● Software Requirements Specification (SRS)

Project Name: El Hamour Market
Prepared By: Mohamed Bekheet, Bassam Mohamed, Ahmed Elrashidy, Seif Aboshanab, Mohamed Mahmoud
Date: 8/11/2024

---

1. Introduction

1.1 Purpose

The purpose of this document is to outline the functional and non-functional requirements for the El-Hamour Market project. This document provides a detailed description of the system's functionality and constraints, serving as a guide for developers, testers, and other stakeholders involved in creating the online marketplace.

1.2 Scope

The El-Hamour Market system is an online marketplace designed to provide users with a seamless shopping experience for a variety of products, including food, vegetables, fruits, and home tools. The system will offer functionality for:

- Secure, multi-factor authenticated user accounts and data protection.
- A user-friendly interface optimized for ease of navigation across all devices.
- Advanced search capabilities, including a smart search with personalized recommendations and filters.
- Customer evaluation of products, writing product reviews and suggestions for product improvement

1.3 Definitions, Acronyms, and Abbreviations

- **API:** Application Programming Interface
- **UAT:** User Acceptance Testing
- **MFA:** Multi-Factor Authentication
- **UI:** User Interface

1.4 References

- **Project Management**: GitHub and Trello for version control and task organization.
- **Frontend Frameworks:** React for interactive UI components and Bootstrap for responsive design.
- **Backend Framework:** Flask for developing RESTful APIs.

- **Database:** Microsoft SQL for secure data storage and retrieval.

2. System Overview

2.1 Product Perspective

This system is a standalone system, and it interacts with users. It consists of:

- **Frontend**: The user-facing part of the system, available as a website or app, where users browse products, add items to their carts, and complete purchases. It is designed for easy use across devices such as computers, tablets, and smartphones.
- **Backend**: The server-side component that handles processes like user authentication, product management, and order processing. It also interacts with external payment services to securely manage transactions.
- **Database**:A structured storage system for user data, product details, and order history. It ensures efficient data retrieval and security, preventing unauthorized access.

2.2 Product Functions

The key functions of the system include:

1. **User Account Management**: The system allows users to create accounts, log in, and manage their profile information. This function includes user authentication and password management to ensure secure access.
2. **Product Browsing and Search**: Users can browse products by categories or use the search feature to find specific items. This function supports filters and sorting options to help users find products easily.
3. **Order Placement and Tracking**: Users can add items to their cart, proceed to checkout, and place orders. This function integrates payment processing and provides order confirmation notifications.

2.3 User Classes and Characteristics

Different types of users who will interact with the system include:

- **Admin Users**: Have full access to all system functions, including managing user accounts, overseeing product listings, handling orders, and configuring system settings.
- **Registered Users**: Can perform standard tasks such as browsing products, adding items to their cart, placing orders,purchase products ,and managing their personal account information
- **Guest Users**: Can only view product listings and basic content but cannot place orders or access user-specific features or purchase products without registering.

2.4 Operating Environment

The system will operate in the following environments:

- **Client Side**: The system runs on web browsers such as Chrome, Firefox, and Safari to ensure compatibility and accessibility for users.
- **Server Side**: The system is hosted on a cloud server running Linux and utilizes a software stack including Flask (Python Library) for server-side processing.
- **Database**: The system uses Microsoft SQL Server to manage and store all persistent data efficiently and securely.

3. Functional Requirements

3.1 Use Case Diagrams / User Stories

- **Use Case 1**: Admin Capabilities
    - **Description**: Admin can add new categories in the website like: (Fruits, Vegetables,Electronics) , and should provide the name of these categories, and can modify these categories he can delete any category of change its name, admin also can add products and should provide the id of the product and the product name
    - **Actors**: Admin
    - **Preconditions**: Admin should have the Authorization to make these.
    - **Postconditions**: these functions should be implemented successfully.
    - **Steps**:
        1. Login as an admin
        2. go to website settings
        3. Modify or add anything you want.

- **Use Case 2**: Customer
    - **Description:** Customer interaction with our system
    - **Actor: Customer**
    - First the customer will register to our system if he doesn't have account or he can just continue as a guest if he want
    - Then he will login or logout to his account
    - He can also update his information
    - Search for the product within the system and he can add or remove it to the cart to make the final checkout.

- **Use Case 3**: Purchase
    - **Description**: The first primary actor can make a purchase. This implies there is a payment process will occur. This system has four types of payment which are : cash, paypal, credit or fawry. Then the secondary actor will coordinate the payment process.
    - **Actors**: primary actor is the user, secondary actor is the payment services coordinator.
    - **Preconditions**: the user must do any purchase .
    - **Postconditions**: the process confirmation and withdrawal.
    - **Steps**:
        1. item selected to be purchased
        2. payment process
        3. selection from four methods of payment.
        4. finally the transaction successfully or unsuccessfully completed.

