Software Requirements Specification

for

Usell

Version 1.0 approved

Prepared by Ahmed Cengiz Yavuz
Bahadır Yavuz YAPAR
Ertuğrul AK
Muzammil Kohistani
Saim Melih Özcan

MvM Software

WE SHAPE FUTURE

02.04.2025

Table of Contents

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document explains all key parts about Usell system - it is online marketplace where peoples can buying and selling second-hand clothes, shoes, bags and other fashion accessories. The platform will works perfectly same on mobile phones and web browsers, so all users gets same good experience no matter which device they using. Main goal of Usell is supporting eco-friendly shopping by makes easy and safe to sell used fashion items instead throwing them away. This document very important because it gives all details and rules for development team and company bosses, helps everyone understand how system should works from first day until final version ready. We writes everything clear with structure so nobody gets confusions during project.

1.2 Document Conventions

This document use special text styles to make important things easy see. When you see bold words, this mean section names or requirement numbers like REQ-1. Italic words we use for example text like <name> or to show something is important. For buttons or codes, we write like this. This help everyone understand fast what is what.

We also have rules for how we write requirements. Every requirement in part 3 have priority level (high, medium, low). Pictures and diagrams go in Appendix B with simple explanations. If we change document, we write all changes in Revision History table with date and reason. This way all team members can follow same rules when they read or update document.

1.3 Intended Audience and Reading Suggestions

This guide is for all peoples working on Usell platform - the developers who builds mobile and web versions, interface designers who makes screens user-friendly, QA testers who checks for problems, managers who keeps project on track, plus security experts and future maintenance teams too. Before looking at technical requirements parts, everybody should first read the Introduction and Overall Description sections because they explains what system does and why important - this helps for better understanding before seeing how everything actually works in real practice with all the functional and non-functional details.

1.4 Project Scope

Usell is online second-hand market what working good on phones and computers both - peoples gets same shopping on every device they using. Users they can make accounts for sell stuffs or finding things with search filters, and can talks safe with buyers/sellers by messages inside. The site got secure pays and feedback system for make more trusts between users. We makes it with new techs and have our own backends with security things for keep user datas safe. Platform can grows bigger later when we adds more smart features like AI suggest things, points rewards programs, and maybe shipping options too when more peoples starts using, so system stays working good always.

1.5 References

- https://www.smashingmagazine.com

Good for understand how to make website easy and nice for users. We use this for make same experience on mobile and computer.

- https://owasp.org

Important for safety, especially when users pay money or share private data. Helps us stop hackers.

- Sommerville, I. (2016). Software Engineering (10th ed.). Pearson.

Old but very useful book for how to write requirements and build system. We follow this for organize our work.

- Rogers, Y., Sharp, H., & Preece, J. (2011). Interaction Design (3rd ed.). Wiley.

Teach how to design buttons, menus so peoples not get confused. Important because our platform work on phones and web.

2. Overall Description

2.1 Product Perspective

Secondhand Shopping App is an independent mobile and web application where customers can sell their old products, buy other's products and talk about the products. The app will try to provide a user-friendly interface, secure shopping experience and adequate communication channels.

The system consists of three main parts: user interfaces (website and mobile app), backend server system and database. The mobile app and website allow users to interact with the system, while the backend server deals with all business logic and communicates with the database. The database stores all user information, product lists, messaging and transaction history.

2.2 Product Features

Secondhand Shopping App will offer the following key features:

- User Management
  - Registration and login
  - Create and edit profiles
  - User evaluation and commenting system
  - Security verification (email, phone)

- Product Listing and Management
  - Add, edit and delete products
  - Upload multiple photos
  - Detailed product descriptions
  - Category and subcategory selection
  - Price setting and negotiation option
  - Add location information

- Search and Discover
  - Advanced filtering (category, price, location, status)
  - Keyword search
  - Recommendation system and popular products
  - Location-based search
  - Favoriting and following

- Communication and Messaging
  - Instant messaging
  - Notification system
  - Video product review
  - Ask the seller a question

- Payment and Secure Shopping
  - Secure payment system
  - Buyer and seller protection
  - Product delivery confirmation
  - Returns and dispute resolution

- Analytics and Reporting
  - User activity monitoring
  - Sales and purchase statistics
  - Popularity and trend analysis

2.3 User Classes and Characteristics

User classes defined for the Secondhand Shopping App:

- Normal Users
  - Target audience: 18-65, with little experience with technology
  - Frequency of use: Several times a week
  - Features: Both can buy and sell, employs basic capabilities
  - Technical awareness: Standard mobile app usage ability
  - Priority level High (primary user segment)

- Professional Sellers
  - Demographics: Small business owners, boutique shops, antique sellers
  - Usage frequency: Daily active use
  - Features: Multiple product listings, requires detailed analytics
  - Technical expertise Medium-high level
  - Priority level: High (target market of premium features)

- Buyers
  - Population: All ages, different economic levels
  - Frequency of use: Need-based, irregular
  - Features: Search, filter, contact and purchase
  - Technical knowledge Low-medium
  - Priority level Middle

- System Administrators
  - Demographics: Technical staff, customer service
  - Usage frequency: Daily (business hours)
  - Characteristics: Content moderation, user support, dispute resolution
  - Technical expertise High level
  - Priority level: Low (few in number but critical)

2.4 Operating Environment

The Secondhand Shopping App will work in the following environments:

- Mobile Platforms:
  - iOS 13.0 and above
  - Android 8.0 (Oreo) and above
  - Minimum 2GB RAM requirement
  - Camera, GPS and notification access required

- Web Platform:
  - Modern web browsers (Chrome, Safari, Firefox, Edge latest versions)
  - Responsive design (mobile, tablet and desktop support)
  - JavaScript must be enabled

- Server Environment:
  - AWS or Google Cloud Platform infrastructure
  - Linux-based servers
  - Load balancing and scalability support
  - SSL security certificates

- Database:
  - Hybrid use of MongoDB (NoSQL) and PostgreSQL (relational)
  - Regular backup and data replication

