

---

# **Software Requirements Specification**

**for**

# **Swappify**

**Version 1.0 approved**

**Prepared by**

Rohail Rathore

Mustan Ali

**SZABIST**

**4th October 2024**

# Table of Contents

<b>1. Introduction</b>	<b>1</b>
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Product Scope	1
1.5 References	2
<b>2. Overall Description</b>	<b>2</b>
2.1 Product Perspective	2
2.2 Product Functions	2
2.3 User Classes and Characteristics	3
2.4 Operating Environment	3
2.5 Design and Implementation Constraints	3
2.6 User Documentation	3
2.7 Assumptions and Dependencies	4
<b>3. External Interface Requirements</b>	<b>4</b>
3.1 User Interfaces	4
3.2 Hardware Interfaces	16
3.3 Software Interfaces	16
3.4 Communications Interfaces	16
<b>4. System Features</b>	<b>16</b>
4.1 Sign Up	16
4.2 Login	18
4.3 Forget Password	19
4.4 Profile Management	21
4.5 List Item	23
4.6 Edit Item	24
4.7 Delete Item	26
4.8 View Item Listings	27
4.9 Search and Filter Item Listings	28
4.10 Add to Wishlist	29
4.11 View Wishlist	31
4.12 Item Detail	32
4.13 Trade Proposals	34
4.14 Accept Trade	35
4.15 Reject Trade	36
4.16 Cancel Trade	38
4.17 Advanced Trade Management	39
4.18 Manage Active trades	41
4.19 Ban User Accounts	42

4.20 Delete List Items	43
4.21 Rate and Review System	45
4.22 Email Notification	46
4.23 Trade History	47
4.24 Send Messages	48
4.25 Remove from Wishlist	49
<b>5. Other Nonfunctional Requirements</b>	<b>50</b>
5.1 Performance Requirements	50
5.2 Safety Requirements	51
5.3 Security Requirements	51
5.4 Software Quality Attributes	51
5.5 Business Rules	51
<b>6. Other Requirements</b>	<b>51</b>
<b>4. Appendix A: Glossary</b>	<b>51</b>
<b>5. Appendix B: Analysis Models</b>	<b>52</b>
a. Use Case Diagram	52
<b>6. Appendix C: To Be Determined List</b>	<b>52</b>

## Table of Figures

Figure 1: Product Perspective Diagram	2
Figure 2: Home Page	4
Figure 3. Sign Up Page	5
Figure 4. Login Page	5
Figure 5. Forget Password	6
Figure 6. Profile Page	6
Figure 7. Update Profile	7
Figure 8. Add Item Page	7
Figure 9. Item Detail Page	8
Figure 10. Edit Item Page	8
Figure 11. User Item Page	9
Figure 12. Wishlist Page	9
Figure 13. Trade Management: Received Request Page	10
Figure 14. Trade Management: Sent Request Page	10
Figure 15. Advance Trade Management: Received Request Page	11
Figure 16. Advance Trade Management: Sent Request Page	11
Figure 17. Advance Trade: Sent Offer	12
Figure 18. Trade History	12
Figure 19. Trade Detail	13

Figure 20. Listed Items Page	13
Figure 21. Trade Proposal Page	14
Figure 22. Admin Login	14
Figure 23. Admin Dashboard	15
Figure 24. Admin Dashboard: Users List	15
Figure 25. Admin Dashboard: Items List	15
Figure 26. Admin Dashboard: Banned Users List	16
Figure 27: Use Case Diagram	52

## **Table of Tables**

Table 1: User Sign-Up Stimulus/Response Sequences	16
Table 2: User Sign-Up Functional Requirements	17
Table 3: User Login Stimulus/Response Sequences	18
Table 4: User Login Functional Requirements	19
Table 5: User Forgot Password Stimulus/Response Sequences	20
Table 6: User Forgot Password Functional Requirements	21
Table 7: Profile Management Stimulus/Response Sequences	21
Table 8: Profile Management Functional Requirements	22
Table 9: List Item Stimulus/Response Sequences	23
Table 10: List Item Functional Requirements	24
Table 11: Edit Item Stimulus/Response Sequences	24
Table 12: Edit Item Functional Requirements	25
Table 13: Delete Item Stimulus/Response Sequences	26
Table 14: Delete Item Functional Requirements	27
Table 15: View Item Listings Stimulus/Response Sequences	27
Table 16: View Item Listings Functional Requirements	28
Table 17: Search and Filter Item Listings Stimulus/Response Sequences	28
Table 18: Search and Filter Item Listings Functional Requirements	29
Table 19: Add to Wishlist Stimulus/Response Sequences	30
Table 20: Add to Wishlist Functional Requirements	30
Table 21: View Wishlist Stimulus/Response Sequences	31
Table 22: View Wishlist Functional Requirements	32
Table 23: Item Detail Stimulus/Response Sequences	32
Table 24: Item Detail Functional Requirements	33
Table 25: Trade Proposals Stimulus/Response Sequences	34
Table 26: Trade Proposals Functional Requirements	35
Table 27: Accept Trade Stimulus/Response Sequences	35

