MSE800-Professional Software Engineering

Assessment 2

July 2025

The Development of Smart Bargain Web Application

—

as partial completion of Master's Degree in Software Engineering

Submitted by:

Palamuni Telma Lakmali De Silva
E-Mail: 270620124@yoobeestudent.ac.nz

Submission date: July 13, 2025

Shiela Marie L. Umali
E-Mail: 270650805@yoobeestudent.ac.nz

Abstract

The Smart Bargain project is a digital marketplace platform for direct and transparent trading of agricultural produce (fruit & vegetable) throughout Aotearoa/New Zealand. The system allows for users to register as buyers or sellers and have specific features to each type of users. Sellers have the ability to either list or update their products and, if necessary, remove them. Buyers can browse, add to cart, and buy ptoducts, with advanced cost calculations, all in a safe transactional environment. One unique feature of Smart Bargain is its strong negotiating module where both buyers and sellers can negotiate on price and quantity. This give-and-take with a single source of truth creates a level playing field and flexible trade agreements. The design of the system is based on the principles of Te Tiriti o Waitangi—partnership, participation, and protection. It is about ensuring that it serves users equitably and safely, while respecting the culture. Smart Bargain is built in Django with tried and tested design patterns for scaling and security. Additional features on the roadmap includes online payment, delivery tracking, dispute resolution, sustainability certification, and analysis dashboards. Through its promotion of transparent negotiation and equitable involvement, Smart Bargain seeks to give communities a voice, encourage fair trade, and increase the productivity of agricultural trade in New Zealand.

Contents

List of Figures

List of Tables

1 System Scope

The Smart Bargain system is a dynamic e-commerce and trading platform for the direct trade of fruit and vegetables in Aotearoa/New Zealand. It is also equipped with functionalities to register as a Buyer or a Seller, and user specific functionalities. Sellers can publish their products through a form containing product description, price per unit, available quantity, unit of measurement (i.e., kg, bundle), and product image. They also have the option to edit or remove these listings whenever they want. Upon visiting Smart Bargain, buyers are able to search for available products, choose the items they want to browse through, and the quantities they want to commit to buy. An order flow causes the upfront price determination, the verification steps, and the monetary transaction to be executed.

One of the main features of Smart Bargain is the ability to reach an agreement by trading in fair and flexible conditions. Buyers can make a negotiation request if they are not satisfied with the listing price per quantity. The negotiation includes different terms such as a lower price per quantity. This request is stored as a Negotiation Request and displayed in buyer and seller dashboards. A negotiation module has been developed in which both sides can accept, reject or make counter-proposals on a sequential renegotiation. This deliberation is structured, and continues until there is either agreement or cessation in negotiation. Status tags such as Active, Pending Buyer Approval, Pending Seller Approval, Cancelled and Rejected help track the progress and current state of each negotiation transparently.

In implementing the principles of Te Tiriti o Waitangi, Smart Bargain is built with the following commitments:

- **Partnership:** Smart Bargain facilitates joint decisions by allowing sellers and buyers to co-create value through negotiated contracts. This can be expanded to Māori farmers, cooperatives, and community groups with businesses respectful of their culture.
- **Participation:** Everyone can negotiate and make counter proposals in Smart Bargain. The system has been designed with low barriers to entry and equitable access, this includes the potential for Te Reo Māori language support and mobile friendly interfaces to reach those in rural communities.
- **Protection:** The application secures its registered members by the best forms of login and data retrievals. Order information is transparent and all trade records are traceable to protect traders from misunderstanding or fraud. It also promotes transparency of the decisions and trust of involved sides.

Smart Bargain built on top of Django framework, features solid patterns such as Singleton for easy database access and Factory for dynamic role management. Planned features

include the support of digital payment, delivery tracking, a dispute resolution system, sustainability certification, and analytics dashboards in order to facilitate decision making and community impact assessment.

2 Project Kick-off

Kick-off Meeting

- **Date:** June 13, 2025
- **Format:** Virtual (MS Teams)
- **Attendees:** Shiela Marie Umali, Palamuni Telma Lakmali De Silva

Agenda

1. **Introductions**: The project kick-off started by each member sharing their backgrounds and expertise in the important fields for this project. These includes full-stack development, project management, QA, and design. This discussion will be essential in work allocation.

2. **System Scope & Vision**: One of the topics emphasized on the meeting was the main objective of this project. Members have discussed and reviewed the project's objectives which focused on: 1) unique negotiation feature that bridges traditional market bargaining with digital commerce, and 2) its commitment to the principles of Te Tiriti o Waitangi.

3. **Feature Prioritization**:
   The Minimum Viable Product(MVP) includes the following:
   - User registration and role management (Buyer/Seller)
   - Product catalog with categories, filtering, search, and detailed listings (including images, pricing, quantity, and measurement units)
   - Buyer browsing and order management
   - Negotiation module supporting iterative offers, counter-proposal, and negotiation status tracking (Active, Pending Buyer/Seller Approval, Cancelled, Rejected)
   - User dashboards for managing listings, negotiations, and sales

4. **Technical Foundation**: Both members have agreed on the tools which will be implemented on this project. It was confirmed that Django will be the backend

framework which employs Singleton and Factory design patterns for scalability and maintainability. Frontend technologies include HTML, CSS, and JavaScript. Database options include SQLite or MySQL. For future deployment, it was planned to use PythonAnywhere.

5. **Collaboration Tools**: MS Teams for communication and meetings, GitHub for code repository and version control, OneNote for documentation, JAM for collaborative brainstorming, Overleaf for technical writing, and planned migration to JIRA for agile project management.

6. **Roles & Responsibilities**: Allocated leadership and supporting roles to cover project management, development, QA, design, and documentation. This allocation ensures redundancy and collaborative ownership.

7. **Milestones & Timeline**: Outlined project phases including research-based requirements gathering, design, development, QA, and pilot launch, targeting a 4-week MVP delivery.

8. **Te Tiriti o Waitangi Commitments**: Smart Bargain incorporates the principles of partnership, participation, and protection throughout its development and features. The platform enables collaborative value creation, ensures equitable access—including Te Reo Māori language support—and maintains transparent, secure trade records. These principles promote fairness, inclusivity, and trust for all users.

9. **Initial Tasks**: Assigned immediate action items such as conducting research for requirements gathering, setting up the development environment, and initiating wireframing of key user flows including registration, product management, and negotiation.

Project Proposal

For the project proposal, please see the document here.

3 Team Formation

3.1 Team Assembly Overview

The project team of Smart Bargain web application was purposefully composed as a cross-functional and strongly-cooperative team which consisted of two aspiring software engineers: Shiela Marie Umali and Palamuni Telma Lakmali De Silva. Both have experience in full-stack development, project management, quality assurance, design, and

specialized roles. In spite of having a small team, there is an efficient coverage of all aspects of project delivery.

3.2 Team Structure and Rationale

Considering small team size, the responsibilities are reassigned parallel to each individual's own capabilities. Redundancy was also taken to consideration to maintain business continuity. The cross-functional model enables flexibility, fast iteration, and joint ownership of project results. Primary and secondary roles are assigned according to each person's assigned roles, which is based on experience and the best fit of strengths. The titles and duties are:

Shiela Marie Umali

- **Head of Project Management**
  Oversees project planning, scheduling, prioritization, and communication. Ensures milestones are met and risks are managed.
- **Frontend & Backend Developer**
  Designs and implements both client-side and server-side features in collaboration with the lead developer.
- **QA Engineer**
  Conducts functional and regression testing, documents defects, and verifies fixes.
- **Design Lead**
  Leads UI/UX design, creates wireframes and prototypes, and sets the overall visual direction.
- **Technical Writer**
  Prepares and maintains technical documentation, user guides, and release notes to ensure clear communication and knowledge transfer.

Palamuni Telma Lakmali De Silva

- **Scrum Master**
  Facilitates Agile processes, removes obstacles, and supports the team in following Scrum practices.
- **Lead Developer (Frontend & Backend)**
  Architects core system components, reviews code, and ensures best practices in software development.

- **QA Lead**
  Defines test strategies, manages the QA process, and ensures product quality standards are met.

- **Designer**
  Collaborates on UI/UX design, refines visual elements, and ensures design consistency.

- **Assistant Project Manager**
  Supports project tracking, documentation, and stakeholder communication. Assists in resource allocation and progress reporting.

There are also different leadership roles, (Scrum Master, Project Management, Lead Developer, QA Lead, Design Lead) assigned based on their primary strength and background. Supporting roles are allocated to provide the coverage, knowledge sharing and continuity during the execution of this project. Shiela and Lakmali were part of the development and design in all major task. It makes working together, and as a result velocity at the project level, very high.

3.3 Collaboration Tools

- **MS Teams:** Communication and meetings
- **Github:** Code repository and version control
- **Overleaf:** Collaborative documentation and technical writing
- **OneNote:** Meeting documentation
- **JAM:** Team collaboration and brainstorming
- **JIRA:** (Planned) Issue tracking and agile project management (migration from GitHub on-going)

4 Requirement Gathering

Figure 1: Smart Bargain requirement gathering process.

At the start of our project, we submitted our proposal to our module professor, who served as both client and key stakeholder. After reviewing our proposal, we connected with him to seek inputs on implementation strategies through out the implementation phase. We also received advice on how to approach specific components. Rather than having formal meetings, our communication was more focused on practical feedback and academic guidance. This helped us align our solution with realistic expectations and academic goals.

Initially the requirements were gathered by conducting quick research online. This helped us understand the eCommerce structure in New Zealand. We explored platforms like Woolsworth and PAK'nSAVE, to understand the pricing and how agricultural products are bought and sold. Also we explored sites like fruit world and farm fresh direct. Through this we wanted to understand the eCommerce aspect as well as to understand if any negotiation or online bargaining component is been used. We realized that most lacked a digital space for price negotiation or direct buyer-seller communication. This is a core aspect of traditional marketplaces that are still prevalent in many communities, especially for fresh produce. It a common aspect when bulk purchases are done.

To understand more on cultural aspect we had informal conversations with staff at Woolsworth and PAK'nSAVE. As a result, we learned that price variations often depend on seasonal demand, and that direct negotiation still plays a crucial role. This is more common at farmers' markets or during bulk purchases. We also noted that physical visits to suppliers are often necessary when a buyer wants to negotiate or build trust. This is something we felt could be improved via technology.

During in store visits to retailers like Woolworths and PAK'nSAVE, we also made notes on product pricing and and we also checked their online listings. We realized fruit and vegetable prices fluctuation depnds on the time of year. This supports our assumption that a negotiation feature could help buyers strike better deals in high demand periods and give sellers more control during surplus seasons. And for sellers the wastage of produced goods will be comparetvly less if there are bargains or negotiations done prior. We did have a look at existing bidding and auction platforms, but realized that this functionality required further investigation and domain specific validation. Hence we could not adapt any feature or functionality. Based on the market domain we came up with our own negotiation platform in terms of user expectations, fairness, and regulatory compliance. We decided to defer bidding functionality to a future sprint and focus on negotiation and core eCommerce features for our initial release.

Throughout the requirement gathering process, we prioritized inclusive collaboration. As a multicultural team, we practiced open communication and actively listened to each other's perspectives. As a team we had both agreements and disagreements on solution approach. Through respectful discussions and learning from one another we came to a common ground on every approach. We also acknowledged the importance of respecting Te Tiriti o Waitangi principles by designing a solution that could potentially support Māori and local community. Smart Bargain provides them with more control over pricing and visibility in the market.

By combining professional consultation, inputs from community, and market research, we gained a good understanding of the challenges and opportunities in New Zealand's agricultural eCommerce space. This gave us a strong foundation for our core features. Such as user registration, product listing, and the unique online negotiation feature that differentiates our platform from conventional solutions.

5 Requirement Analysis and Prioritization

Once the initial requirement gathering and stakeholder input is completed Smart Bargain team conducted a systematic analysis of all proposed features. This was done in order to determine what could realistically be implemented within the project timeline and available resources. Team's goal was to prioritize functionality that delivers immediate value to users and lay a strong foundation for future scalability.
We started by listing a number of features that aligned with traditional eCommerce expectations. To this the team introduced the negotiation and the bid feature which guarantees the uniqueness of the Smart Bargain system. After analyzing the technical complexity and time constraints team categorized the requirements into two groups. as In scope and Out of scope features for the initial implementation.

In Scope Features: The below were identified as the most essential and feasible features for the first deployment.

Homepage: This is the entry point allowing access to other key pages.

Registration and Login: Only two user roles are defined for the initial release and that is the "buyer" and "seller". Based on user role, accessibility to different components is defined.

Product Catalog with Categories: This page outlines the basic product management capabilities. Buyers can view and purchase products — the traditional eCommerce implementation. Sellers can search, add, edit, and delete products for buyers to purchase.

Negotiation Feature: What makes Smart Bargain special from other eCommerce sites is this feature. This feature enables sellers to set bargain thresholds. Buyers, on the other hand, can initiate a negotiation with the seller directly.

Order Confirmation and History: Provides a clear record of orders for each seller and buyer.

Direct Messaging System: Any user, regardless of being registered or not, has access to this feature where users can send direct messages to the seller. This could be with regard to any purchases or even wholesale suppliers reaching out to sellers.

Buyer and Seller Dashboards: These are role-based interfaces for managing offers, orders, negotiations, and messages.

Role Based Access Control: This feature ensures that each user sees and interacts only with features relevant to their role.

These features were chosen because they are both achievable and essential to validating our unique concept. Which is negotiation based e-commerce solution tailored for agriculture industry.

Out Of Scope Features: Several ideas were determined to be better suited for future sprints due to higher complexity or lower immediate value. Below is the identified list.

Bid Feature: Automatic pricing proposals without negotiation flow

Product Recommendations: requires AI or user behavior analytics

Automated Approval Logic: For negotiation feature

Multiple Payment Models: Introduce more payment options like paypal and crypto wallet payments

Mobile Responsive Design: Introduce Smart Bargain to mobile accessible user friendly design.

Advanced Order Management: This is to maintain orders once the negotiations are active on the system.

Invoice Management: Once the Order management is introduced to active negotiations, the invoices are to be auto created or updated based on the orders and negotiations.

Rating and Review Systems: Allows sellers to rate and review product purchases and negotiations

Delivery Management: For orders and and active negotiations a mechanism needs to be introduced to manage transportation of goods.

Each of these features has value but was excluded either because they require advanced logic (e.g., bid automation), integration with third party services (e.g., payment/invoicing

systems), or design considerations that extend beyond the current logic. They remain documented as backlog items to be revisited in future iterations.

