

[AIFoodFloow]

[CSCI313: Software Engineering]



by

Philopater Ayman: 211001925

SARA ABOELYAZEED: 211001641

MICHEAL ADEL: 211001760

SALMA ESSAM: 211001778

HYSSEIN AHMED: 211000504

Our instructor

Dr Radwa Mohamed Tawfik

TA HossamEldin Mostafa

**Table of Contents**

1.Introduction............................................................................................................................ 3  
 1.1. Purpose of the document................................................................................................. 3  
 1.2. Scope of the project......................................................................................................... 3  
 1.3. Technologies Used........................................................................................................... 3  
 1.4. Intended Audience........................................................................................................... 4  
 1.5. Overview of the document.............................................................................................. 4

2. Overall Description............................................................................................................... 5

2.1 Product Perspective.......................................................................................................... 5

2.2 User Characteristics.......................................................................................................... 5

2.3 Operating Environment.................................................................................................... 6

2.4 Constraints.........................................................................................................................6

2.4.1 Software Constraints............................................................................................... 6

2.5 Assumptions and Dependencies....................................................................................... 6

2.6 External Services.............................................................................................................. 7

3. Functional Requirements....................................................................................................... 8

3.1 User Interface (UI) .......................................................................................................... 8

3.2 Admin Dashboard............................................................................................................. 8

3.3 Database........................................................................................................................... 8

3.4 Online Shopping............................................................................................................... 8

3.5 Order Management........................................................................................................... 8

3.6 AI Recipes Generator........................................................................................................ 8

4. Non-Functional Requirements.............................................................................................. 9

4.1 Performance..................................................................................................................... 9

4.2 Security............................................................................................................................ 9

4.3 Reliability and Backup..................................................................................................... 9

4.4 Usability............................................................................................................................ 9

4.5 System Scalability............................................................................................................ 9

4.6 Browser Compatibility..................................................................................................... 9

4.7 Mobile Responsiveness.................................................................................................... 9

5.Interfaces.............................................................................................................................. 10

5.1 System interface............................................................................................................. 10

5.2 Software Interface.......................................................................................................... 17

5.3 Hardware Interface......................................................................................................... 17

6.Diagrams.............................................................................................................................. 18

6.1. Use case Diagram ......................................................................................................... 18

6.1.1. Use case scenarios................................................................................................ 19

6.2. Sequence Diagrams ...................................................................................................... 32

6.2.1. Create Diet Plans.................................................................................................. 32

6.2.2. Generate Recipes.................................................................................................. 33

6.3. Class Diagram............................................................................................................... 34  
7. Test Cases ........................................................................................................................... 36

**1.Introduction**

**1.1 Purpose of the Document:**

The purpose of this Software Requirements Specification (SRS) document is to articulate the precise requirements and specifications for the development of a comprehensive grocery website and a complementary mobile application. This document serves as a definitive guide that outlines the core features, functionality, and technical criteria needed to create a successful online platform for selling fresh and organic food products, providing users with an enriched shopping experience.

**1.2 Scope of the Project:**

The project involves the development of an online platform for selling fresh and organic food products. It offers a range of features such as online purchasing, secure payment processing, and delivery scheduling. The project encompasses two distinct user interfaces: one for users to browse and buy products and another for administrators to manage the product inventory. A database system will be implemented to store product information, user data, orders, and inventory. Furthermore, the project introduces an innovative feature—the AI Recipes Generator. This component uses artificial intelligence to recommend recipes based on user input, which can be an image or the name of a food item. The project also extends to a mobile application that connects seamlessly with the website, allowing users to order groceries, access recipes, and receive personalized recommendations.

**1.3 Technologies Used:**

* **Version Control:** Use version control systems like GitHub and GitLab to manage code repositories and facilitate collaborative development.
* **User Interface (UI):** HTML, CSS, Bootstrap, JavaScript, and front-end framework (React).
* **Database:** A relational database system such as MySQL.
* Backend Frameworks: Using backend frameworks like Laravel (PHP) to facilitate server-side development.
* **Search and Filtering:** Implement search engines like Elasticsearch to enable users to find products quickly. Use filtering technologies to refine search results.
* **Payment Processing:** Integration with secure payment gateways such as PayPal or VISA to be secure online transactions.
* **Delivery Management:** Development of a delivery management system using technologies like GPS tracking and scheduling software.
* **AI Recipes Generator:** Integration with machine learning and natural language processing using web services or API (ChatGPT) to provide accurate and personalized recipe recommendations.
* **Google Maps API:** Integration with Google Maps API for location-based services.

**1.4 Intended Audience:**

* **Client:** To correspond with the team about the adjustments needed throughout each step.
* **Developers and software engineers:** To have a guide with the required tasks to be implemented.
* **Software Tester:** To outline all the test cases that are set for the system and ensure that it meets the client’s requirements.
* **Designer:** tasked with creating a user-friendly and visually appealing interface.
* **Project manager:** To sketch the plan to follow throughout the project and overseeing the project's development and implementation.

**1.5 Overview of the Document:**

The next section, Overall Description, of this document gives an overview of the functionality of the product. The document is organized into several chapters, each chapter focusing on specific aspects of the project. The subsequent chapters will provide detailed information on the system's functional and non-functional requirements, user interface design, database structure, and more. It will also address the AI Recipes Generator, mobile application development, and integration with the website.

# **2. Overall Description**

# **2.1 Product Perspective**

* The AI-Food Flow website is a comprehensive platform designed to facilitate the online purchase of fresh and organic food products.
* It aims to connect customers with a variety of food items. The system operates as an e-commerce platform with additional features such as an AI Recipes Generator and an AI Coach to enhance the user experience.

A diagram of a server

Description automatically generated

**2.2 User Characteristics**

* The key user groups engaging with the Grocery website are as follows:
* Customers: These are individuals who frequent the platform to explore, search, and procure fresh and organic food products.
* Customers are empowered to create accounts, facilitating personalized experiences that encompass profile management and order tracking.
* Administrators: Taking charge of the platform's backend operations, administrators play a pivotal role in overseeing product listings, managing inventory, and ensuring the smooth processing of customer orders. Their responsibilities encompass maintaining the robust functioning of the system to enhance user experience.

## **2.3 Operating Environment**

* The Grocery website thrives in a web-based ecosystem, seamlessly accessible through ubiquitous web browsers across a spectrum of devices—ranging from desktops and laptops to tablets and smartphones.
* By harnessing cutting-edge web technologies, the platform guarantees a fluid and responsive user experience, irrespective of the user's chosen device.

## **2.4 Constraints**

* The Grocery website faces certain limitations crucial to its functionality. It requires a reliable internet connection for users to seamlessly browse, shop, and process orders, as these operations heavily rely on real-time data retrieval and transactions.
* Ensuring compatibility with various web browsers, addressing security concerns, and managing scalability as the platform expands are additional constraints that influence the website's development and operational scope.

