Figure 5 Engage Volunteers

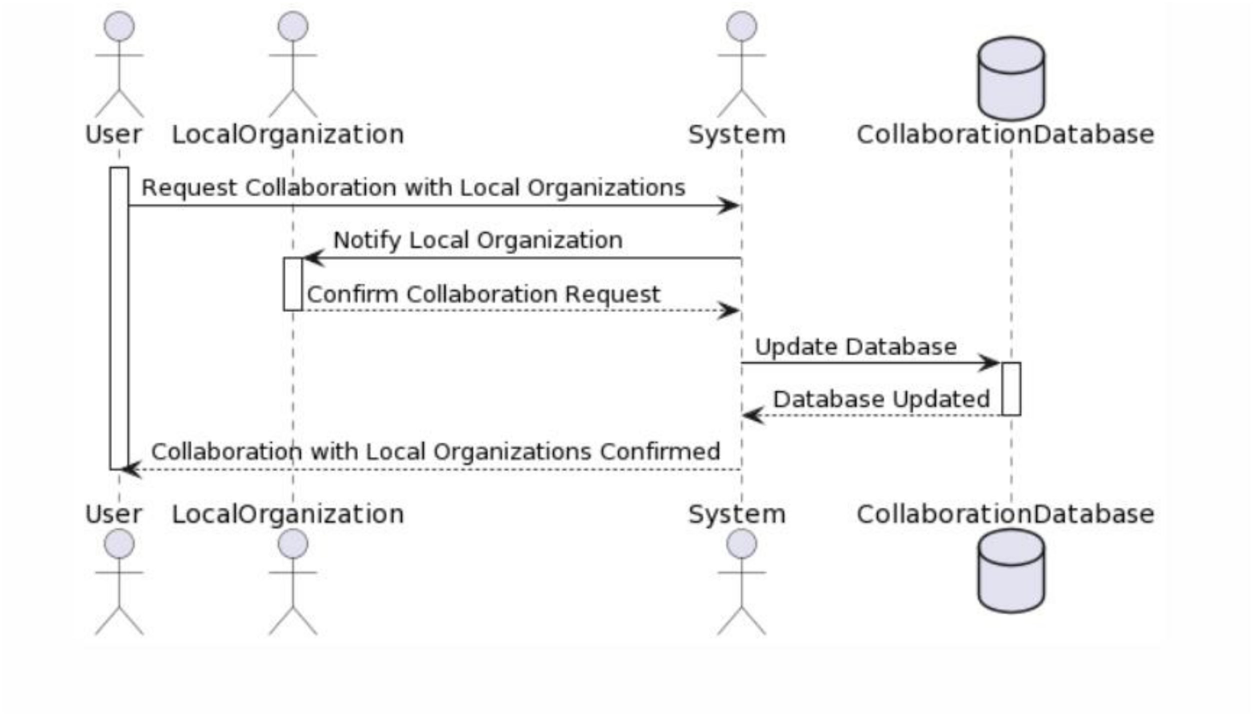
User
LocalOrganization
System
CollaborationDatabase
Request Collaboration with Local Organizations
Notify Local Organization
Confirm Collaboration Request
Update Database
Database Updated
Collaboration with Local Organizations Confirmed
User
LocalOrganization
System
CollaborationDatabase

Figure 6 Collaborate with Local Organizations

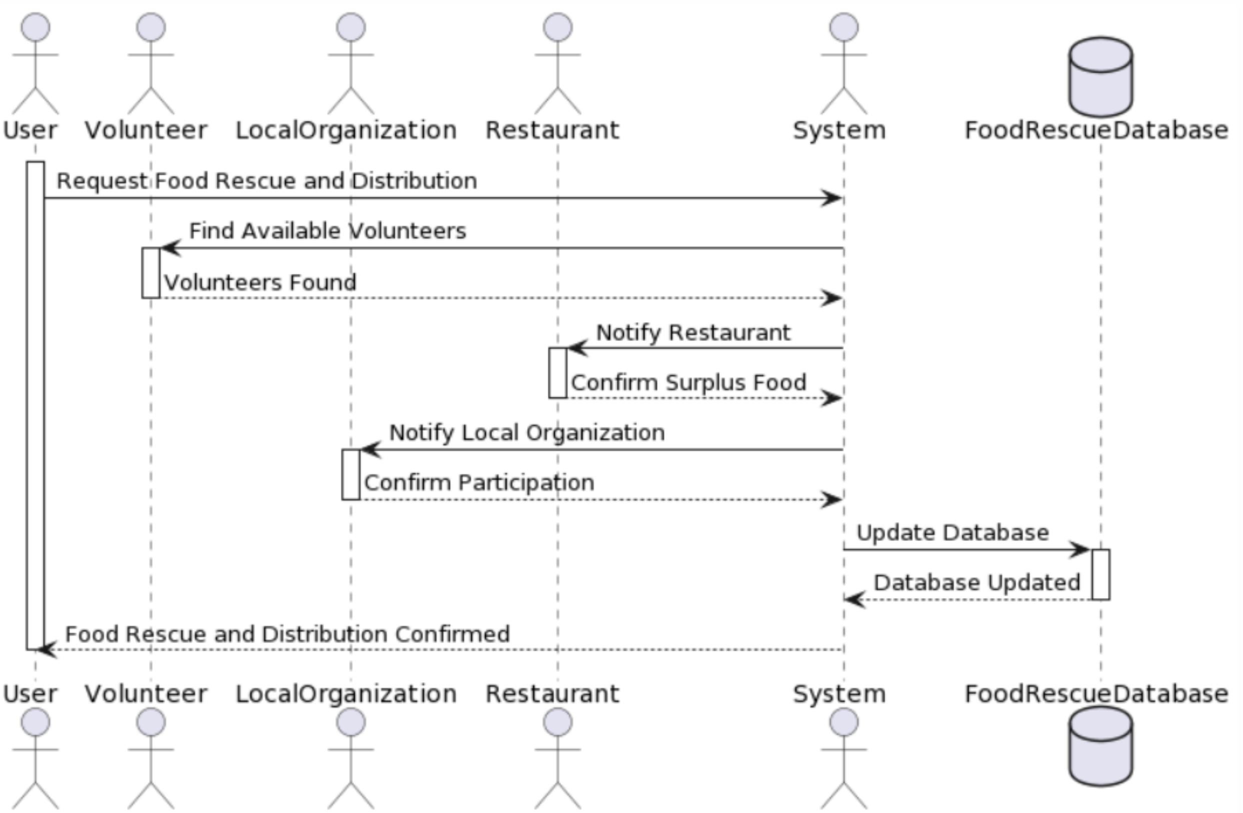

Figure 7 Manage Food Rescue and Distribution

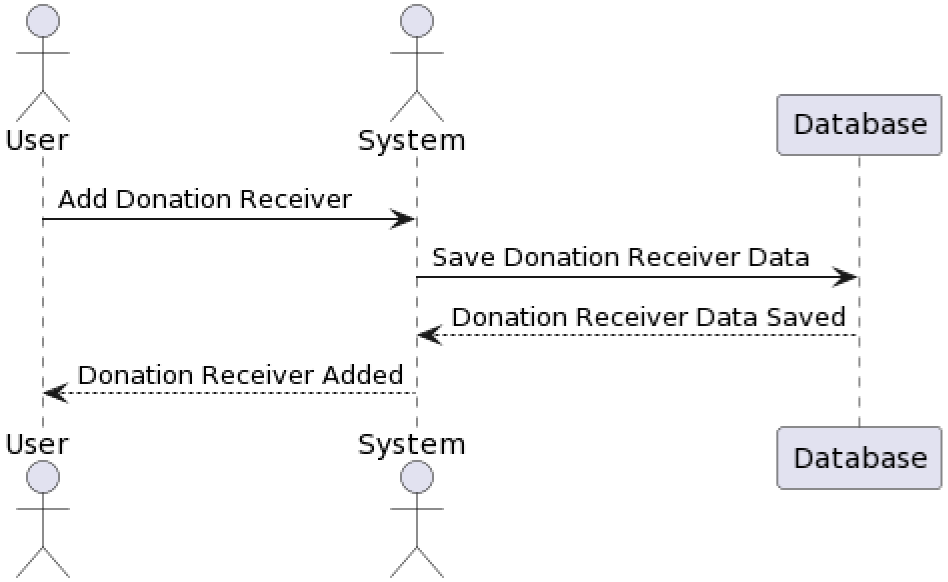

Figure 8 Add Donation Receiver

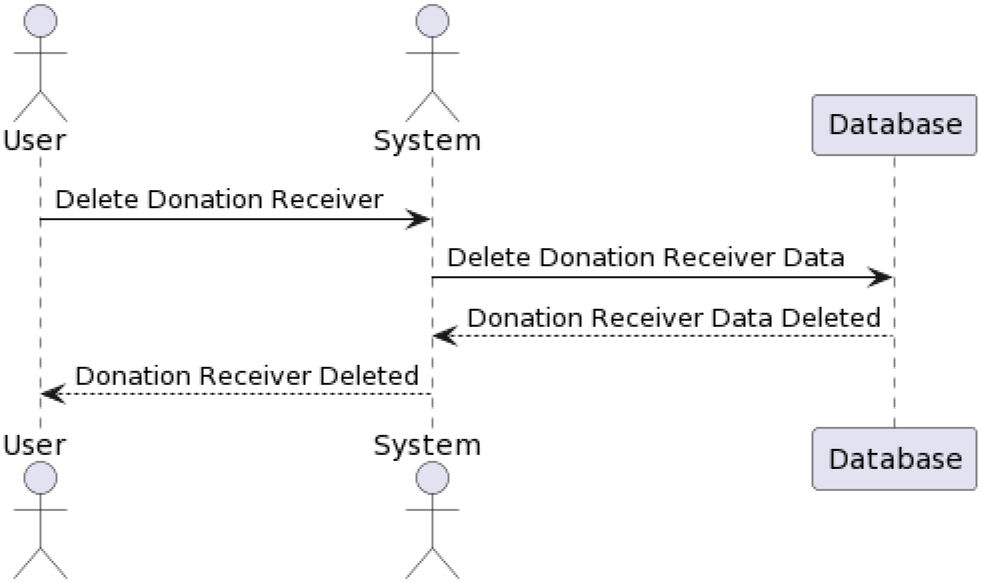
User
System
Database
Delete Donation Receiver
Delete Donation Receiver Data
Donation Receiver Data Deleted
Donation Receiver Deleted
User
System
Database

