Software Requirements Specification

for

Web3 Campus Marketplace

Version 1.0 approved

Prepared by

Yang Yanqi
FanQianYi
Qiu Yixuan
Huang Qiyuan
Hu ShengQuan
Zhang Song

G7

2025.12.15

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

*The purpose of this Software Requirements Specification (SRS) is to define the functional and non-functional requirements for **Version 1.0** of the **Web3 Campus Marketplace**.This document provides a comprehensive description of the system's capabilities, interfaces, and constraints. It covers the full scope of the initial implementation, including the frontend user interface, the backend server logic, the database schema, and the specialized Web3 simulation service designed to mimic blockchain interactions. This SRS is intended to serve as the authoritative reference for the development team (Group 7:G7) and the project supervisor during the design, implementation, and testing phases of the project.*

1.2 Document Conventions

The following conventions are used throughout this document:

***Priorities**: Requirements are classified as High, Medium, or Low priority to guide development sequencing. High-priority items represent the Minimum Viable Product (MVP) core features.*
***Web3 Terminology:** Terms such as "Wallet," "Signing," and "Transaction" in this document refer to the **simulated** behaviors implemented within the project's educational scope, ensuring no real cryptocurrency or mainnet connection is required unless explicitly stated as a future expansion.*

1.3 Intended Audience and Reading Suggestions

***Development Team (Group 7):** To understand the detailed technical requirements for implementing the React.js frontend, Node.js backend, and PostgreSQL database.*
***Testers:** To derive test cases and validate that the system meets the specified functional and safety requirements.*

*Should read the entire document, paying particular attention to **Section 3 (External Interface Requirements)** and **Section 4 (System Features)** for specific implementation details regarding the Web3 simulation and marketplace logic.*

1.4 Product Scope

*The **Web3 Campus Marketplace** is a specialized second-hand trading platform designed specifically for the university community.*
The software aims to address the inefficiencies, lack of trust, and data privacy issues found in general-purpose platforms (e.g., Carousell) or informal channels (e.g., WhatsApp groups). The specific goals include:
***Secure Authentication:** Integrating simulated Web3 wallet login with campus email verification to ensure all participants are verified students.*
***Trustworthy Transactions:** Utilizing a simulated decentralized transaction-signing workflow to demonstrate data integrity and secure payment intent without using real funds.*

Efficient Marketplace: *Providing standard e-commerce features such as listing creation (CRUD), search filtering, and real-time chat to facilitate seamless peer-to-peer trading.*

1.5 References

Group 7 Project Proposal: *Web3 Campus Marketplace: Practical Implementation and Future Expansion Paths, submitted to Dr. Shen Zhiqi, Nanyang Technological University.*

2. Overall Description

2.1 Product Perspective

*The **Web3 Campus Marketplace** is a new, self-contained software product designed to address the specific needs of university students for a secure, efficient, and community-focused second-hand trading platform. Unlike general-purpose marketplaces such as Carousell, this system is tailored to the university ecosystem, integrating Web3 concepts—such as decentralized identity and simulated transaction signing—within a controlled educational environment. The product operates as a standalone web application but is architected to allow future integration with real blockchain networks or external campus systems (e.g., student directory services). It does not replace any existing NTU system but complements the informal trading channels currently used by students.*

2.2 Product Functions

The system provides the following major functions:

1. ***User Authentication & Campus Verification**: Simulated Web3 wallet login combined with university email verification to ensure only verified students can access the platform.*
2. ***Item Listing Management**: Users can create, read, update, and delete (CRUD) listings with images, titles, descriptions, and prices.*
3. ***Search & Filtering**: Browse and search items by keywords, categories, price range, and other attributes.*
4. ***Real-Time Communication**: Integrated chat system between buyers and sellers for negotiation and coordination.*
5. ***Simulated Web3 Transactions**: A mock transaction-signing workflow to demonstrate decentralized verification and data integrity without real cryptocurrency.*
6. ***Community Forum (Optional)**: A dedicated space for users to discuss products, share tips, and interact socially.*
7. ***Administrative Dashboard**: Moderators can manage users, listings, and disputes.*

