Noushin Binte Islam Ananna

Independent University, Bangladesh

CSE307, Autumn2024, Sec: 03

Web Application & Network

Project Name: Idea of Organic Food Service Website Application

Software Requirements Specification (SRS) of Organic Food Service Website

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to outline the functional and non-functional requirements for the "Organic Food Service" website. The software aims to provide a seamless online shopping experience for customers interested in purchasing organic food products. It includes features like product browsing, cart management, checkout, online payment, and customer support. This document will also serve as a reference for developers, testers, and stakeholders to ensure the software meets the defined objectives.

1.2 Document Conventions

This document follows specific conventions. It uses abbreviations like "SRS" for Software Requirements Specification, "UI" for User Interface, and "DB" for Database. For formatting, bold text is used for headings and important terms, bulleted lists are applied for itemized details, and code blocks are used for technical references.

1.3 Intended Audience and Reading Suggestions

The intended audience includes developers who will use this document to understand the functional and non-functional requirements, testers who will create test cases based on these requirements, project managers who will track project scope and deliverables, and stakeholders who will verify that the software aligns with business objectives. It is recommended to start with Section 1 to understand the project scope and purpose, refer to Section 3 for detailed functional requirements, and consult Section 4 for technical and system requirements.

1.4 Project Scope

The "Organic Food Service" website aims to simplify the process of purchasing organic food online while educating customers about healthy diets and sustainable practices. Its key objectives are to provide a platform for browsing and purchasing organic food products, enable seamless cart and checkout functionalities with payment options like cash on delivery and bKash, offer customer support through contact forms with email and phone number fields, educate users through blogs on organic food, healthy diets, and farmer-related topics, and build trust by sharing company details, licenses, and social media links on the "About Us" page.

1.5 References

This document references HTML and CSS specifications for web development, JavaScript and PHP documentation for functionality implementation, payment gateway APIs such as bKash documentation, and guidelines for creating accessible and user-friendly interfaces.

Some links: https://thrivemarket.com/
https://www.misfitsmarket.com/hp14
https://www.farmfreshtoyou.com/
https://ghorerbazar.com/

2. Overall Description

2.1 Product Perspective

The "Organic Food Service" platform is a standalone web-based application designed to serve as an e-commerce solution for organic food products. It is not part of a larger system but integrates with external APIs, such as the bKash payment gateway, for financial transactions. The system includes a frontend for user interactions and a secure backend for managing inventory, transactions, and user data. The platform is intended to function independently, providing a seamless user experience for both customers and administrators.

2.2 Product Features

The website includes several features tailored to improve the shopping experience and enhance functionality:

1. **Home Page**: Highlights promotions and includes a "Shop Now" button for navigation.
2. **Shop Page**: Displays products with details like name, price, and an "Add to Cart" button, alongside filtering options.
3. **Cart Page**: Shows selected products, quantities, and prices, with options to update or remove items and calculates totals dynamically.
4. **Checkout Page**: Supports payments via cash on delivery and the bKash payment gateway.
5. **Contact Us Page**: Allows users to submit queries using a form with email and phone number fields.
6. **Blog Page**: Offers educational content on topics such as organic farming and healthy diets.
7. **About Us Page**: Provides company details, licensing information, and social media links.

2.3 User Classes and Characteristics

The software caters to two main types of users:

1. **Customers**:
    - Primary users who browse and purchase organic food products.
    - Require an easy-to-navigate, visually appealing interface for shopping.
2. **Administrators**:
    - Manage inventory, update blogs, and respond to customer inquiries.
    - Require a secure backend with functionalities to add, edit, or remove content efficiently.

2.4 Operating Environment

The "Organic Food Service" platform operates in the following environment:

- **Frontend**: Built with HTML, CSS, and JavaScript to ensure responsiveness and interactivity.
- **Backend**: Developed using PHP for dynamic content and data handling.
- **Payment Gateway**: Integrated with bKash API for online payment processing.
- **Browsers**: Compatible with Google Chrome, Mozilla Firefox, Safari, and other modern browsers.
- **Devices**: Optimized for desktops, tablets, and mobile devices.