### **2.4.1 Software Constraints**

* For the Grocery website, there are a few software limitations that shape its design and performance. It's important that the website works smoothly with different web browsers like Chrome or Firefox, ensuring everyone can easily use it.
* As more people join and more products are added, the website needs to handle all that extra traffic and data without slowing down—this is what we mean by scalability.
* Security is a top priority, so the site must be super careful with user data and transactions. Choosing the right technology and making sure the website works well on mobile devices are also key considerations. All these factors help create a website that's not just functional but secure and user-friendly.

## **2.5 Assumptions and Dependencies**

* Users are assumed to have basic internet literacy and familiarity with online shopping practices.
* It is assumed that users will provide accurate and valid information during the registration and checkout process.
* The system depends on the continued support and updates of the chosen web development frameworks and libraries.
* The website's performance assumes consistent access to external APIs for features like real-time product availability and pricing.
* The system depends on the reliability of third-party delivery services for the successful completion of the order and delivery process.
* Assumption is made that users will have devices with an acceptable level of security to protect their personal information during online transactions.

## **2.6 External Services**

* AI Recipes Generator: The system assumes a functional link to an external service that generates recipes based on user input, providing information on calories and protein content and then buy the items need from the website.
* AI Coach: Integration with a third-party AI coaching service is assumed, where users can input personal information to receive tailored diet plans with calculated calories and protein intake and then buy the items need from the website.

# **3. Functional Requirements**

# **3.1 User Interface (UI)**

* Allow users to create accounts.
* Provide a user-friendly interface for browsing and searching products.
* Enable users to track and manage their orders.

**3.2 Admin Dashboard**

* Allow administrators to add, update, or delete product listings.
* Provide inventory management functionality.
* Implement a system for processing and managing customer orders.

## **3.3 Database**

* Store details of various products, including name, price, and description.
* Capture and store user information for account management.

## **3.4 Online Shopping**

* Enable users to browse and view product details.
* Implement a shopping cart for users to add, modify, and confirm their orders.
* Secure Payment Processing:
* Integrate secure payment options for credit/debit card payments and online gateways.

## **3.5 Order Management**

* Develop a system for users to track their orders in real-time.

## **3.6 AI Recipes Generator**

* Allow users to input food names to receive recipe recommendations.

# 

# **4. Non-Functional Requirements**

# **4.1 Performance**

* Ensure the website responds within 3 seconds for user interactions.

## **4.2 Security**

* The website must be able to withstand a variety of cyber-attacks.

## **4.3 Reliability and Backup**

* Aim for at least 99% system uptime.
* Implement regular data backups and a robust recovery plan.

## **4.4 Usability**

* Ensure the website is intuitive and easy to navigate for users of all ages.

## **4.5 System Scalability**

* Design the system architecture to handle an increasing number of users and products.

## **4.6 Browser Compatibility**

* Ensure compatibility with popular web browsers (Chrome, Firefox, Safari, Edge).

## **4.7 Mobile Responsiveness**

* Design the website to be responsive and accessible on various devices, especially mobile phones.

**5. Interface**

**5.1 System Interface**

When the user opens the grocery website for the first time, they are going to see a sign-up page as shown in Figure 1. In addition to this, a sign in option is available for users who already have registered accounts as shown in Figure 2.

When the user picks to sign up and create account, he is asked to give his name, email, password, phone-number as shown in Figure 1. After the user signs up, the user will be able to sign in with his account.

A screenshot of a login form

Description automatically generatedFigure 1

A screenshot of a login form

Description automatically generatedFigure 2

The Sign in emails and passwords will send and save to the database to inform the admin how many users in the system and to make system analysis.

A screenshot of a website

Description automatically generated

Figure 3

In figure 3, you can see that after the user signs in, the user will be able to view the home page where he can browse through all the different grocery items and order what he need. The user can search by the name of the item he wants, see all the categories that the website has, offers packages, contact us, send feedback, know more about us, download the app, the user can see his Wishlist, shopping cart and his profile to edit his personal information.

A close-up of a white square with black and white icons

Description automatically generated

Figure 4

In figure 4, you can see that the user can choose any category the user wants like fruits and vegetables, medicine, baby care, stationary, beauty and gardening as will show in figure 5, 6, 7, 8, 9, 10.

**The categories**

A screenshot of a phone

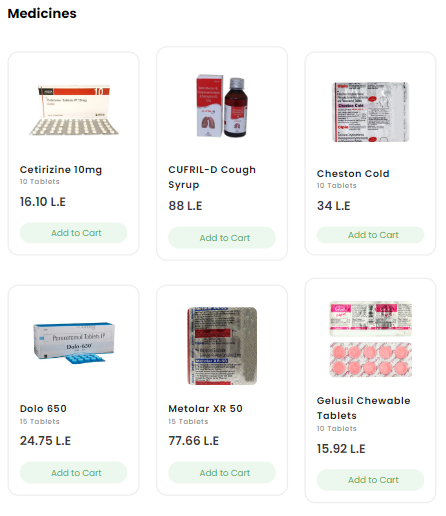
Description automatically generated 

Figure 5 Figure 6

The user can see all the items and select what he wants by clicking on the add to cart button.

A screenshot of a product price list

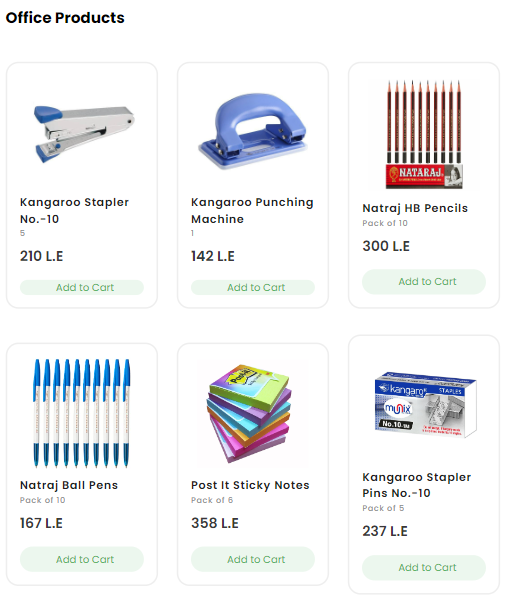
Description automatically generated 

Figure 7 Figure 8

The user can see all the items and select what he wants by clicking on the add to cart button.

A screenshot of a product

Description automatically generatedA screenshot of a phone

Description automatically generated

Figure 9 Figure 10

The user can see all the items and select what he wants by clicking on the add to cart button.

A screenshot of a grocery store

Description automatically generatedFigure 11

In figure 11, you can see that the website offers some popular bundle packs user can choose any of them by clicking on add to cart button.

A screenshot of a phone

Description automatically generated

Figure 12

In figure 12, you can see that the website shows customer reviews and opinions about the site to encourage them and give them enough reassurance to purchase.A screenshot of a phone

Description automatically generatedFigure 13

In figure 13, you can see that the website offers to the client that he can download the app from play store and app store for easier shopping and to use the AI new features like AI recipe generator and AI coach for diet planer.