This prioritization approach ensured that core value of Smart Bargain is culturally relevant negotiation driven e-commerce platform can be delivered with quality and clarity. It also reflects local considerations, such as the importance of building trust in buyer seller relationships, specially in agricultural settings where repeat business and verbal agreement style transactions are common.

6 System Design

6.1 Use Case Diagram

The use case diagram of the Smart Bargain System models the main potential tasks of the users (Buyers, Sellers, Unauthenticated Users) and the system (core functions). Each use case has a target of what the user should be able to do/have access to or the task that the user completes within the platform, to provide a clear understanding of what needs to be built and how different users will interact with the system. Both this diagram and the use case list are perfect inclusions in technical documentation or system design documents. They give broad description of the user interactions, the system boundaries, and the fundamental Smart Bargain features which will assist in requirement analysis and system development.

Key Components

- **Actors**
    1. **Buyer:** Registered User. Able to buy and browse products, and negotiate to seller.
    2. **Seller:** Registered user who is able to create and manage products, negotiate and manage messages.
    3. **Unathenticated User:** Unregistered user with limited access to Smart Bargain system. Can fill-out contact forms. Able to create an account.
- **Main Use Cases**
    1. **Account Management:** Registration, login, and logout for buyers and sellers.
    2. **Product Management:** Add, edit, delete, and search product
    3. **Purchasing Action:** Viewing Products, purchase, purchase confirmation and payment of products.
    4. **Negotiation Management:** Initiating, accepting, rejecting, canceling and counter-proposing in negotiation contexts.
    5. **Orders and Messages management:** Order history, contact forms submission, and messages management.

Note: For a better view of the diagram, access the original pdf file thru this link:
Smart Bargain System - Use Case Diagram

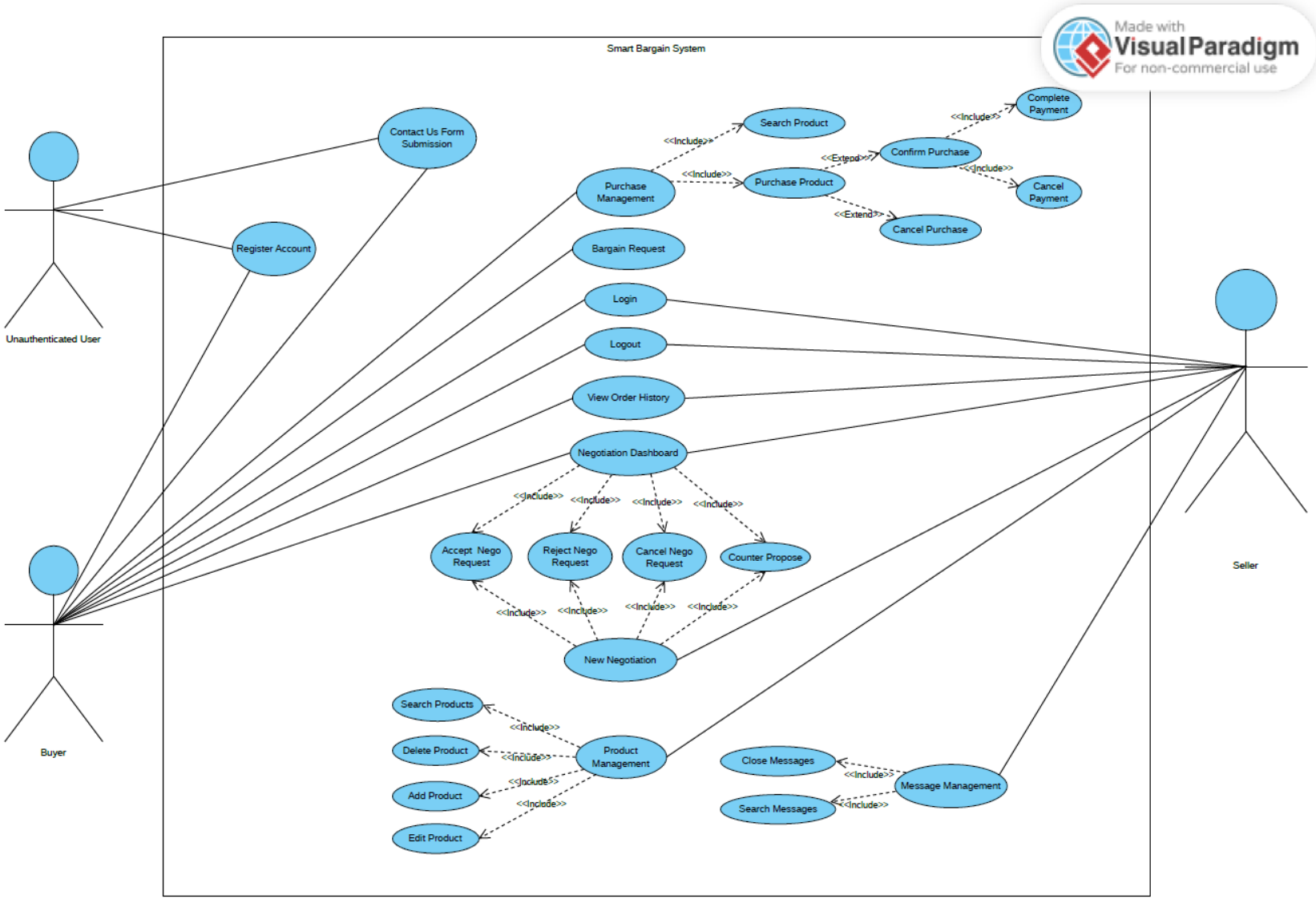

Figure 2: Use Case Diagram of Smart Bargain System

Note: For a better view of the diagram, access the original pdf file thru this link:
Smart Bargain System - Use Case Diagram

Use Case Number	Use Case Name
SB-UC-0001	Register a Buyer or Seller Account
SB-UC-0002	Login to Smart Bargain
SB-UC-0003	Logout from Smart Bargain
SB-UC-0004	Navigate from Home Page
SB-UC-0005	Buyer and Seller Access rights
SB-UC-0006	Search and Filter Products in products and negotiation page
SB-UC-0007	Add Product to Catalog
SB-UC-0008	Edit Product Details
SB-UC-0009	Delete Product Details
SB-UC-0010	View and Purchase Products
SB-UC-0011	Complete Payment via Purchase Modal
SB-UC-0012	Seller Updates Bargain Settings
SB-UC-0013	Buyer Initiates Bargain Request
SB-UC-0014	Seller Reviews and Responds to New Negotiation Request
SB-UC-0015	Seller or Buyer Accepts Negotiation Request
SB-UC-0016	Negotiation Dashboard - Ongoing Negotiation Management
SB-UC-0017	Seller or Buyer Rejects Negotiation Request
SB-UC-0018	Seller or Buyer Cancels Negotiation Request
SB-UC-0019	Seller or Buyer Counter Proposes on Negotiation
SB-UC-0020	View Order History
SB-UC-0021	Contact Us Form Submission
SB-UC-0022	View Contact Messages
SB-UC-0023	Filter Contact Messages on Seller Dashboard

Table 1: Smart Bargain Use Case List

Use Case Number	SB-UC-0001
Use Case Name	Register a Buyer or Seller Account
Actors	New User
Preconditions	1. The user is not already registered. 2. The user has a valid email address.
Description	A new user visits the registration section and chooses whether to register as a buyer or seller by selecting the appropriate option. They then fill in their username, email, password, and confirm password fields. Upon successful registration: • A confirmation message is shown to the user. • The user can proceed to the login section to access their account.
Error Handling	1. Show an error if the email is already in use. 2. Show a warning if the passwords do not match. 3. Show a message if any required field is left empty.

Use Case Number	SB-UC-0002
Use Case Name	Login to Smart Bargain
Actors	Buyer and Seller
Preconditions	1. The user has already registered as a buyer or seller. 2. The user knows their registered email and password.
Description	The user enters their email and password on the login page to access their Smart Bargain account. If the credentials are correct: • A welcome message is displayed with the user's name. • The user is redirected to the home page. • The "Login" option on the navigation bar changes to "Logout", showing the user's name next to it.
Error Handling	1. Show a clear error message if the email or password is incorrect. 2. Show a message if the fields are left empty.

Use Case Number	SB-UC-0003
Use Case Name	Logout from Smart Bargain
Actors	Buyer and Seller
Preconditions	The user is currently logged into their Smart Bargain account
Description	The user clicks the Logout option from the navigation bar to safely exit their account. After logging out: • The user is redirected to the login/registration page. • The navigation bar is updated to show the Login option instead of the user's name.
Error Handling	If a user attempts to access a restricted page after logout, they should be redirected to the login page

Use Case Number	SB-UC-0004
Use Case Name	Navigate from Home Page
Actors	Buyer and Seller
Preconditions	1. The user is on the Smart Bargain Home Page after logging in. 2. Navigation buttons and links are visible and active.
Description	The user interacts with key buttons on the Home Page for quick access to core features: • Clicking "Shop Now" or "Explore Our Catalog" navigates the user to the Product Page. • Clicking "Your Dashboard" navigates the user to their personalized Dashboard Page, showing orders and negotiation history. • Clicking "Start Negotiating" directs the user to the Negotiation Page where they can initiate or update negotiation threshold.
Error Handling	If the user is not logged in and tries to access restricted pages (Dashboard or Negotiation), they are redirected to the Login page.

Use Case Number	SB-UC-0005
Use Case Name	Buyer and Seller Access Rights
Actors	Buyer, Seller
Preconditions	• User logged in as Buyer or Seller • Role-based access implemented
Description	After login, users access pages and actions by role: **Common Access:** • Home: General navigation • Dashboard: Personalized views • Product: View products (access varies) • Negotiate: Participate in negotiations **Seller Access:** • Products: Add/edit/delete catalog items • Negotiate: Set min. quantity/price • Dashboard: View negotiation requests, negotiation, and order tables **Buyer Access:** • Products: Purchase from catalog • Negotiate: Initiate bargain requests • Dashboard: View negotiation and order tables
Error Handling	Hide or disable restricted actions based on user role

Use Case Number	SB-UC-0006
Use Case Name	Search and Filter Products in products and negotiation page
Actors	Buyer and Seller
Preconditions	1. User must be logged in as either Buyer or Seller 2. Products exist in the system with assigned categories (vegetables or fruits) 3. The frontend view includes a search input and category filter dropdown
Description	1. Buyer/Seller Logs into the system 2. Navigates to Products or Negotiate Page By default, all products are displayed. • Users can **type in a product name** to narrow down the list. • Users can also **filter by category** (All, Vegetable, Fruits). • The filter and search criteria work **together**: filtering by a category and typing a name will combine both criteria.
Error Handling	1. If no products match the search/filter criteria, display a message: "No products found" 2. If the search input is blank and filter is set to "All", display all product

Use Case Number	SB-UC-0007
Use Case Name	Add Product to Catalog
Actors	Seller
Preconditions	1. Seller is logged in 2. Seller has navigated to the Product Page
Description	The seller adds a new product to the catalog by providing the following information: • Product Name • Image Path or URL • Available Quantity • Unit (e.g., Kilogram) • Price per unit • Category (selected from dropdown: Vegetables, Fruits) On submitting the form, the product is added to the product list.
Error Handling	1. Show a message if any required field is missing 2. Show a message if price or quantity is invalid (e.g., negative or non-numeric)

Use Case Number	SB-UC-0008
Use Case Name	Edit Product Details
Actors	Seller
Preconditions	1. Seller is logged in 2. At least one product exists in the seller's product list 3. Seller is on the Product Page
Description	The seller selects a product from the list and clicks Edit. An editable form is displayed with pre-filled fields: • Quantity • Unit • Price Seller can update any of these values and click Save to apply the changes, or Cancel to discard them.
Error Handling	1. Show a message if the seller tries to save with empty or invalid fields 2. Show confirmation message upon successful update

Use Case Number	SB-UC-0009
Use Case Name	Delete Product Details
Actors	Seller
Preconditions	1. Seller is logged in 2. At least one product exists in the seller's product list 3. Seller is on the Product Page
Description	The seller selects a product from the list and clicks Delete. A confirmation dialog appears with the message: "Are you sure you want to delete this product?" Seller must choose: • Yes, Delete to confirm and remove the product • Cancel to close the dialog without deleting
Error Handling	Display a success or failure message after deletion

Use Case Number	SB-UC-0010
Use Case Name	View and Purchase Products
Actors	Buyer
Preconditions	1. The user is logged in as a Buyer 2. Products are available in the catalog
Description	The buyer navigates to the Products Page to browse and purchase items. The page displays a list of available products along with details such as image, name, quantity, price per unit, and category. The buyer can: • Search for products by name • Filter products by category (All, Vegetables, Fruits) • Click the cart button to open the purchase modal • In the modal, specify the desired quantity • View the unit price and total cost • Click Confirm to proceed with purchase or Close to cancel the action • If user confirms the purchase, another dialog will appear for the user to confirm to proceed with payment
Error Handling	1. Show a message if the entered quantity is invalid (e.g., empty, zero, or exceeds available stock) 2. Show a success message after confirming the purchase

Use Case Number	SB-UC-0011
Use Case Name	Complete Payment via Purchase Modal
Actors	Buyer
Preconditions	1. Buyer has selected a product and confirmed the purchase quantity 2. The purchase confirmation dialog has been accepted 3. The payment modal is triggered
Description	After confirming the product and quantity in the purchase modal, the buyer is shown a payment modal where they must enter their card details to complete the transaction. The form includes: • Card Number • Name on Card • MM/YY (Expiry Date) • CVV The buyer can click Pay Now to finalize the purchase, or Cancel to abort the payment process.
Error Handling	1. Show field-level validation if any entry is missing or invalid 2. Display an error message if payment is declined or fails 3. Show a success message upon successful payment 4. If the buyer clicks Cancel, close the modal and return to the product page without saving any data

Use Case Number	SB-UC-0012
Use Case Name	Seller Updates Bargain Settings
Actors	Seller
Preconditions	1. Seller is logged into their Smart Bargain account 2. The product must exist in the catalog and be listed by the seller
Description	The seller navigates to the Negotiation Page and uses the search or filter option to find a specific product. Once selected, the seller clicks "Update", which opens a modal window displaying the current minimum quantity and price for that product. The seller can then adjust the Minimum Quantity and Minimum Price, and either: • Click "Save" to update the bargain settings • Or click "Cancel" to discard the changes
Error Handling	1. Show validation messages if fields are left empty or non-numeric values are entered 2. If the update fails due to a system error, show a message like: "Unable to save changes. Please try again later."