3.2 Feature Requirements

Feature 1: Search for a product.

- **Description**: This feature must allow the user to find and discover the list of products.
- **Inputs**: Query
- **Outputs**: all relevant products for the query.

- **Error Handling**:if the product does not exist the search box will show an empty search result.

Feature 2: Shopping Cart

- **Description**: The user can be able to add, drop or review products.
- **Inputs**: product quantity, delivery address payment method
- **Outputs**: confirmation string with cart summary.
- **Error Handling**: The added product is not available at the current time the system will show an error message to the user.

Feature 3: Payment

- **Description**: various Payment gateways to support more than one option.
- **Inputs**: payment information such as payment method, card number expiration date, CVV
- **Outputs**: Successful or failure message
- **Error Handling**: alternative payment methods. error message

Feature 4: Promotions and sales Page

- **Description**: the user shall be able to get sales and promotion with various types like : product bundle offers, buy one get one and discount.
- **Inputs**: promotion types and details
- **Outputs**: go to the promotion page
- **Error Handling**: If any of the input fields fail validation, display error messages explaining the issue and how to solve it.

Feature 5: Order Management and Tracking

- **Description**: the user must be able to place an order for products that exist. Also, the user can track his order. in addition the user should be able to view his/her order history.
- **Inputs**: order id
- **Outputs**: order details
- **Error Handling**: if the order with invalid details there will be an error message describing the problem.

Feature 6: Location and Delivery

- **Description**: The user can enter their delivery address during checkout, and the system will ensure the order is shipped to the provided address. The user's location will be used to calculate delivery charges and estimated delivery time.

- **Inputs**: User's address, User's contact information Order details like products selected and quantities.
- **Outputs**: Confirmation of the delivery address entered by the user. Estimated delivery time and shipping charges based on the address.
- **Error Handling**: If the delivery address is outside the serviceable area, the system will display: "Sorry, we do not deliver to this location."

Feature 7: sign up via facebook

- **Description**: The system should allow users to sign up and create an account by using their existing Facebook account. This feature provides a fast registration process for users.

- **Inputs**: facebook login data
- **Outputs**:account creation
- **Error Handling**:if the authentication failed the system will provide an error message.

Feature 8: Return Product

- **Description**: The system allows users to return a purchased product and receive a full refund within a specified return period. The return process includes verifying the product's condition and confirming the user's eligibility for a refund.
- **Inputs**: User request to return a product, Product ID, Order ID.Condition of the product (e.g., unused, damaged, or defective). User's reason for the return.
- **Outputs**: Confirmation of return request, Notification of refund approval or rejection.
- **Error Handling**: If the return request is outside the allowed return period, the system will notify the user with an error message: "Return period has expired."

Feature 9: Categories Page

- **Description**: The user can choose a product category from a list and be directed to a page displaying all products within that category. The products will be organized and displayed for easy browsing.
- **Inputs**: User selects a product category from the category list.
- **Outputs**: Display of products within the selected category.
- **Error Handling**: If the selected category has no products, the system will display: "No products available in this category right now."

Feature 10: user review

- **Description**: The user can write reviews and post comments on products they have purchased. Reviews include a rating from 1 to 5 stars and a comment to share feedback about the product.
- **Inputs**: User's rating from 1 to 5 stars, User's written comment on the product, Product ID.
- **Outputs**: Displayed review on the product page including the rating and comment., Average rating updated on the product page based on all reviews, Confirmation message: "Your review has been submitted."
- **Error Handling**: If the user has not purchased the product, the system will display: "You must purchase the product to leave a review.".

4. Non-Functional Requirements

4.1 Performance Requirements

- The product search result should be viewed within 3 seconds.

4.2 Security Requirements

- The system must secure all users' data by encrypting it.