Table 28: Accept Trade Functional Requirements	36
Table 29: Reject Trade Stimulus/Response Sequences	36
Table 30: Reject Trade Functional Requirements	37
Table 31: Cancel Trade Stimulus/Response Sequences	38
Table 32: Cancel Trade Functional Requirements	39
Table 33: Advanced Trade Management Stimulus/Response Sequences	39
Table 34: Advanced Trade Management Offer Functional Requirements	40
Table 35: Manage Active trades Stimulus/Response Sequences	41
Table 36: Manage Active trades Functional Requirements	42
Table 37: Ban User Accounts Stimulus/Response Sequences	42
Table 38: Ban User Accounts Functional Requirements	43
Table 39: Delete List Items Stimulus/Response Sequences	43
Table 40: Delete List Items Functional Requirements	44
Table 41: Rate and Review System Stimulus/Response Sequences	45
Table 42: Rate and Review System Functional Requirements	46
Table 43: Email Notification Stimulus/Response Sequences	46
Table 44: Email Notification Functional Requirements	47
Table 45: Trade History Stimulus/Response Sequences	47
Table 46: Trade History Functional Requirements	48
Table 47: Send Messages Stimulus/Response Sequences	48
Table 48: Send Messages Functional Requirements	49
Table 49: Remove from Wishlist Stimulus/Response Sequences	49
Table 50: Remove from Wishlist Functional Requirements	50

## Revision History

Name	Date	Reason For Changes	Version

# 1. Introduction

## 1.1 Purpose

The purpose of Swapify is to revolutionize the barter trading process by providing a digital platform that efficiently connects users for direct exchanges of goods. Focusing on sustainability, Swapify enables users to trade items without the need for monetary transactions, promoting the reuse and recycling of goods that may otherwise go to waste. What sets Swapify apart is its user-friendly trade management system and community driven trust building mechanisms, including a rating system that fosters secure and reliable transactions. By redefining the way people perceive and engage in barter, Swapify creates an alternative marketplace centered on utility, value, and sustainability.

## 1.2 Document Conventions

The document is created using Google Sheets and has the font "Times New Roman". The fixed font size that has been used for the body text is 11pt. For the primary headings, we use font size 18, and for the subheadings, font size 14 has been used.

## 1.3 Intended Audience and Reading Suggestions

This SRS is intended for developers, project managers, users, testers, and documentation writers. Developers will focus on technical specifications, while project managers will use system objectives and use cases to manage timelines and testers will prepare test cases based on functional requirements. Documentation writers will use the feature descriptions to create guides. To fully understand the system's capabilities, readers should start with the platform overview before proceeding to the detailed technical specifications.

## 1.4 Product Scope

This project is focused on creating a digital platform that offers a reliable and user-friendly environment for barter trading, where users can easily exchange goods without involving money. The project scope includes the development of several key features to enhance user experience and community trust.

- **Advanced Search:** Users can filter trade items by category to quickly find specific goods.
- **User Authentication:** Includes verification processes to ensure user security and trustworthiness.
- **Trade Proposal System:** Enables users to propose, negotiate, and finalize trades directly with others.
- **Wishlist:** Users can bookmark items of interest and share them with friends for future reference.
- **Review & Rating System:** Users can rate and review trading partners to build community trust.
- **Messaging Feature:** An integrated messaging system allows users to communicate and negotiate trade details in real-time.

By focusing on these key areas, Swapify aims to provide a secure, efficient, and transparent barter trading experience that promotes sustainability and community engagement. The platform will concentrate on creating a user-friendly environment where trust and value are prioritized over monetary transactions.