Use Case Number	SB-UC-0013
Use Case Name	Buyer Initiates Bargain Request
Actors	Buyer
Preconditions	1. Buyer is logged into their Smart Bargain account 2. The selected product is eligible for bargaining (seller has set minimum quantity and price)
Description	The buyer navigates to the Negotiation Page, uses search or filters to find a desired product, and clicks the bargain icon for that item. This opens a bargain modal displaying: • Product image, name, and original price • Seller-defined Minimum Quantity and Minimum Price for Bargain The buyer enters: • Bargain Quantity • Bargain Price As the user inputs values, the system automatically calculates the Total Price (Bargain Quantity × Bargain Price) and displays it. The buyer then clicks "Submit" to send the bargain request to the seller, or "Close" to cancel.
Error Handling	1. Show validation messages if input fields are empty, zero, or invalid 2. If bargain quantity is less than the minimum threshold, show an appropriate message like: "Quantity must meet or exceed the seller's minimum requirement"

Use Case Number	SB-UC-0014
Use Case Name	Seller Reviews and Responds to New Negotiation Request
Actors	Seller
Preconditions	1. A Buyer has submitted a negotiation request for a product 2. The Seller is logged into their Smart Bargain account and has navigated to dashboard
Description	When a buyer submits a negotiation request, the request appears in the Seller Dashboard under the "New Negotiation Requests" table. This table is only visible to the seller and shows the Product Name, Quantity, Proposed Price, and Status (initially set to "Pending Seller Approval") The seller can view and take an action on each record. Once an action is taken, the record is moved to the Negotiation Tasks section for ongoing interaction between buyer and seller. **Note:** Specific seller actions (Accept, Reject, Counter Propose, Cancel) are handled in separate use cases.
Error Handling	1. If any action fails due to system issues, show: "Action failed. Please try again later." 2. Validate changes in quantity and price when using the Counter Propose option

Use Case Number	SB-UC-0015
Use Case Name	Negotiation Dashboard – Ongoing Negotiation Management
Actors	Buyer, Seller
Preconditions	• User logged in as Buyer or Seller • Negotiation moved to Dashboard (ongoing) • User is party to negotiation
Description	Negotiation Dashboard is visible only to involved parties. Both can: • View negotiation record (product, quantity, price, status) • Adjust quantity/price as allowed **Action Rules:** • Buyer proposal: status = Pending Seller Approval • Seller counter: status = Pending Buyer Approval • Negotiation continues until Accept/Reject/Cancel • If Cancelled: no further actions allowed
Error Handling	Show: "This negotiation has been cancelled. No further actions are allowed."

Use Case Number	SB-UC-0016
Use Case Name	Seller or Buyer Accepts Negotiation Request
Actors	Buyer and Seller
Preconditions	1. Negotiation request is visible in the "New Negotiation Requests" table or in the "Negotiation Dashboard" table 2. Seller/Buyer is logged in and has opened the request
Description	**New Negotiation Requests:** The Seller chooses to accept the buyer's proposed quantity and price without changes. The system updates the negotiation record's status to "Active", and the record is moved to the Negotiation Tasks table. No further actions are expected from either party unless the business process requires any changes. **Note:** Initial status of the record in new negotiation request table always remains as "Pending Seller Approval" **Negotiation Dashboard:** The seller or Buyer can accept the record. The system updates the negotiation record's status to "Active" despite of its previous status. No further actions are expected from either party unless the business process requires any changes. **Note:** Records in Cancelled status cannot be accepted in "Negotiation Dashboard"
Error Handling	If user action fails due to a backend error, display appropriate message

Use Case Number	SB-UC-0017
Use Case Name	Seller or Buyer Rejects Negotiation Request
Actors	Buyer and Seller
Preconditions	1. Negotiation request is visible in the "New Negotiation Requests" table or in the "Negotiation Dashboard" table 2. Seller/Buyer is logged in and has opened the request
Description	**New Negotiation Requests:** The Seller chooses to reject the buyer's proposed quantity and price without changes. The system updates the negotiation record's status to "Rejected", and the record is moved to the Negotiation Tasks table. No further actions are expected from either party unless the business process requires any changes. **Note:** Initial status of the record in new negotiation request table always remains as "Pending Seller Approval" **Negotiation Dashboard:** The seller or Buyer can reject the record. The system updates the negotiation record's status to "Rejected" despite of its previous status. No further actions are expected from either party unless the business process requires any changes. **Note:** Records in Cancelled status cannot be rejected in "Negotiation Dashboard"
Error Handling	If user action fails due to a backend error, display appropriate message

Use Case Number	SB-UC-0018
Use Case Name	Seller or Buyer Cancels Negotiation Request
Actors	Buyer and Seller
Preconditions	1. Negotiation request is visible in the "New Negotiation Requests" table or in the "Negotiation Dashboard" table 2. Seller/Buyer is logged in and has opened the request
Description	**New Negotiation Requests:** The Seller chooses to cancel the buyer's proposed quantity and price without changes. The system updates the negotiation record's status to "Cancelled", and the record is moved to the Negotiation Tasks table. **Note:** Initial status of the record in new negotiation request table always remains as "Pending Seller Approval" **Negotiation Dashboard:** The seller or Buyer can cancel the record. The system updates the negotiation record's status to "Cancelled" despite its previous status. Records in Cancelled status cannot be edited.
Error Handling	If user action fails due to a backend error, display appropriate message

Use Case Number	SB-UC-0019
Use Case Name	Seller or Buyer Counter Proposes on Negotiation
Actors	Buyer and Seller
Preconditions	1. Negotiation request is visible in the "New Negotiation Requests" table or in the "Negotiation Dashboard" table 2. Seller/Buyer is logged in and has opened the request
Description	**New Negotiation Requests:** The seller modifies either the proposed quantity, price, or both. The new values are submitted with optional comments. The negotiation status changes to "Pending Buyer Approval", and the buyer is notified of the counter offer via the Negotiation Dashboard. **Negotiation Dashboard:** When a seller submits a counter proposal (i.e., changes the quantity, price, or both), the negotiation status is updated to "Pending Buyer Approval". The buyer can view the updated offer and seller's comments in the Negotiation Dashboard, where they may choose to: • Accept the counter offer • Reject it • Counter Propose again with new terms • Cancel the negotiation The back-and-forth continues until the negotiation is either accepted, rejected, or cancelled by either party.
Error Handling	1. If user action fails due to a backend error, display appropriate message 2. If no change is made to quantity or price, disable the Counter Propose button

Use Case Number	SB-UC-0020
Use Case Name	View Order History
Actors	Buyer and Seller
Preconditions	1. The user is logged in as a Buyer or Seller 2. Orders have been successfully placed through the products page by the Buyer
Description	Users can view order records through the Dashboard → Orders section: A Buyer can only view orders that they have placed. A Seller can view all orders related to products they have listed and sold. The order history displays: • Product Name • Quantity • Total Price • Status • Date on which the order was placed This feature provides transparency and allows users to review and track transaction history at any time.
Error Handling	If there are no orders to display, show a message: "No orders found"

Use Case Number	SB-UC-0021
Use Case Name	Contact Us Form Submission
Actors	Any User (Unauthenticated or Logged-in User)
Preconditions	The user is on the "Contact Us" page.
Description	The user fills in their name, email, and message and clicks the Send Message button. The system validates the fields and saves the message to the database with status set to "open", along with the current timestamp.
Error Handling	If any required field is empty or email is invalid, an inline validation error is shown.

Use Case Number	SB-UC-0022
Use Case Name	View Contact Messages
Actors	Seller
Preconditions	Seller is logged in and navigates to the dashboard. At least one contact message has been submitted.
Description	The system displays a Contact Messages table to the seller, listing all customer messages. Each row shows the customer name, date, and message status. The seller can click on any row to open a modal with full message details (read only), including email, full message content, and options to Reply (placeholder), Close Message, or Exit.
Error Handling	When reply button is clicked a user-friendly message display to advice the user on future implementation

Use Case Number	SB-UC-0023
Use Case Name	Filter Contact Messages on Seller Dashboard
Actors	Seller
Preconditions	Seller is on the dashboard and the contact messages section is visible.
Description	The seller uses filter controls to narrow down contact messages. Filters include Status (Open, Closed) and Date (specific date via calendar picker). The system applies the filters dynamically to update the message list based on the selected criteria.
Error Handling	If no messages match the filters, display "No messages found"

6.2 Activity Diagram

The diagram above shows the working cycle of a Smart Bargain System, identifying the interactions among various stakeholders: Seller, Buyer, Unauthorized User, and the system. The flow is divided into four main columns (swimlanes), one for each type of user or system piece. It would be appropriate to document this diagram in technical or system design document such that visual description of end-to-end process and user interactions in the Smart Bargain System is available. It provides a summary of how authentication, product management, negotiation, and messaging fit together in the platform.

Key Components

1. Seller Section
    - Sellers can register, log-in, update products, view order and negotiation history, and can respond to buyer messages.
    - Sellers can make or accept counter proposals during negotiations.
2. Buyer Section
    - Register, log in, browse products, buy products, and participate in negotiations (bargain, bid, or send messages to sellers).
    - View the order and negotiation history in the dashboard. Make or accept counter proposals.
3. Smart Bargain System
    - Handles authentication, stores credentials, validates logins, and manages all data storage and updates.
    - Acts as the central hub for all data exchange between buyers and sellers.
4. Unauthorized User
    - Limited to sending messages to sellers, without access to other system functionalities.

Flow Summary

1. Buyers and sellers, are required to register and login to use the system.
2. Back and forth counter offers and messages are exchanged between buyers and sellers.
3. The system monitors and updates actions to make the data consistent and the workflow suitable for all types of users.

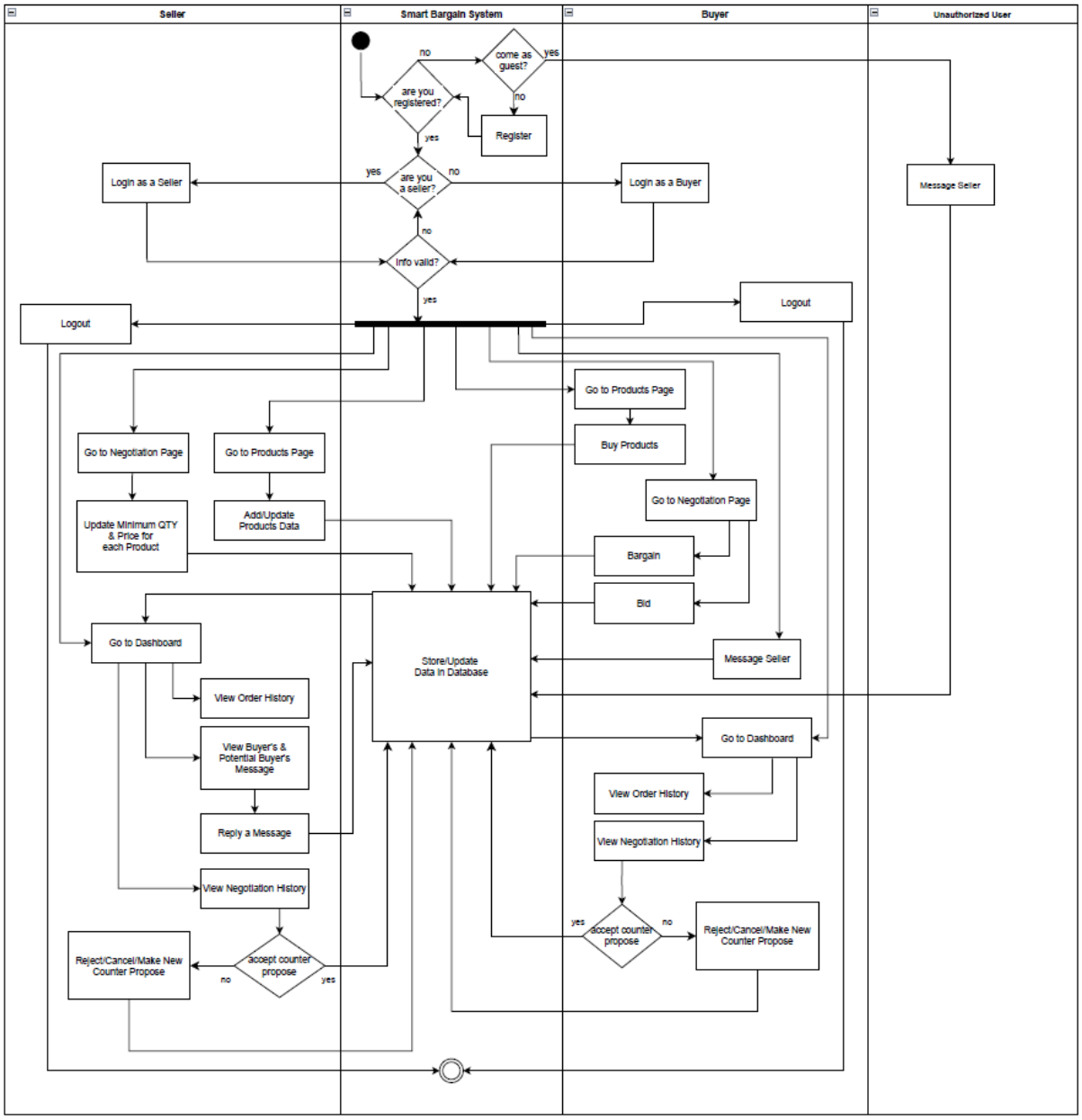

Figure 3: Activity Diagram for the Overview of Smart Bargain System

Note: For a better view of the diagram, access the original pdf file thru this link:
Overview - Smart Bargain System - Activity Diagram

6.2.1 Activity Diagram for Smart Bargain Negotiation Feature

The negotiation based feature of the Smart Bargain system is modeled in terms of end-to-end flow through the Buyer, Seller and the Smart Bargain System. The diagram is divided into three main lanes:

- Buyer: Initiates bargain requests.
- Smart Bargain System: Store data and coordinate process.
- Seller: Reviews, responds, and manages incoming bargain requests.

Buyer Actions

- View Product Catalog
- Decision Point – Satisfied with Price?
  - If yes: Proceeds directly to purchase.
  - If no: Considers making a bargain.
- Set Bargain Quantity and Price
- Add Comment (Optional)
- Submit Bargain Request
- Cancel Bargain Request (Optional)
- Review Counter Proposal from Seller
- Purchase Product if terms are accepted

Smart Bargain System Actions

- Data Storage: Every action (submit, accept, reject, counter, or cancel) is logged and updated in the system database to ensure process integrity and traceability.