- Network Requirements:
  - Stable internet connection (minimum 1 Mbps)
  - Communication via RESTful API
  - WebSocket support (for instant messaging)

2.5 Design and Implementation Constraints

The following constraints will be taken into account in the development of the Second Hand Shopping App:

- Legal Constraints:
  - KVKK (Personal Data Protection Law) compliance is mandatory
  - Compliance with e-commerce law is mandatory
  - Legal regulations regarding selling used goods need to be taken into account
  - Age limit for users below the age of 18 needs to be enforced

- Technical Constraints:
  - React Native for cross-platform mobile application development
  - Node.js and Express framework back-end use
  - Use of JWT (JSON Web Token) to secure the APIs
  - Photo size optimization (max 5MB/photo)
  - Monthly data storage limits (1GB for normal users)

- Hardware Limitations:
  - Performance optimization for low-medium mobile devices
  - Data optimization for low bandwidth users

- Security Constraints:
  - Two-factor authentication requirement
  - PCI DSS compliance for storing payment details
  - Regular security tests and penetration tests

- User Interface Constraints:
  - Conformity with Material Design and iOS Human Interface Guidelines
  - Conformity with accessibility guidelines (WCAG 2.1) required

2.6 User Documentation

The following user documentation will be included with the Secondhand Shopping App:

- User Manuals:
    - Electronic user manual (PDF)
    - Frequently Asked Questions (FAQ) section
    - Video tutorials (short, feature-based)
    - Step-by-step guide to product listing

- Online Help:
    - In-app help center
    - Context-sensitive tips
    - Live support service (business hours)
    - Community forum

- Security and Usage Policies:
    - Terms and conditions of use
    - Privacy Policy
    - Guide to safe shopping
    - Guide to reporting fraud and suspicious behavior

- Onboarding Materials:
    - Interactive guidance for first-time use
    - “Quick start” guide
    - Demos for new features

2.7 Assumptions and Dependencies

The following dependencies and assumptions were determined for the Secondhand Shopping App:

- Assumptions:
  - The assumption is made that the majority of users use smartphones or tablets
  - The audience is assumed to be technologically literate with general technology awareness
  - It is being assumed that the majority of the users will provide camera and location service access
  - Users are being assumed to have a typical internet connection
  - It is being assumed that user counts and transaction volume will increase over time

- Dependencies
  - Google Maps API (for geolocation search)
  - Firebase (for analytics and notifications)
  - Iyzico and PayTR payment platforms
  - AWS S3 (for photo storage)
  - Twilio (for SMS verification)
  - Algolia (for SEO)
  - Apple App Store and Google Play Store review processes
  - Third-party authentication services (login with Apple, Google, Facebook)

- Risks
  - Risk of market saturation
  - Fraud attempts and fake accounts
  - Risk of illegal or inappropriate products being listed
  - Changes in legal regulations
  - Possible changes or interruptions in third-party APIs

Changes to any of these assumptions and dependencies may impact the project timeline, costs or feature set.

3. System Features

3.1 User Registration and Authentication

3.1.1. Description and Priority

This feature allows users to create an account and enter the application securely. Users can sign up using traditional registration models such as email-password registration or rely on third-party authentication methods such as Google and Apple accounts as alternatives. As user authentication is a crucial aspect of having access to personalized services, such as creating listings, messaging, and manipulating their profile, this is a high priority for the application. A smooth and secure login experience can also help instill user trust in the application and promote user engagement.

3.1.2. Stimulus/Response Sequences

The user opens the application, or website, and is given a prompt to log in or create an account. If a user chooses to register an account, then the system prompts for some basic information: email and password and asks if they would like to register with pre-existing 3rd party accounts. Once the user has registered their account, the system emails a confirmation to the user. When a user logs in, the system authenticates a user's account. Once a user is successfully logged into their account, they are in the application or website or an error message will occur if the user was not successfully logged in. When a user forgets their password, the system will prompt the user to reset their password.

3.1.3. Functional Requirements

REQ-1: The system will allow users to register with an email and password.

REQ-2: The system will support social logins using platforms such as Google and Apple.

REQ-3: The system will send a confirmation email upon registration to verify the account.

REQ-4: The system will validate input fields and provide error messages for invalid formats.

REQ-5: The system will allow users to reset their password using a verification email.

REQ-6: The system will block access after multiple failed login attempts to ensure security.

3.2 Product Listing and Search

3.2.1. Description and Priority

This essential feature enables users to upload their second-hand items for sale, as well as browse the uploads from other users. The listings will include information about second-hand items including title, category, description, price, condition, location, and photos. This feature is high-priority, as the whole premise of the platform is around exchanging used items. A user-friendly, filter-friendly listings interface is important to ensure a fun and productive experience for both buyers and sellers.

3.2.2. Stimulus/Response Sequences

When a user is interested in selling an item, they tap the “Sell” button and the system will launch a form for them to fill in product information and upload pictures. After submitting, the listing is published and viewable by other users of the app. The buyers may utilize the search bar or navigate categories and filters such as price used, condition, and distance in order to narrow down searches to relevant items. When they tap on a product, it opens a detailed view where they can see full product information and contact the seller.

3.2.3. Functional Requirements

- REQ-1**:** The system will allow users to create product listings with images, text, and categories.
- REQ-2**:** The system will support uploading multiple images for each listing.
- REQ-3**:** The system will provide real-time search with filters such as category, location, price, and condition.
- REQ-4**:** The system will display detailed product pages when listings are clicked.
- REQ-5**:** The system will support editing or deleting listings by their owner.

3.3 Messaging and Communication

3.3.1. Description and Priority

This function offers a way for buyers and sellers to communicate synchronously through an in-app messaging system. This function is intended for real-time communication purposes to facilitate negotiations, answer questions, and arrange delivery/pick-up for the product. Messaging is a medium priority feature in that it does improve trust and improves the efficiency of the buying process because the buyers and the sellers can have a conversation that stays within the app.

