**Software Requirement Specification**

**NAME OF SYSTEM : Bakery**

**DATE : 4/7/2024**

Version 1.0.0

**Logo :**

**Presented To: Dr.Mohamed Ramdan**

**Submitted By: Technical team**

Group names :2ABK

**TABLE OF CONTENTS**

**REVISION HISTORY**

| **Date** | **Author** | **Distributed to** | **Version** | **Description** |
| --- | --- | --- | --- | --- |
| 4/7/2024 | doc Team | Relevant stakeholders | 1.0 | first version of  documentation  initializing  requirements |

**1 Introduction**

The project aims to develop a bakery website with a selling system for online orders. The Software Development Plan outlines the project scope, objectives, schedule, resources, and risk management. Its purpose is to guide the development team and ensure alignment with stakeholder expectations for successful project execution.

**1.2 Scope**

### Personas / Actors

#### 1. Customer

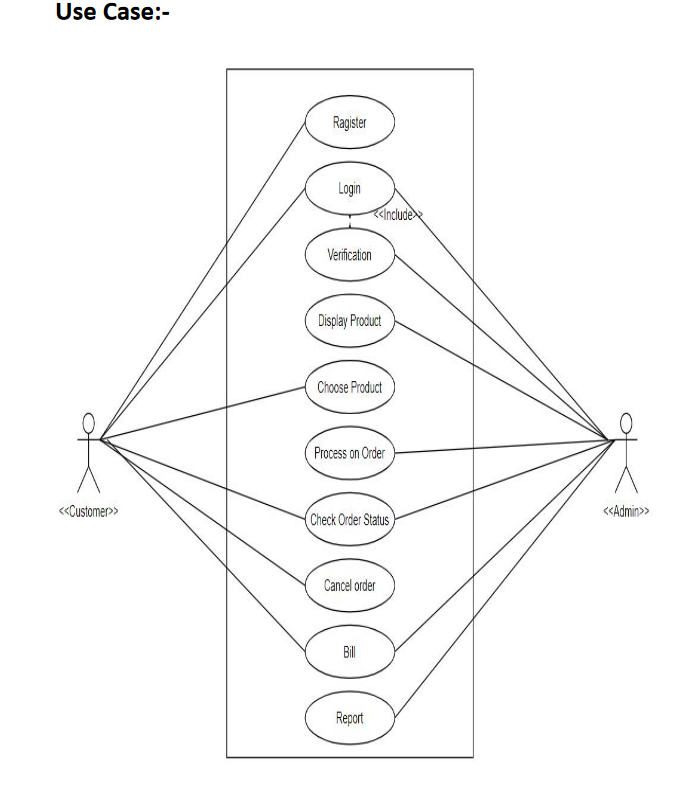
* **Description**: Represents individuals who visit the bakery website to browse and purchase bakery products.
* **Needs/Goals**: Convenient browsing, easy checkout process, secure payment options, access to product information and reviews.
* **Actions**: Browsing products, adding items to cart, proceeding to checkout, making payments, leaving reviews and feedback.

#### 2. Bakery Staff/Administrator

* **Description**: Represents individuals responsible for managing the bakery's online presence, including inventory, orders, and customer interactions.
* **Needs/Goals**: Efficient management of inventory, processing orders, monitoring sales, updating product information, managing customer inquiries.
* **Actions**: Adding/editing/deleting products, managing inventory levels, processing orders, responding to customer inquiries, analyzing sales data.

#### 3. Payment Processor (Third-Party)

* **Description**: Represents the external service used for processing online payments securely.
* **Needs/Goals**: Securely process payments from customers on behalf of the bakery website.
* **Actions**: Handling payment transactions securely, providing confirmation of successful payments to the website.



**2 Related Documents**

Requirements Traceability matrix reference :

| **Requirement ID** | **Description** | **Design Document** | **Implementation** |
| --- | --- | --- | --- |
| REQ-001 | The system shall allow users to create an account. | DD-001 | Implemented |
| REQ-002 | Users shall be able to browse bakery products. | DD-002 | Implemented |
| REQ-003 | The system must support online payments. | DD-003 | In Progress |
| REQ-004 | Customers should receive email notifications for order confirmations. | DD-004 | Not Implemented |
| REQ-005 | Admins should be able to manage inventory. | DD-005 | Implemented |
| REQ-006 | The system should generate sales reports. | DD-006 | In Progress |

**3 Requirements**

**Customer:**

* As a customer, I want to browse through various bakery products.
* As a customer, I want to create an account to facilitate easier checkout for future orders.
* As a customer, I want to be able to add items to my shopping cart.
* As a customer, I want to view detailed information about each bakery product, including ingredients, price, and availability.
* As a customer, I want to be able to place an order and choose a convenient delivery or pickup option.