## 1.5 References

- React Documentation <https://react.dev/learn>
- Node.js v22.9.0 Documentation <https://nodejs.org/docs/latest/api>
- Express 5.x API Reference <https://expressjs.com/en/5x/api.html>
- MongoDB Documentation <https://www.mongodb.com/docs>

## 2. Overall Description

### 2.1 Product Perspective

Swapify is a new web application designed to facilitate seamless barter trading among users, promoting sustainability through direct exchanges of goods without monetary transactions. It is not a follow-on product or replacement but an innovative solution to the growing demand for sustainable living and community engagement.

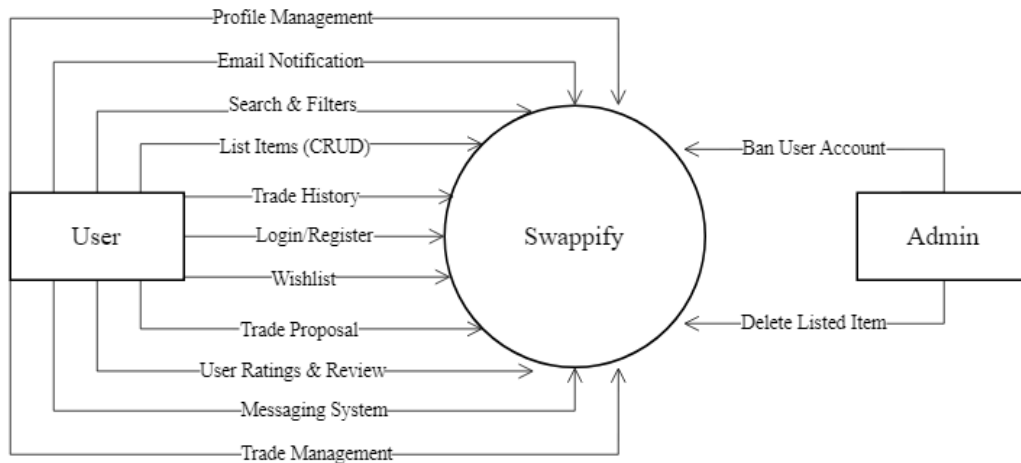


Figure 1: Product Perspective Diagram

### 2.2 Product Functions

- **User Authentication & Profile Management:** Allow users to create their accounts and manage profiles.
- **Item Listings & Search and Filters:** Enable users to list items for trade and utilize search and filtering options to find specific goods.
- **Wishlist:** Allow users to add an item to favorites.
- **Trade Proposal:** Allow users to propose trades to others based on their listings.
- **Trade Management:** Facilitate the management of ongoing trades, including acceptance, rejection, and completion processes.
- **Admin Portal:** Offer administrators tools to manage users, monitor trades, & maintain platform integrity.

- **User Ratings and Reviews:** Enable users to rate and review to build trust within the community.
- **Email Notification:** Provide users with notifications regarding important updates.
- **Trade History:** Allow users to view their past trades and interactions for reference and tracking.
- **Messaging System:** Implement a communication system for users to discuss trade details and negotiate terms directly.

## 2.3 User Classes and Characteristics

- **User (Trader):** Users can view and search for items, propose trades, and manage their profiles. They can list items for trade, add items to their wishlist, rate other users, and receive notifications regarding important updates. Additionally, users can engage with the community through a messaging system to discuss trades.
- **Admin:** Admins can view item listings, delete items as necessary, and manage user accounts. They also oversee trade activities, monitor feedback and ratings.

## 2.4 Operating Environment

- **Hardware:**
  - Operating System: Windows 10 (64-bit)
  - Processor: Core i7 - 8th Generation
  - RAM: 16gb
  - Storage: 40gb
- **Software:**
  - Visual Studio Code
  - MongoDB Compass
  - Postman
  - React JS
  - Node JS
  - Express JS
  - MongoDB

## 2.5 Design and Implementation Constraints

- **Browser Compatibility:** Swapify should support major web browsers, such as Google Chrome, Firefox, Safari, and Microsoft Edge, to ensure accessibility for a broader user base.
- **Minimal System Requirements:** Swapify should be designed to operate efficiently on devices with a minimum of 16GB of RAM and modern processors to ensure optimal performance and responsiveness across a variety of systems.

## 2.6 User Documentation

The SRS document provided to users will offer a clear understanding of how to interact with Swappify. It will be written in simple, understandable language to ensure accessibility for all users. Additionally, the documentation will include context and use case diagrams to enhance comprehension and provide insight into the app's functionalities and future developments.