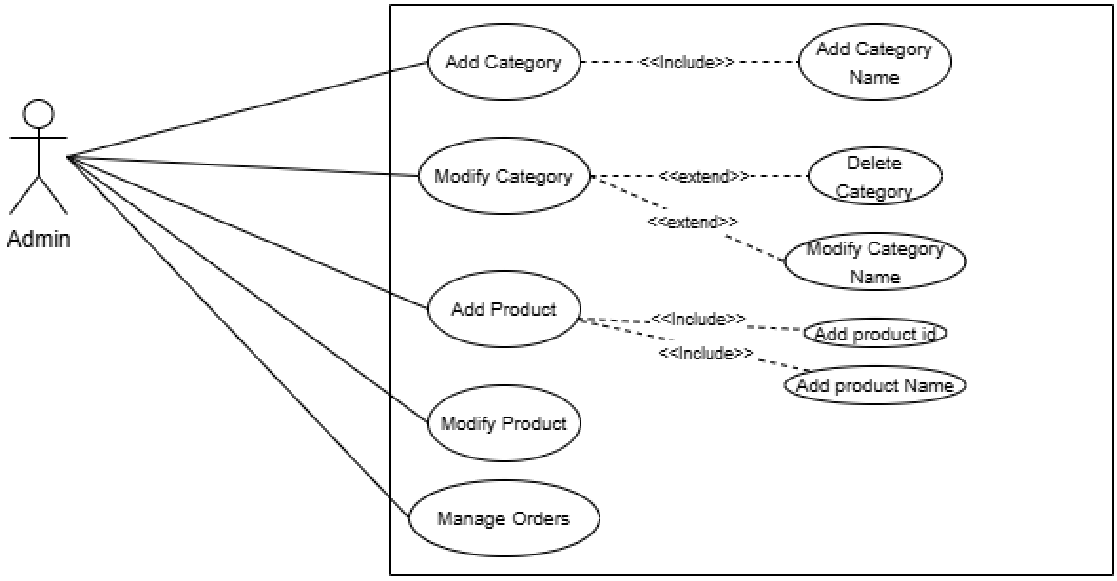
4.3 Reliability and Availability Requirements

- The system shall be available to any transaction during the 24 working hours.