2.3 User Classes and Characteristics

***Student Buyers**: Frequent users who browse, search, and purchase items. They value ease of use, trust, and quick communication.*
***Student Sellers**: Users who list items for sale. They need efficient listing tools, visibility, and secure transaction mechanisms.*

***Administrators**: University staff or designated moderators responsible for content moderation, user verification, and system oversight. They require elevated privileges and audit capabilities.*
***Guests**: Unauthenticated visitors who can view listings but cannot interact (e.g., chat, buy, or sell). They represent potential new users.*

2.4 Operating Environment

***Frontend**: React.js-based single-page application (SPA) running in modern web browsers (Chrome 90+, Firefox 88+, Safari 14+).*
***Backend**: Node.js with Express.js server, running on a Linux-based environment (e.g., Ubuntu 20.04 LTS).*
***Database**: PostgreSQL 14+ for persistent storage of users, listings, messages, and transactions.*
***Web3 Simulation**: Local services using ethers.js or web3.js libraries to mimic wallet authentication and signing, without connection to live blockchains.*
***Deployment**: Docker containers for consistent development, testing, and production deployment (e.g., on NTU lab servers or cloud VMs).*
***Network**: Standard HTTP/HTTPS protocols; WebSocket (via Socket.IO) for real-time chat.*

2.5 Design and Implementation Constraints

***Technology Stack**: Must use React.js (TypeScript), Node.js/Express.js, and PostgreSQL as specified in the proposal.*
***Web3 Simulation**: No connection to real blockchain mainnets is allowed during development; all wallet and transaction behaviors must be simulated.*
***Campus-Only Access**: Platform access restricted to users with verified university email addresses (@e.ntu.edu.sg or equivalent).*
***Development Timeline**: Project must be completed within the 12-week schedule outlined in the proposal.*
***Data Privacy**: Compliance with NTU data protection policies; no real private keys or sensitive personal data stored on server.*
***No Real Currency**: All transactional features are simulated; no real money or cryptocurrency may be transferred.*

2.6 User Documentation

The following user documentation will be provided:
***Online User Guide**: Integrated help system within the web application, covering login, listing creation, searching, chatting, and transaction steps.*
***Administrator Manual**: Separate guide for moderators covering user management, dispute handling, and system monitoring.*
***API Documentation**: For developers, describing backend endpoints and Web3 simulation service interfaces.*
***Deployment Guide**: Instructions for setting up the system in a Docker environment.*

2.7 Assumptions and Dependencies

***Assumptions**:*

Users have basic web browsing skills and access to a modern browser.
University email addresses are available and can be used for verification.

The NTU network provides stable internet connectivity during demonstrations.
No real blockchain integration is required for the scope of this course project.
Team members have proficiency in the selected technology stack (React, Node.js, PostgreSQL).

***Dependencies**:*

Availability of Docker for environment consistency.
Use of open-source libraries (e.g., ethers.js, Socket.IO, React) under their respective licenses.
Access to NTU lab servers or cloud credits for deployment and testing.
Guidance and feedback from project supervisor during weekly reviews.

3. External Interface Requirements

3.1 User Interfaces

This system adopts a web-based user interface, with a browser serving as the primary access point, providing a unified and intuitive second-hand trading platform for students within the campus. Users can access the system and complete all core operations through desktop or mobile browsers without the need to install additional clients.

The system is mainly targeted at students who have undergone campus identity verification. Users can simultaneously hold the identities of both buyers and sellers in the system and freely switch between different functions based on their usage scenarios. The design of the user interface focuses on lowering the usage threshold, enabling first-time users to quickly understand the system structure and operation logic.

In terms of functional structure, the user interface is mainly composed of the following core pages: login and wallet connection interface, campus email verification interface, market homepage, product detail page, product release and editing page, instant chat interface, and user personal information page. The market homepage is used to centrally display product lists and supports basic browsing and filtering functions; the product detail page is used to display detailed product information and provide an entry for communication with the seller; the chat interface supports one-on-one real-time communication between buyers and sellers to facilitate transaction completion.

Users typically log in through a simulated Web3 wallet and complete campus email verification during their first use. After completing the verification, users can browse products, post products, or initiate communication with other users. When users initiate a transaction, the system will guide them through a simulated transaction signature process to reflect the basic concepts of decentralized identity and transaction confirmation.