3.3.2. Stimulus/Response Sequences

When a buyer has seen an item, they can press the "Message Seller" button, which will present a chat window to the buyer. The buyer has an option to type a message and send it through the chat. Once they do, their message will be sent to the seller, and the seller will get a notification. Seller can reply to the buyer through the app, and the conversation continues. Buyer and seller are able to access all messages in their message history at any time. Users can report a message when it is inappropriate or harmful.

3.3.3. Functional Requirements

REQ-1: The system will allow users to send and receive messages related to a product listing.

REQ-2: The system will notify users when they receive a new message.

REQ-3: The system will store all conversations in a secure chat history.

REQ-4: The system will allow users to report abusive messages or block other users.

REQ-5: The system will display timestamps and read indicators in conversations.

3.4 Transaction and Order Status Management

3.4.1. Description and Priority

With this feature, sellers have a method to update the status of their items and allow buyers to see whether the other products they are interested in purchasing are available or not. Marking items "Sold" or "Reserved" helps prevent confusion and keeps the website accurate and current. This is a medium priority feature, supporting transparency, and contributes to a better overall user experience through possibly reducing miscommunication.

3.4.2. Stimulus/Response Sequences

When an item is no longer available, the seller opens their listing management panel and marks the item as “Sold.” The system then removes the item from public search results and notifies any users who previously showed interest. Buyers who previously contacted the seller will see the updated status in their conversation or listing history.

3.4.3. Functional Requirements

REQ-1: The system will allow sellers to update the availability status of their listings.

REQ-2: The system will hide sold items from search results and general browsing.

REQ-3: The system will notify buyers who have interacted with the listing about its status change.

REQ-4: The system will allow users to view their transaction history, both as a buyer and seller.

4. External Interface Requirements

4.1 User Interfaces

The user interface of our software product will follow a simplistic design pattern to ensure a smooth user experience. Our team follow standard GUI conventions and design principles, with a fixed top menu, uniform font styles, and clearly labeled buttons. Each screen will include common navigation elements to guide users. Buttons such as Home, Search, Sell, My Account, and Get Help will remain accessible at all times, ensuring easy navigation throughout the application.
Our main interface components will include the following:

- Home Page: Displays items based on the user's recent browsing activity. Users can like items or leave comments on listings they are interested in.

- Navigation Bar: Provides quick access to essential pages such as *Home*, *Categories*, *Sell an Item*, *My Account*, and *Get Help*.

- Search Functionality: A responsive search bar enables users to find items by keyword, brand, or category.

- Product Listings: Each item will display a thumbnail image, product name, price, and a brief description, visible on the Home page and in search results.

- Product Detail Page: Offers comprehensive item information including multiple images, a detailed description, seller contact info, and buying options.

- User Dashboard: Allows users to manage their profile, view their active and past listings, track purchases, and monitor sales performance.

- Accessibility: The interface will follow accessibility standards to support users with disabilities, including features such as color contrast adjustments for users with color blindness.

4.2 Hardware Interfaces

The software product is designed to operate on commonly used consumer hardware, including smartphones, tablets, and personal computers such as laptops and desktops. It requires a stable internet connection and a modern web browser. Except these basic requirements, no additional hardware is necessary for the application to run smoothly.
However, some optional hardware components provide better user experience. For example, when listing an item for sale, users may utilize their device's built-in camera to capture photos. The software will access the camera through standard device permissions and APIs, allowing images to be uploaded directly to the platform. While the absence of such optional hardware will not prevent the application from working, its accessibility provides improved usability.

4.3 Software Interfaces

The software application will communicate with a dedicated back-end server that manages business logic, data processing, and communication with the database. The front-end app will talk to the server for sending and receiving RESTAPIs through HTTPS. The APIs will manage tasks such as user authentication, product listing, browsing through products, image upload, purchase transactions, and altering messages between users.

Back-end server will be deployed with Node.js (version 18.0) and it will talk to relational database management like PostgreSQL (version 14.0). Persistent data such as images, product description, comments, user details, transaction record, etc. shall be stored in the database. Also application will utilize shared libraries and SDKs for tasks such as image processing, form validation, and rendering UI. Third-party service PayPal will be used. Communication among components will be stateless, request-response based.
User and item information will be transferred between front-end and back-end components via typed json payloads.

4.4 Communications Interfaces

The software product will employ HTTPS for all the secure communication between the front-end application and the back-end server. All the API requests, including user authentication, product listings, and payment transactions, will be made over the HTTPS 1.3 protocol, which will ensure encrypted communication. This will prevent unauthorized interception of sensitive data during transmission.

Apart from HTTPS, the product will also use email communication to notify users of major events such as registration confirmation, password reset, purchase order confirmation, and shipping details. Emails will be sent using an SMTP protocol, with appropriate message format for accessibility by various email clients. All email communication will be transmitted securely using SSL/TLS encryption to prevent leakage of information.

For data transfer rate, the system should support standard internet rate, with low-latency processing given top priority for the processing of real-time updates. The system should ensure that emails are sent out within minutes of triggering actions and no perceptible delay will occur in request processing from the users through the HTTPS connection.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The system need to work fast and smooth all time, even when many peoples using together. It should handle minimum 10,000 users at same time without getting slow. When peoples looking at products or sending messages, it should take less than 2 seconds. Searching must be quick too - show results in under 1.5 seconds. First screen must load fast when open app, maximum 3 seconds. Same speed needed for both website and mobile app. System can't crash much - must stay working 99.5% of time every month. We will test hard to make sure it don't break when suddenly many peoples come.

5.2 Safety Requirements

We need protect users from losing important datas by mistakes. When somebody try delete product or make payment, system must ask "you sure?" first. Deleted things stay in backup for 1 week in case need recover. System should tell users clearly when they make wrong inputs. Everything peoples do - like login, upload, buy - must save in logs with time and date so we can check later if problems happen.