Figure 9 Delete Donation Receiver

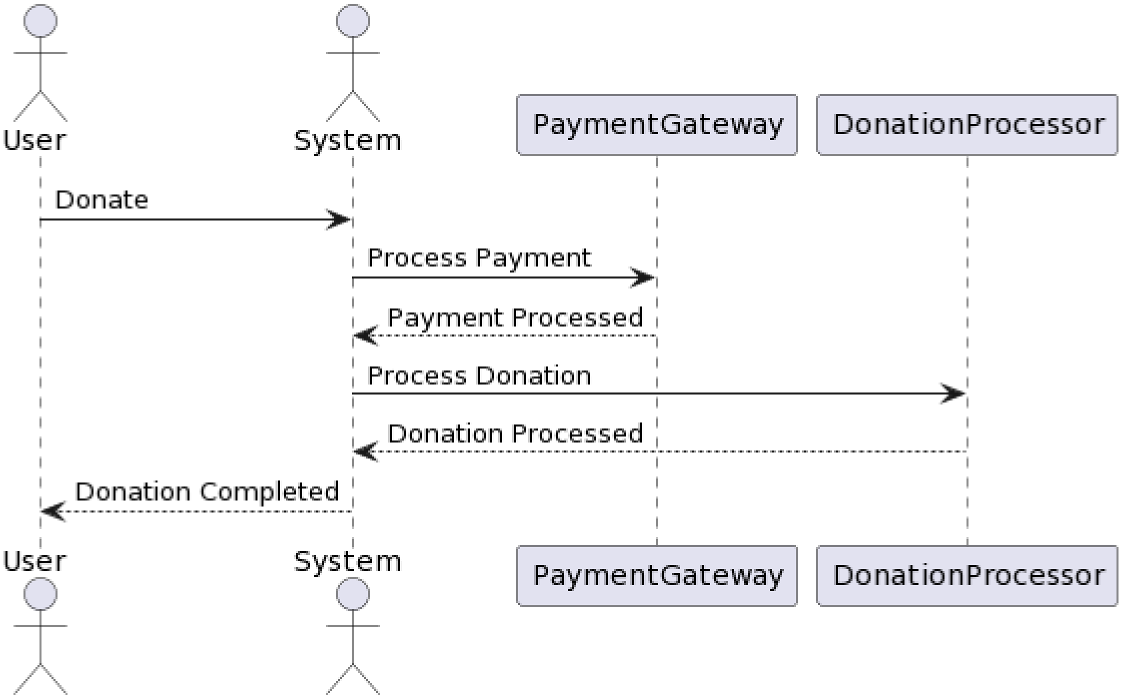

Figure 10 Donate

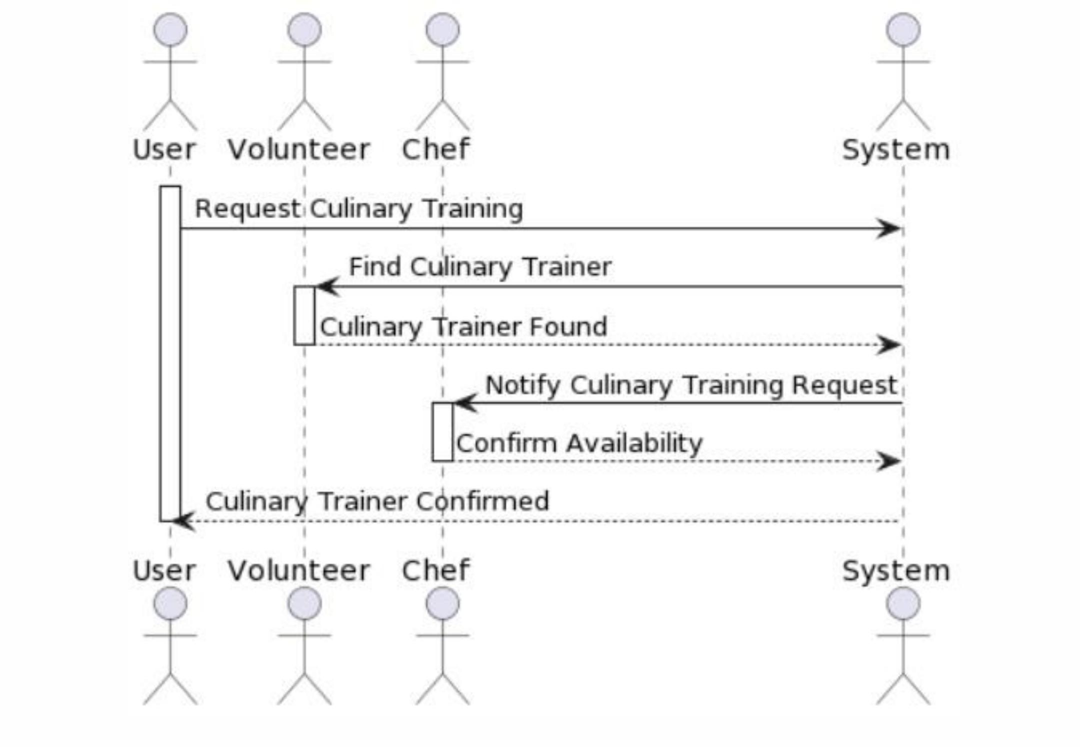

Figure 11 Conduct Culinary Training

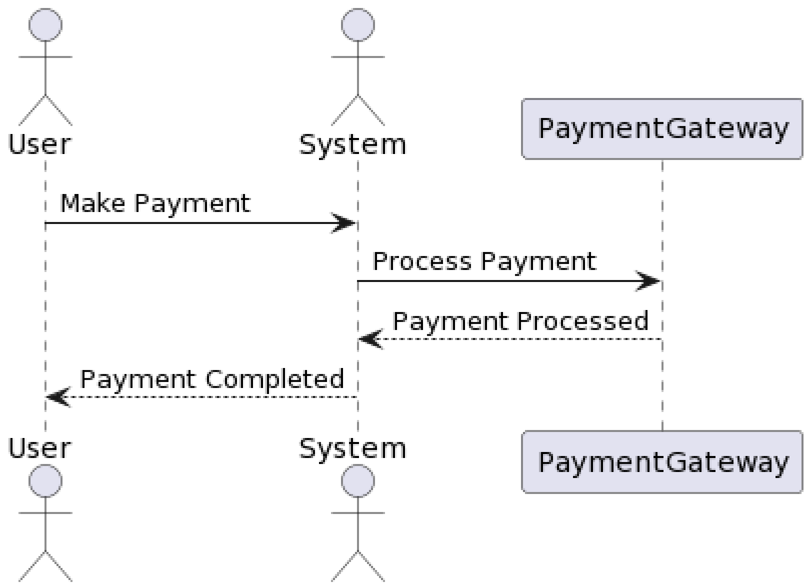

Figure 12 Make Payment

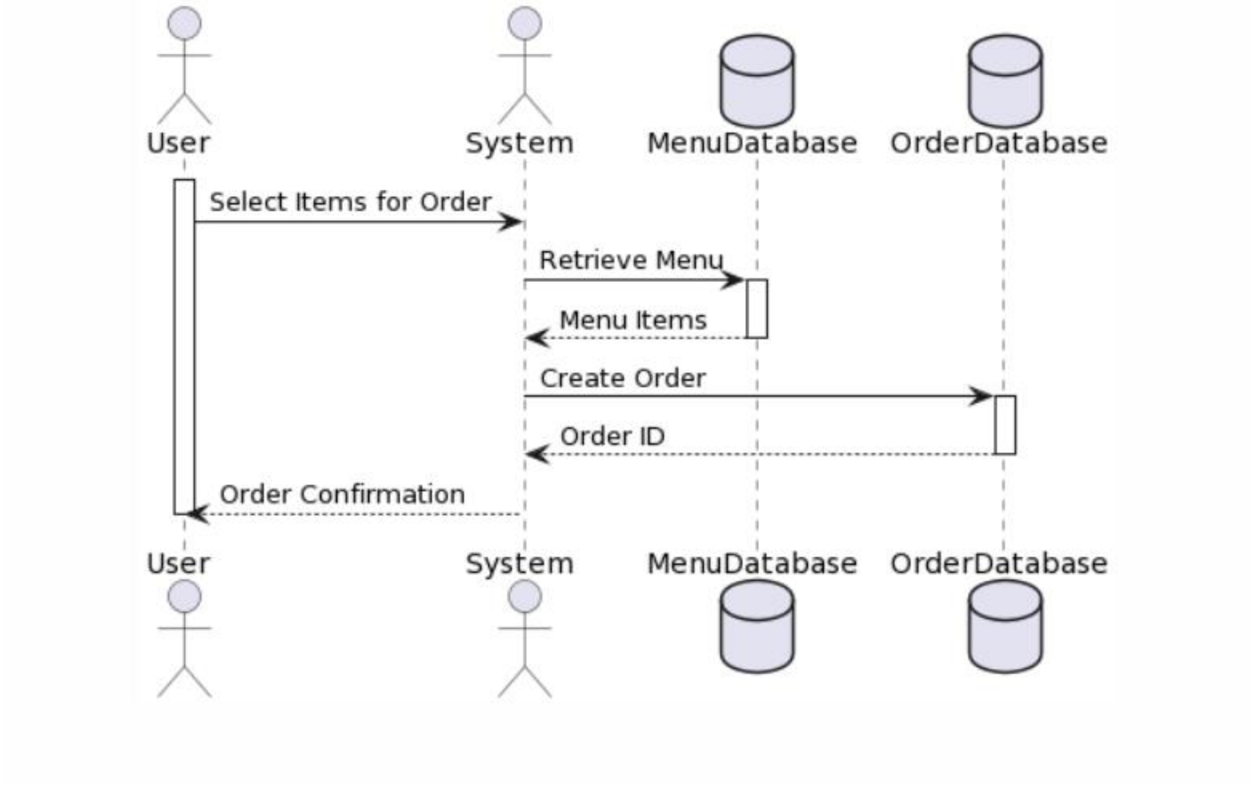

Figure 13 Implement Online Food Ordering

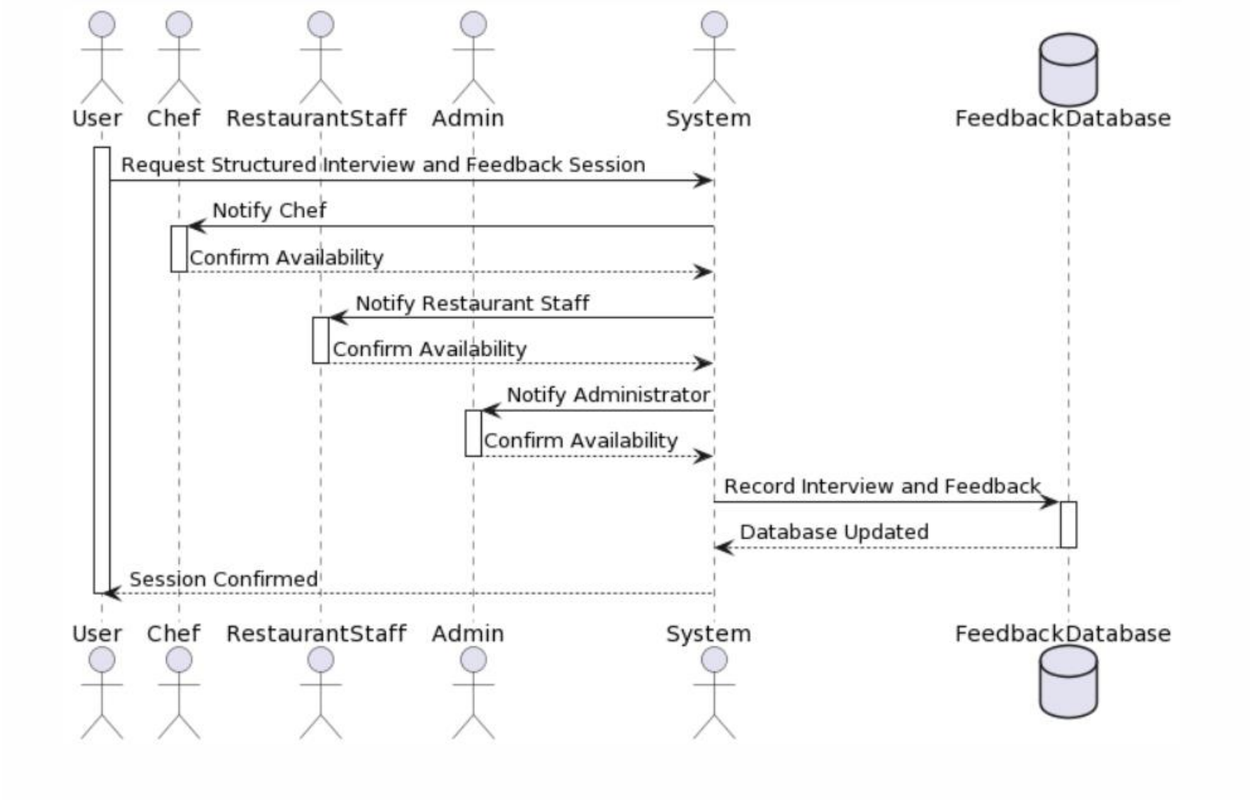

Figure 14 Conduct Structured Interviews and Feedback Sessions