A screenshot of a phone

Description automatically generated

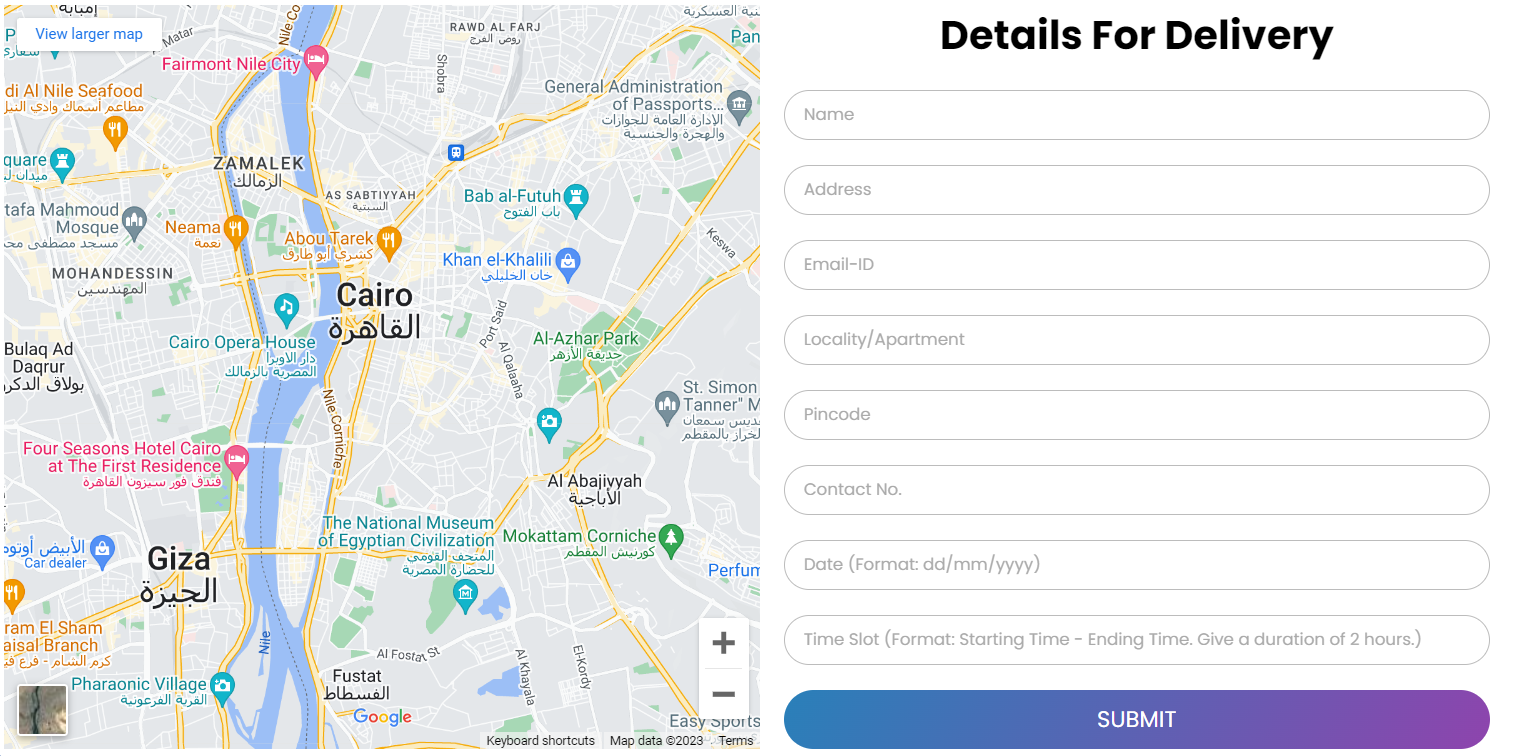
In figure 14, users can contact the admin and send their feedback by clicking on contact us, feedback buttons. The user must fill the form with his name, subject, phone, email, and the message he wants to send.

Figure 14

A screenshot of a computer

Description automatically generatedFigure 15

In figure 15, you can see that user can see his Wishlist by clicking on the Wishlist button and can remove any of the selected items any time.

Figure 16

In figure 16, after the user clicked on buy now button and proceed to checkout the user must fill in details for delivery like his name, address, email, location, pin code, contact number, date, and his preferred time slot to deliver his delivery items.

Then the AI recipe generates according to the user ingredients

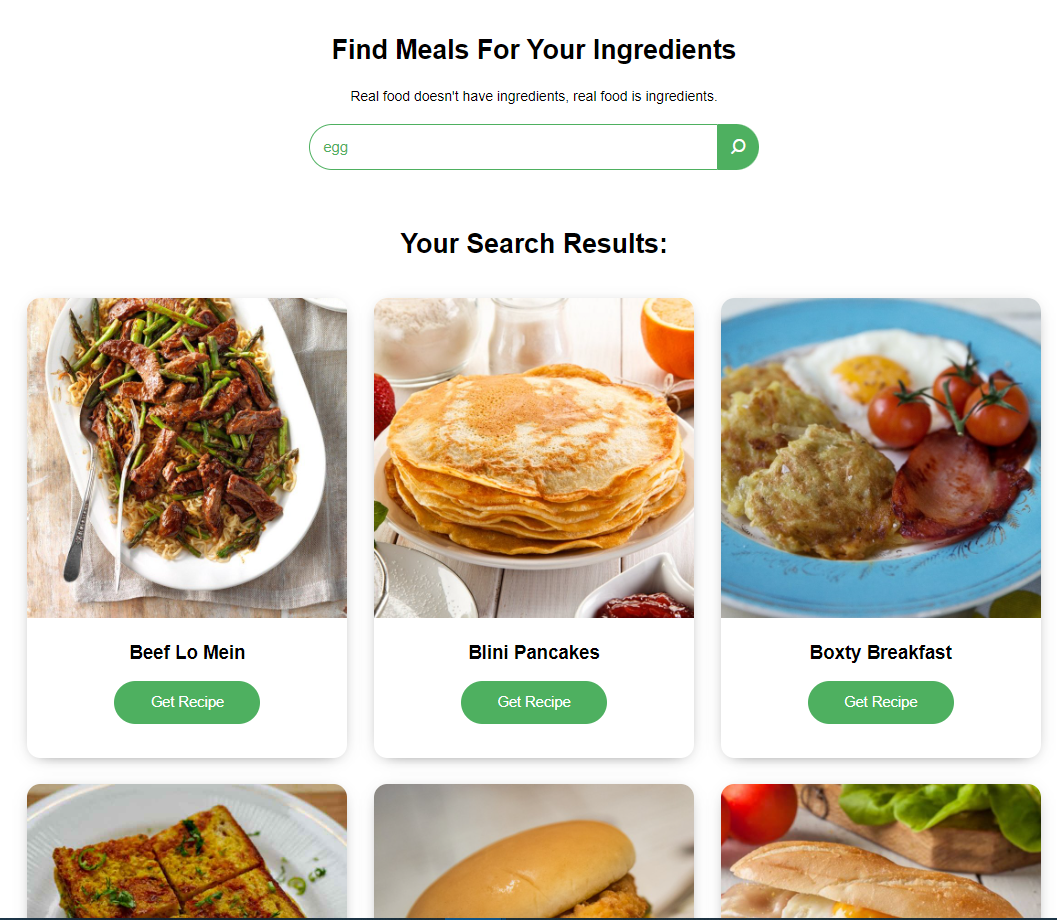
**The AI recipe generates System Interface:**