**Administrator:**

* As an administrator, I want to manage the bakery's product inventory, including adding new products, updating existing ones, and removing discontinued items.
* As an administrator, I want to view and process customer orders, including confirming orders, updating order statuses, and generating invoices.
* As an administrator, I want to manage user accounts, including creating new accounts, updating user information, and deactivating accounts if necessary.

**3.1 Punch List**

The following list outlines the remaining items that need to be addressed:

* Implementing a search functionality to allow customers to easily find specific bakery products.
* Enhancing the user interface design for a more visually appealing and user-friendly experience.
* Integrating a secure payment gateway to facilitate online transactions.
* Implementing user authentication and authorization mechanisms to ensure the security of customer and administrator accounts.
* Optimizing website performance to ensure fast loading times and smooth navigation.

**3.2 Use Case 1**

**3.2.1 Use Case Description**

**Description: This section provides a detailed description of a specific use case within the bakery website system.**

**Use Case: Browse Bakery Products**

**Name: BrowseProducts**

**Participating Actor(s): Customer**

**Entry: The customer accesses the bakery website.**

**Exit: The customer finds the desired bakery product or completes the browsing session.**

**Flow of Events:**

1. **The customer opens the bakery website in their web browser.**
2. **The system displays the homepage, showcasing featured bakery products and promotions.**
3. **The customer navigates to the product catalog section.**
4. **The system retrieves and displays a list of available bakery products, including images, names, prices, and brief descriptions.**
5. **The customer can filter and sort the product list based on criteria such as category, price range, and popularity.**
6. **The customer selects a specific product to view more detailed information.**
7. **The system presents the product details page, showing additional information such as ingredients, nutritional facts, customer reviews, and related products.**
8. **The customer can add the selected product to their shopping cart or continue browsing.**
9. **If the customer adds the product to their cart, the system updates the cart summary to reflect the added item.**
10. **The customer may choose to proceed to checkout or continue browsing for more products.**
11. **If the customer decides to check out, the system redirects them to the checkout process.**

**Special Requirements:**

* **The system should load product images and information quickly to provide a seamless browsing experience.**
* **The product catalog should be easily navigable, allowing customers to find desired items efficiently.**

**3.2.2 User Interface Requirements**

* **The homepage should feature visually appealing images and clear navigation options to guide users to the product catalog.**
* **The product catalog page should display product images, names, prices, and brief descriptions in a grid or list format.**
* **Each product should have a clickable element to view more details.**
* **The checkout button should be prominently displayed on relevant pages to encourage conversions.**

**3.2.3 Other Details Requirements**

* **Non-Happy Path Flows:**
  + **If the customer encounters a slow-loading page, the system should display a loading indicator or error message to indicate the delay and reassure the customer.**
  + **If a selected product is out of stock or unavailable, the system should notify the customer and suggest similar alternatives.**
  + **If the customer encounters any errors during the browsing process, such as page not found or server errors, the system should provide a clear error message and guide the customer to navigate back to a functional page.**

**3.2 Use Cases**

**3.2.1 Use Case Description**

**Description: This section provides a detailed description of a specific use case within the bakery website system.**

**Use Case: Browse Bakery Products**

**Name: BrowseProducts**

**Participating Actor(s): Customer**

**Entry: The customer accesses the bakery website.**

**Exit: The customer finds the desired bakery product or completes the browsing session.**

**Flow of Events:**

1. **The customer opens the bakery website in their web browser.**
2. **The system displays the homepage, showcasing featured bakery products and promotions.**
3. **The customer navigates to the product catalog section.**
4. **The system retrieves and displays a list of available bakery products, including images, names, prices, and brief descriptions.**
5. **The customer can filter and sort the product list based on criteria such as category, price range, and popularity.**
6. **The customer selects a specific product to view more detailed information.**
7. **The system presents the product details page, showing additional information such as ingredients, nutritional facts, customer reviews, and related products.**
8. **The customer can add the selected product to their shopping cart or continue browsing.**
9. **If the customer adds the product to their cart, the system updates the cart summary to reflect the added item.**
10. **The customer may choose to proceed to checkout or continue browsing for more products.**
11. **If the customer decides to check out, the system redirects them to the checkout process.**

**Special Requirements:**

* **The system should load product images and information quickly to provide a seamless browsing experience.**
* **The product catalog should be easily navigable, allowing customers to find desired items efficiently.**

**Use Case: Manage Product Inventory**

**Name: ManageInventory**

**Participating Actor(s): Administrator**

**Entry: The administrator logs into the bakery website admin dashboard.**

**Exit: The administrator successfully manages the bakery's product inventory.**

**Flow of Events:**

1. **The administrator accesses the admin dashboard after logging in.**
2. **The system presents various administrative options, including inventory management.**
3. **The administrator selects the inventory management option.**
4. **The system displays a list of current bakery products along with their details such as name, quantity, and status.**
5. **The administrator can add new products to the inventory by providing necessary information such as name, description, price, and quantity.**
6. **The administrator can update existing product information, including price, quantity, and status (e.g., available, out of stock).**
7. **The administrator can remove discontinued or outdated products from the inventory.**
8. **The system updates the inventory database with any changes made by the administrator.**