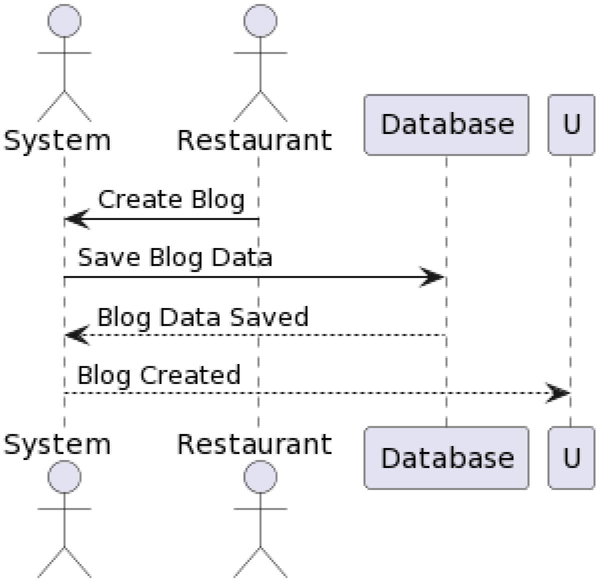

Figure 15 Create Blog

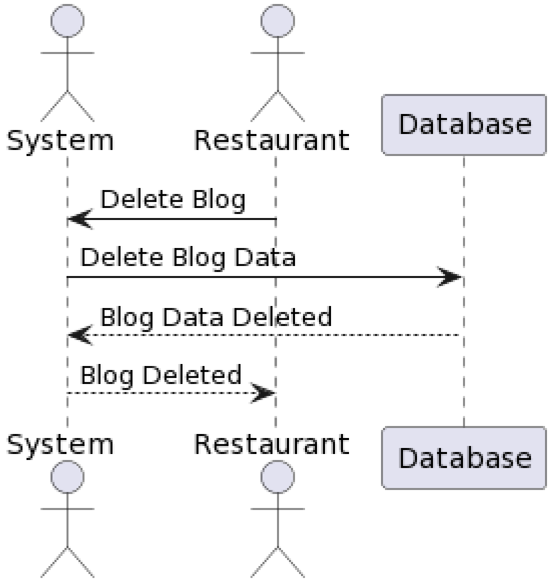

Figure 16 Delete Blog

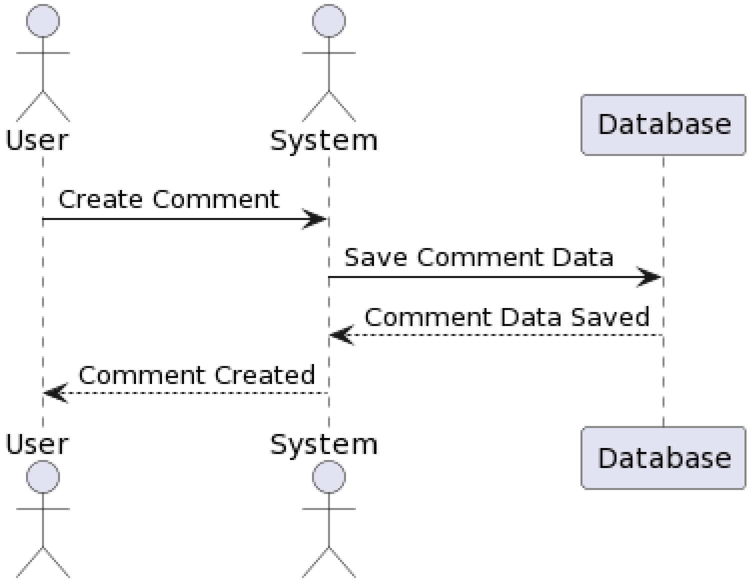

Figure 17 Create Comment

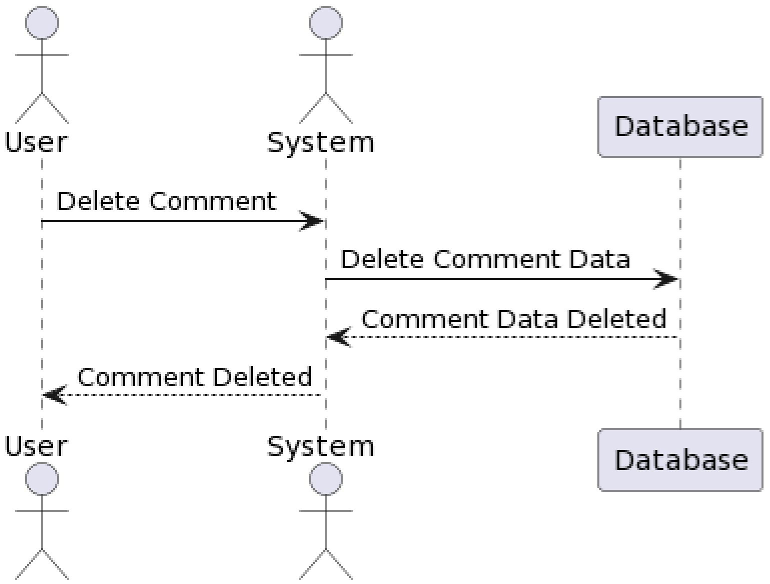

Figure 18 Delete Comment

6.4. State Diagram

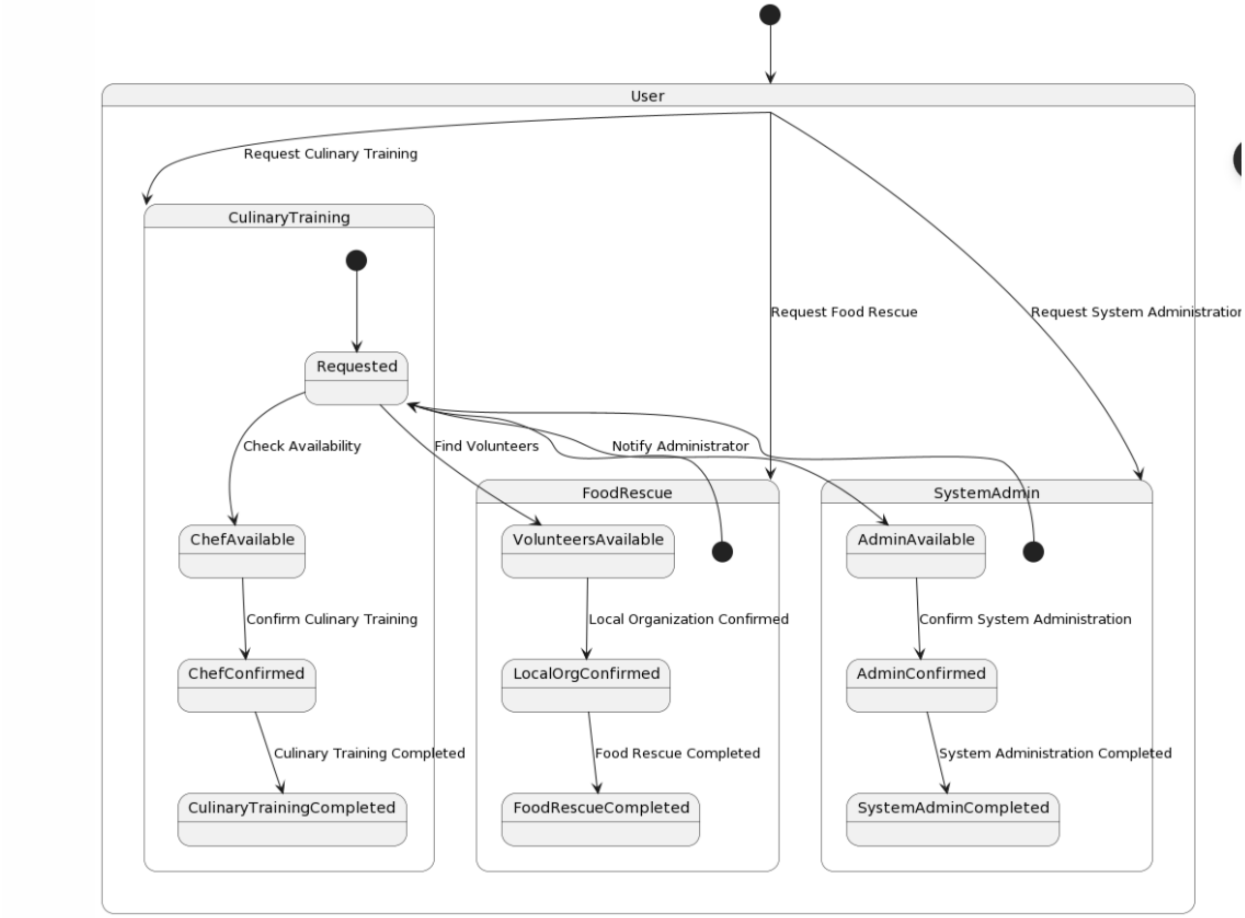

Figure 19 State diagram

6.5. Activity Diagram