2.5 Design and Implementation Constraints

Several constraints influence the design and implementation of the platform. The platform must adhere to web standards to ensure accessibility and usability. Integration with the bKash payment gateway limits the payment options to those supported by the API. The website must be mobile-friendly and responsive to cater to users across devices. Additionally, all functionalities must comply with data security and privacy regulations.

2.6 User Documentation

The following documentation will be provided to assist users. User manuals will include step-by-step guides for navigating and using the platform, targeted at customers. An administrator guide will provide detailed instructions for managing the backend, including adding or updating content and resolving customer inquiries. A FAQs section will be available on the website to address common user questions about the platform and its features.

2.7 Assumptions and Dependencies

The platform assumes that users will access the website via a stable internet connection. It is dependent on the bKash payment gateway remaining operational and available for integration. Hosting services must provide sufficient resources to handle traffic and data storage. The platform also relies on the latest versions of web browsers and PHP to ensure compatibility. Lastly, it is assumed that customers and administrators will have basic technical knowledge to interact with the platform effectively.

3. System Features

3.1 User Login System

Description and Priority: The User Login System allows customers and administrators to access their respective accounts securely. This feature is of high priority as it ensures data privacy and provides personalized experiences. Customers can view their order history, and admins can manage the backend.

Stimulus/Response Sequences:

1. **Stimulus**: A user enters their credentials (email and password) on the login page.
   Response: The system verifies the credentials and grants access to the corresponding dashboard.
2. **Stimulus**: A user enters incorrect credentials.
   Response: The system displays an error message and prompts the user to retry or reset their password.

Functional Requirements: Users must be able to log in using their registered email and password. The platform must provide a "Forgot Password" option to allow users to reset their credentials. Administrators should have access to a secure admin panel upon successful login.

3.2 Product Browsing and Filtering

Description and Priority: This feature allows users to browse and filter products by categories or search keywords. It is essential for providing a seamless shopping experience and is of high priority.

Stimulus/Response Sequences:

1. **Stimulus**: A user navigates to the "Shop" page and selects a category filter.
   Response: The system displays products that match the selected category.
2. **Stimulus**: A user enters a search term in the search bar.
   Response: The system displays relevant products based on the keyword.

Functional Requirements: The platform must display all available products on the "Shop" page. Users should be able to filter products by categories, such as fruits, vegetables, and grains. Additionally, the platform must enable users to search for products using a search bar.

3.3 Shopping Cart

Description and Priority: The Shopping Cart feature enables users to select and manage products they intend to purchase. It is crucial for the platform's functionality and has the highest priority.

Stimulus/Response Sequences:

1. **Stimulus**: A user adds a product to the cart.
   Response: The system updates the cart with the selected item and displays the updated cart contents.
2. **Stimulus**: A user adjusts the quantity or removes an item from the cart.
   Response: The system recalculates the subtotal, tax, and total cost accordingly.

Functional Requirements: Users should be able to add or remove products from the cart. The cart must dynamically update quantities and calculate the total cost. Additionally, the contents of the cart must be stored until the user proceeds to checkout.

3.4 Checkout and Payment System

Description and Priority: This feature allows users to complete their purchases by providing payment options. It is vital for order processing and is of the highest priority.

Stimulus/Response Sequences:

1. **Stimulus**: A user selects "Checkout" and chooses a payment method (cash on delivery or bKash).
   Response: The system processes the order and sends a confirmation email.
2. **Stimulus**: A user cancels the checkout process.
   Response: The system retains the cart contents for future use.

Functional Requirements: Users must be able to select a preferred payment method. The system must securely process payments via the bKash API. Additionally, order confirmation emails must be sent to the user upon successful payment.

3.5 Contact Us System

Description and Priority: The Contact Us feature allows users to submit queries or feedback via a form. This is a medium-priority feature for ensuring customer support.

Stimulus/Response Sequences:

1. **Stimulus**: A user fills out the contact form with their name, email, phone number, and message.
   Response: The system sends the message to the administrator and displays a success notification.

Functional Requirements: The form must validate user input to ensure that all required fields are completed. Messages must be stored in the database for admin review. Additionally, the system must display a confirmation of successful submission to the user.