Seller Actions

- View Bargain Request in Dashboard
- Review and Check Buyer's Bargain Request
- Add Comment (Optional)
- Decision Points:
  - Reject Bargain Request

  - Accept Bargain Request
  - Make Counter Proposal
  - Cancel Bargain Request
  - Review Bargain Request Again
- Reject Bargain Request (if rejected or cancelled, process ends)

Negotiation Loop

The process supports iterative bargaining: buyers and sellers can exchange counter proposals and comments until either party accepts, rejects, or cancels the negotiation. All actions and decisions are tracked within the system for transparency

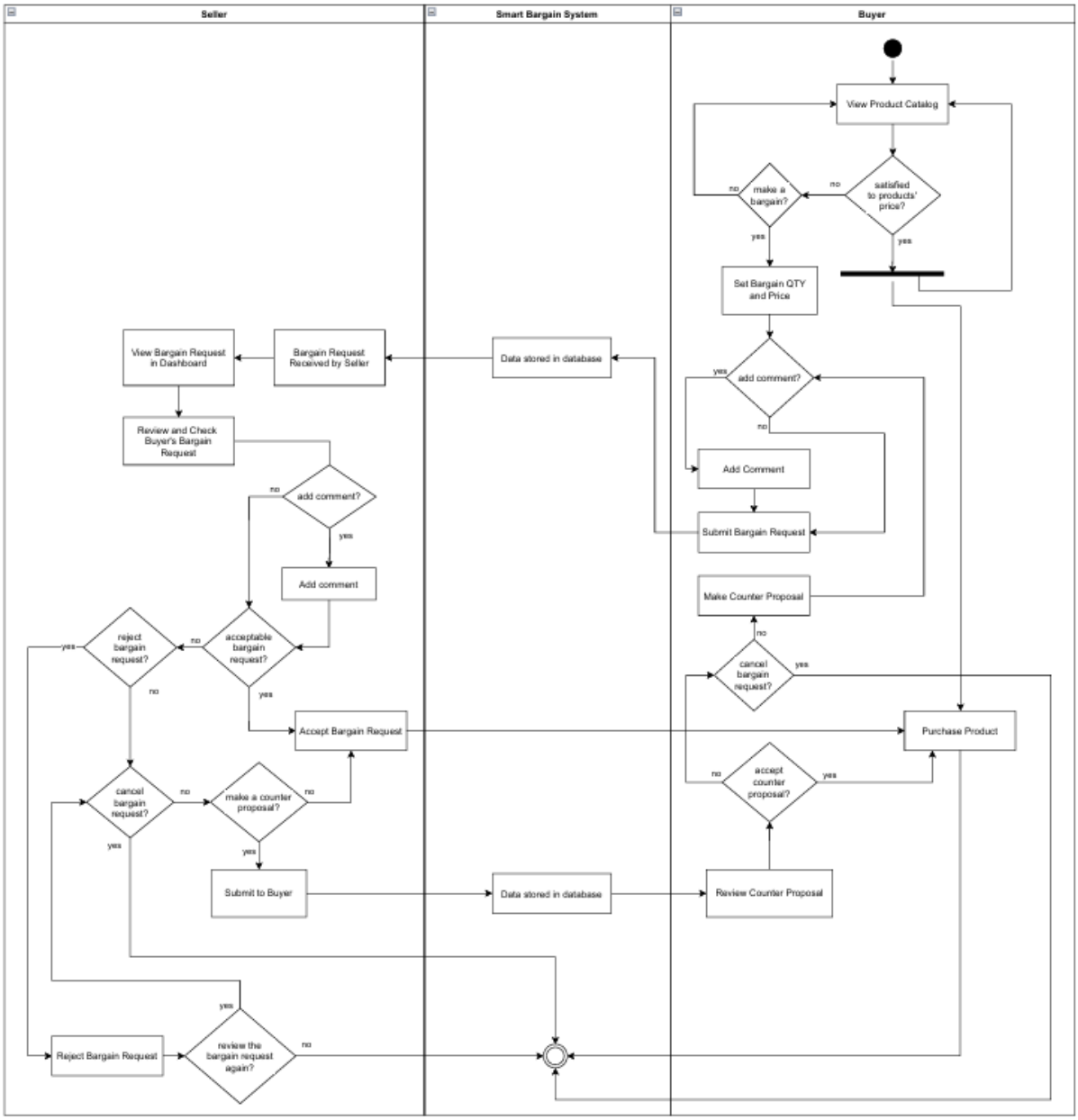

Figure 4: Activity Diagram for Smart Bargain Negotiation Feature

Note: For a better view of the diagram, access the original pdf file thru this link:
Smart Bargain System - Activity Diagram for Negotiation Feature

6.3 Seller Interface Diagram

The seller interface diagram shows how the sellers can explore the Smart Bargain and its features. It describes interaction of sellers with the system to handle product, deals, dash-board analytics etc. The diagram illustrates the seller's journey by mapping links between entering the system, managing products, negotiations, and sales activities and so on. Every action is associated with navigation pointers (buttons, icons) and confirmatory steps to facilitate a seamless operation experience.

Main Components

- **Home Page:** Features header and banners which maps to various pages. Also features a footer with contact information. Users can scroll up and down to navigate through the main sections.
- **Product Management:**
  - Click 'Add New Product' button to add new products.
  - Click catalog icon to open product modal where seller can edit, update, or delete existing products by clicking corresponding action buttons.
  - Confirm actions.
- **Negotiation Management:**
  - Open the Negotiation Page to check buyer's bargain requests and messages.
  - Respond to buyers' messages, accept or refuse price negotiations, or send out price counter offers by quantity or amount.
  - Cancel negotiations if necessary.
- **Dashboard and Tables:**
  - For a full tracking lists of all of: order history, negotiation, and messages tables.
- **Future Work:** The diagram shows future functionalities as well. It implied that the seller interface is dynamic and constantly evolving.

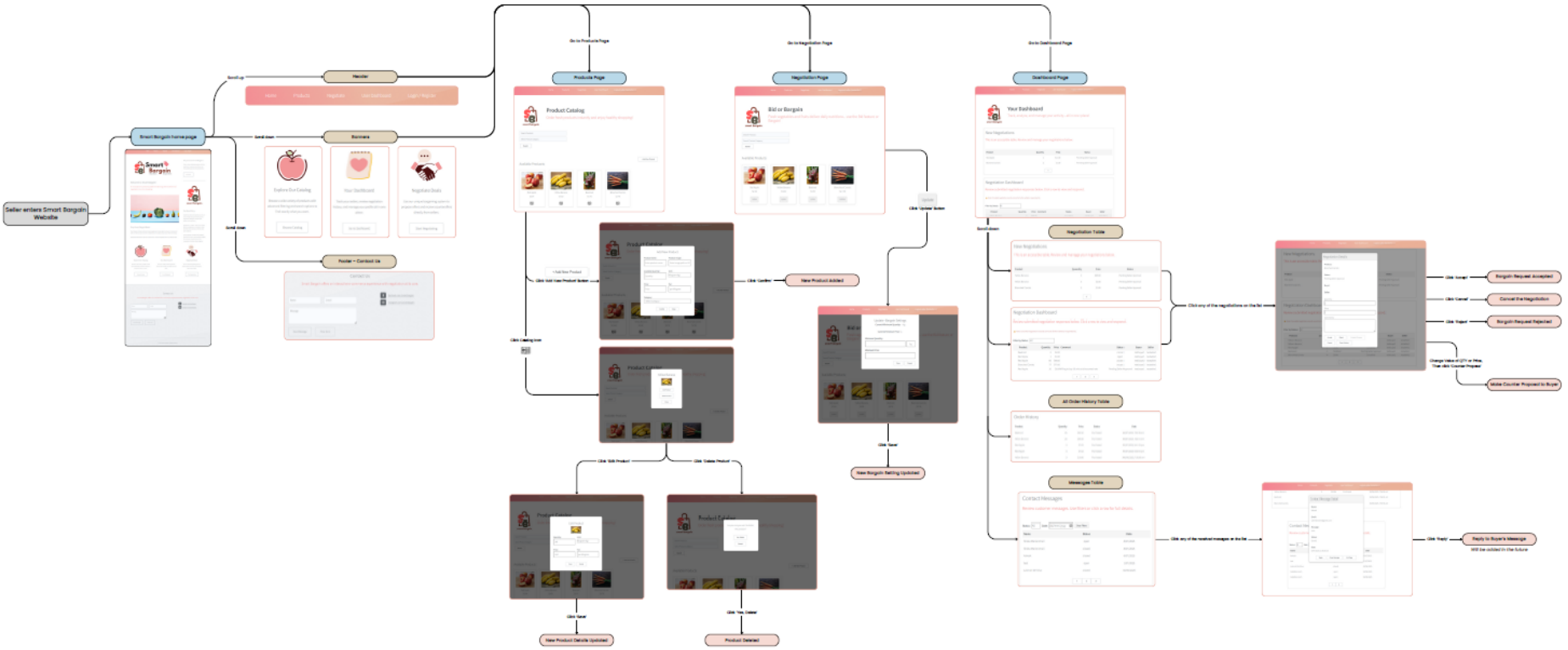

Figure 5: Seller Interface Diagram of Smart Bargain

Note: For a better view of the diagram, access the original pdf file thru this link:
Smart Bargain System - Seller Interface Diagram

6.4 Buyer Interface Diagram

The Buyer Interface Diagram presents a representation of the navigation functionality and interactivity accessible for buyers or potential buyers through the Smart Bargain market. It describes how users browse items, make purchases, start and manage negotiations. The map is divided into key stages of the buyer's journey. This include entering the market and trying on products, and negotiating and completing a purchase. Every step is linked to clear navigation actions (buttons or icons) and confirmation messages, to lead the user through the way.

Main Components

- **Home Page:** Features header and banners which maps to various pages. Also features a footer with contact information. Users can scroll up and down to navigate through the main sections.
- **Product Interaction:**
  - Browse available products on the Products Page.
  - Click 'Cart Icon' icon to start purchase.
  - Click 'Bargain' button to initiate a bargain.
- **Negotiation Management:**
  - Go to Negotiation Page to view all negotiations.
  - From the negotiation dashboard, select any negotiation from the list to view details.
  - Submit new bargain requests or respond to seller counter-proposals.
  - Confirm orders and proceed to payment.
  - Cancel negotiation.
- **Order and History Tracking:**
  - View order history and negotiation records in the Dashboard Page.
  - Access the negotiation dashboard for detailed tracking.
- **Negotiation Actions:**
  - Accept, reject, or make counter proposals by adjusting quantity or price.
  - Confirm, submit, or cancel actions as needed.

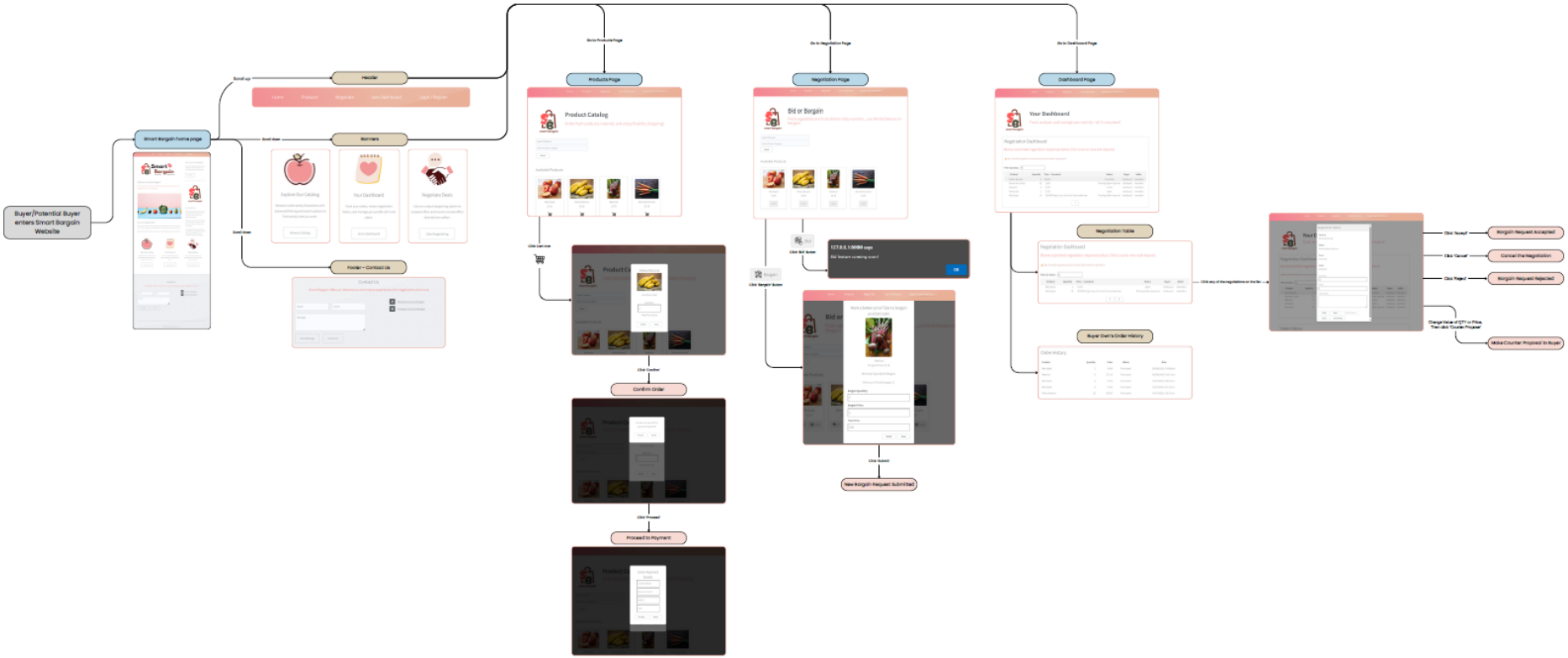

Figure 6: Buyer Interface Diagram of Smart Bargain

Note: For a better view of the diagram, access the original pdf file thru this link:
Smart Bargain System - Buyer Interface Diagram

6.5 Sequence Diagrams

The Buyer and Seller sequence diagrams show the main ways that users interact with the Smart Bargain system. The buyer flow includes starting negotiations (which are checked against the seller's minimum amount), view all negotiation records, responding to negotiations by accepting, rejecting, canceling, or counter-proposing , direct messaging sellers and view own order history. The seller flow includes logging in, adding, editing, and removing products, managing bargain settings, viewing direct messages, managing ongoing negotiations , handling new negotiation requests (accept, reject, cancel, or counter-propose), and view all order history.