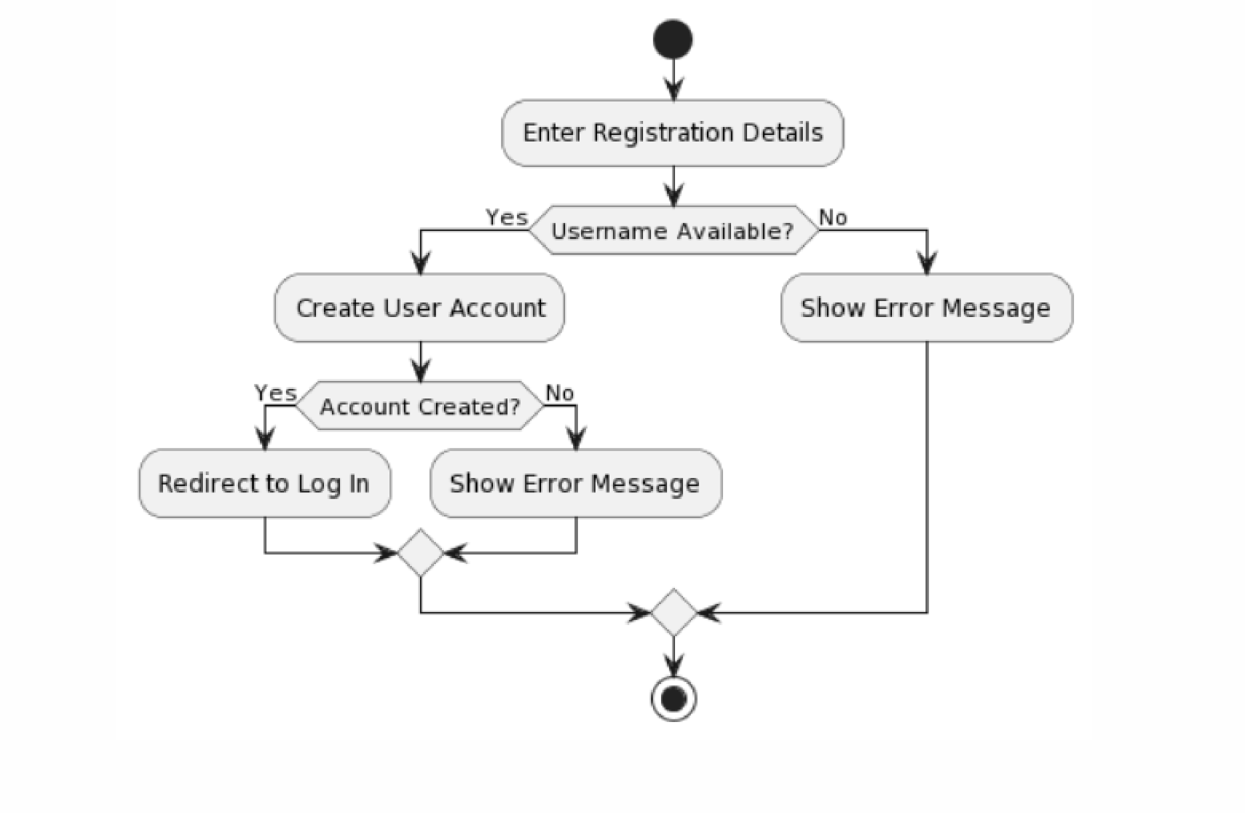

Figure 20 Registration

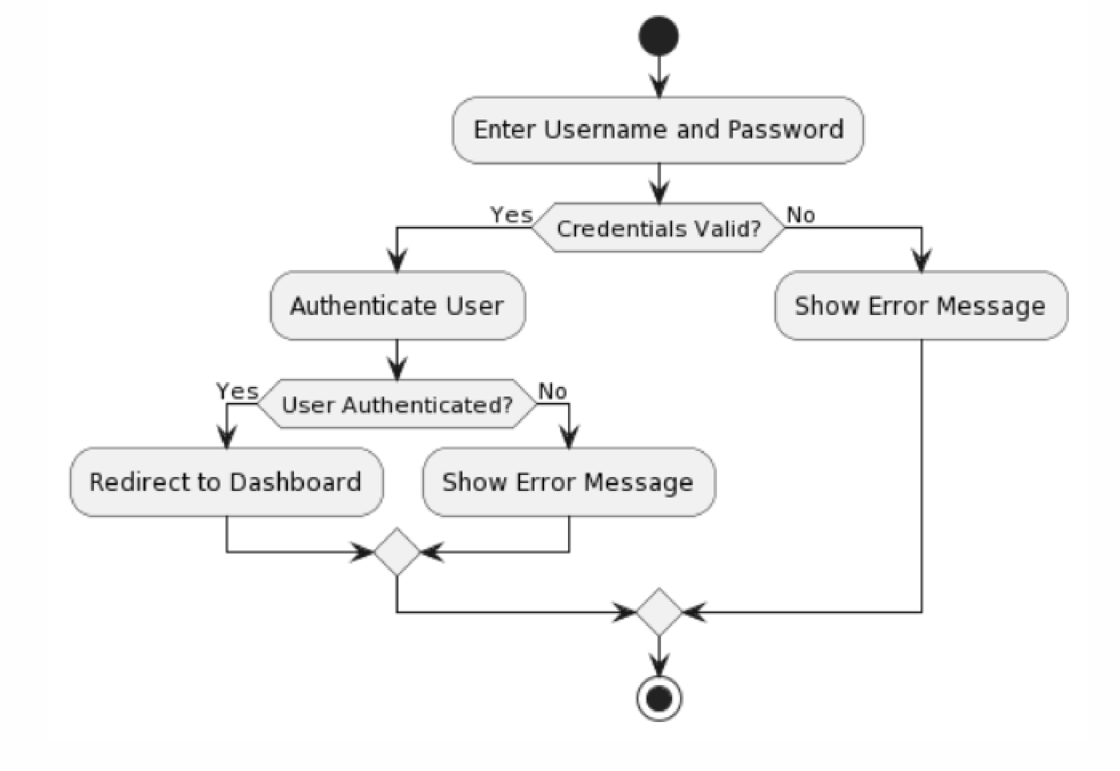

Figure 21 Log in

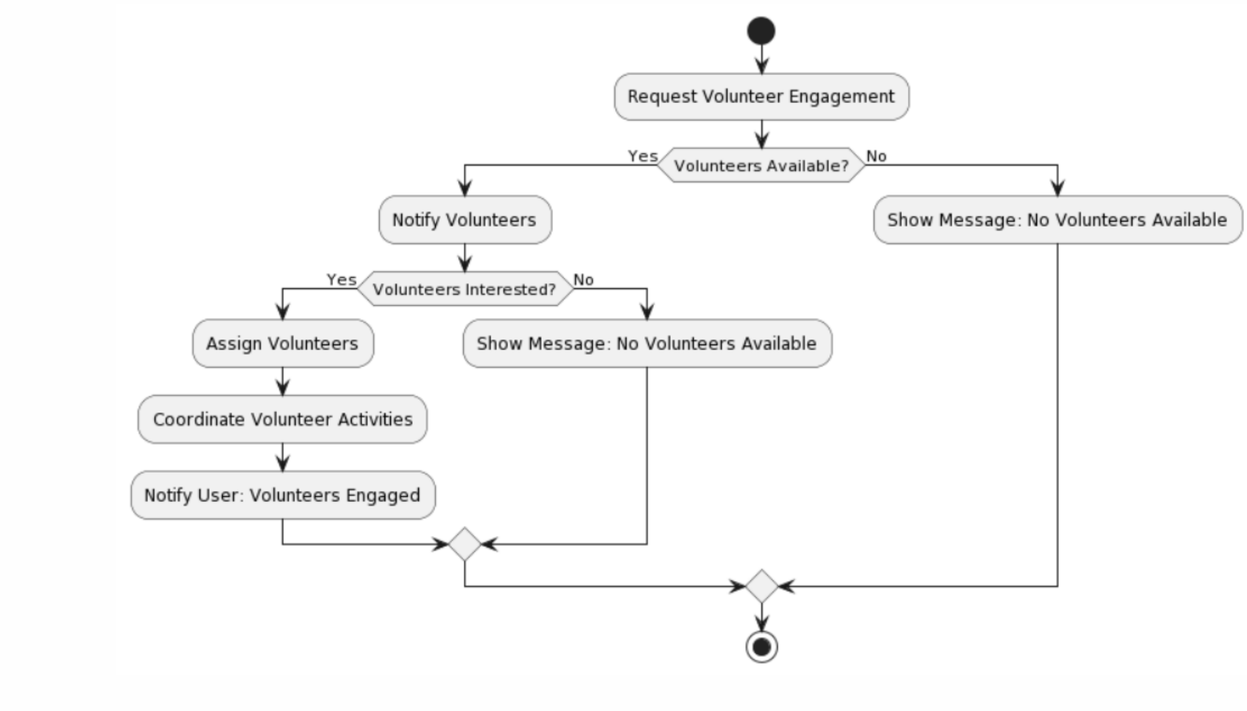
Request Volunteer Engagement
Yes
Volunteers Available?
No
Notify Volunteers
Show Message: No Volunteers Available
Yes
Volunteers Interested?
No
Assign Volunteers
Show Message: No Volunteers Available
Coordinate Volunteer Activities
Notify User: Volunteers Engaged