3.6 Blog Management System

Description and Priority: This feature provides educational content and updates for customers. It is a medium-priority feature designed to engage users and enhance brand credibility.

Stimulus/Response Sequences:

2. **Stimulus**: A user visits the "Blog" page and clicks on an article title. **Response**: The system displays the full blog post with its content.
3. **Stimulus**: An admin adds a new blog post from the backend. **Response**: The system updates the blog list visible to users.

Functional Requirements: The platform must display a list of blogs with titles, images, and short descriptions. Admins should be able to add, edit, or delete blog posts via the backend. Users must be able to navigate and read individual blog posts.

3.7 About Us Page

Description and Priority: The "About Us" page provides information about the company, including licensing and social media links. It is a low-priority feature aimed at building trust.

Stimulus/Response Sequences:

1. **Stimulus**: A user clicks on the "About Us" page link.
   Response: The system displays detailed company information, licenses, and social media links.

Functional Requirements: The platform should display company information, including its history and mission. It must provide links to social media pages and include licensing and certification details.

4. External Interface Requirements

4.1 User Interfaces
The system will provide a graphical user interface (GUI) for customers and administrators. The interface will include the following elements:

- **Navigation bar**: Includes links to Home, Shop, Cart, Blog, Contact Us, and About Us pages.
- **Shop page**: Displays products with images, names, prices, and an "Add to Cart" button.
- **Cart page**: Shows selected items, allows quantity updates, and calculates costs dynamically.
- **Admin panel**: Allows administrators to manage products, blogs, and customer inquiries.

4.2 Hardware Interfaces
The platform is designed to operate on standard consumer devices:

- **User Devices**: Desktop computers, laptops, tablets, and smartphones.
- **Server Requirements**: A web server with at least 4 GB RAM, 100 GB storage, and PHP support.

4.3 Software Interfaces
The system will interact with the following software components:

- **Payment Gateway**: Integration with the bKash API for online payments.
- **Web Browsers**: Compatibility with Google Chrome, Mozilla Firefox, and Safari.

4.4 Communications Interfaces
The system will use standard internet protocols to facilitate communication:

- **HTTP/HTTPS**: For secure data transfer between the client and server.
- **SMTP**: For sending order confirmation and password reset emails.
- **API Calls**: For interacting with the bKash payment gateway.

5. Other Nonfunctional Requirements

5.1 Performance Requirements
The system must meet specific performance criteria, ensuring that all web pages' load within 3 seconds under normal load conditions. It should support up to 100 concurrent users without experiencing performance degradation.

5.2 Safety Requirements
To ensure safety and fault tolerance, the system will handle failed payment transactions gracefully by notifying users of the failure and providing instructions for retrying. Regular database backups will be performed to prevent data loss.

5.3 Security Requirements
Security measures will include encrypting sensitive data, such as payment information, using SSL/TLS. Role-based access control will be implemented to secure admin functionalities. Regular vulnerability assessments will be conducted to maintain data integrity and privacy.

5.4 Software Quality Attributes

The system will adhere to several software quality attributes:

- **Maintainability:** The codebase will be modular, well-documented, and structured to accommodate future updates and feature additions.
- **Portability:** The platform will work seamlessly across different web browsers and devices without compatibility issues.
- **Usability:** The user interface will be intuitive, accessible, and responsive, ensuring a seamless experience for all users, including those with disabilities.

6. Other Requirements

Any additional requirements that do not fit into the categories above will be documented as they arise during the project development lifecycle.

Appendices

A. Glossary

- **SRS:** Software Requirements Specification.
- **UI:** User Interface.
- **DB:** Database.
- **API:** Application Programming Interface.
- **SSL/TLS:** Secure Sockets Layer/Transport Layer Security.
- **SMTP:** Simple Mail Transfer Protocol.