**Special Requirements:**

* **The admin dashboard should provide a user-friendly interface for easy navigation and efficient inventory management.**
* **Changes to the inventory should be reflected immediately in the system to ensure accurate stock levels.**

**Use Case: Process Customer Orders**

**Name: ProcessOrders**

**Participating Actor(s): Administrator**

**Entry: The administrator accesses the admin dashboard.**

**Exit: The administrator successfully processes customer orders.**

**Flow of Events:**

1. **The administrator logs into the admin dashboard.**
2. **The system presents options for order management.**
3. **The administrator selects the order management option.**
4. **The system displays a list of pending customer orders, including order details such as customer name, order items, and total amount.**
5. **The administrator reviews each order and verifies the availability of products in the inventory.**
6. **If all products are available, the administrator confirms the order and proceeds to generate an invoice.**
7. **The system updates the order status to "confirmed" and generates an invoice for the customer.**
8. **The administrator notifies the customer of the order confirmation and provides details regarding delivery or pickup options.**

**Special Requirements:**

* **The order management system should provide real-time updates on order statuses to ensure timely processing.**
* **Invoices generated should be accurate and include relevant details such as order items, prices, taxes, and total amount.**

**3.2.2 User Interface Requirements**

* **The user interface should be intuitive and easy to navigate for both customers and administrators.**
* **Clear and visually appealing design elements should enhance the user experience.**
* **Relevant actions and options should be easily accessible from their respective pages.**

**3.2.3 Other Details Requirements**

* **Non-Happy Path Flows:**
  + **Error handling mechanisms should be in place to handle scenarios such as failed login attempts or database errors.**
  + **If an administrator tries to process an order with insufficient stock, the system should notify them and suggest appropriate actions such as contacting the customer or updating the inventory.**
  + **In case of network issues or system downtime, the system should display a friendly error message and provide instructions for users to try again later.**

**4 Non-Functional Requirements**

**4.1 Other Systems**

**Description:** The bakery website must integrate with other systems to enhance functionality and streamline operations.

* **Measurable Requirement:** The integration must be completed within [specified time frame], ensuring seamless communication between the bakery website and external systems such as payment gateways, inventory management systems, and email notification services.

**4.2 Security**

**Description:** The bakery website must prioritize security measures to protect user data and ensure safe transactions.

* **Measurable Requirement:** The website must utilize SSL (Secure Sockets Layer) certification to encrypt data transmissions, providing a secure browsing experience for users.

**4.3 Performance**

**Description:** The bakery website must deliver consistent performance to meet user expectations and maintain satisfaction.

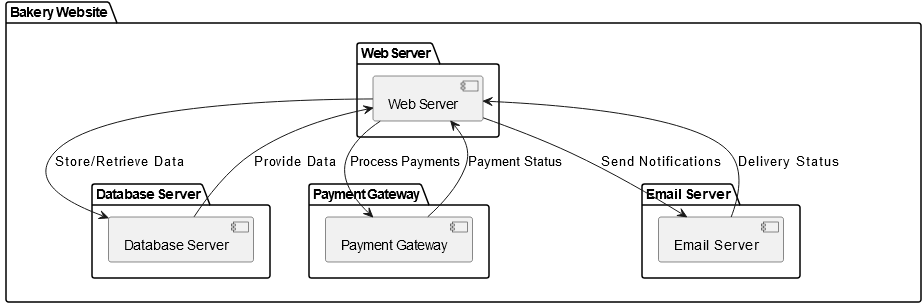
* **Measurable Requirement:**
  + **Uptime:** The website must maintain at least 99% uptime, ensuring minimal downtime for users to access services.
  + **Response Time:** The average page load time should not exceed [specified time], ensuring a responsive browsing experience for users across different devices and network conditions.

**4.4 Maintainability**

**Description:** The bakery website must be designed and developed in a way that facilitates easy maintenance and updates.

* **Measurable Requirement:**
  + **Regular Updates:** The website must undergo regular maintenance updates to address security vulnerabilities, optimize performance, and introduce new features or improvements.
  + **Documentation:** Comprehensive documentation outlining the website's architecture, codebase, and operational procedures must be maintained to aid future development and troubleshooting efforts.

**5 Architecture**



**Signature Page**

AGREED TO:

**Customer?**

**Signature**

**Name:**

**Title:**

**Date:**

AGREED TO:

**Team members**

**Signature**

**Name:**

**Title:**

**Date:**