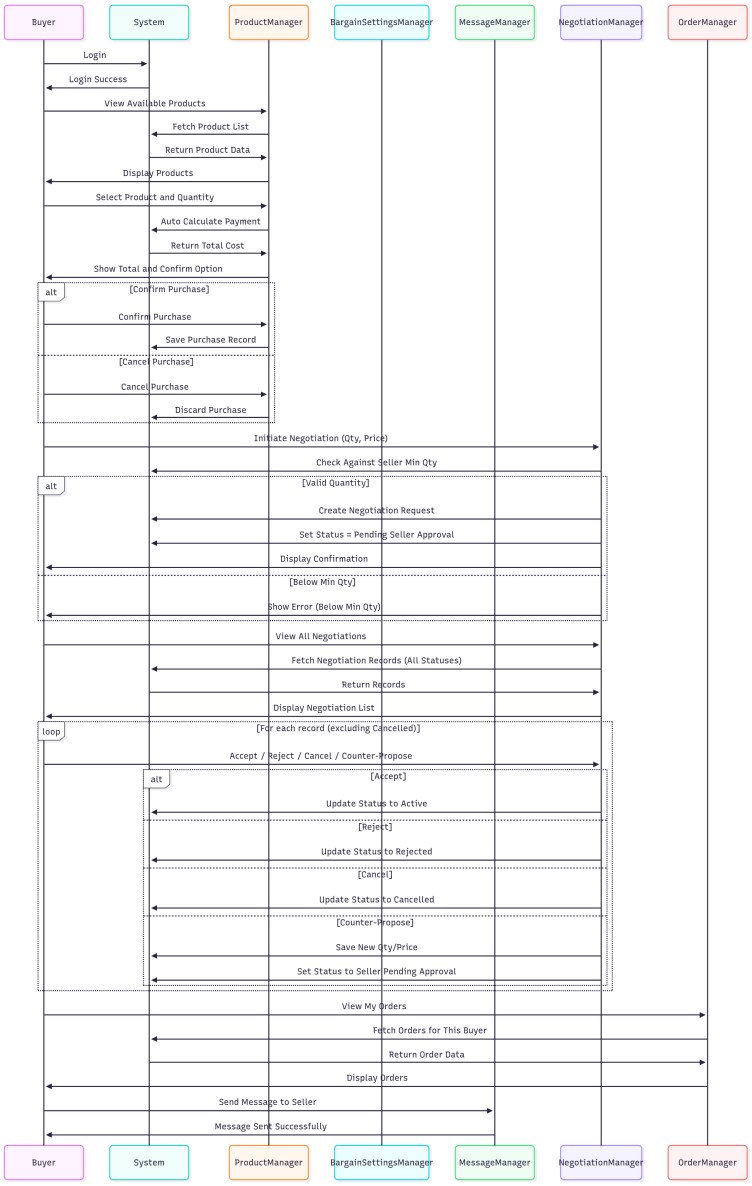

Figure 7: Buyer Sequence Diagram of Smart Bargain
Note: For a better view of the diagram, access the original pdf file thru this link:
Smart Bargain Buyer Sequence Diagram

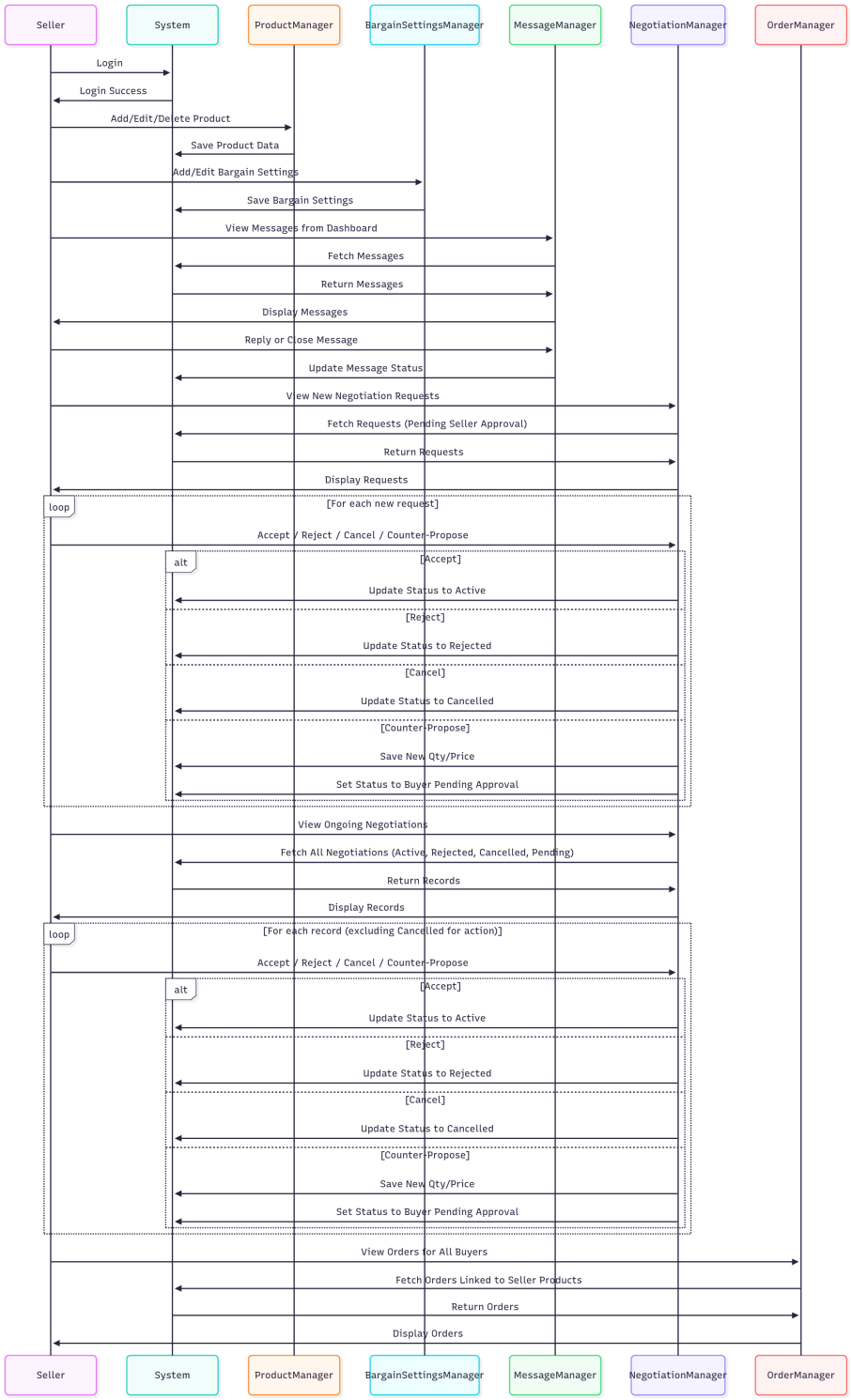

Figure 8: Seller Sequence Diagram of Smart Bargain

Note: For a better view of the diagram, access the original pdf file thru this link:

Smart Bargain Seller Sequence Diagram

6.6 Class Diagram

The Smart Bargain class diagram is the class diagram that explains the design of a software system that would allow product negotiations between buyers and sellers to take place. The diagram illustrates the core classes, attributes, and methods, and the connections between classes in a modular, object-oriented design. This class diagram represents an organized system for controlling users, products, bargaining process, and user communication system of an e-commerce negotiation platform.

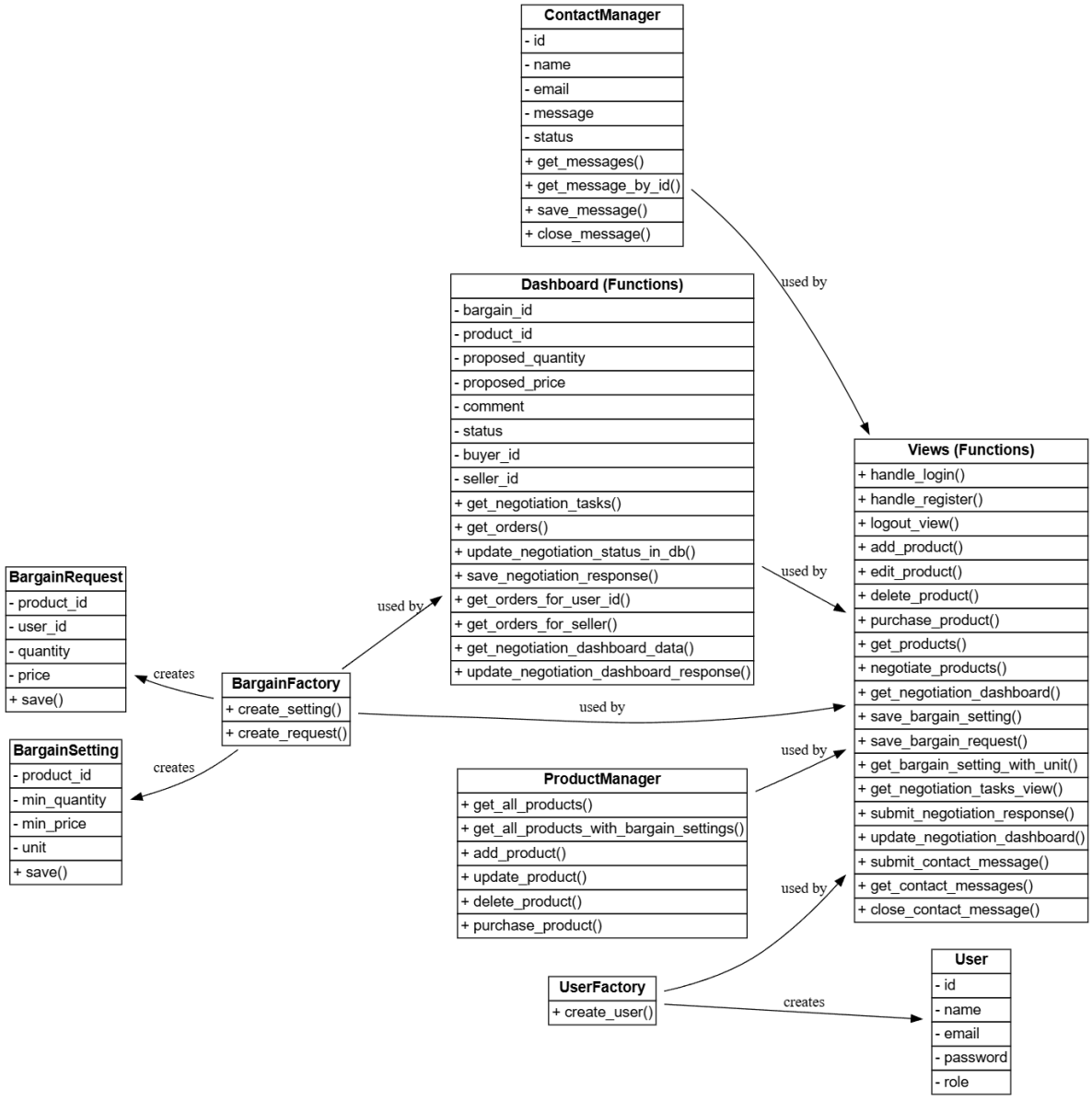

Figure 9: Class Diagram of Smart Bargain
Note: For a better view of the diagram, access the original pdf file thru this link:
Smart Bargain - Class Diagram

7 Agile Sprint Development

This chapter documents the Agile sprint process for the project, referencing actual sprint features, durations, and the use of GitHub for sprint management.

7.1 Sprint Planning and Goals

GitHub Projects was used to implement the sprint planning. On this platform, Smart Bargain features were organized and tracked for each sprint. Below are the details for each sprint, including the features and their scheduled durations:

Sprint	Features
Sprint 1	Home Page | Initial Feature Design
	Product Page | Initial Feature Design
	Negotiation Page | Initial Feature Design
	Dashboard Page | Initial Feature Design
	Product and Negotiation Page | Search and Filter Feature for Catalog and Negotiation
	All pages | Color coordinate
	All pages | Add and replace icons where necessary
	Login Page | Initial Feature Design
Sprint 2	All pages | Design Fine tuning
	Session Management | Apply functionality as per Logged in User
	All pages | Add Logo
	Messaging Feature | Contact Us Message Handling (Seller Dashboard Integration)
	Bargain Function | Introduce Qty threshold
Sprint 3	Product Page | Error Handling
	Payment Modal | Enhance Payment Modal to Support Multiple Payment Methods
	Product Page | Product catalog – Advanced filters
	Negotiation Page | Implement Bargain Feature

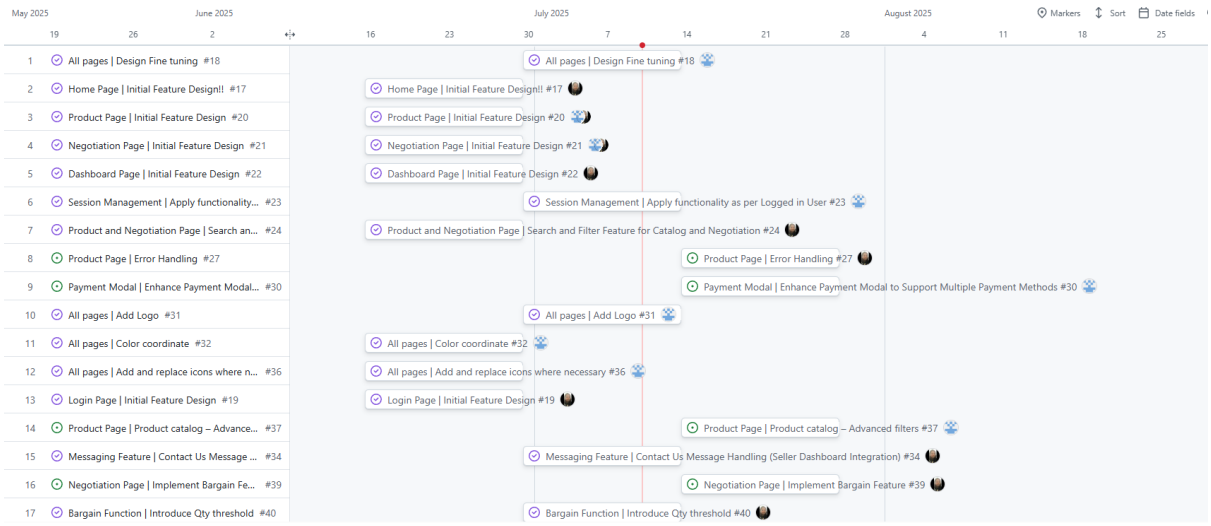

Figure 10: Sprints Duration as shown in Github Projects

Sprints Duration Overview

The project is structured into three consecutive sprints. Each sprint are planned to be done in two weeks time. These sprints are designed to organize and pace the team's work, ensuring steady progress and clear milestones.