B. Analysis Models

1. Use Case Diagram

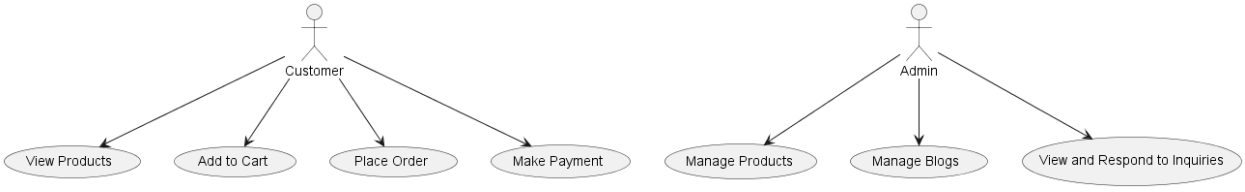

Fig: Use Case Diagram of Organic Food Service Website

The use case diagram is to illustrate interactions between users and the system to represents the interactions between the users (Customer, Admin) and the system.

Elements:

- **Actors**:
  - Customer
  - Admin

- **Use Cases**:
  - View Products
  - Add to Cart
  - Place Order
  - Make Payment
  - Manage Products (Admin only)
  - Manage Blogs (Admin only)
  - View and Respond to Inquiries (Admin only)

Diagram Description:

The Customer actor is linked to use cases such as "View Products," "Add to Cart," "Place Order," and "Make Payment." The Admin actor is connected to use cases like "Manage Products," "Manage Blogs," and "View and Respond to Inquiries."

2. Sequence Diagram

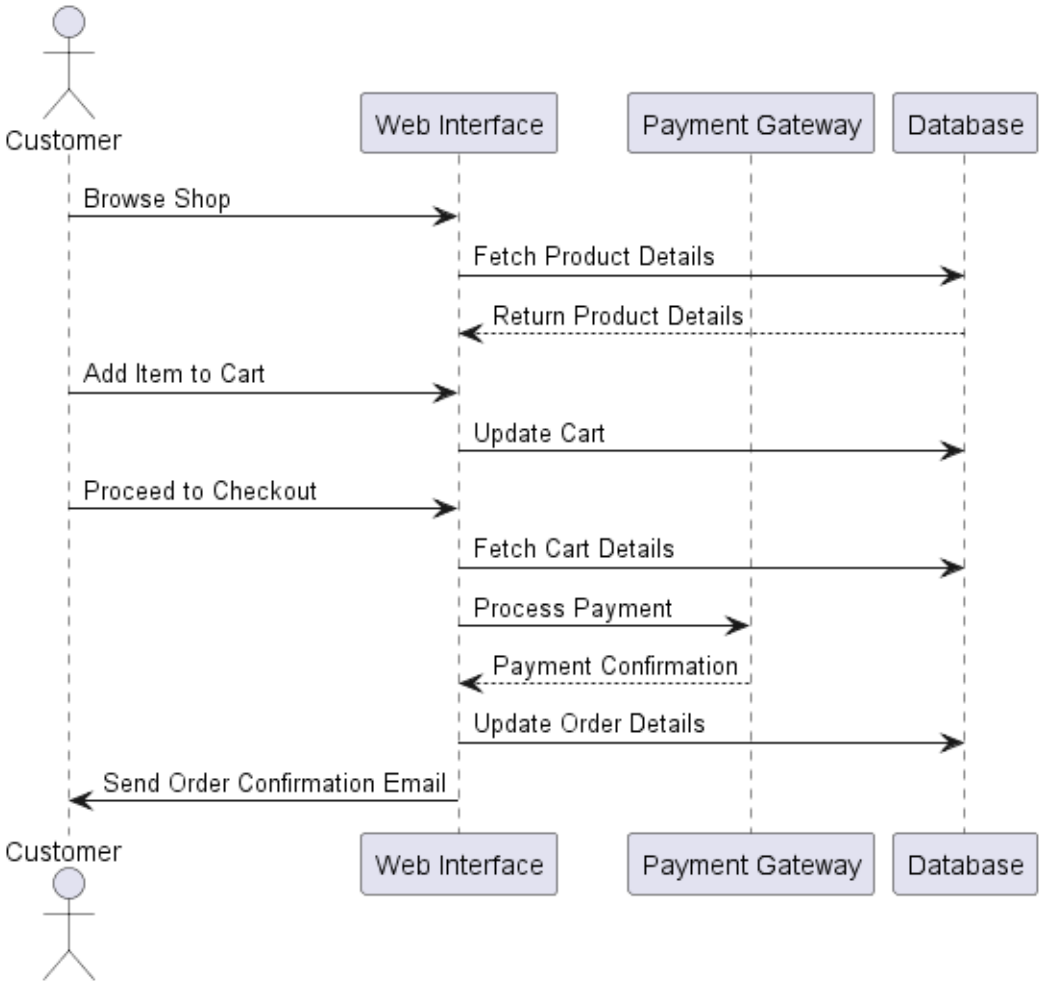

Fig: Sequence Diagram of Organic Food Service Website

The sequence diagram is to show the flow of operations in details that the flow of operations for a customer placing an order.