The system as a whole adopts a simple and modern design style, referring to the interface layout of mainstream consumer-level e-commerce and second-hand trading platforms. The interface layout emphasizes clear information hierarchy, intuitive navigation structure, and has good responsive design capabilities to adapt to devices of different screen sizes. At the same time, during the design process, the principle of consistency is emphasized to ensure that the operation logic and visual style of different pages remain unified, thereby enhancing the overall user experience.

3.2 Hardware Interfaces

This system does not involve dedicated hardware user interfaces and does not directly interact with any specific physical hardware devices. On the client side, users can access the system using personal computers, laptops, or smart mobile devices. Common input devices include keyboards, mice, or touch screens, and the output is mainly presented through the device's display screen. The system does not set special configuration requirements for client hardware and only requires basic network connection and browser operation capabilities. On the server side, the system is deployed in cloud servers or general-purpose computing server environments. The details of server hardware are transparent to the application layer, and the system communicates with clients through standard network interfaces, without relying on specific brand or model hardware devices. In summary, this system has good universality and portability at the hardware level and can operate stably in different hardware environments.

3.3 Software Interfaces

Based on the system's "three-layer architecture + Web3 simulation service layer", the core interfaces are divided into four categories, covering the entire transaction process requirements:

3.3.1 Front-end - Back-end business interface
Using the RESTful specification, focusing on the core business scenarios:
User authentication interface: including POST /api/auth/register, POST /api/auth/login, achieving "Web3 wallet + campus identity" dual verification, ensuring that only authenticated students can access.
Product management interface: supporting POST /api/items/create, GET /api/items/list, PUT/DELETE /api/items/:itemId, covering all CRUD operations of products, supporting image upload and multi-condition filtering.
Transaction process interface: through POST /api/transactions/create-payload, POST /api/transactions/verify-signature, simulating the Web3 transaction process, ensuring transaction transparency and traceability.
Real-time communication interface: based on Socket.IO, designing chat:connect, chat:send events, achieving real-time communication between buyers and sellers, synchronizing messages and storing them in the database to support historical backtracking.

3.3.2 Web3 simulation service interface
Independently encapsulating the core capabilities of Web3, not relying on the live blockchain network:
Generating simulated key pairs: the front-end generates wallet addresses and private keys using ethers.js, and stores the private keys locally;
Generating login challenge value: the backend generates a 32-bit nonce for login requests to prevent replay attacks;
Verifying signature validity: the backend verifies the signature and the matching of the wallet address, the data to be verified, and confirms the legality of the operation.
3.3.3 Back-end - Database interface
Designed based on PostgreSQL and ORM framework, achieving data persistence:
User model (User): supports querying and updating the campus authentication status by wallet address / user ID;

Product model (Item): supports filtering queries by category / price, updating the product status (available / sold);
Transaction model (Transaction): records transaction ID, information of both buyers and sellers, signature result, supporting transaction status tracking.

3.3.4 Reserved external service interface

Reserved for future expansion of standardized interfaces:
Real blockchain connection interface: follows the JSON-RPC protocol, supports connecting to the Ethereum test network;
Campus identity system interface: adopts OAuth2.0, realizes unified campus authentication automatic verification;
OCR book recognition interface: supports passing in the cover image of the book, returning ISBN, book name, etc. to simplify product listing.

3.3.5 Exception handling

The unified exception return format is {code: error code, message: description information, data: null}, covering key exception scenarios: signature verification failure (401) returns "Signature invalid, please retry"; no product operation permission (403) prompts "Not the product publisher, no operation permission"; database query exception (500) feedback "System busy, please try later", ensuring users clearly perceive the problem and guide correct operations.