Figure 22 Engage Volunteers

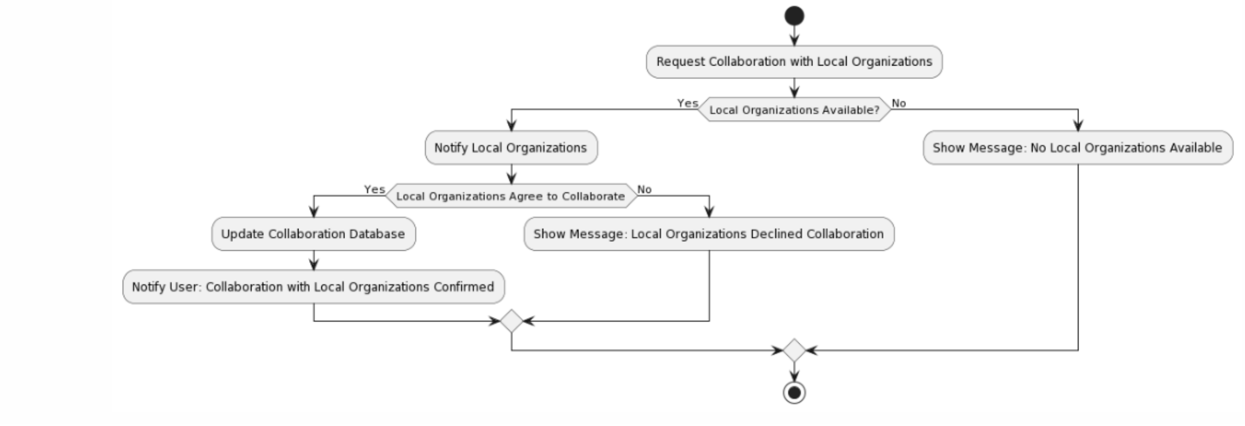
Request Collaboration with Local Organizations
Yes
Local Organizations Available?
No
Notify Local Organizations
Show Message: No Local Organizations Available
Yes
Local Organizations Agree to Collaborate
No
Update Collaboration Database
Show Message: Local Organizations Declined Collaboration
Notify User: Collaboration with Local Organizations Confirmed

Figure 23 Collaborate with Local Organizations

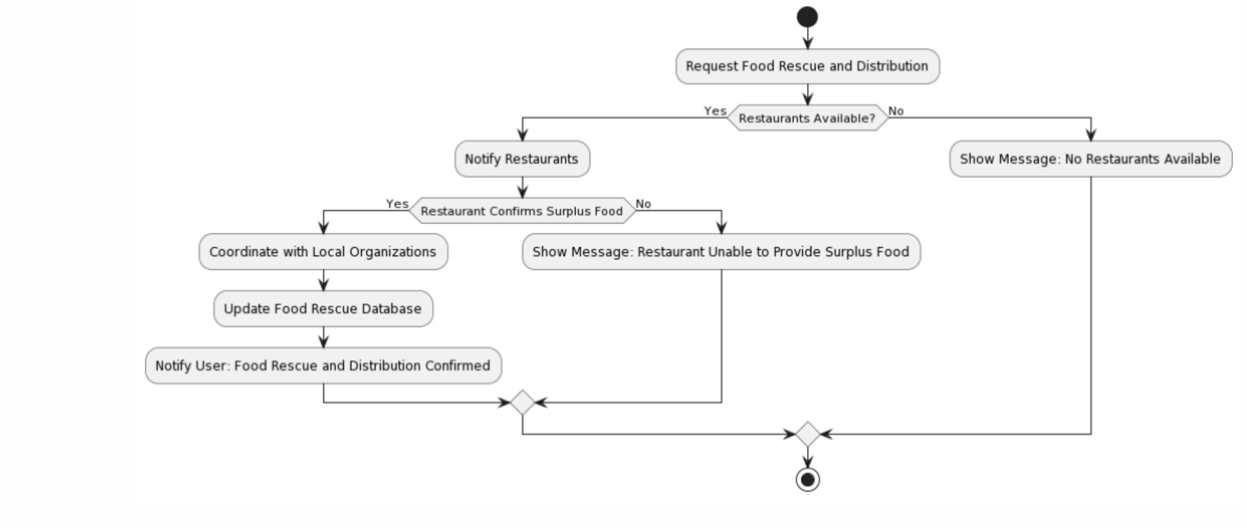
Request Food Rescue and Distribution
Yes
Restaurants Available?
No
Notify Restaurants
Show Message: No Restaurants Available
Yes
Restaurant Confirms Surplus Food
No
Coordinate with Local Organizations
Show Message: Restaurant Unable to Provide Surplus Food
Update Food Rescue Database
Notify User: Food Rescue and Distribution Confirmed

Figure 24 Manage Food Rescue and Distribution

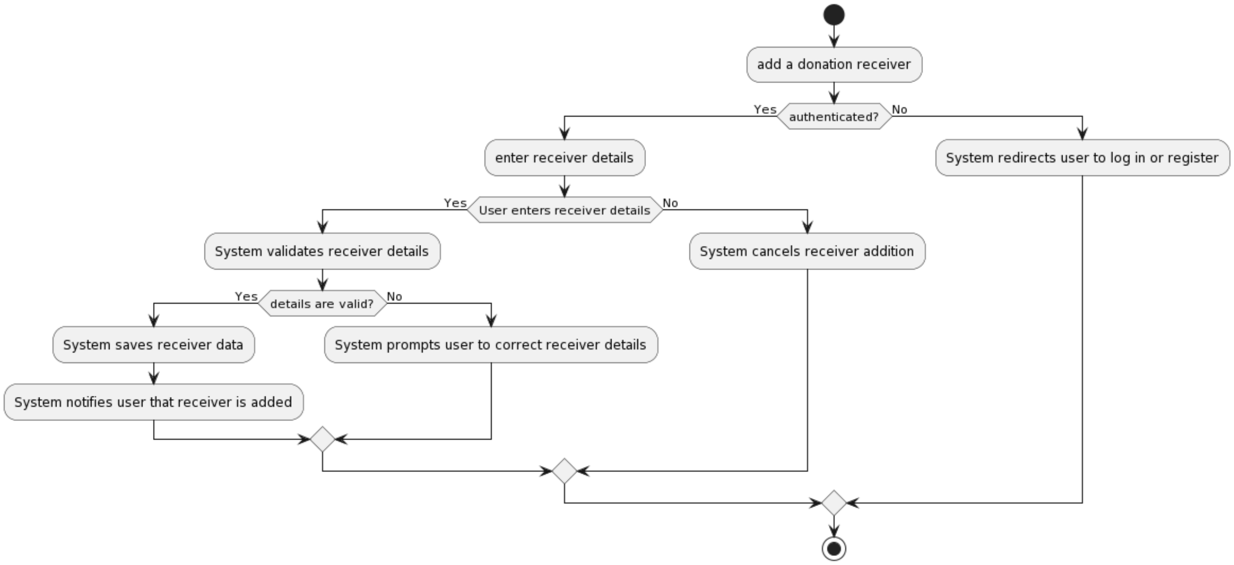
add a donation receiver
Yes
authenticated?
No
enter receiver details
System redirects user to log in or register
Yes
User enters receiver details
No
System validates receiver details
System cancels receiver addition
Yes
details are valid?
No
System saves receiver data
System prompts user to correct receiver details
System notifies user that receiver is added