Steps:

1. The Customer browses the Shop page.
2. The Customer adds items to the Cart.
3. The Customer proceeds to Checkout.
4. The system calculates the total cost.
5. The Customer makes a payment via the payment gateway.
6. The system confirms the payment and sends an order confirmation email.

Diagram Description:

The objects involved include the Customer, Web Interface, Payment Gateway, and Database. Messages exchanged include item selection, cart updates, payment processing, and email notifications.

3. Class Diagram

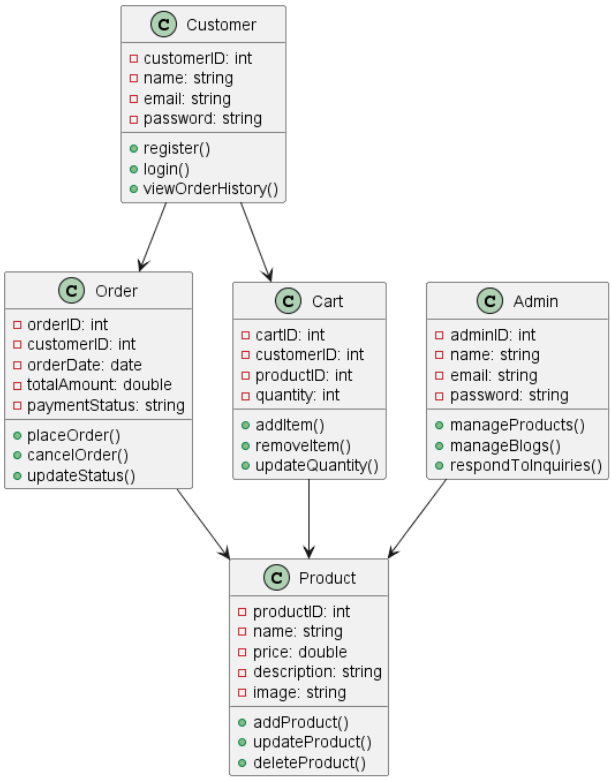

Fig: Class Diagram of Organic Food Service Website

The class is to represent the structure of the database and application in the diagram outlines of the database and application structure.

Classes:

- **Customer**:
  - Attributes: customerID, name, email, password
  - Methods: register(), login(), viewOrderHistory()
- **Product**:
  - Attributes: productID, name, price, description, image
  - Methods: addProduct(), updateProduct(), deleteProduct()
- **Order**:
  - Attributes: orderID, customerID, orderDate, totalAmount, paymentStatus
  - Methods: placeOrder(), cancelOrder(), updateStatus()
- **Cart**:
  - Attributes: cartID, customerID, productID, quantity
  - Methods: addItem(), removeItem(), updateQuantity()
- **Admin**:
  - Attributes: adminID, name, email, password
  - Methods: manageProducts(), manageBlogs(), respondToInquiries()