3.4 Communications Interfaces

This system adopts the typical client-server communication mode, and realizes the data interaction between the front-end user interface and the back-end application services through the network. The design of the communication interface focuses on ensuring the reliability, security of data transmission, and the collaborative operation among different functional modules.
At the basic communication level, the main communication method between the system's front end and back end is based on the HTTP/HTTPS protocol. All requests related to user identity authentication, product information management, and transaction processes are transmitted through a secure HTTPS channel to prevent data from being stolen or tampered with during transmission. The client completes page data loading and user operation processing by sending requests and receiving responses from the back end.
In terms of interface design, the front-end and back-end of the system communicate through a set of well-structured application programming interfaces (APIs). The interfaces follow a resource-oriented design concept and are used to support core functions such as user login, product posting and querying, and user information management. The data structures of interface requests and responses adopt a unified data format to ensure the consistency and parsability of data among different modules of the system.
For functions that require real-time interaction, such as instant chatting between buyers and sellers, the system adopts a communication mechanism based on persistent connections to support the real-time transmission of messages. This communication interface is used to transfer chat messages and related status information between the client and the server, thereby reducing communication latency and enhancing the user interaction experience. The necessary communication data will be recorded by the backend service to support subsequent message queries and system auditing.
In terms of data representation, the system uniformly uses a structured data format to encapsulate and transmit information during the communication process. The communication interface performs

necessary format verification and error handling when receiving and sending data to ensure the stability and fault tolerance of the communication process.
Overall, the communication interface design of this system adheres to the principles of standardization and security. Through clear communication protocols and data formats, it enables efficient collaboration among all components of the system, providing a communication-level guarantee for the reliable operation of the platform.

4. System Features

This section defines the core functional features of the Web3 Campus Marketplace. These modules collectively form the system's Minimum Viable Product (MVP), covering the complete business flow from user authentication and item management to transaction execution and platform oversight. Each feature is elaborated upon through its description, stimulus/response sequences, and specific functional requirements.

4.1 User Authentication and Campus Verification

4.1.1 Description and Priority

This system feature defines the mechanisms by which users authenticate themselves and are verified as legitimate members of the university community before accessing protected marketplace functionalities. The feature combines a simulated Web3 wallet–based authentication process with campus email verification to establish a trusted, campus-only user environment.
The purpose of this feature is to ensure that only verified university users are allowed to create listings, initiate transactions, and communicate with other users, thereby reducing fraud risks and improving trust within the marketplace.
This feature is considered High Priority, as it serves as the entry point to the system and underpins all subsequent trading and interaction features.

4.1.2 Stimulus/Response Sequences

Stimulus 1:
A user attempts to access the platform for the first time and initiates the login process.
Response 1:
The system prompts the user to connect a simulated Web3 wallet and generates a unique authentication challenge for the session.
Stimulus 2:
The user signs the authentication challenge using their simulated wallet.
Response 2:
The system verifies the signature and associates the wallet address with a user account.
Stimulus 3:
The user submits a campus email address for verification.
Response 3:
The system sends a verification request and updates the user's verification status upon successful confirmation.
Stimulus 4:
An unverified or unauthenticated user attempts to access restricted features (e.g., posting a listing or initiating a transaction).

Response 4:
The system denies access and displays a message indicating that authentication and campus verification are required.

4.1.3 Functional Requirements

FR-AUTH-1: The system shall allow a user to initiate authentication using a simulated Web3 wallet.
FR-AUTH-2: The system shall generate a unique authentication challenge for each wallet login attempt.
FR-AUTH-3: The system shall verify the authenticity of a signed challenge by validating the signature against the provided wallet address.
FR-AUTH-4: The system shall create or retrieve a user account associated with a successfully authenticated wallet address.
FR-AUTH-5: The system shall require users to complete campus email verification before accessing trading-related features.
FR-AUTH-6: The system shall restrict unverified users from creating item listings, initiating transactions, or sending messages.
FR-AUTH-7: The system shall maintain and store the authentication and verification status of each user account.
FR-AUTH-8: The system shall allow authenticated and verified users to access all features permitted by their assigned role.

4.2 User Profile and Account Management

4.2.1 Description and Priority

This system feature defines how authenticated users manage their personal profiles and account-related information within the platform. It covers the viewing and updating of basic user information, the management of wallet and verification status visibility, and the enforcement of role-based access distinctions.
The purpose of this feature is to provide users with transparency and control over their account information while ensuring that account data remains consistent with the system's authentication and authorization mechanisms. By clearly managing user profiles and roles, the system supports secure access control and prepares the foundation for administrative oversight.
This feature is considered Medium Priority, as it supports and complements core marketplace operations but does not directly enable trading activities.