- Sprint 1 (June 16 – June 29)
- Sprint 2 (June 30 – July 13)
- Sprint 3 (July 14 – July 27)

Sprint Board Reference:
All sprints and their features were managed and tracked using GitHub Projects which can be accessed thru this link: Github Projects - Sprints Timeline

7.2 Tools and Technologies

The following tools and technologies supported the Agile development in Smart Bargain Project:

- **GitHub Projects**: Plan and track tasks and display progress.
- **Git:** For source code version control.
- **VS Code:** Developmet environment.
- **Communication Tools:** MS Teams - used for team discussions and stand-ups.
- **Online Meeting Platforms** MS Teams for scrum meeting and reviews.

- **JIRA:** Coming soon. Migration of various project data from Github Projects to JIRA will be done in the future to explore useful features.

7.3 Tasks and Assignments

Tasks were assigned to team members based on expertise and availability. Every action taken are well-coordinated with each member. Pull requests were done through constant communication. All of the changes done to the codes are tracked, updated, and documented in GitHub Projects. Each sprint item was broken down into actionable tasks, ensuring accountability and clarity. The assignments and statuses were updated in real time on the GitHub board which can be accessed on the links provided on the table below.

Feature/Task	Assigned To	Status	URL
Home Page | Initial Feature Design	Lakmali	Done	1
Product Page | Initial Feature Design	Lakmali/Shiela	Done	2
Negotiation Page | Initial Feature Design	Lakmali/Shiela	Done	3
Dashboard Page | Initial Feature Design	Lakmali	Done	4
Product and Negotiation Page | Search and Filter Feature for Catalog and Negotiation	Lakmali	Done	5
All Pages | Color Coordinate	Shiela	Done	6
All pages | Add and replace icons where necessary	Shiela	Done	7
Login Page Initial Feature Design	Lakmali	Done	8
All pages | Design Fine tuning	Shiela	Done	9
Session Management | Apply functionality as per Logged in User	Shiela	Done	10
All Pages | Add Logo	Shiela	Done	11
Messaging Feature | Contact Us Message Handling (Seller Dashboard Integration) | Add Logo	Lakmali	Done	12
Functionality Integration	Lakmali	Done	13
Bargain Function | Introduce Qty threshold	Lakmali	Done	14
Product Page | Error Handling	Lakmali	To Do	15
Payment Modal | Enhance Payment Modal to Support Multiple Payment Methods	Shiela	To Do	16
Product Page | Product Catalog Advanced Filters	Shiela	To Do	17
Negotiation Page | Implement Bargain Feature	Lakmali	To Do	18

Table 2: Task Assignments with URLs

7.4 Scrum Meeting (MOM)

From 20th of June until 9th of July 2025, the Scrum team had organized several scrum meetings to make the "Smart Bargain" project design and development. Each of these sessions was chaired by the Scrum Master and aimed at UI/UX, functionality, integration achievement, etc. Below are the tables for the MOM details for each of the scrum meeting:

Scrum Meeting 1	
Date	June 20, 2025
Attendees	Shiela, Lakmali
Facilitator	Lakmali
Agenda	1. Introduce project and team roles 2. Review initial requirements for Home Page 3. Discuss design inspirations 4. Set timeline for Home Page design
Discussion Highlights	Team aligned on Home Page objectives and style; wireframe ideas exchanged; agreed on minimalistic approach for initial design.
Action Items	Lakmali to create Home Page wireframe; Shiela to gather reference designs for review.

Scrum Meeting 2	
Date	June 23, 2025
Attendees	Shiela, Lakmali
Facilitator	Lakmali
Agenda	1. Review Product Page requirements 2. Discuss integration with Home Page 3. Identify the issues 4. Assign design responsibilities
Discussion Highlights	Product modal has some issues; discussed layout for product cards; integration points with Home Page clarified.
Action Items	Shiela to draft Product Page layout. Lakmali and Shiela to fix bugs and add feature on the modal.

Scrum Meeting 3	
Date	June 24, 2025
Attendees	Shiela, Lakmali
Facilitator	Lakmali
Agenda	1. Present Negotiation Page concept 2. Discuss user flow for negotiation 3. Identify UI elements for negotiation 4. Plan for initial feature implementation 5. Review Dashboard Page requirements 6. Plan user role-based dashboard views 7. Assign widget design tasks
Discussion Highlights	Negotiation workflow mapped out; clarified steps for user offers and counter-offers; decided to use modal dialogs for negotiation actions. Identified the data will be shown on dashboard; agreed to the current views.
Action Items	Shiela to design negotiation modal; Lakmali to implement initial negotiation logic. Lakmali and Shiela to fix bugs and add feature on the modal.

Scrum Meeting 4	
Date	June 24, 2025
Attendees	Shiela, Lakmali
Facilitator	Lakmali
Agenda	1. Plan search and filter features for catalog 2. Discuss integration with negotiation system 3. Review filter UI options 4. Detection of issues/bugs
Discussion Highlights	Chose filter sidebar design; clarified catalog search requirements; to review and identify bugs.
Action Items	Lakmali to fix bugs and add feature on the modal; Shiela to design filter sidebar UI.

Scrum Meeting 5	
Date	June 25, 2025
Attendees	Shiela, Lakmali
Facilitator	Lakmali
Agenda	1. Review color coordination across all pages 2. Discuss accessibility standards 3. Agree on color palette 4. Plan for color updates in UI
Discussion Highlights	Finalized color palette for the application; discussed contrast and accessibility; agreed on color usage guidelines.
Action Items	Shiela to update UI components with new colors.

Scrum Meeting 6	
Date	June 26, 2025
Attendees	Shiela, Lakmali
Facilitator	Lakmali
Agenda	1. Review the Logo for Smart Bargain 2. Discuss how to implement consistency of Smart Bargain branding
Discussion Highlights	Agreed on the logo design and branding
Action Items	Shiela to implement the discussed logo and branding for Smart Bargain

Scrum Meeting 7	
Date	June 30, 2025
Attendees	Shiela, Lakmali
Facilitator	Lakmali
Agenda	1. Review Login Page design 2. Test login functionality 3. Discuss error handling for login 4. Plan for multi-user support
Discussion Highlights	Login page design approved; login tested with valid/invalid users; discussed improvements for error messaging.
Action Items	Lakmali to refine error messages; Shiela to test multi-user login scenarios.

Scrum Meeting 8	
Date	July 1, 2025
Attendees	Shiela, Lakmali
Facilitator	Lakmali
Agenda	1. Review design fine-tuning tasks 2. Discuss points for improvement 3. Plan further UI enhancements 4. Assign fine-tuning responsibilities
Discussion Highlights	User feedback highlighted spacing issues; typography improvements discussed; prioritized UI fine-tuning tasks.
Action Items	Shiela to adjust spacing and fonts; Lakmali to review and approve changes.

Scrum Meeting 9	
Date	July 2, 2025
Attendees	Shiela, Lakmali
Facilitator	Lakmali
Agenda	1. Discuss session management requirements 2. Review user login/logout flows 3. Plan session timeout handling 4. Assign session management tasks
Discussion Highlights	Defined session timeout policy; mapped login/logout user flows; planned for persistent user sessions.
Action Items	Lakmali to implement session handling; Shiela to test session timeout scenarios.

Scrum Meeting 10	
Date	July 4, 2025
Attendees	Shiela, Lakmali
Facilitator	Lakmali
Agenda	1. Plan logo integration across all pages 2. Review logo placement 3. Test logo visibility on different backgrounds 4. Assign logo update tasks
Discussion Highlights	Logo placement standardized; minor adjustments suggested.
Action Items	Shiela to update logo assets.

Scrum Meeting 11	
Date	July 5, 2025
Attendees	Shiela, Lakmali
Facilitator	Lakmali
Agenda	1. Review messaging feature requirements 2. Plan contact us message handling 3. Integrate messaging with seller dashboard 4. Discuss logo display in messaging area
Discussion Highlights	Messaging feature requirements finalized; integration points with seller dashboard mapped; logo placement in messaging area confirmed.
Action Items	Lakmali to implement message handling logic; Shiela to design messaging UI with logo.

Scrum Meeting 12	
Date	July 7, 2025
Attendees	Shiela, Lakmali
Facilitator	Lakmali
Agenda	1. Review overall functionality integration 2. Identify integration issues 3. Test cross-feature workflows 4. Plan for final integration testing
Discussion Highlights	Cross-feature workflows tested; minor integration bugs found; agreed on checklist for final integration.
Action Items	Lakmali to fix integration bugs.

Scrum Meeting 13	
Date	July 9, 2025
Attendees	Shiela, Lakmali
Facilitator	Lakmali
Agenda	1. Discuss bargain function requirements 2. Plan quantity threshold logic 3. Review UI for bargain offers 4. Assign bargain feature tasks
Discussion Highlights	Quantity threshold logic for bargains defined; UI for bargain offers reviewed; agreed on validation rules for offers.
Action Items	Lakmali to implement quantity threshold logic; Shiela to update bargain offer UI and validation.

7.5 Sprint Review & Retrospective

Sprint 1: Review

Completed Features	Initial design and implementation of the Home Page, Product Page, Negotiation Page, Dashboard, and Login Page.
	Search and filter functionality for catalog and negotiation pages.
	UI enhancements: color coordination, icon updates, and accessibility improvements.
Feedback	Minimalistic design for Home Page approved.
	Product Page layout and modal integration discussed and iteratively improved.
	Negotiation workflow and modal dialogs mapped out and validated.
	Dashboard requirements clarified, with role-based views agreed upon.
	Color palette and accessibility standards finalized.
Challenges	Product modal encountered issues requiring multiple bug fixes.
	Integration points between pages needed clarification and further refinement.
	Some UI enhancements (e.g., icon replacement) identified for future sprints.

Sprint 1: Retrospective

What Went Well	Strong alignment on design direction and key page objectives.
	Effective collaboration in wireframing and UI decision-making.
	Quick feedback cycles enabled rapid iteration and consensus.
Areas for Improvement	More thorough initial bug detection could have reduced rework on modals.
	Earlier clarification of integration points between features would have streamlined development.
	Need for clearer documentation of action items and responsibilities.
Action Items	Implement a more structured bug tracking and resolution process.
	Schedule dedicated integration review sessions mid-sprint.
	Improve documentation of design decisions and task assignments.

Sprint 2: Review

Completed Features	UI fine-tuning across all pages based on user feedback (spacing, typography).
	Session management: user login/logout flows, session timeout policy, and persistent sessions.
	Logo integration and standardization across the application.
	Messaging feature: requirements finalized, integration with seller dashboard mapped, and UI updated with branding.
	Bargain function: quantity threshold logic defined, UI for bargain offers reviewed and updated.
Feedback	Positive feedback on improved UI consistency and branding.
	Session management flows tested and validated for robustness.
	Messaging feature and bargain logic presented, with validation rules agreed upon.
Challenges	Minor integration bugs identified during cross-feature workflow testing.
	Some adjustments to logo placement and messaging UI required post-demo.

Sprint 2: Retrospective

What Went Well	Responsive adaptation to feedback, especially for UI fine-tuning.
	Clear division of responsibilities for session management and branding tasks.
	Successful integration of messaging and bargain features with existing dashboards.
Areas for Improvement	Integration testing will be done on early stages of sprints to catch cross-feature issues sooner.
	More proactive planning for branding consistency across new features.
	Enhanced coordination needed for simultaneous UI and logic updates.
Action Items	Start integration testing in parallel with feature development.
	Establish branding guidelines to streamline future UI work.
	Hold regular sync-ups for teams working on interdependent features.

7.6 Backlogs

The product backlog is a dynamic document that describes everything that might be needed in the product and is understood to do so by the stakeholders. The backlog is held openly on GitHub Projects and continuously reviewed and reprioritized toremain flexible to stakeholder demand and changes in the market. This approach is designed to enable flexibility, continuous value and a best practices alignment to Agile.

Current Backlog Items

- **Multi-language user access:** The platform must be made available for a broad multilingual user community.
- **Advanced analytic dashboard:** Offer an advanced reporting and insights on buyer and supplier end that include sales trends, customer behavior, inventory analytics etc.
- **User profile personalization:** Enable user to customize their profile, to define preferences and notifications.
- **Integration with more payment gateways:** Add more currencies and payment gateways like PayPal, Stripe, mobile wallets, etc. for more usability.
- **Advanced order management:** Provide tools for batch order management, order adjustments, and intricate order flow management.
- **Express checkout:** Simplify the purchase experience for repeat buyers, decreasing cart abandonment rates, and increasing conversion rates.
- **Improved product search and filter:** Upgrade search algorithms, screen advanced filters (for example, by region, fresh, certified), and add auto complete suggestions.
- **Inventory/stock tracking:** Real-time inventory updates and low stock alerts help to prevent overselling and improve the accuracy of fulfilment.
- **Customer Subscription:** Let customers subscribe to regular deliveries of products, such as a weekly produce box, to create regular revenue.
- **Mobile-responsive and app-based access:** Make the user experience as smooth and integrated across devices or consider building a mobile app.
- **Knowledge center and help resources:** Provide tutorials, crop-specific tipsheets, FAQs and details about customer support to help users and establish trust.

- **Customer loyalty and rewards program:** Apply points, discount or refer-rebate to help customer loyalty.
- **Recommendation system:** Build a recommendation system by following user behavior and purchase history.
- **Rating / review system:** Both buyers and sellers to be able to rate and review the transaction to each other, giving more visibility and trust.
- **Vendor/supplier management:** Provide sellers tools to manage offers. Also, track performance and communication flow with buyers.
- **Logistics and delivery tracking:** Real-time shipment tracking and delivery alerts will be embedded to the system.
- **Bug fixes and technical debt:** Triage and fix all open bugs and refactor any portion of our codebase to continue to increase platform stability and scale.