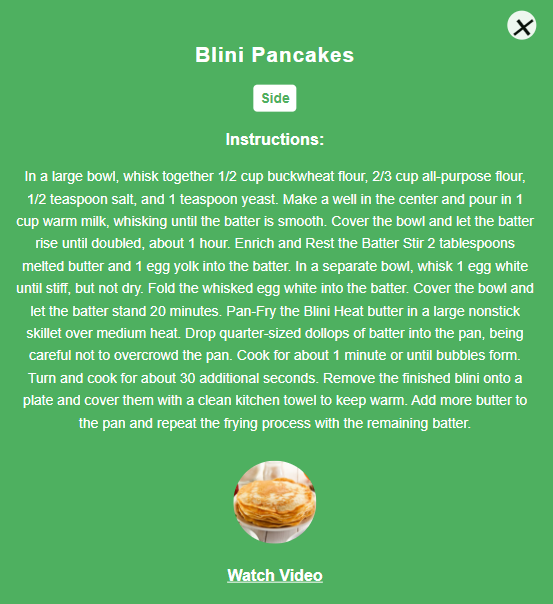
A screenshot of a recipe search

Description automatically generated

Figure 17

In figure 17, after the user clicking on the ai recipe generator button the user will go to new website page, the user will be able to put ingredient he has to search for meals he wants to make like pancake he will search with egg or dough.



Then the user clicks on get recipe button to know the ingredients of the selected meal. the recipe ingredients and the instruction appears in detailed to help the user know what the missing items he wants to buy from the grocery website to complete making the meal and there is a link to watch how to make the recipe via a YouTube video.

**5.2 Software Interface**

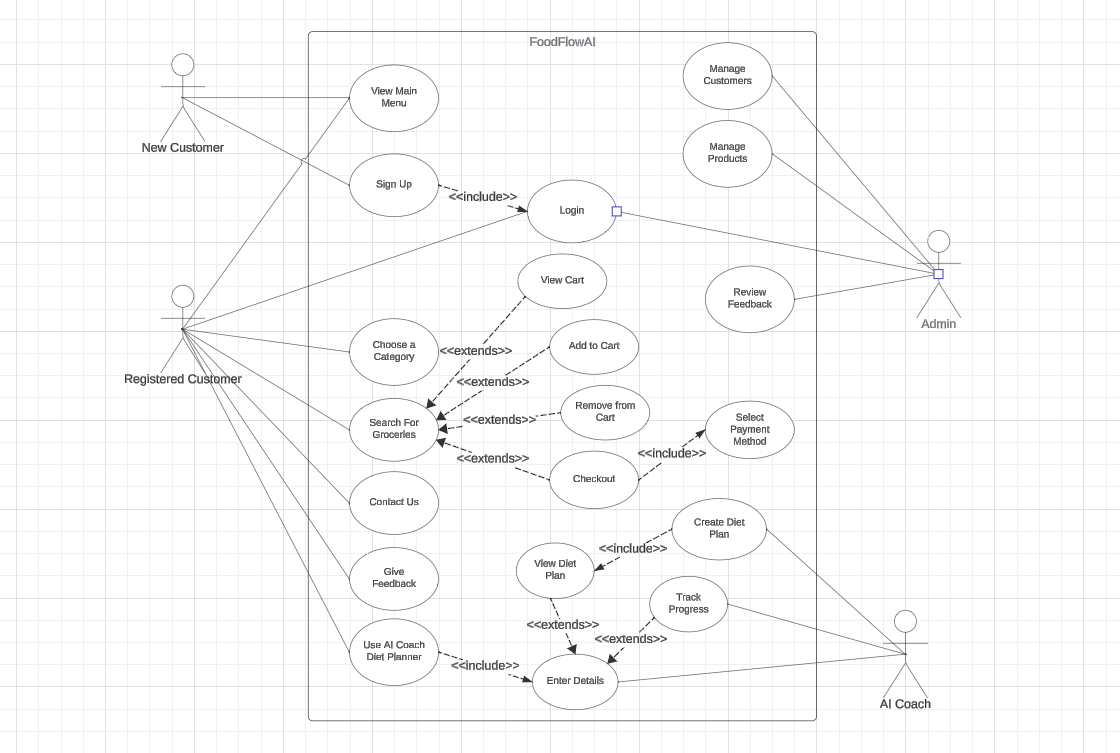
The 'AIFoodFlow' website is designed as a responsive platform crafted using a blend of technologies, including HTML, CSS, and JavaScript. It leverages frameworks like React.js for seamless user interactions and dynamic content rendering. Back-end functionalities are developed to manage and store user profiles, product listings, and order data. The platform ensures a fluid exchange of information between users and the server, handling outgoing data, such as user interactions, searches, and orders, and receiving incoming data, including product details, user profiles, and notifications.

**5.3. Hardware Interface**

“AIFoodFlow” website will require internet connection to function. Another crucial thing is that the laptop or phone device need to support GPS, internet connection.

# **6. Diagrams**

## **6.1 Use case Diagram**



**6.1.1. Use case scenarios.**

Use Case #1

A black arrow pointing to a blue oval

Description automatically generated

|  |  |
| --- | --- |
| User Case Name | View Main Menu |
| Actors | New Customer / Registered Customer |
| Main success scenario | 1. The user accesses the website/application and navigates to the main menu. 2. The system displays the main menu options, including categories like 'Fresh Produce,' 'Meat & Poultry,' 'Dairy Products,' etc. 3. The user selects a category or browses through various sections within the main menu. 4. The system displays the selected category/products along with their details (such as name, description, price, etc.). |
| Actions | * User opens the application/website. * User navigates to the main menu section. * User explores different categories or sections within the main menu. * User selects a specific category or product to view details. |
| Pre-Conditions | * The user has a functional device (PC, smartphone, etc.) with internet access. * The user is logged in as a registered customer or is a new visitor to the site. |
| Post-Conditions | * The user gains an understanding of available product categories and their details. * The user can proceed to browse, add items to the cart, or explore more features of the website. |