Figure 25 Add Donation Receiver

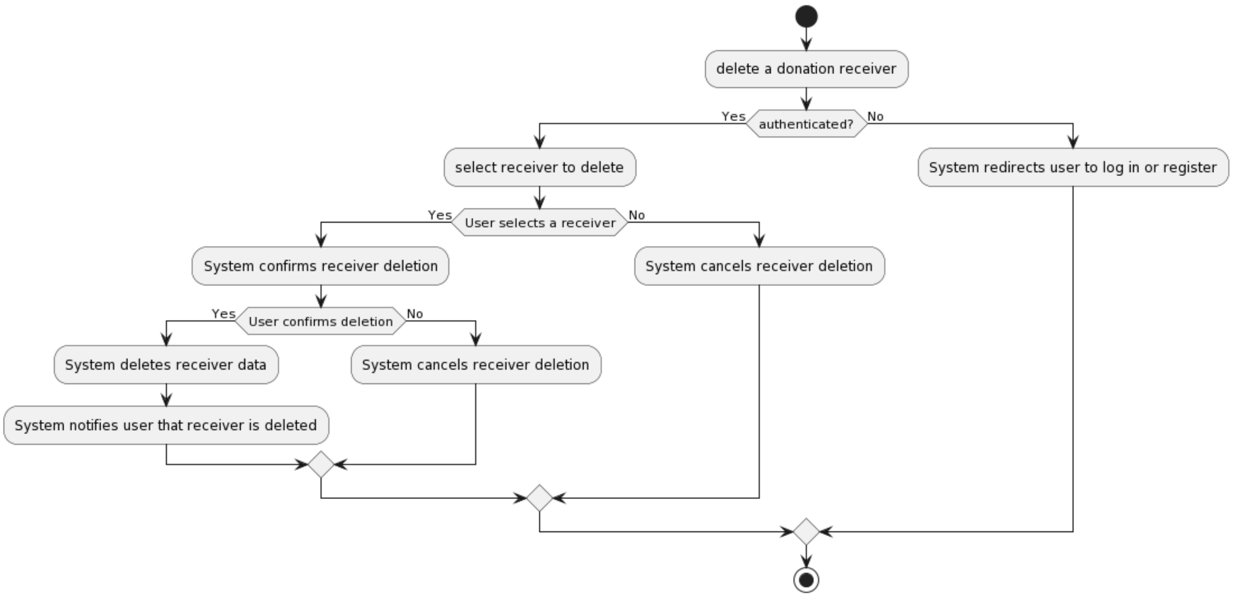
delete a donation receiver
authenticated?
Yes
No
select receiver to delete
System redirects user to log in or register
User selects a receiver
Yes
No
System confirms receiver deletion
System cancels receiver deletion
User confirms deletion
Yes
No
System deletes receiver data
System cancels receiver deletion
System notifies user that receiver is deleted

Figure 26 Delete Donation Receiver

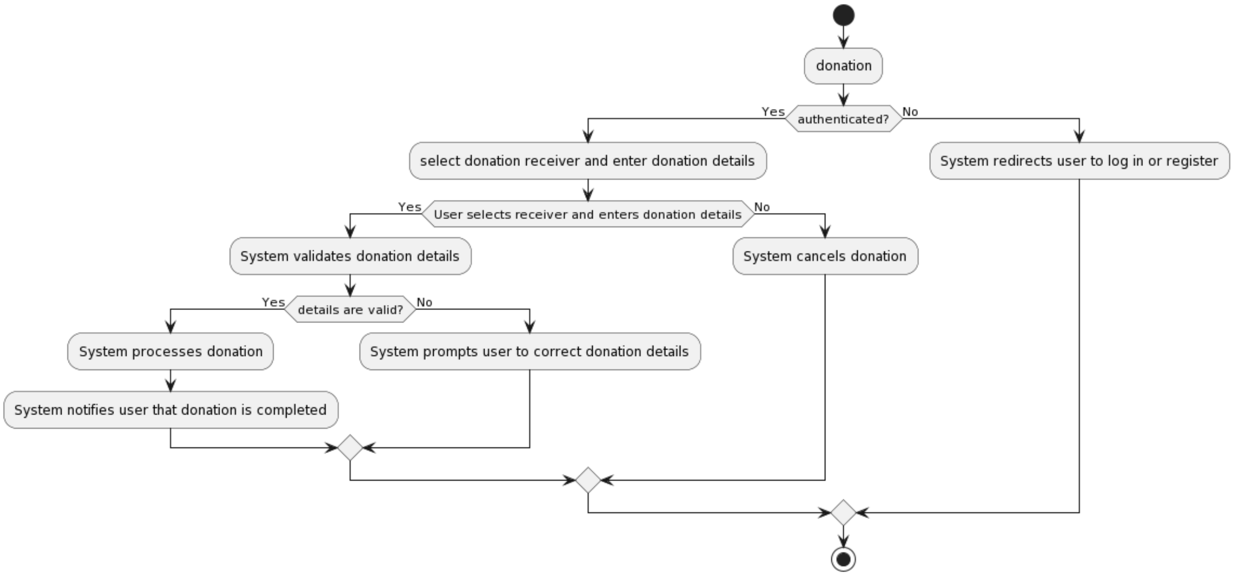
donation
authenticated?
Yes
No
select donation receiver and enter donation details
System redirects user to log in or register
User selects receiver and enters donation details
Yes
No
System validates donation details
System cancels donation
details are valid?
Yes
No
System processes donation
System prompts user to correct donation details
System notifies user that donation is completed

Figure 27 Donate

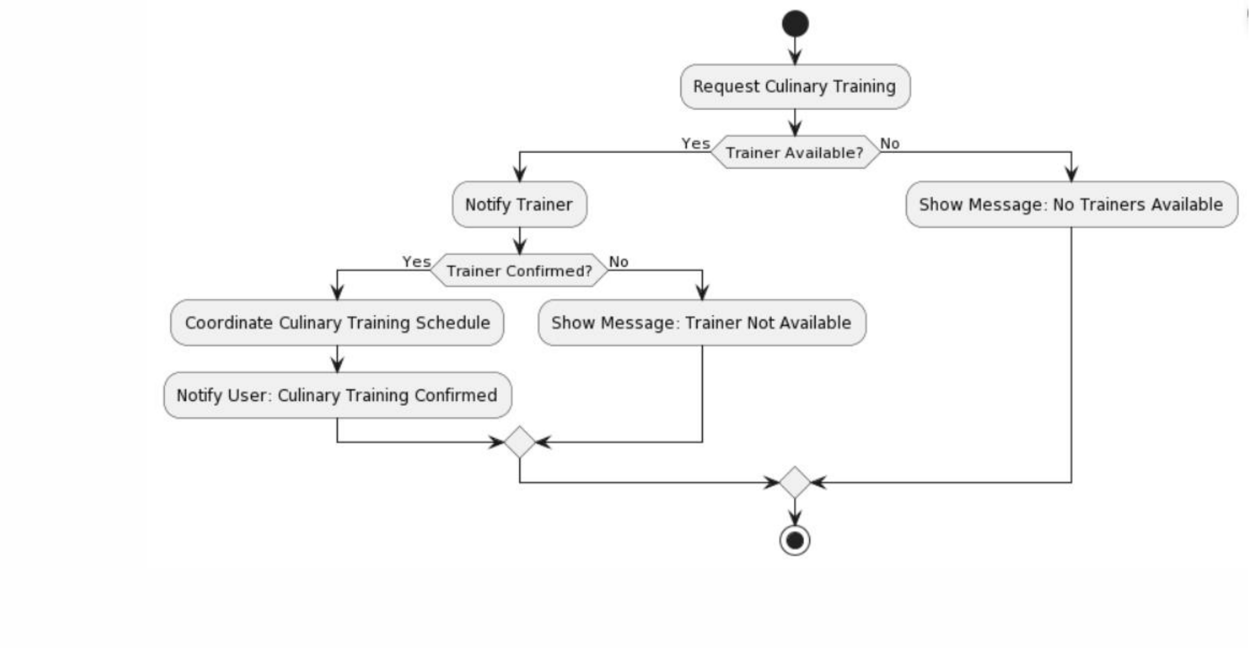
Request Culinary Training
Yes
Trainer Available?
No
Notify Trainer
Show Message: No Trainers Available
Yes
Trainer Confirmed?
No
Coordinate Culinary Training Schedule
Show Message: Trainer Not Available
Notify User: Culinary Training Confirmed

Figure 28 Conduct Culinary Training

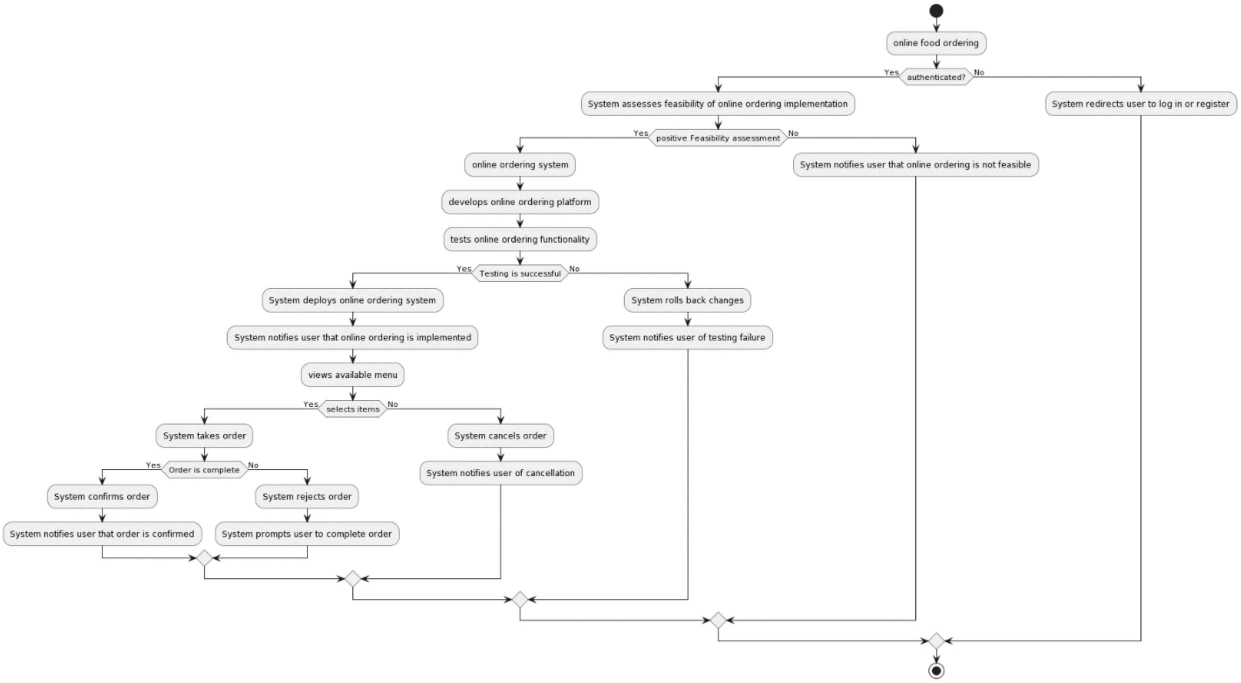
online food ordering
Yes
authenticated?
No
System assesses feasibility of online ordering implementation
System redirects user to log in or register
Yes
positive Feasibility assessment
No
online ordering system
System notifies user that online ordering is not feasible
develops online ordering platform
tests online ordering functionality
Yes
Testing is successful
No
System deploys online ordering system
System rolls back changes
System notifies user that online ordering is implemented
System notifies user of testing failure
views available menu
Yes
selects items
No
System takes order
System cancels order
Yes
Order is complete
No
System notifies user of cancellation
System confirms order
System rejects order
System notifies user that order is confirmed
System prompts user to complete order