4.2.2 Stimulus/Response Sequences

Stimulus 1:
An authenticated user navigates to their personal profile or account settings page.
Response 1:
The system displays the user's profile information, including wallet address, campus verification status, and assigned role.
Stimulus 2:
The user submits a request to update editable profile information.
Response 2:
The system validates the submitted information and updates the user profile upon successful validation.
Stimulus 3:

The user attempts to access account management functions that exceed their assigned role privileges.
Response 3:
The system denies the request and displays a message indicating insufficient permissions.
Stimulus 4:
An administrator accesses the profile of another user for moderation or oversight purposes.
Response 4:
The system displays the selected user's profile information according to administrative access permissions.

4.2.3 Functional Requirements

FR-PROFILE-1: The system shall allow an authenticated user to view their own profile information.
FR-PROFILE-2: The system shall display the user's wallet address, campus verification status, and assigned role within the profile view.
FR-PROFILE-3: The system shall allow a user to update permitted profile information.
FR-PROFILE-4: The system shall validate user-submitted profile updates before applying changes.
FR-PROFILE-5: The system shall prevent users from modifying profile attributes that are restricted by system policy.
FR-PROFILE-6: The system shall enforce role-based access control when users attempt to access account management functions.
FR-PROFILE-7: The system shall allow administrators to view user profiles for moderation and management purposes.
FR-PROFILE-8: The system shall store and maintain updated user profile information in persistent storage.

4.3 User Profile and Account Management

4.3.1 Description and Priority

This system feature defines how verified users create, view, update, and manage item listings within the campus marketplace. It governs the full lifecycle of a listing, from initial creation to modification, status updates, and removal, while enforcing ownership and access control rules.
The purpose of this feature is to enable sellers to efficiently publish and manage second-hand items and to ensure that all listings presented on the platform are accurate, traceable to a verified user, and compliant with marketplace policies. Proper listing management is essential for maintaining data integrity and trust between buyers and sellers.
This feature is considered High Priority, as it represents a core capability of the marketplace and directly supports trading activities.

4.3.2 Stimulus/Response Sequences

Stimulus 1:
A verified user submits a request to create a new item listing.
Response 1:
The system validates the submitted listing information and creates a new item listing associated with the user's account.

Stimulus 2:
A user views their existing item listings.
Response 2:
The system retrieves and displays all listings owned by the user, including their current status.
Stimulus 3:
The user submits a request to edit or update an existing item listing.
Response 3:
The system verifies the user's ownership of the listing, validates the updated information, and applies the changes upon successful validation.
Stimulus 4:
The user submits a request to delete an item listing.
Response 4:
The system confirms the request and removes the listing from active marketplace visibility.
Stimulus 5:
An unverified user attempts to create or modify an item listing.
Response 5:
The system denies the request and displays a message indicating that campus verification is required.

4.3.3 Functional Requirements

FR-LIST-1: The system shall allow a verified user to create a new item listing.

FR-LIST-2: The system shall require each item listing to include mandatory attributes such as title, description, and price.

FR-LIST-3: The system shall associate each item listing with the user account that created it.

FR-LIST-4: The system shall allow a user to view all item listings that they have created.

FR-LIST-5: The system shall allow a user to edit information of an item listing that they own.

FR-LIST-6: The system shall prevent a user from modifying or deleting item listings owned by other users.

FR-LIST-7: The system shall allow a user to delete an item listing that they own.

FR-LIST-8: The system shall update the status of an item listing to reflect its current availability.

FR-LIST-9: The system shall restrict unverified users from creating, editing, or deleting item listings.

FR-LIST-10: The system shall store and maintain all item listing information in persistent storage.

4.4 User Profile and Account Management

4.4.1 Description and Priority

This system feature defines how users browse, search, and filter item listings available on the campus marketplace. It enables users to efficiently discover items of interest through

structured listing presentation and query mechanisms without requiring prior interaction with sellers.
The purpose of this feature is to improve information accessibility and usability by allowing users to locate relevant listings based on keywords, categories, and other attributes. Effective browsing and search capabilities are essential for reducing transaction friction and supporting informed purchasing decisions.
This feature is considered High Priority, as it directly affects user experience and is fundamental to the usability of the marketplace.