4. Deployment Diagram

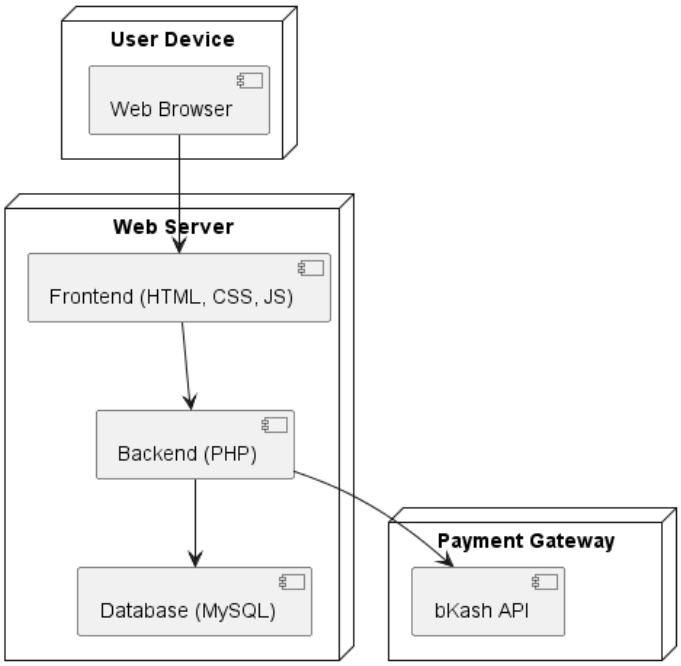

Fig: Deployment Diagram of Organic Food Service Website

The deployment diagram represents the physical architecture of the system. It shows how the components of the system are deployed on different nodes. In this case, it highlights the interaction between the user's device (via a web browser), the web server hosting the frontend (HTML, CSS, JavaScript), the backend (PHP), the MySQL database, and the external payment gateway (bKash API).

5. Rich Picture

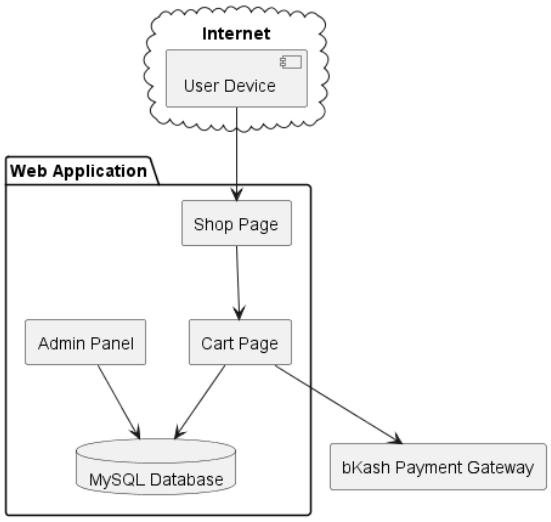

Fig: Rich Picture of Organic Food Service Website

The rich picture provides a conceptual overview of the system's flow and structure. It depicts key elements such as the user device, core application components (Shop Page, Cart Page, Admin Panel), the MySQL database, and the bKash payment gateway. Arrows indicate the flow of interactions between these components, giving a high-level understanding of the system's operation.

6. Flow Chart

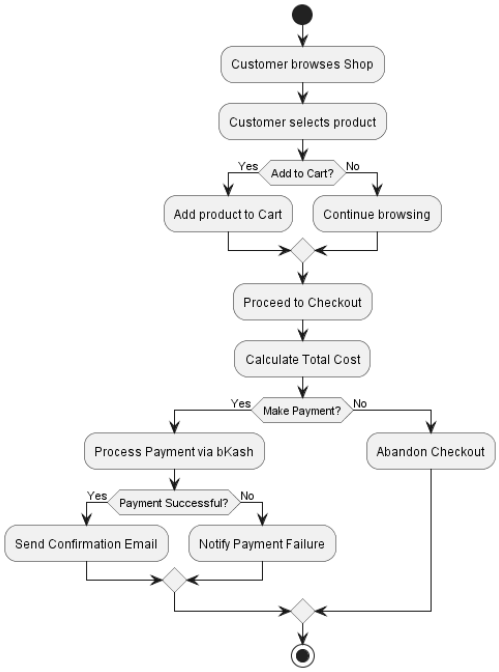

Fig: Flow Chart of Organic Food Service Website

The flowchart illustrates the logical sequence of operations for a customer placing an order. It starts with browsing products, adding items to the cart, proceeding to checkout, calculating the total cost, making a payment via the bKash gateway, and concludes with either a confirmation email (on success) or a failure notification (on failure). This diagram visually represents the decision-making process and operational flow within the system.

C. Issues List

Open issues related to the requirements will be documented and tracked here, including:

- Payment gateway integration challenges.
- Mobile responsiveness testing.
- Scalability concerns for future expansion.