Figure 29 Implement Online Food Ordering

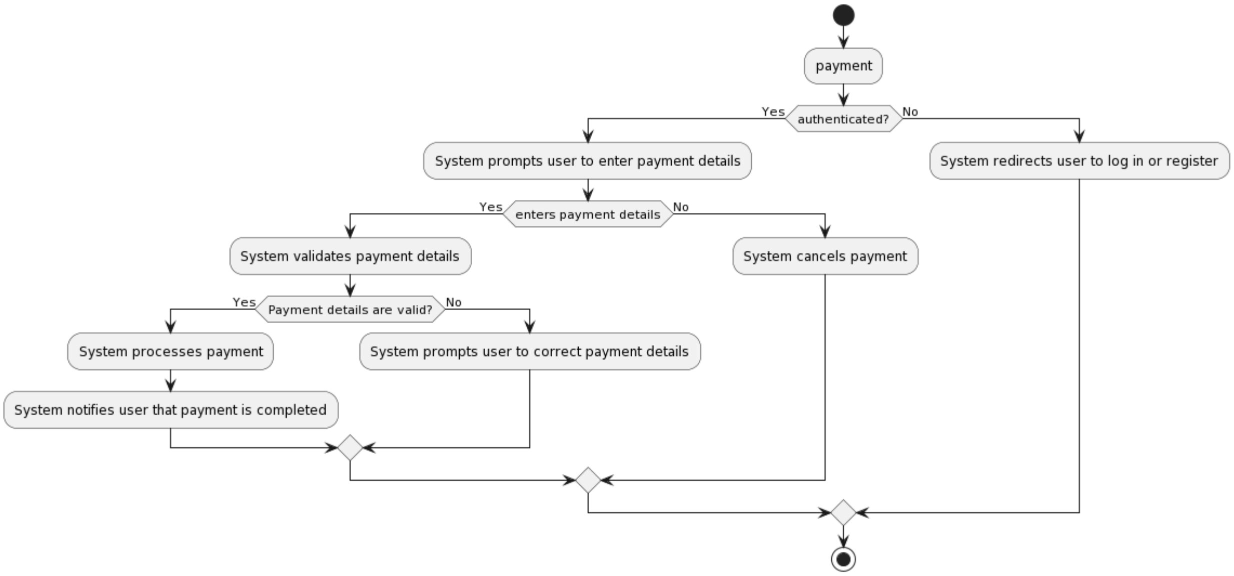
payment
authenticated?
Yes
No
System prompts user to enter payment details
System redirects user to log in or register
enters payment details
Yes
No
System validates payment details
System cancels payment
Payment details are valid?
Yes
No
System processes payment
System prompts user to correct payment details
System notifies user that payment is completed

Figure 30 Make Payment

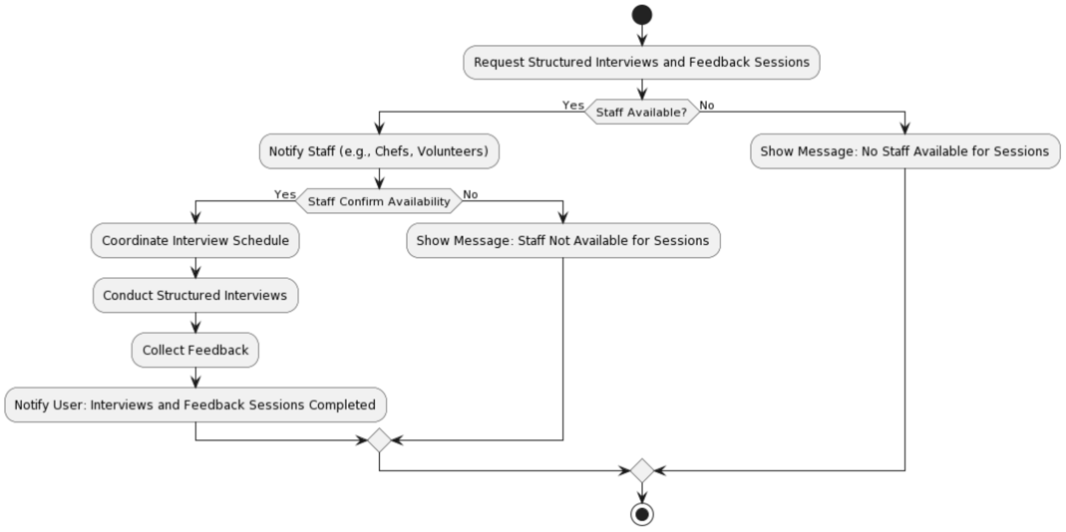
Request Structured Interviews and Feedback Sessions
Staff Available?
Yes
No
Notify Staff (e.g., Chefs, Volunteers)
Show Message: No Staff Available for Sessions
Staff Confirm Availability
Yes
No
Coordinate Interview Schedule
Show Message: Staff Not Available for Sessions
Conduct Structured Interviews
Collect Feedback
Notify User: Interviews and Feedback Sessions Completed