Use Case #2

A black arrow pointing to a blue oval

Description automatically generated

|  |  |
| --- | --- |
| User Case Name | Sign up |
| Actors | New Customers |
| Main success scenario | 1. When a new unregistered customer opens the app, they can’t do anything without being registered. 2. The user clicks on the sign up which takes him to another page. 3. The system asks him to enter First name, Last name, email address, phone no., and password. 4. The user clicks on register. 5. The system takes the user to the login page. 6. In the login page the user enters his email and password to confirm registration. |
| Exceptions | * User enters a wrong email or phone number. * Email already registered. |
| Actions | * System displays to the user an alert ‘‘Please fill in all the required fields’’. * User fills missing information. * System display ‘‘Email already registered’’. * User enters another email. |
| Pre-Condition | * The user is a new visitor or non-registered customer. * The user has a valid email address for account verification. * The user has access to the website/application. |
| Post-Condition | * User is added to the system. * User is successfully registered. * User information is stored in the database. |

Use Case #3

A black arrow pointing to a blue oval

Description automatically generated

|  |  |
| --- | --- |
| Use ase Name. | Login |
| Actors | Registered Customers / Admin |
| Main success scenario | 1. Access Login Page: The user navigates to the login page. 2. Enter Credentials: The user enters their username and password in the designated fields. 3. Validate Credentials: The system validates the entered credentials against the stored database. 4. Successful Login: If the credentials are valid, the system grants access and redirects the user to the main dashboard or desired section. 5. Session Initiation (For Admins): A secure session is established, allowing the admin to perform authorized actions within the system. |
| Exceptions | * Invalid Credentials: If the username or password is incorrect, the system displays an error message and prompts the user to try again. * Account Locked: If the user exceeds the maximum allowed login attempts with invalid credentials, their account might be temporarily locked up for security purposes. |
| Actions | * User selects the 'Log In' option. * User provides registered credentials (username/email and password). |
| Pre-condition | * The user has previously created an account on the platform. * The user knows their registered username/email and password. |
| Post-condition | * Successful login provides access to the user's account and its associated features. * The system authenticates the user's credentials for subsequent interactions. |

Use Case #4

A black arrow pointing to a blue oval

Description automatically generated

|  |  |
| --- | --- |
| User case Name | Choose a category |
| Actors | Registered Customer |
| Main success scenario | 1. User presses the ‘Choose a category’ button. 2. System gives the user options (e.g. Fruits, vegetables, meat, etc.) 3. User chooses between these options then clicks search. 4. System shows products depending on the chosen option. |
| Exceptions | * The user doesn’t choose a category. |
| Actions | * User navigates to the main menu or category section. * User selects a category from the available list. |
| Pre-condition | * The user has access to the website/application. * The system has multiple categories with respective items available for selection. |
| Post-condition | * Upon selection, the chosen category's items are displayed to the user. * The system loads and presents the relevant products within the selected category. |

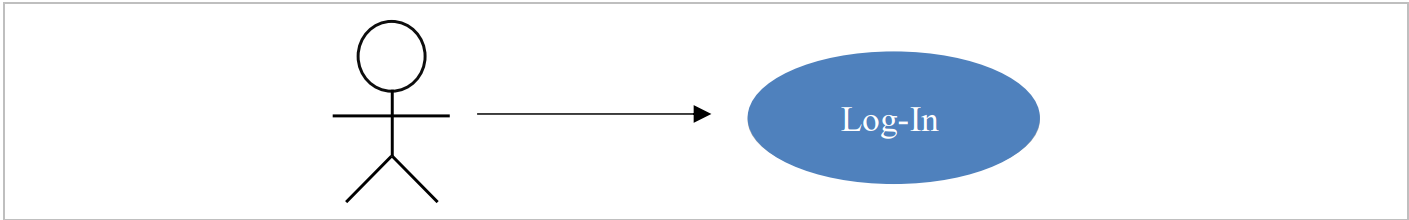
Use Case #5

A black arrow pointing to a blue oval

Description automatically generated

|  |  |
| --- | --- |
| User case Name | Search for Groceries / View Cart / Add to Cart / Remove from Cart / Checkout / Select payment method |
| Actors | Registered Customers |
| Main success scenario | 1. User presses the search for groceries button after typing the grocery name. 2. The system shows multiple products related to the name entered with an add to cart button and a remove from cart if the product is already in the cart. 3. There is also a view cart button to view the products in the cart and to checkout. 4. If the user clicks checkout the system asks for a payment method whether it’s by card or by cash. 5. After the payment method is chosen the user clicks confirm and the payment is done. |
| Exceptions | * The user doesn’t choose a payment method. * Products are out of stock. |
| Actions | * System shows the cart and each item in it. * System edits the items inside the cart according to the user. * System displays a message saying ‘please enter a payment method to proceed. |
| Pre-condition | * User must be logged-in |
| Post-condition | * Payment confirmed |

Use Case #6



|  |  |
| --- | --- |
| User case name | Contact Us |
| Actors | Registered Customer |
| Main success scenario | 1. The user navigates to the "Contact Us" section within the website/application. 2. The system presents a form or interface for the user to input their query or message. 3. The user fills in their name, email, subject, and message details. 4. The system validates the information and sends the query to the support team. 5. The support team receives the query and responds accordingly. |
| Exceptions | * User enters unsupported characters in the message |
| Actions | * User navigates to the "Contact Us" section. * User fills in the contact form with necessary details. * User submits the form. |
| Pre-Conditions | * The user has access to the "Contact Us" section. * The system has a functional contact form. * The support team is available to receive and respond to queries. |

Use Case #7

A black arrow pointing to a blue oval

Description automatically generated

|  |  |
| --- | --- |
| User case name | Give Feedback |
| Actors | Registered Customer |
| Main success scenario | 1. The user accesses the "Feedback" section on the website/application. 2. The system presents a form or interface for the user to provide feedback. 3. The user enters their feedback, ratings, and any additional comments. 4. The system validates the input and records the feedback. 5. Optionally, the system acknowledges successful submission. |
| Exceptions | * User enters unsupported characters in the message |
| Actions | * User navigates to the "Feedback" section. * User fills in the feedback form. * User submits the feedback. |
| Pre-Conditions | * The user has access to the "Feedback" section. * The system has a functional feedback form. * The system can handle and store user feedback. |

Use Case #8

A black arrow pointing to a blue oval

Description automatically generated

|  |  |
| --- | --- |
| User case name | Manage Products |
| Actors | Admin |
| Main success scenario | 1. Access Products: The admin logs in to the system and navigates to the "Manage Products" section. 2. View Product List: The admin is presented with a list of all products in the system, including their names, descriptions, prices, images, and inventory levels. 3. Search & Filter: The admin can search for specific products using keywords, filters by category, brand, or other criteria. 4. View Product Details: The admin can click on a product to view its detailed information, including specifications, reviews, and sales data. 5. Add New Product: The admin can click the "Add Product" button and enter details like name, description, price, category, brand, specifications, images, and inventory. 6. Edit Existing Product: The admin can click "Edit" on a product and modify its details, including name, description, price, category, brand, specifications, and images. 7. Set Inventory Levels: The admin can update the inventory levels for each product to reflect current stock. 8. Publish/Unpublish Products: The admin can control product visibility by publishing them to the storefront or unpublishing them for temporary removal. 9. Bulk Actions: The admin can perform bulk actions on products like deleting, changing prices, or updating categories. 10. Review and Approve Products: (Optional) If a product approval workflow exists, the admin can review and approve newly added or edited products before they become visible to customers. |
| Exceptions | * Insufficient permissions: Some actions might require specific admin roles or permissions. * Product data validation errors: Adding or editing a product might fail if mandatory fields are missing or invalid data is entered. * Inventory updates exceeding stock: Updating inventory levels might not be allowed if it exceeds the available stock. |
| Actions | * View product list and details. * Search and filter products. * Add, edit, and delete products. * Set inventory levels. * Publish/unpublish products. * Perform bulk actions. * Review and approve products (optional). |
| Pre-Condition | * Admin is logged in to the system. * The product management system is accessible. |
| Post-Condition | * Product list is updated with any changes made by the admin. * Product details are updated with any edits. * Inventory levels are adjusted. * Product visibility is controlled based on publishing/unpublishing actions. * New products are added, or existing products are modified. * Product approvals are completed (optional). |