4.4.2 Stimulus/Response Sequences

Stimulus 1:
A user accesses the marketplace homepage.
Response 1:
The system displays a list of available item listings according to default sorting rules.
Stimulus 2:
The user submits a search query using keywords.
Response 2:
The system retrieves and displays item listings that match the search criteria.
Stimulus 3:
The user applies one or more filtering conditions (e.g., category or price range).
Response 3:
The system updates the displayed listings to reflect the selected filters.
Stimulus 4:
The user selects a specific item listing from the results.
Response 4:
The system displays the detailed information associated with the selected item listing.

4.4.3 Functional Requirements

FR-BROWSE-1: The system shall allow users to browse a list of available item listings.
FR-BROWSE-2: The system shall display item listings with essential summary information such as title, price, and availability status.
FR-BROWSE-3: The system shall allow users to search item listings using keyword-based queries.
FR-BROWSE-4: The system shall allow users to filter item listings based on predefined attributes.
FR-BROWSE-5: The system shall update search and filtering results dynamically based on user input.
FR-BROWSE-6: The system shall allow users to view detailed information for a selected item listing.
FR-BROWSE-7: The system shall exclude unavailable or restricted listings from default browsing results.
FR-BROWSE-8: The system shall retrieve and present browsing and search results within acceptable response time limits.

4.5 Buyer–Seller Real-Time Messaging

4.5.1 Description and Priority

Description: Implements one-on-one instant messaging functionality between buyers and sellers, supporting real-time message exchange to help users negotiate transaction details, confirm product status, and ultimately complete transactions. This feature ensures that only the buyer and seller involved in a specific transaction can communicate, protecting user privacy and reducing irrelevant information interference.

Priority: High (Core transaction function directly related to buyer-seller communication and transaction completion)

4.5.2 Stimulus / Response Sequences

Stimulus 1:
User A clicks the "Contact Seller" button on the product details page
Response 1:
System checks if User A and the seller are participants in the same transaction, establishes a chat session upon successful verification
Stimulus 2:
User A enters and sends a message in the chat input box
Response 2:
System validates message content and pushes the message to User B (seller) in real-time
Stimulus 3:
User B receives the message and replies
Response 3:
System pushes the reply message to User A in real-time
Stimulus 4:
User A or User B views chat history
Response 4:
System retrieves and displays complete chat records from the database

4.5.3 Functional Requirements

FR-CHAT-1: The system shall support one-on-one chat sessions between buyers and sellers
FR-CHAT-2: The system shall implement real-time message sending and receiving functionality
FR-CHAT-3: The system shall restrict chat permissions to only allow relevant transaction participants
FR-CHAT-4: The system shall save all chat records for users to view at any time
FR-CHAT-5: The system shall handle message sending failures and notify users
FR-CHAT-6: The system shall support message read status display

4.6 Simulated Transaction and Signing Workflow

4.6.1 Description and Priority

Description: Implements the complete process of transaction confirmation and recording, demonstrating decentralized transaction verification mechanisms through simulated Web3 core functions. This feature demonstrates data integrity and secure payment intent, allowing users to understand the basic principles of blockchain transactions while ensuring no real funds are transferred.

Priority: High (Core showcase feature of the Web3 Campus Marketplace, embodying project characteristics)

4.6.2 Stimulus / Response Sequences

Stimulus 1:
User clicks the "Buy Now" button on the product details page
Response 1:
System generates transaction content including product information, price, and buyer-seller information
Stimulus 2:
User confirms transaction content and submits
Response 2:
System simulates digital signature process, displaying signature confirmation interface
Stimulus 3:
User clicks "Confirm Signature" in the simulated wallet interface
Response 3:
System updates transaction status to "Confirmed" and records transaction details
Stimulus 4:
User views transaction history in personal center
Response 4:
System displays list of all completed transactions and their status

4.6.3 Functional Requirements

FR-TRANS-1: The system shall support user purchase requests
FR-TRANS-2: The system shall automatically generate standardized transaction content
FR-TRANS-3: The system shall simulate digital signature confirmation process
FR-TRANS-4: The system shall record and update transaction status
FR-TRANS-5: The system shall save all transaction records for subsequent queries
FR-TRANS-6: The system shall handle errors during transaction process

4.7 Community Forum and Social Interaction (Optional)

4.7.1 Description and Priority

Description: Provides non-transactional campus communication features to enhance platform social attributes. This feature allows users to discuss products, share usage tips, and exchange campus life experiences, creating an active campus community environment and improving user stickiness and platform value.