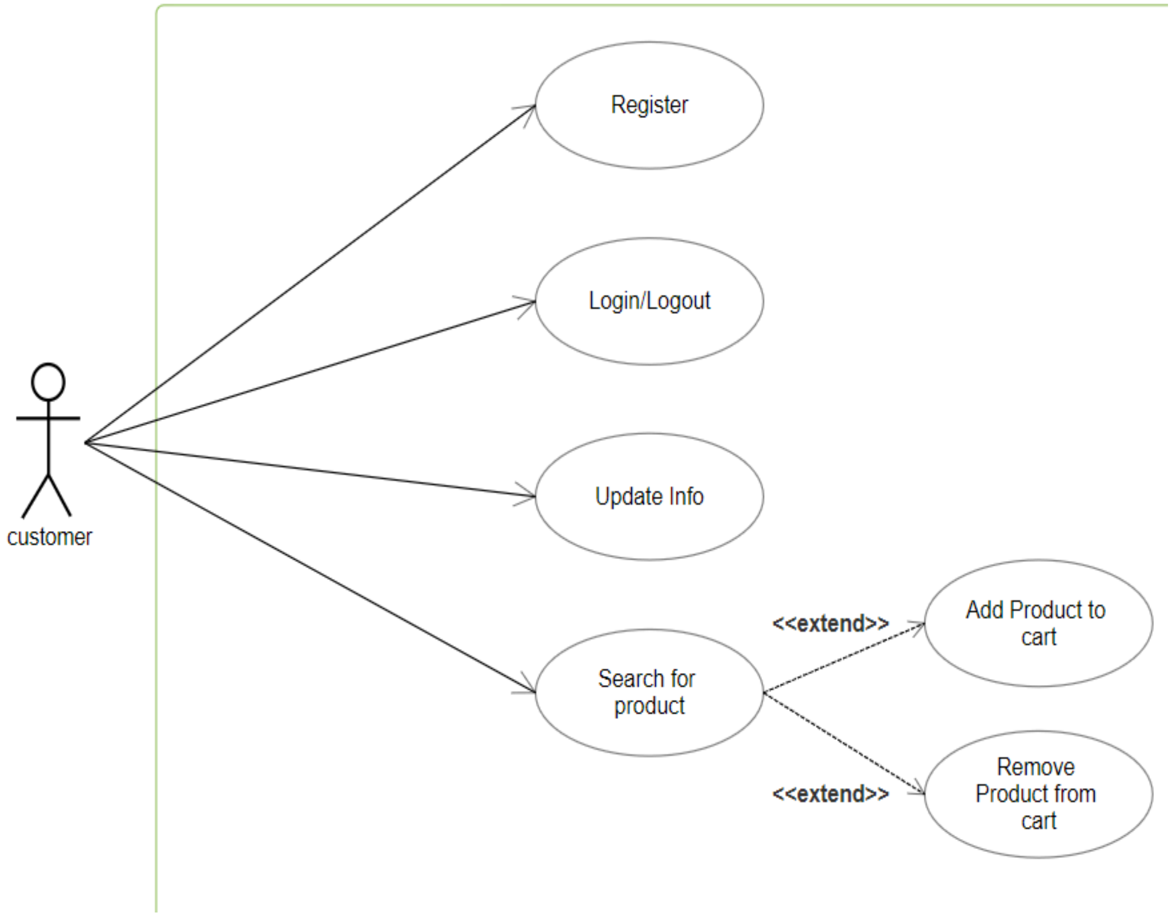
5. System Models

5.1 Use Case Diagrams

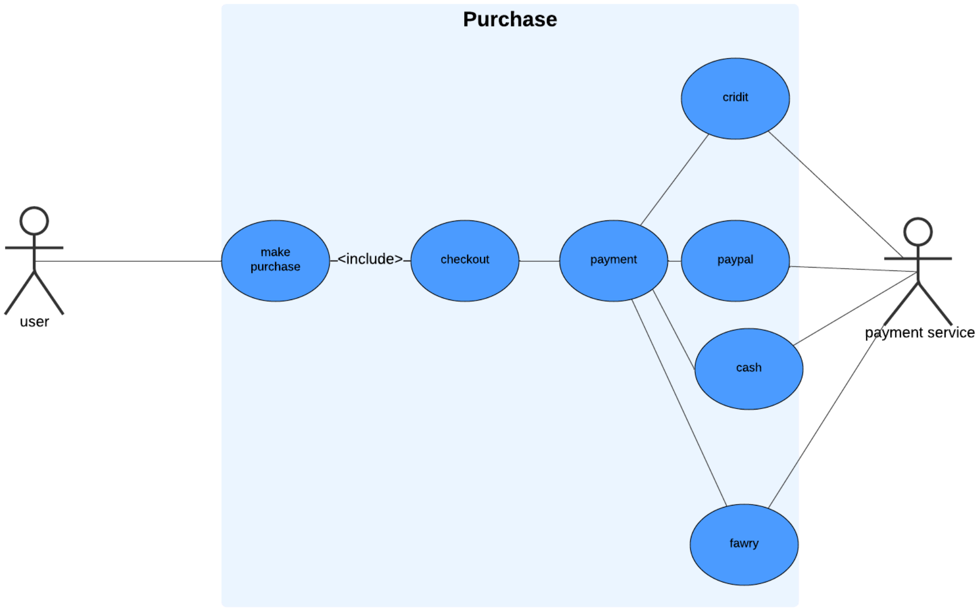
- **Figure 1:**



- **Figure 2:**



- **Figure 3:**



5.2 Data Flow Diagrams

[Insert Data Flow Diagrams to represent how data moves through the system.]

5.3 Class Diagrams

[Insert Class Diagrams to represent the system's object model.]

6. External Interface Requirements

6.1 User Interfaces

- The system should have an intuitive UI with the following major components:

  1. **Home Screen**: Displays a summary of user activity, including the latest searches and shopping history for users.
  2. **Dashboard**: Allows the user to perform actions such as browsing, adding products to the cart, and managing their account. It also includes advanced search options with filters, a smart search experience that learns from user behavior.

6.2 API Interfaces

- Data Retrieval: Enable users to retrieve product data and user information.

- User Management: APIs to handle user authentication such as registration, and profile management.

- Search and Recommendations: APIs that support advanced search features and data mining techniques to improve search and user experience.

6.3 Hardware Interfaces

- The system will interact with external hardware as required as example:
- Printers: To print order details, receipts as part of the order fulfillment process.

7. Other Requirements

7.1 Legal and Regulatory Requirements

- The system must comply with regulations related to online marketplaces and user data protection, such as: General Data Protection Regulation (GDPR): The system will comply with and adhere to the GDPR regulations.

7.2 Documentation Requirements

- user manuals: The project will provide comprehensive documentation, including: Explanation for users on how to navigate the platform, conduct searches, and make purchases.
- API Documentation: Detailed API references for developers, to ensure user information management, data retrieval, and product management for easy integration and maintenance.

7.3 Data Backup Requirements

- The system shall back up important data, including product information, user data, and transaction records, on a daily basis.
- Retention Period: Backups will be securely stored for at least 30 days, ensuring that data can be restored in the event of unexpected problems or data loss.

8. Conclusion

This SRS document outlines the necessary functional and nonfunctional requirements for the El Hamour Market project. By following specific rules, we hope to provide a flexible system for users, ensuring a secure and reliable online marketplace experience. This comprehensive specification works to deliver a powerful system with high quality.