Use Case #9

A black arrow pointing to a blue oval

Description automatically generated

|  |  |
| --- | --- |
| User case name | Manage Customers |
| Actors | Admin |
| Main success scenario | 1. Access Customers: The admin logs in to the system and navigates to the "Manage Customers" section. 2. View Customer List: The admin is presented with a list of all customers in the system, including their names, emails, contact information, purchase history, and account status. 3. Search & Filter: The admin can search for specific customers using keywords, filters by location, activity level, purchase history, or other criteria. 4. View Customer Details: The admin can click on a customer to view their detailed information, including addresses, orders, interactions with support, and any notes or tags. 5. Update Customer Information: The admin can edit customer details like name, email, phone number, and address. 6. Manage Account Status: The admin can activate, deactivate, or suspend customer accounts based on specific criteria or actions. 7. Send Messages: The admin can initiate communication with customers via email, in-app notifications, or other channels to send announcements, updates, or support messages. 8. Review & Resolve Customer Issues: The admin can access and manage customer support tickets, investigate issues, and provide resolutions or escalate cases when needed. 9. Export Customer Data: The admin can export customer data in various formats for reporting, analysis, or marketing purposes. 10. Manage Customer Groups: (Optional) The admin can create and manage customer groups based on demographics, purchase behavior, or other criteria to target them with specific campaigns or offers |
| Exceptions | * Insufficient permissions: Some actions might require specific admin roles or permissions. * Customer data privacy: Exporting or sharing customer data might require adherence to privacy regulations and policies. * Unsuccessful communication attempts: Message delivery might fail due to invalid email addresses or other technical issues. |
| Actions | * View customer list and details * Search and filter customers * Update customer information * Manage account status. * Send messages to customers. * Review and resolve customer issues. * Export customer data * Manage customer groups (optional) |
| Pre-Condition | * Admin is logged in to the system. * Customer management system is accessible. |
| Post-Condition | * The customer list is updated with any changes made by the admin. * Customer details are updated with any edits. * Account status is changed as per admin action. * Customer communication is initiated or resolved. * Customer data is exported or managed as required. * Customer groups are created or managed (optional). |

Use Case #10

A black arrow pointing to a blue oval

Description automatically generated

|  |  |
| --- | --- |
| User case name | Create Diet Plan |
| Actors | User |
| Main success scenario | 1. User accesses the AI Coach section for diet planning. 2. The system prompts the user to input personal details (age, weight, height, goal). 3. AI Coach analyzes user data to generate a personalized diet plan. 4. The system displays the diet plan including meal schedules, nutritional information, and food suggestions. 5. User can review and accept the generated diet plan. 6. The system saves the diet plan in the user's profile. |
| Actions | * User accesses the AI Coach section for diet planning. * User inputs personal details required for planning (age, weight, height, goal). * AI Coach processes the data and creates a tailored diet plan. * The system presents the generated plan including meals, portion sizes, and nutritional data. * User reviews the plan and accepts it for implementation. |
| Pre-Condition | * User is logged into the system. * AI Coach functionality is accessible and operational. * User provides accurate personal data for effective planning. |
| Post-Condition | * The generated diet plan is saved in the user's profile for future reference. * User receives guidance for meals based on the created diet plan. |