5.3 Security Requirements

Very important keep user informations safe. Passwords must save in special coded way. If user not active for 15 minutes, system make them login again. Nobody can see phone numbers or emails without permission. Can use SMS code for extra security if want. All payments go through safe outside companies. Website must use HTTPS so bad peoples can't steal datas.

5.4 Software Quality Attributes

App need be easy use - new peoples should understand fast without help. Must work good normal time and fix itself if problems come. Code write clean so developers can change easy later. When more peoples use, system should grow bigger without breaking. Same good work on phones and computers. We make sure can test everything good - both automatic and manual tests possible.

6. Other Requirements

6.1 Technical Stuff & Database

We use MongoDB for messy data like photos and chats, PostgreSQL for important things like user accounts and payments. System takes backup every night (keeps for 30 days) and must handle 10k+ users without lagging. Queries need be fast - if take more than 0.5 second then problem. Having sync issues between databases during tests. Users should at least see last viewed listings when offline.

6.2 Legal & Local Requirements

We must follow KVKK rules for protect user datas (no want big fines!) but hard part is stop underage sellers - if we ask ID scan users complain, if just put "I'm 18+" button kids can lie. Payment systems need be more secure (still waiting Iyzico finish their check). For languages we start only English and Turkish because more easy, show dates/prices automatic in right format for each country (like 01.04.2023 for Turkey, 04/01/2023 for US). Biggest problem now is how check age for real without make good users angry - legal guys want ID but UX team say will lose many users, still no find good solution, maybe need think different way.

6.3 Future Proofing & Extras

Writing code in modules so can reuse later (if we document properly - which probably we won't lol). Adding Firebase analytics to track what users actually do. Must work with screen readers for blind people (WCAG compliance). Performance sucks on old phones - need optimize. Also weird bugs where listings not updating properly sometimes.

Appendix A: Glossary

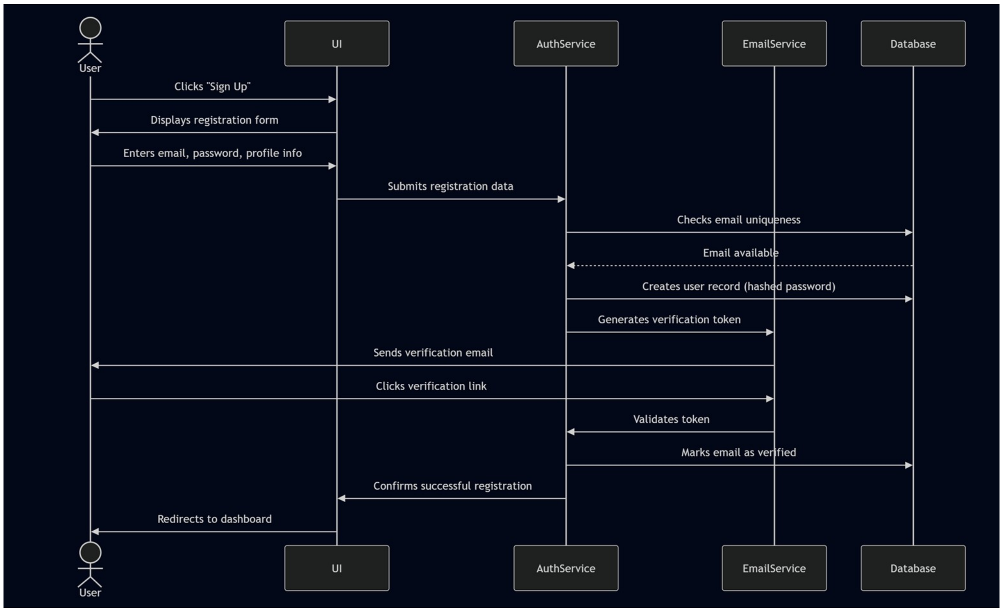
- **SRS** - "Software Requirements Stuff" - Big document that says what system must do
- **KVKK** - Turkish law about keeping user datas safe (very important!)
- **JWT** - Fancy password token that keeps logins secure
- **REST API** - How app talks to server (like ordering food from menu)
- **NoSQL** - Database for messy datas like photos and chats
- **SQL** - Database for neat datas like user accounts
- **PCI DSS** - Super strict rules for payment things
- **WCAG** - Rules to make app work for blind/disabled peoples
- **UI** - What users see and click (buttons, screens etc)
- **UX** - How easy app is to use (should not make people angry)
- **Backend** - Magic behind scenes that makes app work
- **Frontend** - Pretty part users actually see
- **HTTPS** - Secure internet connection (no hackers stealing data)
- **SSL** - Extra lock for security (like bank website has)
- **Firebase** - Google tool for analytics and notifications
- **AWS** - Amazon's cloud servers where we host stuff
- **API** - App's way of talking to other services
- **JSON** - Format for sending datas (like digital paperwork)
- **Responsive** - Works good on both phone and computer
- **Cache** - Temporary storage to make app faster
- **Two-factor auth** - Extra login security with SMS code

- **Scalability** - Ability to handle more users without crashing
- **Bandwidth** - Internet speed needed for app to work smooth
- **PostgreSQL** - Strong database for keeping user accounts and payments safe (like digital filing cabinet)