To keep the backlog well-maintained and prioritixzed, the team makes sure that high-value features and improvements are ready for upcoming sprints. This framework continually drives innovation, user happiness, and the platform's future potential.
Note: All sprint features, tasks, and progress tracking is done through the GitHub Project Board so everything is clear and accountable as each particular feature is being developed.
Reference:
Backlogs are managed and tracked using GitHub Projects which can be accessed thru this link: Github Projects - Backlogs

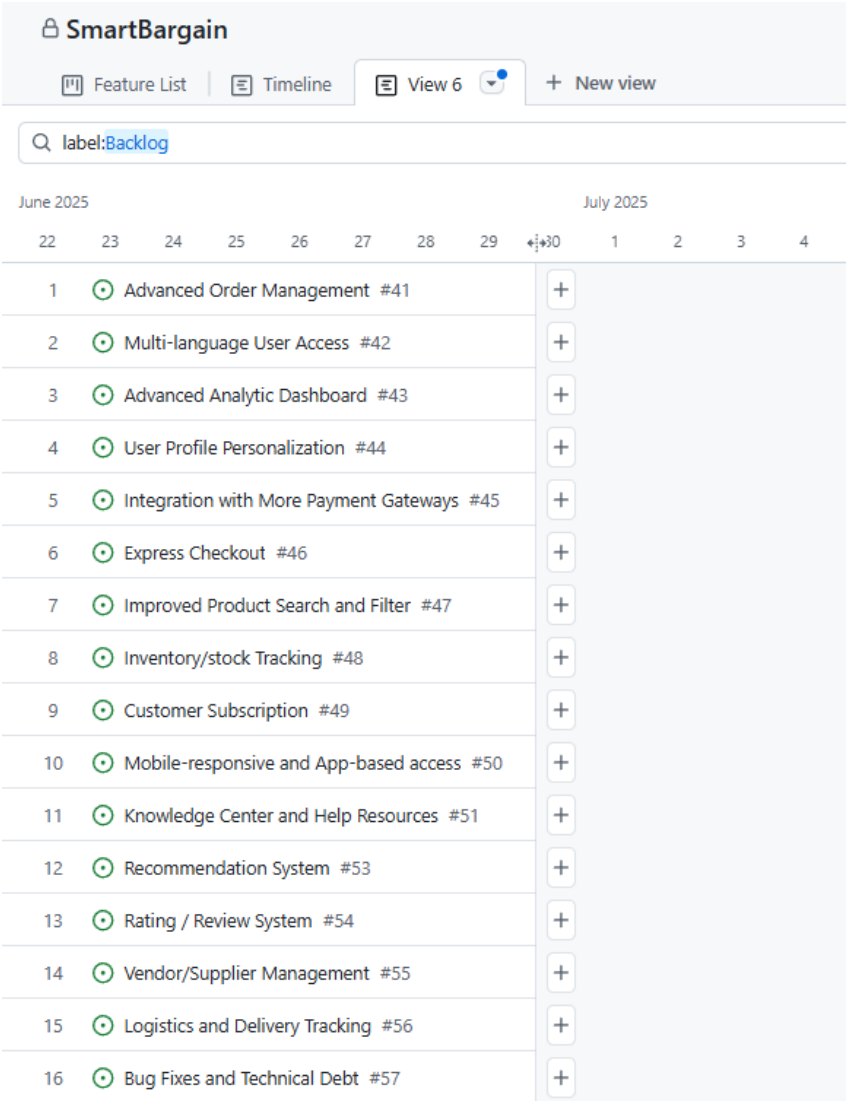

Figure 11: Backlogs as shown in Github Projects

8 Cost Estimation

The following chapter describes a thorough assessment of the cost estimation in the Smart Bargain project. This includes cost, time, resource, and tools and technology estimation. The methodology mirrors industry-recognized best practices for software projects' budget and resource planning making it both transparent and precise for the stakeholders. Smart Bargain is an open source project and considers economy prices of resources. The estimated cost is in the range of industry benchmarks for web-applications of a similar kind, and the stack allows for both scalability and low initial investment. Once the project scales up, ongoing costs will mostly amount for more feature development or cloud infrastructure, plus potential growth of the team or another tool.

8.1 Time and Resource Allocation

Estimation Approach

The project uses a bottom-up estimation method, breaking down all major features and tasks for a granular and realistic projection of effort and resources. This approach is highly known for its precision, especially in Agile software development, where work is clearly outlined and measured.

Project Phases and Time Allocation:

Phase	Estimated Effort (Person-Days)	Key Activities
Requirements & Planning	5	Stakeholder meetings, requirements gathering, backlog setup
Design (UI/UX, Architecture)	5	Wireframes, UI/UX mockups, architectural planning
Core Development (Sprint 1 & 2)	20	Homepage, Product Catalog, Negotiation, Dashboard, Login, Search/Filter
UI Enhancements & Branding	4	Color coordination, accessibility, logo, branding
Session Management & Security	3	Login/logout, session timeout, role-based access
Messaging & Bargain Features	4	Direct messaging, bargain logic, integration
Integration & Testing	5	Cross-feature testing, bug fixes, user feedback
Documentation & Deployment	2	User guides, deployment scripts, project reports

Table 3: Time and Resource Allocation by Project Phase

Total Estimated Effort:

The total estimated effort is **43 person-days**. This means that for a two-member team, there is approximately 3–4 weeks of full-time work aligning to sprint durations.

Resource Allocation Best Practices:

- Tasks are divided based on expertise
- Regular sprint reviews and retrospectives
- Non-coding activities are explicitly allocated to avoid quality or schedule risks.

8.2 Tools/Technology Costs

Development Tools & Platforms:

Tool/Technology	Purpose	Cost Estimate (NZD)
GitHub	Code repository, project management	Free (Student/Education tier) or ~$0–$10/month
Visual Studio Code	Primary code editor	Free
Django (Python Framework)	Backend	Free (Open Source)
SQLite/MySQL	Database	Free (Open Source)
MS Teams	Communication	Free (Education)
Overleaf	Collaborative documentation	Free (Basic tier)
OneNote	Meeting notes, documentation	Free
JAM	Brainstorming, collaboration	Free (Basic)
JIRA (planned migration)	Advanced Agile project management	~$0–$14/user/month (may be free for small teams/education)
Cloud Hosting (PythonAnywhere, planned)	Deployment & hosting	Free–$10/month (starter tier)
Third-Party Libraries	UI, icons, accessibility enhancements	Free (Open Source)

Table 4: Tools and Technology Cost Estimates

Key Notes:

- The project leverages open-source and educational/free-tier tools to minimize direct technology costs.
- The primary expenses are anticipated for future scaling (e.g., cloud hosting, advanced project management tools).
- No proprietary software or paid licenses are required for the MVP phase, keeping technology costs minimal.

Indirect and Hidden Costs:

- Team members use personal or institution-provided equipment, so no additional hardware costs are incurred.
- Maintenance, technical debt, and knowledge transfer are recognized as ongoing indirect costs, but are not significant at the MVP stage.

8.3 Estimated Cost for Backlog Implementation

As Smart Bargain grows as a platform, the backlog should include incremental resource requirements, time, and any other investment that the business may need to make in order to enable new technologies or services. This section also lists other backlog features that should facilitate strategic prioritization and visible project management. The backlog right now is doable in development. The direct cost of technology is also low for some product but some features may incur implementation costs. Order of project selection, and staged execution can be a good way to manage synchronization between the implementation backlog vs. available resources vs. strategic objectives. This proactive approach to monitoring ensures that Smart Bargain is lean, scalable to any size, and responsive to fluctuating user and business needs.

Backlog Item	**Estimated Effort (Person-Days)**	**Additional Tools/Costs (NZD)**
Multi-language user access	8–12	Translation management tools (e.g., Crowdin, Lokalise), professional translation services ($200–$500 per language)
Advanced analytic dashboard	10–15	BI libraries (e.g., Chart.js, D3.js), integration with analytics platforms (Google Analytics, Power BI), cloud compute/storage ($0–$50/month)
User profile personalization	5–7	Minor database upgrades; no significant tool cost
Integration with more payment gateways	7–10	Payment gateway fees (Stripe, PayPal: $0–$50/month depending on usage), compliance and security upgrades
Advanced order management	8–12	Workflow/process management libraries; no significant tool cost

Backlog Item	**Estimated Effort (Person-Days)**	**Additional Tools/Costs (NZD)**
Express checkout	4–6	Minor third-party API integration; no significant tool cost
Improved product search and filter	7–10	Search libraries (e.g., Elasticsearch: open source or $0–$30/month for managed services)
Inventory/stock tracking	5–8	Real-time database/notification services; no significant tool cost
Customer subscription	6–9	Recurring payment integration; possible additional gateway fees
Mobile-responsive and app-based access	12–18	Mobile frameworks (React Native: free), App Store/Play Store fees ($25–$100 one-time)
Knowledge center and help resources	4–6	Documentation tools (e.g., GitBook: free/basic tier)
Customer loyalty and rewards program	6–9	Integration with rewards APIs; no significant tool cost
Recommendation system	10–15	Machine learning libraries (scikit-learn, TensorFlow: free), possible cloud compute costs ($0–$30/month)
Rating / review system	4–6	No significant tool cost
Vendor/supplier management	6–9	Dashboard enhancements; no significant tool cost
Logistics and delivery tracking	8–12	Integration with logistics APIs (courier tracking: $0–$50/month)
Bug fixes and technical debt	4–8 (ongoing)	No direct cost; essential for platform stability

Table 5: Estimated Effort and Costs for Backlog Features

Key Insights and Recommendations:

- **Scalability:** All features are relatively straight forward - that said, some (multi-language support, analytics, controllable via mobile, recommendation systems) may need specialised know-how or third- party services.
- **Cost Management:** Taking full advantage of open-source services, free starter tiers, and similar concessions will keep costs low, however some enhancements

(translations, analytics, payments, logistics) might represent recurring operational costs.

- **Strategic Prioritization:** Features that directly impact user experience, platform scalability, or revenue (e.g., improved search, express checkout, mobile access) should be considered for early implementation.
- **Continuous Investment:** To attain long-term stability, bug fixes and reduction of technical debt should be a priority.

9 Acceptance Criteria and Testing

This section talks about the acceptance criteria and the testing approach considered when building Smart Bargain. After obtaining an understanding on how the current market functions, especially in relation to agricultural trade and the increasing demand for direct, transparent transactions between buyers and sellers, this system was designed. The needs and priorities of the system were influenced by these observations from the real world.

We are expecting our future platform users to be small scale farmers, vendors, and regular consumers. We are also expecting the same groups to be our main stakeholders since they stand to gain from straightforward negotiations, reasonable prices, and easily available trading tools. The uniqueness of smart bargain relies on the fact that, this system caters negotiation functionality unlike other ecommerce websites available in the New Zealand agricultural section. Having said that even though the primary focus is on agricultural sector the functionality of smart bargain can be implemented on any other industry as well.

Detailed use cases are discussed in the previous sections of this report. Referring the use cases, the acceptance criteria for each use case is mentioned below to guarantee that features of smart bargain satisfy user expectations.

The team has used thorough acceptance testing to support these results and made sure the project was inclusive and suitable for local communities, including Māori communities. In order to preserve cultural relevance to the tenets of Te Tiriti o Waitangi, team conducted a quick market study during use case design.

9.1 Acceptance Criteria

Each user story has clearly defined acceptance criteria, derived from the use cases available in previous chapters.

Use Case Number	**Use Case Name**	**Acceptance Criteria**
SB-UC-0001	Register a Buyer or Seller Account	A new user can choose to be either a seller or a buyer. User must complete out all of the mandatory fields on the form, which are user's email address, username, password, and password confirmation. The system needs to ensure that the emails are unique, that the email formats are correct, and that the password fields match. A notification saying "success" will show up if the user is successful. Messages that check for errors are shown.
SB-UC-0002	Login to Smart Bargain	Users who have signed up can get into their account simply typing in their email address and password. If they are successful, they are taken to their dashboard. When the credentials are wrong, certain error messages are shown. When the Login button turns to Logout, the user name shows up.
SB-UC-0003	Logout from Smart Bargain	Users who are logged in can end the session by clicking Logout. After being sent to the login page, they need to log in again to see pages that are off-limits. The navigation is up to date.
SB-UC-0004	Navigate from Home Page	If the user is logged in, user can navigate to the Products, Dashboard, or Negotiation pages from the Home Page. If a visitor user tries to get to a restricted page and isn't logged in, they are sent to the login page.
SB-UC-0005	Buyer and Seller Access rights	Based on user role, users see only permitted pages and functions: Buyers cannot add or edit products, and Sellers cannot initiate purchases.

Use Case Number	**Use Case Name**	**Acceptance Criteria**
SB-UC-0006	Search and Filter Products in products and negotiation page	Users can filter products on the Products or Negotiation pages by either choosing a category or typing in the name of the product. The filtered results must meet the criteria. The default view should show all of the products.
SB-UC-0007	Add Product to Catalog	The seller must submit correct product information, such as the name, price, quantity, category, unit, and picture. When seller send in the form, all fields should be checked. When the addition is successful, a success message should appear and the catalog should be updated. Errors should show up when input is missing or wrong.
SB-UC-0008	Edit Product Details	The seller clicks Edit to edit the quantity, unit, or price of a product. Seller can change the form that is already filled out. Changes must be checked before they can be saved. A notification confirming success will show up. Cancel should remove changes.
SB-UC-0009	Delete Product Details	The seller can delete a product with the help of a confirmation dialog. When you click "Yes, Delete," the product should be erased and a notification saying "success" should show.
SB-UC-0010	View and Purchase Products	Buyers can see product listings that show the name, picture, price, quantity, and category. When Buyer click the cart button, a box should pop up where buyer may enter the quantity. Calculations of costs are done in real time. Errors should be shown for inputs that aren't valid or that go beyond the limit.