Use Case #11

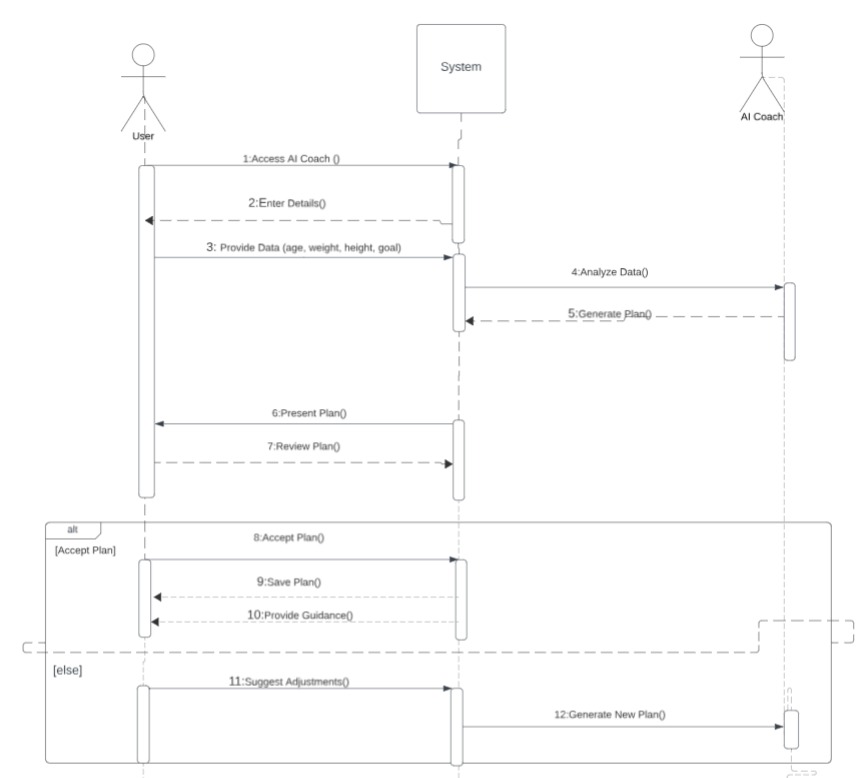
A black arrow pointing to a blue oval

Description automatically generated

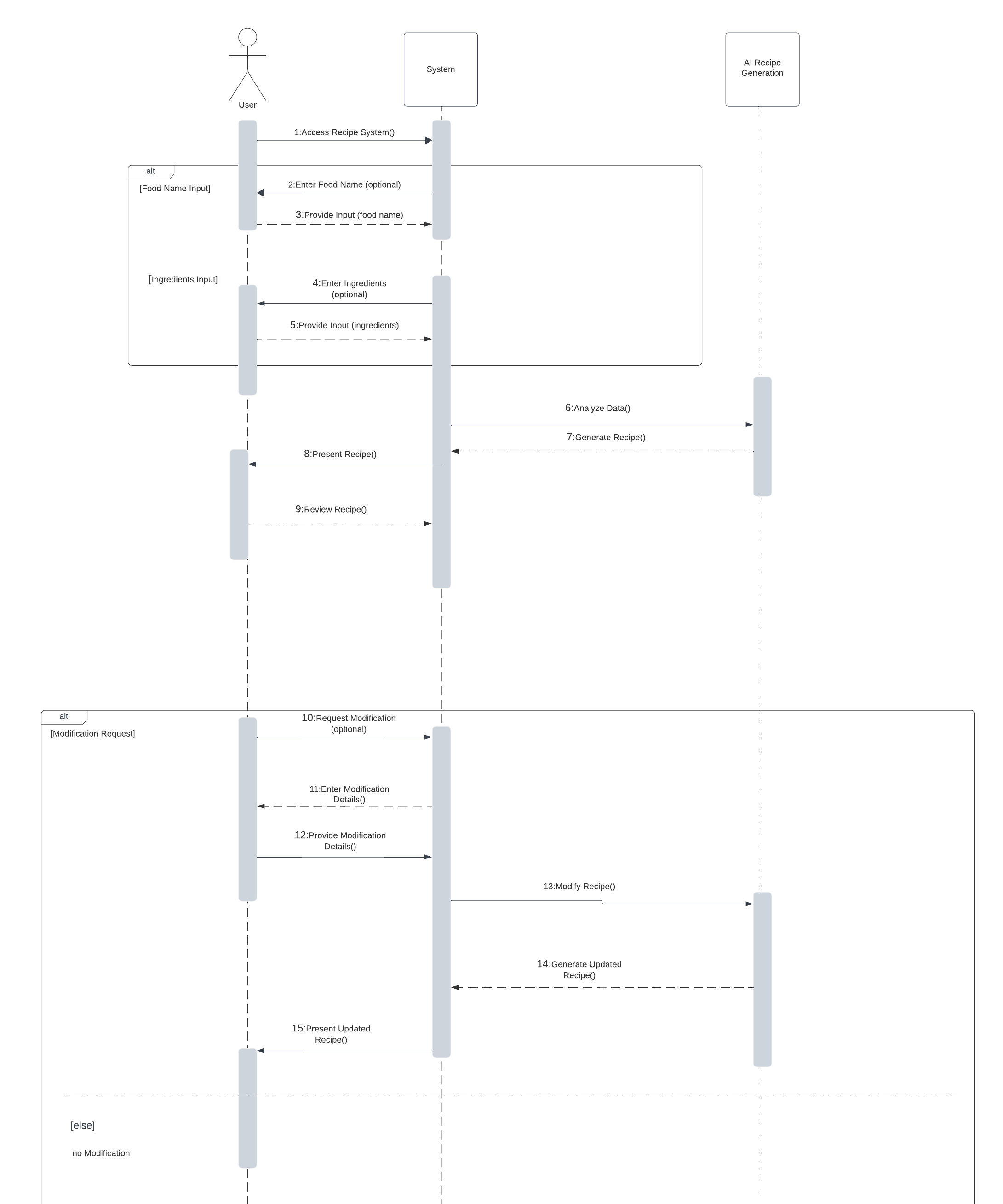
|  |  |
| --- | --- |
| User case name | Generate Recipes |
| Actors | User |
| Main success scenario | 1. User accesses the AI Recipe Generation system. 2. The system prompts the user to input food name or ingredients. 3. AI analyzes the input data. 4. The system generates a recipe based on the analyzed data. 5. User receives the recipe with ingredients, instructions, and nutritional details. |
| Actions | * User accesses the AI Recipe Generation system. * User inputs food name. * AI system processes the data using ML. * The system formulates a recipe with detailed instructions and ingredients. * User receives the generated recipe for the input food. |
| Pre-Condition | * User is logged into the system. * AI Recipe Generation system is accessible and operational. * User provides clear input data food name for recipe generation. |
| Post-Condition | * The generated recipe is displayed to the user for reference. * User can view ingredients, cooking instructions, and nutritional information. |

**6.2 Sequential diagrams**

**6.2.1 Create Diet Plan**



**6.2.2 Generate Recipes**



**6.3 Class diagram**

**A diagram of a system

Description automatically generated**

**Link for team work video**<https://nileuniversity-my.sharepoint.com/:v:/g/personal/s_aboalyazeed2141_nu_edu_eg/EUs950n8dI1DquiJWRkhT88BN7gbnFad5M0s6Wqvtro1kg?nav=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAiOiJPbmVEcml2ZUZvckJ1c2luZXNzIiwicmVmZXJyYWxBcHBQbGF0Zm9ybSI6IldlYiIsInJlZmVycmFsTW9kZSI6InZpZXciLCJyZWZlcnJhbFZpZXciOiJNeUZpbGVzTGlua0NvcHkifX0&e=MSqBRI>

|  |  |
| --- | --- |
|  | |
| **TEST CASE 1:**   |  |  | | --- | --- | |  | | | **Test Case ID: LOGIN-0.1** | **Test Designed by: sara abaalyazeed** | | **Test Priority (Low/Medium/High): high** | **Test Designed date: 22/12/2023** | | **Module Name: login** | **Test Executed by: sara abaalyazeed** | | **Test Title: log-in** | **Test Execution date: 22/12/2023** | | **Description: verify log-in functionality with valid username and password** |  | |  |  | |  |  | | **Pre-conditions:**  User has a valid account | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | | 1 | Open the website. |  | Able to the sign-up page | As expected | Pass | | 2 | Choose log-in |  | Able to see the login page | As expected | Pass | | 3 | Enter username | Username: [test@gmail.com](mailto:test@gmail.com) | Credential can be entered | As expected | Pass | | 4 | Enter password | Password: \*\*\*\*\*\*\* | Credential can be entered | As expected | Pass | | 5 | Click on the log-in button |  | User logged in | User logged in successfully | Pass |  |  |  | | --- | --- | |  | | | **Test Case ID: LOGIN-0.2** | **Test Designed by: sara abaalyazeed** | | **Test Priority (Low/Medium/High): high** | **Test Designed date: 22/12/2023** | | **Module Name: login** | **Test Executed by: sara abaalyazeed** | | **Test Title: log-in** | **Test Execution date: 22/12/2023** | | **Description: verify log-in functionality with invalid username and password** |  | |  |  | |  |  | | **Pre-conditions:**   * User has a valid account | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | | 1 | Open the website. |  | Able to the sign-up page | As expected | Pass | | 2 | Choose log-in |  | Able to see the login page | As expected | Pass | | 3 | Enter username | Username: [test@gmail.com(invalid](mailto:test@gmail.com(invalid)) | Credential can be entered | As expected | Pass | | 4 | Enter password | Password: \*\*\*\*\*\*\*  (invalid) | Credential can be entered | As expected | Pass | | 5 | Click on the log-in button |  | User logged in | unsuccessfully login, system displays the message “invalid” | Fail | | |
| **TEST CASE 2:**  **Test Case ID: AI Recipe Generation-0.1** | **Test Designed by: sara abaalyazeed** |
| **Test Priority (Low/Medium/High): high** | **Test Designed date: 22/12/2023** |
| **Module Name: ai recipe generator** | **Test Executed by: sara abaalyazeed** |
| **Test Title: AI Recipe Generation** | **Test Execution date: 22/12/2023** |
| **Description: verify AI Recipe Generation functionality with valid data** |  |
|  |  |
|  |  |
| **Pre-conditions:**   * **User is on the AIFoodFloow recipe generator page.** * **Ingredient search field is visible and accessible.** * **The AI recipe generator is enabled and functioning.** | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| 1 | Ingredient Input | ingredients ("eggs, milk, flour") | User should be able to write in the ingredient input | As expected, | pass |  |
| 2 | Click the "Search" button |  | The system displays a generated recipe that includes the specified ingredients. | As expected, | Pass |  |