Priority: Medium (Optional feature that enhances user experience but is not a core transaction function)

4.7.2 Stimulus / Response Sequences

Stimulus 1:
User clicks the "Post New Thread" button on the forum page
Response 1:
System displays thread editing interface where user fills in title and content
Stimulus 2:
User submits thread content
Response 2:
System reviews thread content (optional), publishes to relevant category upon approval
Stimulus 3:
Other users browse thread and click "Comment" button
Response 3:
System displays comment input box where user enters comment content
Stimulus 4:
User submits comment
Response 4:
System adds comment below thread and displays it to other users in real-time
Stimulus 5:
User selects specific category (e.g., "Electronics", "Books") on forum page
Response 5:
System displays list of all threads in that category

4.7.3 Functional Requirements

FR-FORUM-1: The system shall support user posting and browsing threads
FR-FORUM-2: The system shall implement comment interaction functionality
FR-FORUM-3: The system shall provide content categorization and search functionality
FR-FORUM-4: The system shall save all social interaction records
FR-FORUM-5: The system shall handle inappropriate content (optional)

4.8 Administrative Moderation and Oversight

4.8.1 Description and Priority

Description: Provides platform management functionality to ensure content quality and user behavior compliance. This feature grants administrators the ability to monitor and constrain platform content, maintain platform order, handle violations, protect user rights, and ensure healthy platform operation.

Priority: High (Platform management and security function ensuring normal platform operation and user safety)

4.8.2 Stimulus / Response Sequences

Stimulus 1:
Administrator logs into system and accesses admin backend
Response 1:
System verifies administrator identity and permissions, displays admin interface
Stimulus 2:
Administrator views content pending review in admin interface (e.g., user reports, newly posted threads)
Response 2:
System displays list and detailed information of content pending review
Stimulus 3:
Administrator selects to process specific content (approve/reject/delete)
Response 3:
System executes corresponding action and records the operation

4.8.3 Functional Requirements

FR-ADMIN-1: The system shall provide administrator permission control mechanism
FR-ADMIN-2: The system shall implement content review functionality
FR-ADMIN-3: The system shall support user and product management
FR-ADMIN-4: The system shall record all administrative operations and violation handling
FR-ADMIN-5: The system shall provide management operation log query functionality
FR-ADMIN-6: The system shall handle administrator operation exceptions

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The system is designed to provide an acceptable level of performance for a campus-scale user base. Under normal NTU network conditions, the marketplace listing page should load within about **2 seconds** for the vast majority of requests, and search results over up to **5,000 active listings** are expected to return within roughly **3 seconds**. The backend should be able to support around **100 concurrent logged-in users** performing typical actions such as browsing and chatting without server errors. Real-time interaction is also important: chat messages are expected to reach the peer within about **1 second** on average, and when a user confirms a purchase or escrow initiation in the wallet interface, the system should acknowledge the request within about **3 seconds**, not counting any simulated on-chain confirmation time. For deployment and demonstration, the combined services (database, backend, Web3 simulation) should be able to start up within approximately **2 minutes** after being triggered.

5.2 Safety Requirements

From a safety perspective, the prototype is intended to minimise the risk of accidental loss of data or assets, especially when blockchain-like interactions are involved. Before executing any irreversible operation—such as deleting a listing or initiating an escrow payment—the system will always request explicit confirmation from the user through a clear confirmation dialog. All Web3-related behaviour will run only on **simulated services or test networks**, ensuring that no real fiat currency or mainnet tokens are transferred during the course project. Whenever a user performs an action that could affect locked funds or transaction status, the interface will display a prominent risk warning message. In addition, key records such as user accounts, listings and transaction logs will be backed up periodically (e.g., at least once per day in the demo environment) so that information can be recovered in case of system failure.