Figure 31 Conduct Structured Interviews and Feedback Sessions

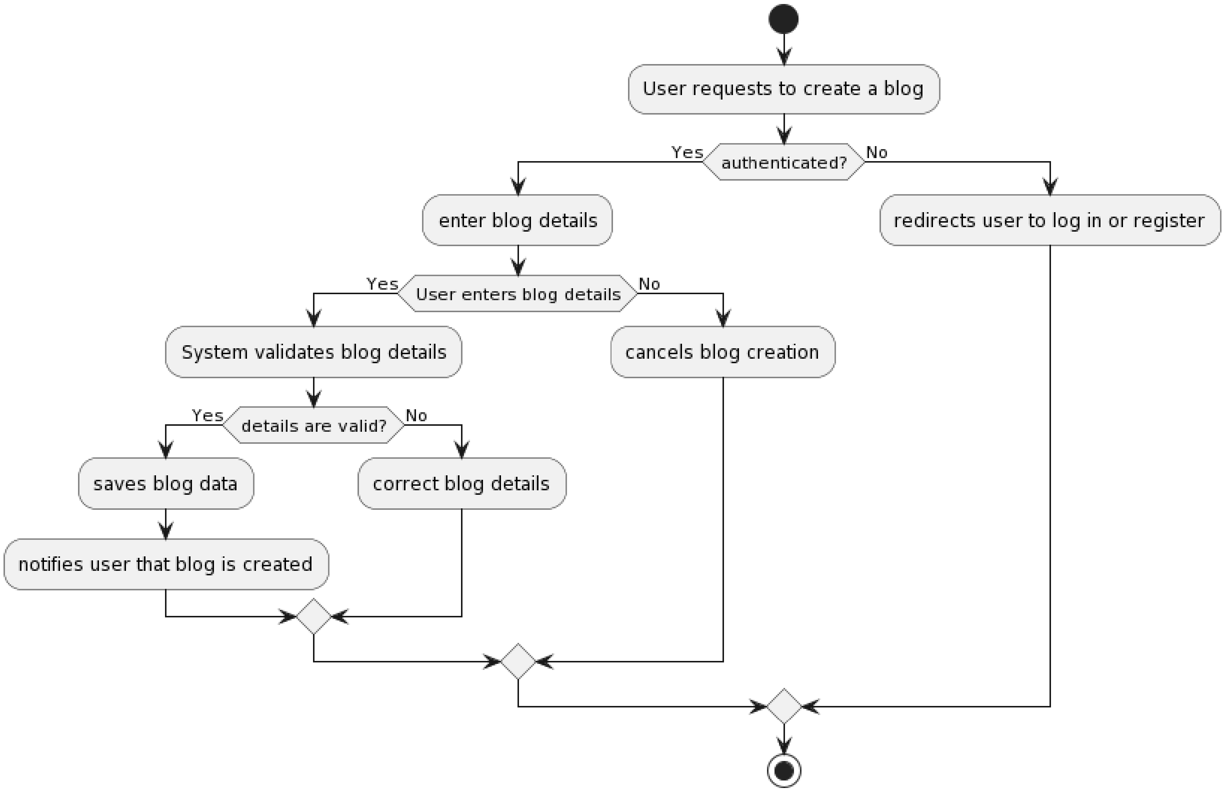

Figure 32 Create Blog

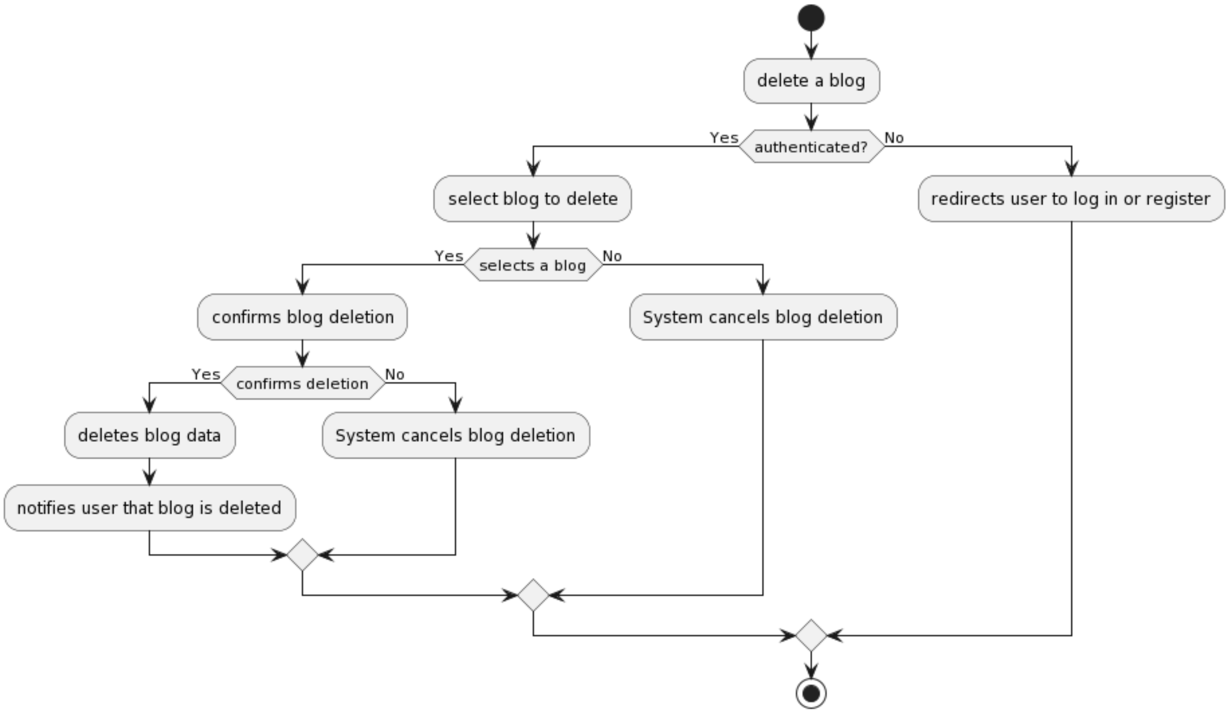

Figure 33 Delete Blog

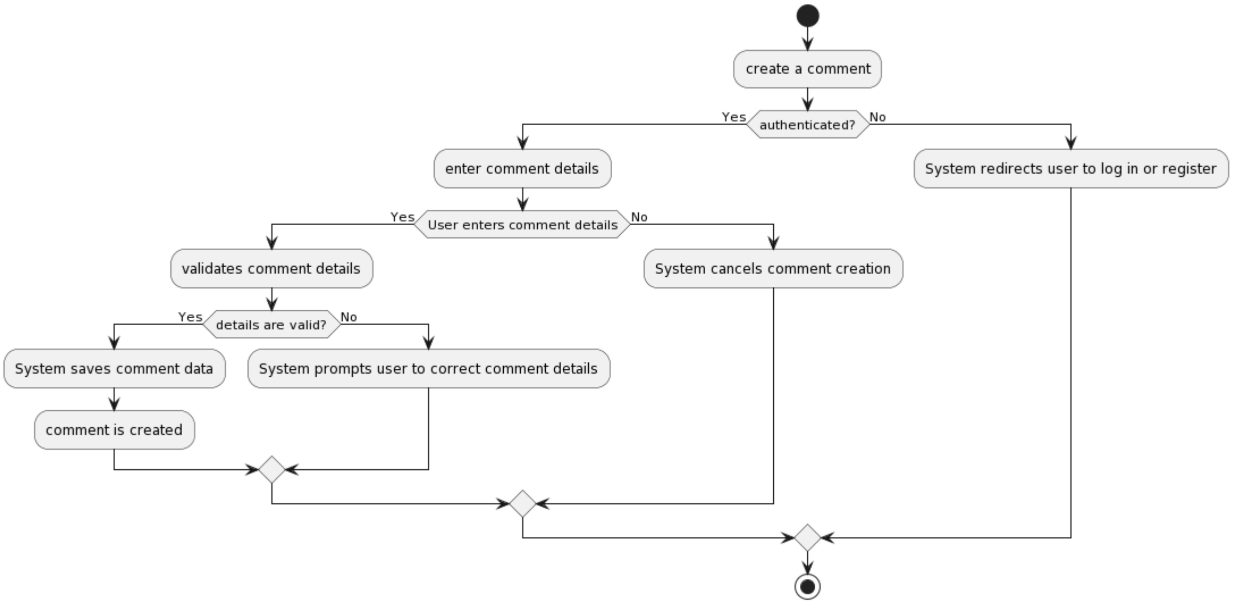
create a comment
authenticated?
Yes
No
enter comment details
System redirects user to log in or register
User enters comment details
Yes
No
validates comment details
System cancels comment creation
details are valid?
Yes
No
System saves comment data
System prompts user to correct comment details
comment is created

Figure 34 Create Comment

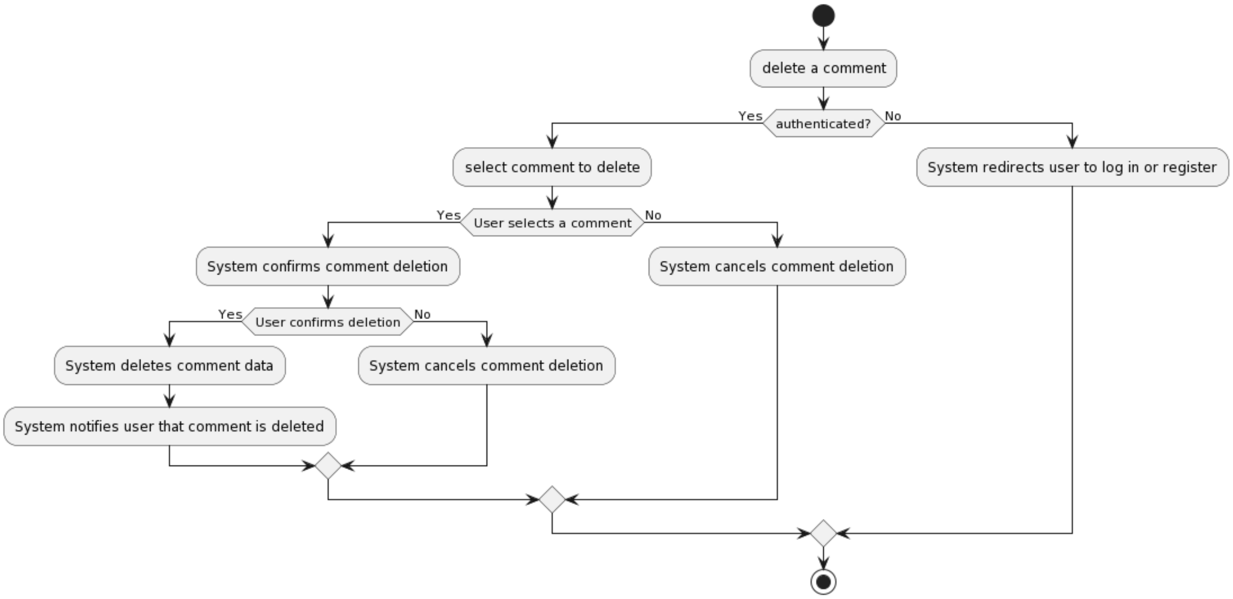
delete a comment
authenticated?
Yes
No
select comment to delete
System redirects user to log in or register
User selects a comment
Yes
No
System confirms comment deletion
System cancels comment deletion
User confirms deletion
Yes
No
System deletes comment data
System cancels comment deletion
System notifies user that comment is deleted

Figure 35 Delete Comment

6.6. Swim Lane Diagram

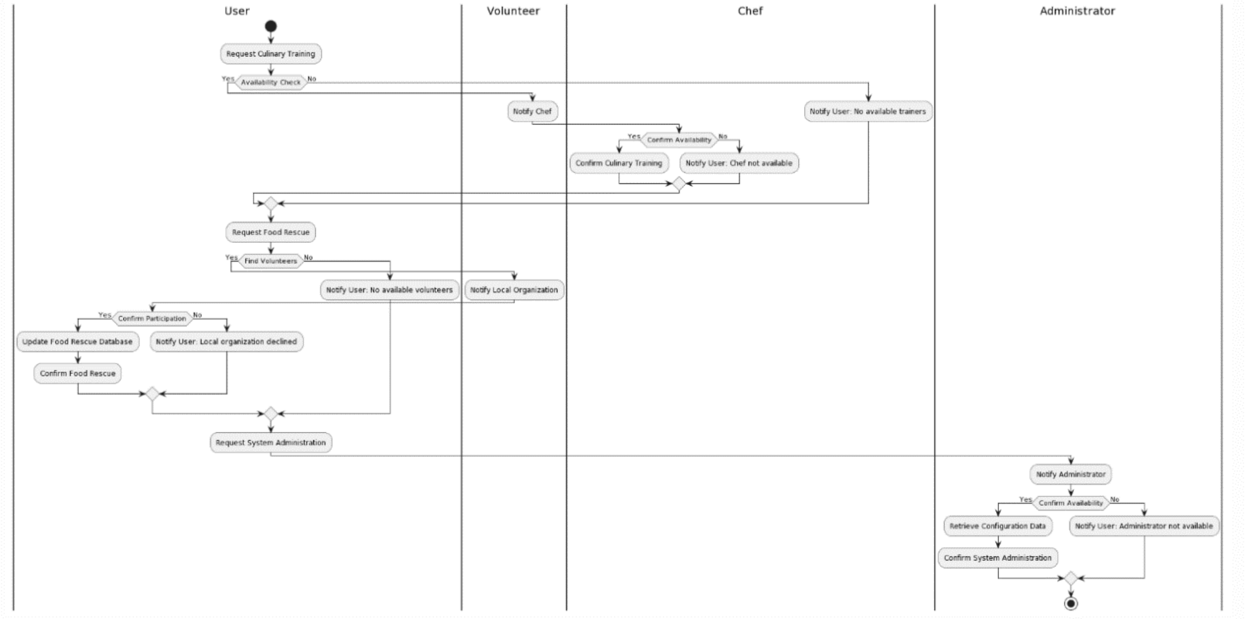
User
Volunteer
Chef
Administrator
Request Culinary Training
Yes
Availability Check
No
Notify Chef
Notify User: No available trainers
Yes
Confirm Availability
No
Confirm Culinary Training
Notify User: Chef not available
Request Food Rescue
Yes
Find Volunteers
No
Notify User: No available volunteers
Notify Local Organization
Yes
Confirm Participation
No
Update Food Rescue Database
Notify User: Local organization declined
Confirm Food Rescue
Request System Administration
Notify Administrator
Yes
Confirm Availability
No
Retrieve Configuration Data
Notify User: Administrator not available
Confirm System Administration

Figure 36 Swim Lane Diagram