Yarn Web Shop - Project Overview

Vision

The goal of this project is to create a simple e-commerce platform for a local yarn shop. The platform allows customers to browse available products, make purchases, and interact with the system seamlessly. The shop should be able to manage inventory, track orders, and update product availability without hassle.

Functional Requirements

1. **Customer Registration and Login**: Customers can register, log in, and save their personal and payment details for future orders.
2. **Product Catalog**: The store staff can add, update, and remove yarn, knitting needles, and accessories from the catalog.
3. **Customer Orders**: The system handles customer orders, showing the status of each order (e.g., "Processing," "Shipped").
4. **Payment**: Customers can make payments securely using various payment methods such as credit cards or Swish.
5. **Inventory Management**: The system automatically updates inventory levels after each sale and allows store staff to adjust inventory manually.

Non-Functional Requirements

1. **Performance**: The system should handle up to 100 simultaneous users without any function response time exceeding 3 seconds.
2. **Security**: The system should encrypt sensitive information like customer payment details, adhering to security standards for e-commerce.
3. **Availability**: The system must be available 24/7 with no more than 1% downtime per month for maintenance.

MoSCoW Prioritization

- **Must**:
    - Customer registration and login
    - Product catalog
    - Payment system
- **Should**:
    - Customer order management

    - Inventory management

- **Could**:
    - Integration with shipping companies for automatic shipment tracking.
    - Customer reviews for products.

Reflection

The critical features for this system to function are customer registration, product catalog management, and payment handling. Without these, the e-commerce platform would not be complete. Customer orders and inventory management are also important but can be released in future updates if not urgent from the start. Integration with shipping companies and customer reviews would enhance user experience but can be added later.

User Stories

1. **Customer Registration and Login**:
    - "As a customer, I want to register and log in to the web shop so I can make purchases and save my address and payment details for future orders."
2. **Product Catalog**:
    - "As store staff, I want to add, update, and remove products from the catalog so customers always see the most up-to-date product availability."
3. **Payment**:
    - "As a customer, I want to pay securely using payment options like credit cards or Swish, so I can complete my purchase safely."

Use Case: Payment of Order

- **Name**: Payment of Order
- **Actor**: Customer
- **Preconditions**:
    - The customer has selected products and added them to their cart.
    - The customer is logged in or has registered before the purchase.
- **Main Flow**:

1. Customer goes to checkout and selects a payment method.
2. The system displays available payment options.

3. Customer chooses a payment method and enters payment details.
4. The system sends payment information to the payment gateway.
5. The payment gateway processes the transaction and returns an approval/denial.
6. The system confirms the payment and displays an order confirmation.

- **Alternative Flow**:
    - If payment fails (e.g., insufficient funds), the system shows an error message and prompts the customer to try again or select another payment method.

Mini Backlog

1. **User Story: Customer Registration & Login**
    - Task: Implement registration form and login functionality.
    - Link: [GitHub Issue #1]
2. **User Story: Product Catalog**
    - Task: Build functionality to add, update, and remove products.
    - Link: [GitHub Issue #2]
3. **User Story: Payment**
    - Task: Integrate with payment gateway (e.g., Klarna/Swish).
    - Link: [GitHub Issue #3]
4. **Task: Test Payment Flow**
    - Link: [GitHub Issue #4]
5. **Task: UI for Cart and Checkout Pages**
    - Link: [GitHub Issue #5]

Differences Between User Story & Use Case

- **User Story**: A short description of a feature from the user's perspective, focusing on the need and expectations. It answers the "who," "what," and "why" of a feature.
- **Use Case**: A more detailed and technical description, including flows and conditions, often broken down into step-by-step actions of how the user and system interact.

- **User Stories** help establish the needs and directions of the project, while **Use Cases** break those needs down into actionable steps for implementation.

It is important to balance both to understand both user requirements and technical feasibility.