**post-condition:** The generated recipe should accurately incorporate all the entered ingredients without omissions or substitutions.

|  |  |
| --- | --- |
|  | |
| **Test Case ID: AI Recipe Generation-0.2** | **Test Designed by: sara abaalyazeed** |
| **Test Priority (Low/Medium/High): high** | **Test Designed date: 22/12/2023** |
| **Module Name: ai recipe generator** | **Test Executed by: sara abaalyazeed** |
| **Test Title: AI Recipe Generation** | **Test Execution date: 22/12/2023** |
| **Description: verify AI Recipe Generation functionality with invalid data** |  |
|  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Ingredient Input | Enter a number (e.g., "123") into the ingredient search field. | User should be able to write in the ingredient input | As expected, | Pass |  |
| 2 | Click the "Search" button |  | The system displays a generated recipe that includes the specified ingredients. | The system displays a “sorry, we didn’t find any meal.” | Fail |  |

**TEST CASE 3:**

|  |  |
| --- | --- |
|  | |
| **Test Case ID: buy items 0.1** | **Test Designed by: sara abaalyazeed** |
| **Test Priority (Low/Medium/High): high** | **Test Designed date: 22/12/2023** |
| **Module Name: login** | **Test Executed by: sara abaalyazeed** |
| **Test Title: buy items** | **Test Execution date: 22/12/2023** |
| **Description: Verify user can successfully search for an item, add it to cart, and view the cart after login** |  |
|  |  |
|  |  |
| **Pre-conditions:**  User is logged in  There are items available for purchase in the inventory | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** |
| 1 | Search for an item | Enter a search term (e.g., "apples") in the search bar | Able to Search for an item | As expected | Pass |
| 2 | Click the "Search" button |  | Search results page displays a list of items matching the search term | As expected | Pass |
| 3 | Add item to cart | Click the "Add to Cart" button next to the desired item | Selected item is added to the cart, and a confirmation message is displayed | As expected | Pass |
| 4 | View cart | Click the "Cart" button | Cart page displays the added item, its quantity, and total price | As expected | Pass |

**Post-conditions:**

The cart contains the added item

**TEST CASE 4:**

|  |  |
| --- | --- |
|  | |
| **Test Case ID: Wishlist add or remove** | **Test Designed by: sara abaalyazeed** |
| **Test Priority (Low/Medium/High): high** | **Test Designed date: 22/12/2023** |
| **Module Name: login** | **Test Executed by: sara abaalyazeed** |
| **Test Title: Wishlist add or remove** | **Test Execution date: 22/12/2023** |
| **Description: Verify user can view and remove items from their wishlist** |  |
|  |  |
|  |  |
| **Pre-conditions:**  User is logged in.  User has at least one item in their Wishlist  Wishlist page is accessible | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** |
| 1 | Access Wishlist | Click on the "Wishlist" button | Wishlist page loads successfully | As expected | Pass |
| 2 | View Wishlist items |  | Wishlist displays the correct items with accurate details | As expected | Pass |
| 3 | Remove an item | Click the "Remove" button next to the item to be removed | Removed item disappears from the Wishlist | As expected | Pass |
| 4 | View cart | Click the "Cart" button | Total price in the Wishlist is updated accordingly | As expected | Pass |

**Post-conditions:**

Wishlist no longer contains the removed item.

Total price in the Wishlist is adjusted.

**TEST CASE 5:**

|  |  |
| --- | --- |
|  | |
| **Test Case ID: delivery details0.1** | **Test Designed by: sara abaalyazeed** |
| **Test Priority (Low/Medium/High): high** | **Test Designed date: 22/12/2023** |
| **Module Name: login** | **Test Executed by: sara abaalyazeed** |
| **Test Title: delivery details** | **Test Execution date: 22/12/2023** |
| **Description: Verify user can successfully fill in delivery details and submit the form** |  |
|  |  |
|  |  |
| **Pre-conditions:**  User is logged in.  User has items in their cart.  User has clicked the "Buy Now" button and proceeded to checkout.  Delivery details form is displayed. | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** |
| 1 | Fill in delivery details | Name: (sara)  Address: (haram)  Email-ID: (sara@gmail.com)  Locality/Apartment: (2)  Pin code: (1929)  Contact No.: (01020202)  Date: (22/1/2024)  Time Slot: (1-3 pm) | User is able to write the details | As expected | Pass |
| 2 | Click the "SUBMIT" button |  | Form accepts valid input for all fields | As expected | Pass |
| 3 | Upon clicking "SUBMIT" |  | Form data is successfully submitted.  A confirmation message indicating successful submission is displayed | As expected | Pass |

**Post-conditions:**

Delivery details are stored in the system.

User's order is ready for processing.

**TEST CASE 5:**

|  |  |
| --- | --- |
|  | |
| **Test Case ID: delivery details0.2** | **Test Designed by: sara abaalyazeed** |
| **Test Priority (Low/Medium/High): high** | **Test Designed date: 22/12/2023** |
| **Module Name: login** | **Test Executed by: sara abaalyazeed** |
| **Test Title: delivery details** | **Test Execution date: 22/12/2023** |
| **Description: Verify user can fill in delivery with missing data** |  |
|  |  |
|  |  |
| **Pre-conditions:**  User is logged in.  Users have items in their cart.  User has clicked the "Buy Now" button and proceeded to checkout.  Delivery details form is displayed. | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** |
| 1 | Fill in delivery details | Name: (sara)  Address: ()  Email-ID: (sara@gmail.com)  Locality/Apartment: ()  Pin code: (1929)  Contact No.: (01020202)  Date: ()  Time Slot: (1-3 pm) | User can write the details | As expected, | Pass |
| 2 | Click the "SUBMIT" button |  | Form accepts valid input for all fields | Form displays clear error messages for missing data. | Fail |

**Link GitHub:** [**https://github.com/Michael-2003/AIFoodFlow**](https://github.com/Michael-2003/AIFoodFlow)