5.3 Security Requirements

Security requirements focus on protecting user identities, access rights and data integrity in a Web2 + Web3 hybrid environment. Users must authenticate using a **campus email–based account or equivalent NTU identity** before they can access trading features, and the system also supports a wallet-based login mechanism where the user proves ownership of a Web3 wallet by signing a challenge message. All client–server communication is protected with **HTTPS (TLS 1.2 or higher)** to prevent eavesdropping and tampering. Access control rules ensure that users can create, edit and delete only their own listings, while administrative functions such as removing reported content or viewing platform-level statistics are restricted to accounts with an explicit admin role. To prevent manipulation of payment parameters, critical transaction fields (such as amount, recipient and order ID) are reconstructed on the backend or inside the smart contract, rather than trusting values directly from the browser.

The system deliberately avoids storing wallet private keys on the server; any local test keys used for simulation are kept only in encrypted form. All important security-relevant events, including login success or failure, listing deletion, payment initiation and dispute creation, are recorded in an audit log with timestamps and user identifiers to support later analysis. Sensitive APIs such as login, wallet binding and payment initiation are additionally protected by rate limiting, in order to reduce the impact of brute-force attempts and abusive traffic.

5.4 Software Quality Attributes

Beyond pure security and performance, the system must satisfy several quality attributes to remain usable and maintainable throughout the project. During scheduled demo sessions, the platform is expected to achieve an availability level of around 99%, avoiding unnecessary downtime in front of users and evaluators. In terms of usability, a new user with basic web experience should be able to complete the flow of logging in, browsing listings and initiating a chat with a seller within about 5 minutes, without any prior training. The codebase is organised so that frontend, backend and Web3 simulation logic reside in separate modules, which improves maintainability and allows developers to change one module with minimal impact on the others. The architecture is designed to support horizontal scaling of the backend—such as adding more instances—without modifying existing APIs or smart contract interfaces, and the entire system can be packaged with Docker so that it can be deployed on different machines (for example an NTU lab server or a cloud VM) with only minor configuration adjustments.

5.5 Business Rules

Several business rules define how the marketplace should behave in a university context. Only verified members of the university community—such as students and staff—are allowed to create listings and participate in trades, ensuring that the platform remains a campus-only service. To prevent abuse and control storage usage, each user is expected to hold no more than about **20 active listings** at any given time; if a user wishes to sell additional items, they must reuse or replace existing listings. For trades conducted through the platform, the default payment path is the **escrow mechanism** implemented by the smart contract or Web3 simulation service; private off-platform transfers are considered outside the system's responsibility. The marketplace will also respect campus regulations by disallowing prohibited items (e.g., controlled goods, exam papers or other banned categories), and such listings can be removed by administrators when detected. In case of disputes between buyers and sellers, the system records the dispute creation and status changes but does not automatically decide outcomes—final resolution is handled offline by designated campus staff.

6. Other Requirements

Appendix A: Glossary

This glossary defines key terms, acronyms, and abbreviations used in this Software Requirements Specification (SRS) for the Web3 Campus Marketplace system.

Escrow
A mechanism by which funds are temporarily held by a neutral party (here, a smart contract or simulation service) and released to the seller only after agreed conditions, such as delivery confirmation, are satisfied.

Testnet
A blockchain network used purely for testing and development, where tokens have no real-world value, allowing smart contracts and transactions to be experimented with safely.

Web3 Wallet
A software component or browser extension that manages blockchain accounts and private keys and can sign transactions or messages on behalf of the user. In this project, wallet behaviour may be simulated.

Campus Marketplace
The web-based platform developed in this project that allows university students and staff to list, browse, and trade second-hand items, with selected flows enhanced by Web3 concepts.

Listing
A record representing an item offered for sale on the marketplace, including details such as title, description, price, owner, and current status.

Transaction
A logical record describing the process by which a buyer and a seller agree to exchange an item for payment, potentially involving an escrow mechanism and dispute handling.

Dispute
A state in which a transaction is contested by one or both parties, for example due to non-delivery or item mismatch; in this project, disputes are recorded in the system but resolved offline by campus staff.

Admin (Administrator)
A privileged user role responsible for moderating content, handling reported listings, and supervising platform operations and health.

Web3 Simulation Service
A backend component that imitates essential blockchain behaviours (such as escrow and event logging) without connecting to a real mainnet, used for demonstration and testing in this course project.

Appendix B: To Be Determined List

TBD-1: Team Role Assignment
TBD-2: Final Technology Choices
TBD-3: System Architecture Details

TBD-4: Database Schema Design
TBD-5: Deployment and Environment Setup