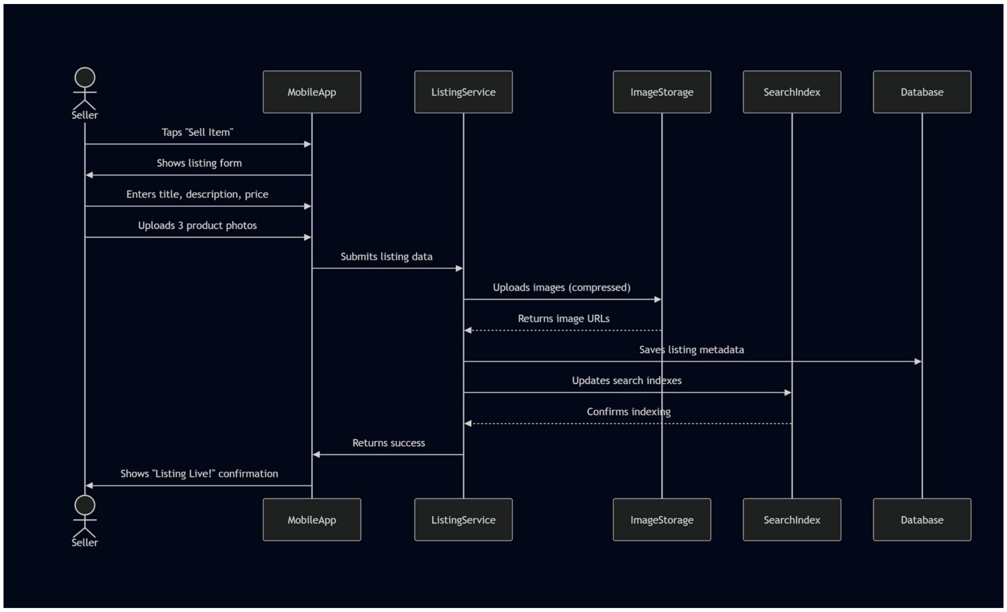
Appendix B: Analysis Models

SEQUENCE DIAGRAMS

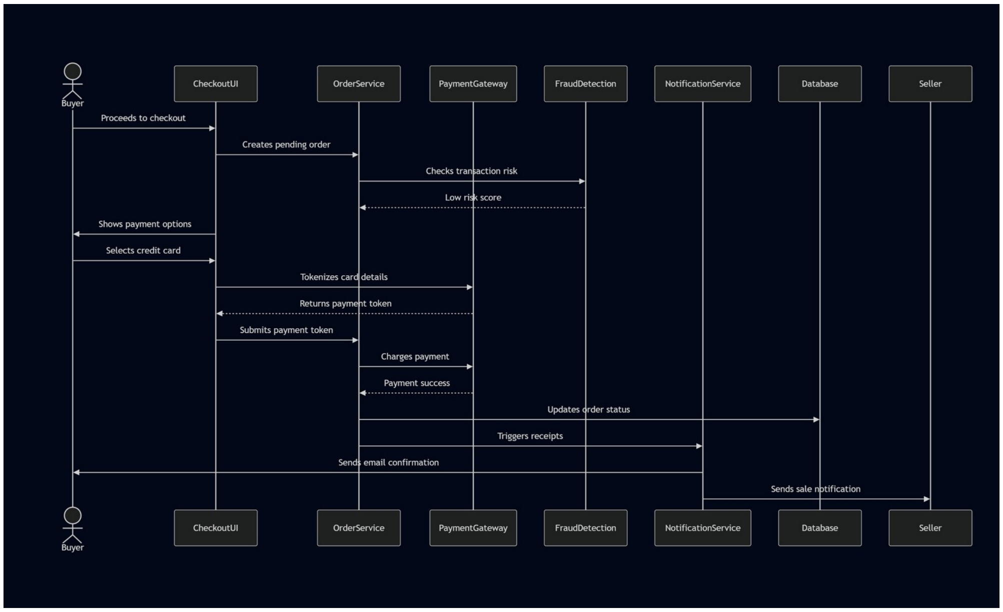
1. User Registration with Email Verification