Use Case Number	Use Case Name	Acceptance Criteria
SB-UC-0011	Complete Payment via Purchase Modal	Buyers must input valid card details: number, name, expiry, and CVV. The form validates each field and only allows payment if all fields are correct. On success, order is recorded and success message shown. Errors for invalid or failed payment displayed.
SB-UC-0012	Seller Updates Bargain Settings	Sellers can open the Negotiation Page, search for a product, and update its minimum quantity and price thresholds for bargaining. Form validates numeric values and required fields. Confirmation message shown if update is successful.
SB-UC-0013	Buyer Initiates Bargain Request	Buyers open the bargain modal and input new quantity and price. Total cost should be auto calculated. Submit triggers request to seller. Form must validate minimum thresholds and display real time error messages if values exceed the boundary values of the threshold.
SB-UC-0014	Seller Reviews and Responds to New Negotiation Request	Sellers receive new negotiation requests in their dashboard. Each record includes product, quantity, and proposed price. Seller must be able to accept, reject, cancel, or counter. Once an action is taken, status is updated and moved to Ongoing Negotiations.
SB-UC-0015	Negotiation Dashboard - Ongoing Negotiation Management	Ongoing negotiations are shown in a table visible only to the buyer and seller involved. Statuses update with each counter. Parties can exchange proposals until accepted, rejected, or cancelled. Cancelled negotiations become read-only.
SB-UC-0016	Seller or Buyer Accepts Negotiation Request	Seller or Buyer accepts the negotiation by clicking Accept. Status changes to 'Active', and the task is finalized. System ensures cancelled records cannot be accepted.

Use Case Number	Use Case Name	Acceptance Criteria
SB-UC-0017	Seller or Buyer Rejects Negotiation Request	Seller or Buyer rejects the negotiation by clicking Reject. Status changes to 'Rejected'. System ensures cancelled records cannot be accepted.
SB-UC-0018	Seller or Buyer Cancels Negotiation Request	Seller or Buyer cancels a negotiation, changing the status to 'Cancelled'. Cancelled records cannot be modified further. UI and backend validations prevent further interaction.
SB-UC-0019	Seller or Buyer Counter Proposes on Negotiation	Counter Propose is available to both parties. Form validates changes to quantity or price. If values are unchanged, the submit button is disabled. Status should change to either 'Pending Buyer Approval' or 'Pending Seller Approval'.
SB-UC-0020	View Order History	Buyers can only view their own orders and not any other buyer's order. Sellers on the other hands can see orders involving their products. Table should display all order details: product name, quantity, price, status, and order date.
SB-UC-0021	Contact Us Form Submission	Any user, despite of being logged in or not logged in, can access the Contact Us form. Fields include name, email, and message. Form should validate required entries and correct email format. Upon successful submission, message should be stored in DB with 'open' status.
SB-UC-0022	View Contact Messages	Sellers should be able to view messages on their dashboard. Messages include sender name, status, and date. Clicking a row opens a modal with message details. After responding offline to the message, seller has the option to close the message.

Use Case Number	Use Case Name	Acceptance Criteria
SB-UC-0023	Filter Contact Messages on Seller Dashboard	Sellers should be able to filter contact messages by date or status (open/closed). The filtered list should get updated real time. If no matches are found, a message should be displayed: 'No messages found'.

9.2 Community and Maori Consideration

Acceptance criteria is a part of the validation process of Smart Bargain. This makes sure that the system is designed in such a way that it delivers both functional correctness and to align with Te Tiriti o Waitangi principles. These rules served as a guide for testing features with fairness, accessibility, and cultural relevance, mainly with regard to Māori communities.

Partnership: Acceptance testing makes sure that the system enables open and honest business transactions, including for community vendors, and buyers.

Participation: Irrespective of position ,experience or gender, buyers and sellers have equal access to platform features (such as dashboards, messaging, and negotiation).

Protection: Safe authentication, system feedback, user communication and negotiations are also a part of the acceptance criteria. By protecting user identity and trade agreements, the system reduces the possibility of misunderstandings or exploitation, especially for inexperienced users or members of vulnerable groups.

9.3 Test Strategies

The test strategy for the Smart Bargain system is outlined in the SmartBargain_TestPlan.pdf and includes a layered approach to make sure system performs functional well. Pleas refer to this pdf in the supporting work folder.

Unit testing was conducted and test scripts are also attached with the project submission in the supporting work folder.

Figure 12: Unit Test Script Snap Shot

Test cases are clearly defined and are conducted in different phases. At present, test cases are maintained in a spreadsheet named **SmartBargain_TestCases.xlsx** and there is a migration plan in the near future to a test management software like TestRail, where test cases will be centralized and used for automation. This document is also attached with the project submission in the supporting work folder.

Test migration strategy is to be further outlined to make the testing process more efficient.

Testing Tools and Environment

Smart Bargain is being tested in a defined environment with manual testing adaption and planned automation support:

Current Tools and Practices

Supported browsers: Chrome, Firefox
Backend: Django (Python) with SQLite
Frontend: HTML, CSS, JavaScript, and Bootstrap for responsive layout and styling using predefined template.
Test Case Execution: Manual testing is taken place in different stages based on the testing type using predefined datasets for Buyers, Sellers and Products. Test data will be recorded on excel.
Test Session Recording: JAM Extension is used during testing sessions to capture UI flows recordings and snap shots with debug friendly layout.

Issue Tracking: GitHub Issues is used for bug tracking (expected to migrate to jira in the future sprints)
Test Documentation: Test plan is shared in SharePoint and is planed to migrated to confluence in the future sprints.

Planned (Future) Tools and Enhancements

BDD Format Adoption: Manual test cases will be migrated to Behavior-Driven Development (BDD) in Given–When–Then format to simplify automation in the coming sprints
UI Test Automation: Selenium will be introduced for automated regression testing, specially across critical workflows (e.g., login, negotiation, payment)
API Testing: Feasibility adapting of postman will be investigated to validate backend API endpoints, specially for login, product management, and negotiation flows
Test Management: Tools like TestRail Adapter will be adopted to manage test cases and data.
CI/CD Integrations: Automation testing will be integrated into the build pipeline.
Jira Integration: All existing GitHub issues will be migrated to Jira for sprint based QA management.
Confluence Integration: Confluence will be linked with Jira to maintain centralized documentation.

9.4 Defect Management

At present the bug life cycle is simple and is straightforward. Once a bug is detected it created on git hub, label is added as bug and a detailed explanation of bug is provided. The bug will be originally set to open and once testing is completed it will be closed and the status of the issue will also become closed. If the bug rephased tester can reopen the bug instead of creating a new bug.

There is the feasibility to add the bug into projects and link to features as well .Defects are recorded in github. Below is a snap shot of some of the closed and open bugs.

Please refer to the detailed list of bugs recorded: Bug List

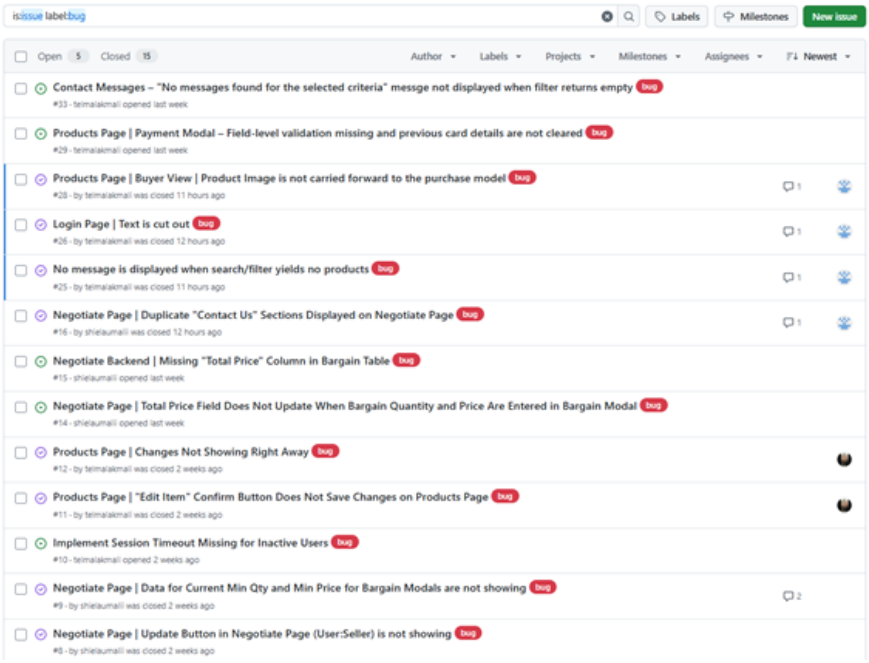

Figure 13: Bug List as shown in GitHub

There is a defined structure of the bugs recorded for easy tracking and understanding. Below is a snap shot of one of the bugs recorded. It clearly defines the steps to reproduce and a detailed explanation of what the bug is.
Refer: Bug Format

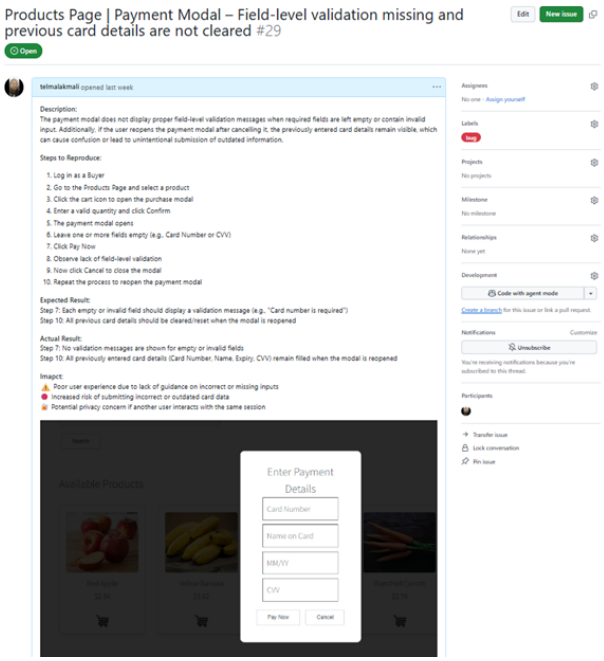

Figure 14: Bug Format

10 Version Control and Maintenance

Versioning is maintained when building Smart Bargain during development and testing. As part of the quality assurance process, all test documents, such as test plans and test cases, are versioned and kept up to date. In terms of development, GitHub is used for all source code management.

Version Number	Date	Contributor	Description
v1.0.0	07 July 2025	Lakmali	Test plan creation
v1.0.1	08 July 2025	Sheila	Test plan review

Figure 15: Test Plan Version History - 1

Version Number	Date	Contributor	Description
v1.0.0	8-Jul-25	Lakmali	Test cases creation
v1.0.1	9-Jul-25	Sheila	Test cases review

Figure 16: Test Case Version History - 2

Source Control Management

There are two developers worked on the initial version of Smart Bargain and shared the same repository to commit the code. Here separate branches for each feature and code is committed there. This makes it easier to work together, keep track of things, and go back to a previous version if necessary. Branches are only merged into the main branch after code reviews and validation have been done.

Maintenance Approaches

To ensure long-term sustainability, reliability, and scalability of the Smart Bargain platform, the following practices are being followed or are planned for future sprints:

1. **Regression Testing**
    - Carried out at the end of each sprint and before major releases.
    - The purpose here is to make sure that new changes do not break existing functionality.

- Automated testing coverage will be expanded where applicable in future phases. Since regression cases are mostly repeated cases, automation sits well here.

2. **Review Phase in Feature Design**
   - **Requirement Reviews:** Requirements are looked over before they are put into action. As the team grows in the next sprint, the review should be a part of the fixed process to make sure that what we're developing is exactly what the customer requested and that there are no assumptions or ambiguities.
   - **Design Reviews:** This is equally important as it checks the feasibility, technical alignment, UI flow, and consistency with project goals.
   - **Code Reviews:** Ensure clean, efficient, and maintainable code through peer review before merges.
3. **Code Housekeeping**
   - Remove redundant code, unused assets, and dead functions.
   - Enforce consistent naming conventions and file structures.
   - Refactor complex or duplicate logic to improve readability and maintainability.
   - Add comments where appropriate so that when a new developer is introduced to the team, it will be easier to understand the code.
4. **Test Data Management**
   - Maintain separate environments. Having the same environment may cause many conflicts where broken areas might be missed even with a solid regression plan.
   - Maintain a traceability matrix so that there is a central point to refer to requirements and respective test scenarios.
   - Carry out static testing before dynamic testing is conducted to uncover any bugs in design against the requirement.
   - Plan for scripts for faster environment setup.
5. **Issue and Defect Management**
   - All bugs, enhancements, and tasks are logged and tracked via GitHub Issues, which will later be migrated to Jira. This helps in maintaining a history of bugs, aiding in investigating similar issues.
   - Bugs are categorized and prioritized based on impact and severity.
   - Each issue is linked to both the requirement and the test case, where applicable.
6. **Documentation Versioning**

- Main documents such as test plans, user manuals, release notes, and requirement entries (e.g., functional solution documents) are version-controlled.
- Certain changes are tracked via Git or document history to ensure alignment across stakeholders.

11 References

Farm Fresh Direct. (n.d.). Retrieved June 16, 2025, from https://farmfreshdirect.co.nz/

Fruit and Veg Delivery Auckland | Fruit and Vegetable Store – Fruit World | Fruit World Royal Oak. (n.d.). Retrieved June 16, 2025, from https://fruitworld.co.nz/

Codemy.com (Director). (2023, August 17). Let's Build an Ecommerce Website! - Django Wednesdays ECommerce #1 [Video recording]. https://www.youtube.com/watch?v=u6R4vBa7ZK4

ProgrammingWithHarry (Director). (2020, May 21). How Django Works with HTML, CSS & JavaScript | Django Tutorial #2 [Video recording]. https://www.youtube.com/watch?v=pHQj9dVw5OA

Telephasic by HTML5 UP. (n.d.). HTML5 UP. Retrieved June 20, 2025, from http://html5up.net/telephasic

Collaboration software for software, IT and business teams. (n.d.). Retrieved July 9, 2025, from https://www.atlassian.com/

Open-source tool that uses simple textual descriptions to draw beautiful UML diagrams. (n.d.). Retrieved July 10, 2025, from https://plantuml.com/

Draw.io. (n.d.). Retrieved July 10, 2025, from https://www.drawio.com/

Canva: Visual Suite for Everyone. (n.d.). Retrieved July 11, 2025, from https://www.canva.com/

Mermaid Chart. (n.d.). Retrieved July 11, 2025, from https://mermaidchart.com

ChatGPT. (n.d.). Retrieved July 12, 2025, from https://chatgpt.com

Perplexity. (n.d.). Retrieved July 12, 2025, from https://www.perplexity.ai/

GitHub. (n.d.). Retrieved July 12, 2025, from https://github.com/

Visual Paradigm—Online Productivity Suite. (n.d.). Retrieved July 12, 2025, from https://online.visual-paradigm.com/