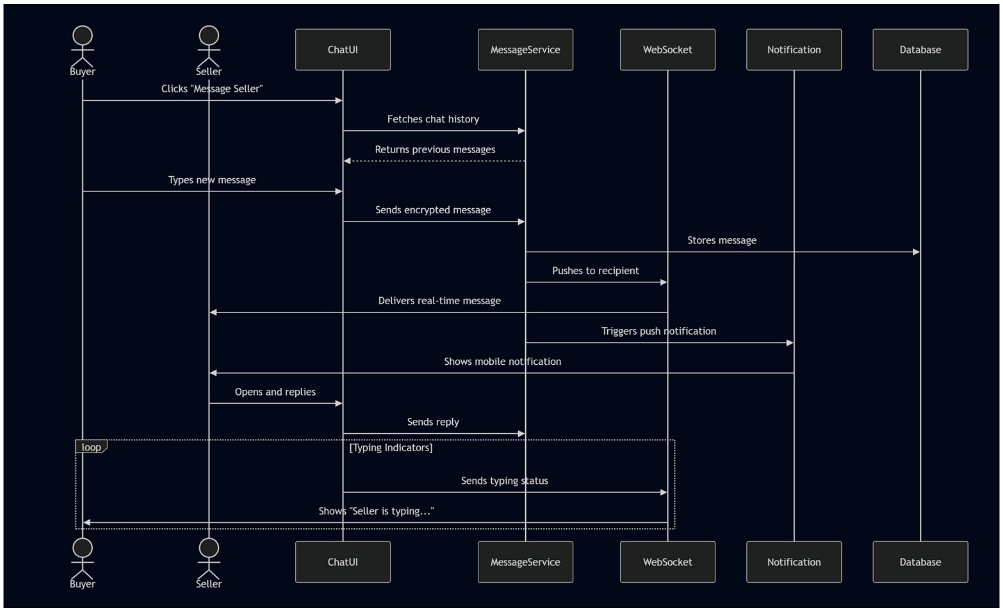
2. Product Listing Creation



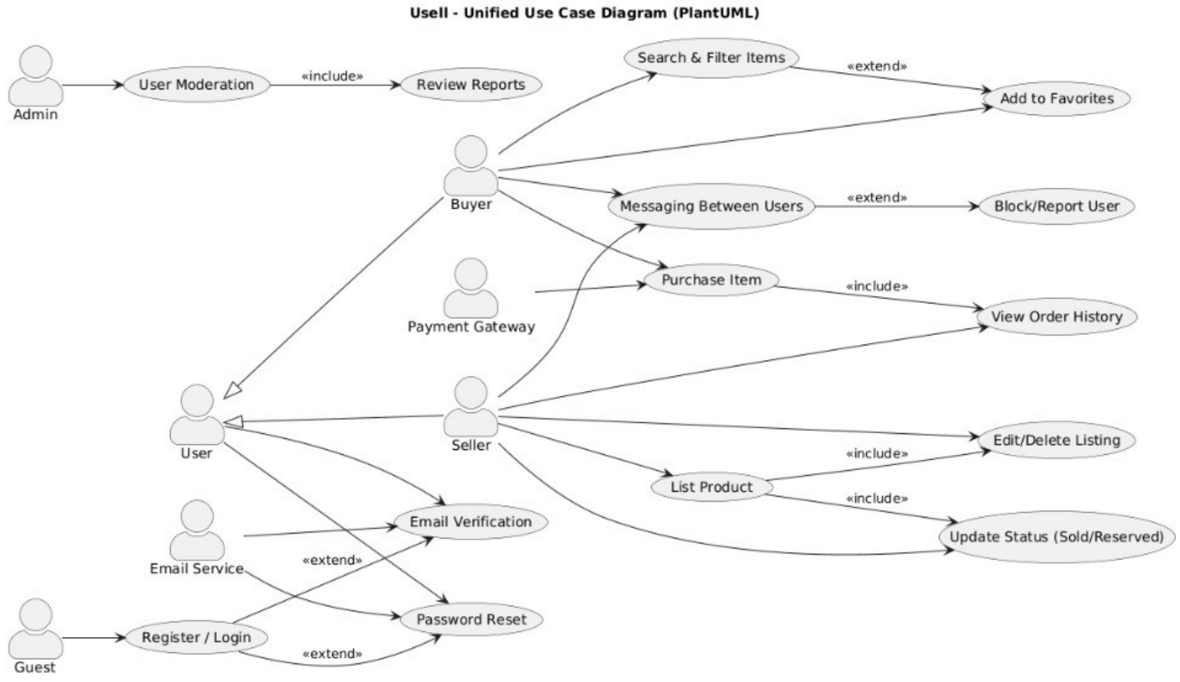
3. Secure Payment Transaction



4. Real-time Messaging

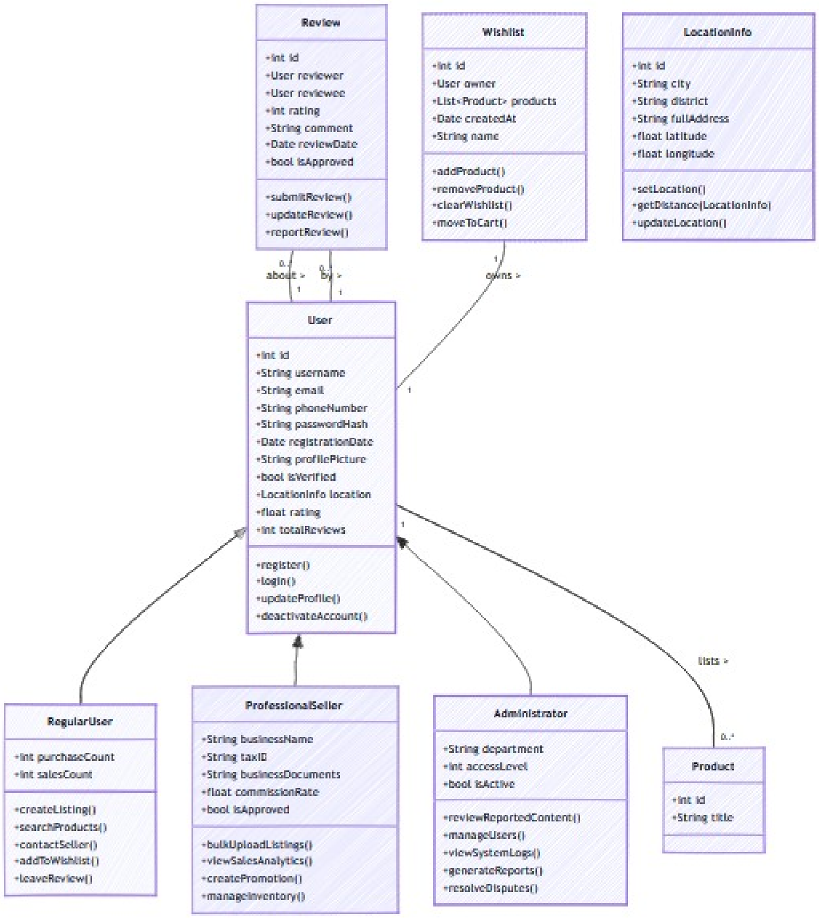


USE CASE DIAGRAM

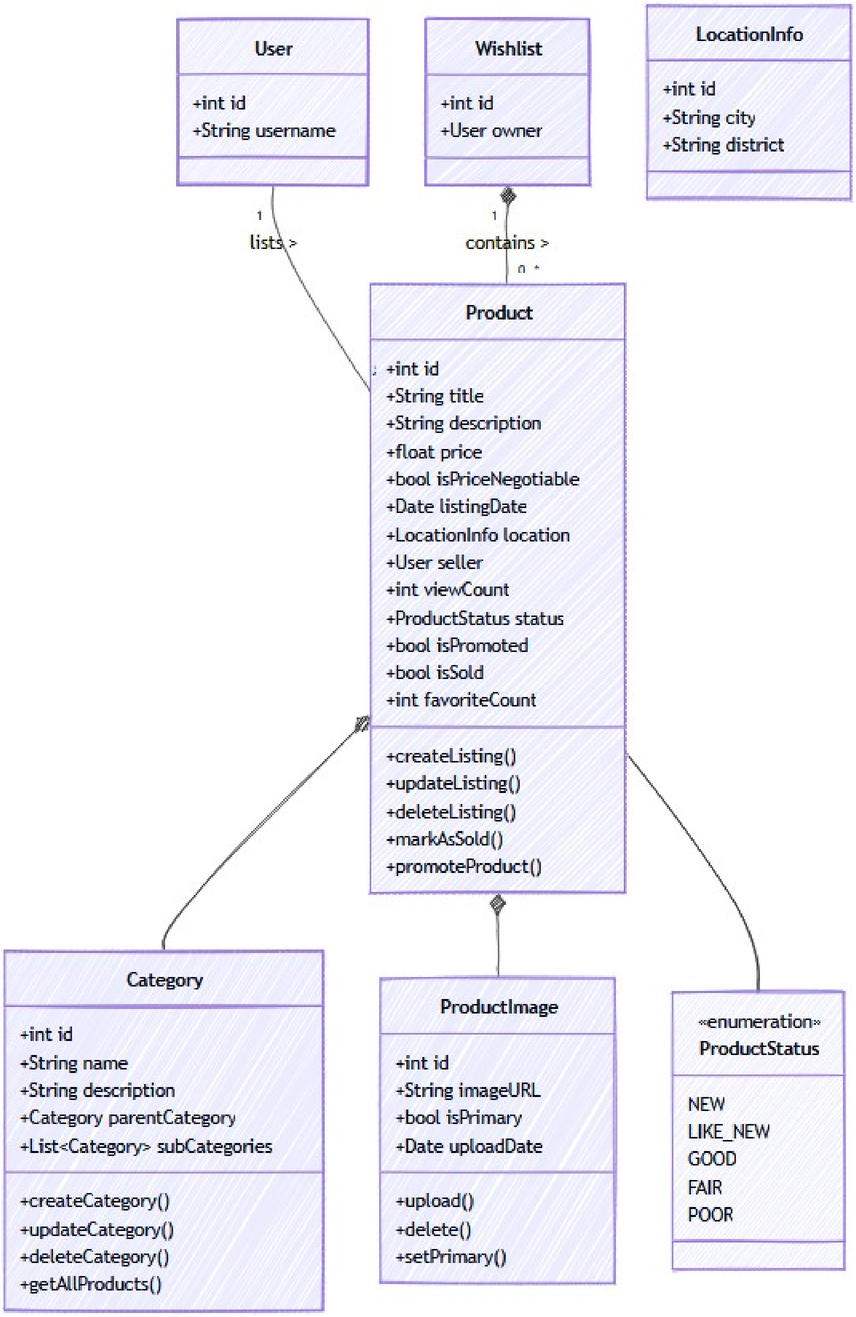


CLASS DIAGRAMS

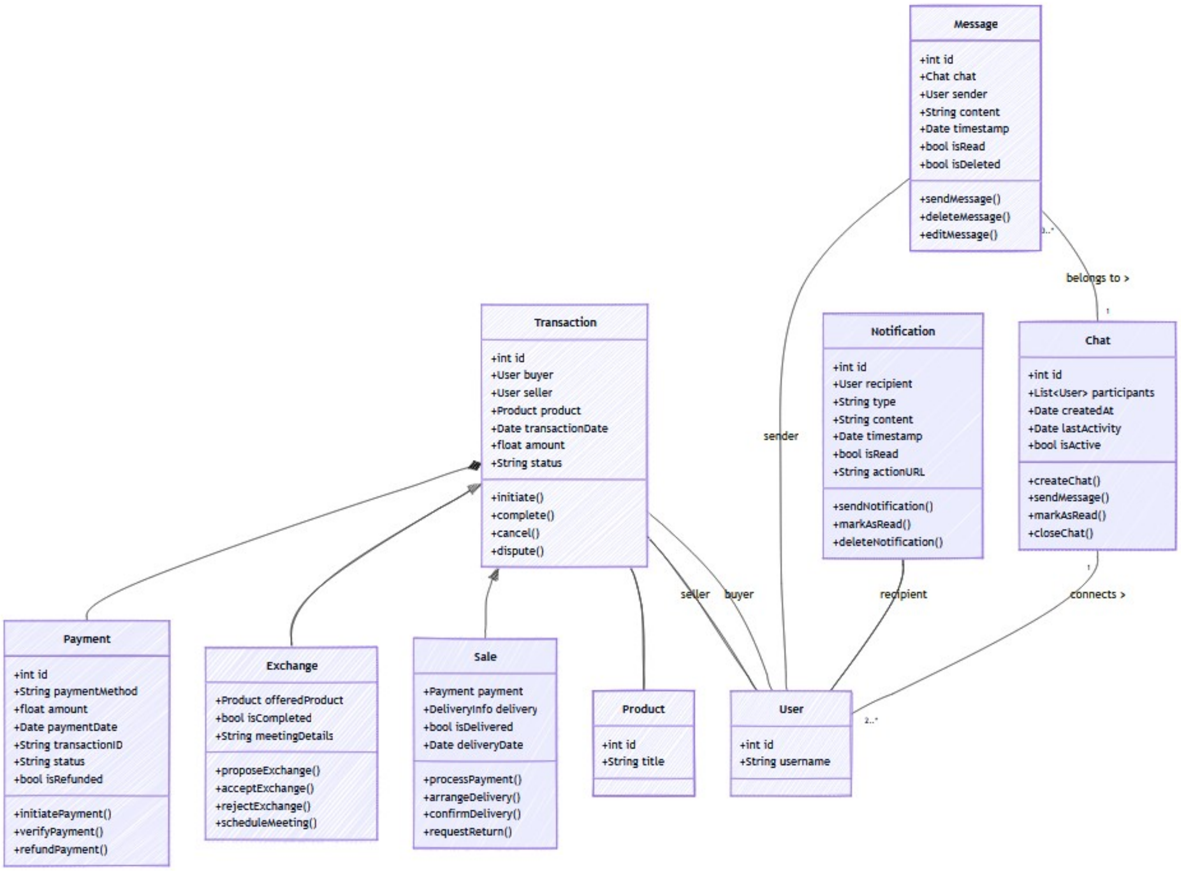
1. User and Profile Classes



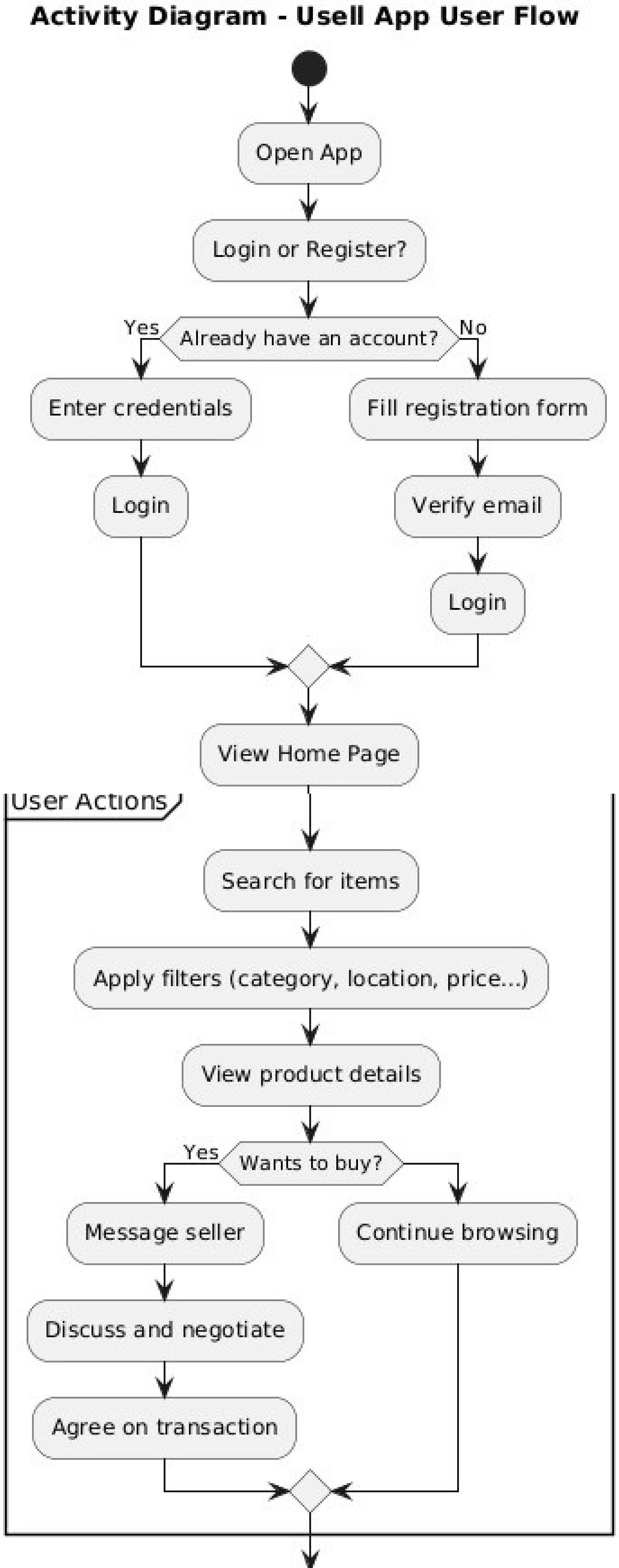
2. Product and Category Classes

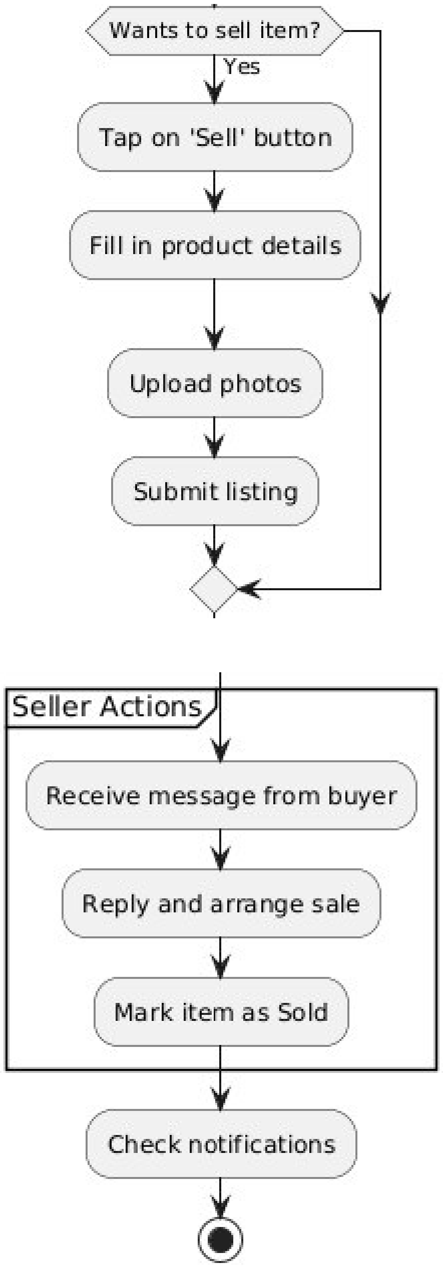


3. Process and Communication Classes



ACTIVITY DIAGRAM




Wants to sell item?
Yes
Tap on 'Sell' button
Fill in product details
Upload photos
Submit listing
Seller Actions
Receive message from buyer
Reply and arrange sale
Mark item as Sold
Check notifications

Appendix C: Issues List

There are several external service integration decisions yet to be made. The payment gateway provider is yet to be decided, with the likes of Stripe, PayPal, or local services being considered. The same goes for the image storage approach whether local server storage, AWS S3, or Firebase Storage will be used. These decisions will be based on pricing, scalability, and simplicity of integration.

The design of the notification system is not decided. Whether it will be real-time push notifications or in-app and email notifications only is not decided. The app needs more planning regarding user and product listing reporting . Finally, accessibility and data privacy, are important but certain standards and regulations still need to be studied more.

Appendix D: Meetings List

- Meeting 1 - 15.03.2025 - Iyzico Payment Team, Legal Advisor
- Meeting 2 - 20.03.2025 - KVKK Consultant, UX Designer
- Meeting 3 - 01.04.2025 - App Store Representative