## 2.7 Assumptions and Dependencies

We assume that users will have a stable internet connection and an up-to-date, modern browser to ensure optimal performance of the Swappify system. Poor connectivity or using an outdated browser may affect functionality. Furthermore, users should have a basic understanding of how to navigate the internet to effectively use the application.

# 3. External Interface Requirements

## 3.1 User Interfaces

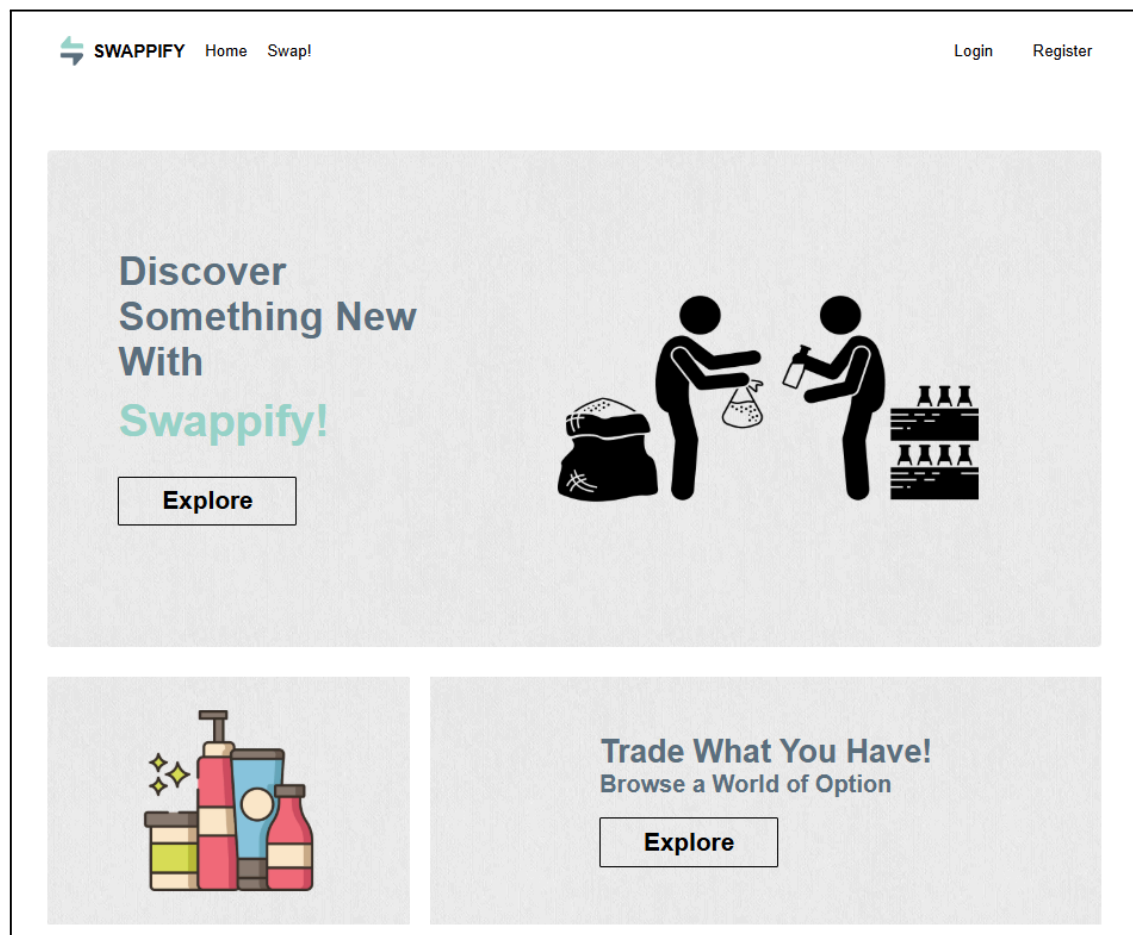



Figure 2. Home Page

 **SWAPPIFY** [Home](#) [Swap!](#)

[Login](#) [Register](#)


## Create an Account

[Register](#)

[Already have an account? Login](#)



Figure 3. Sign Up Page

 **SWAPPIFY** [Home](#) [Swap!](#)

[Login](#) [Register](#)

## Welcome Back

[Login](#)

[Don't have an account? Register](#)

[Forgot Password? Reset Here](#)



Figure 4. Login Page

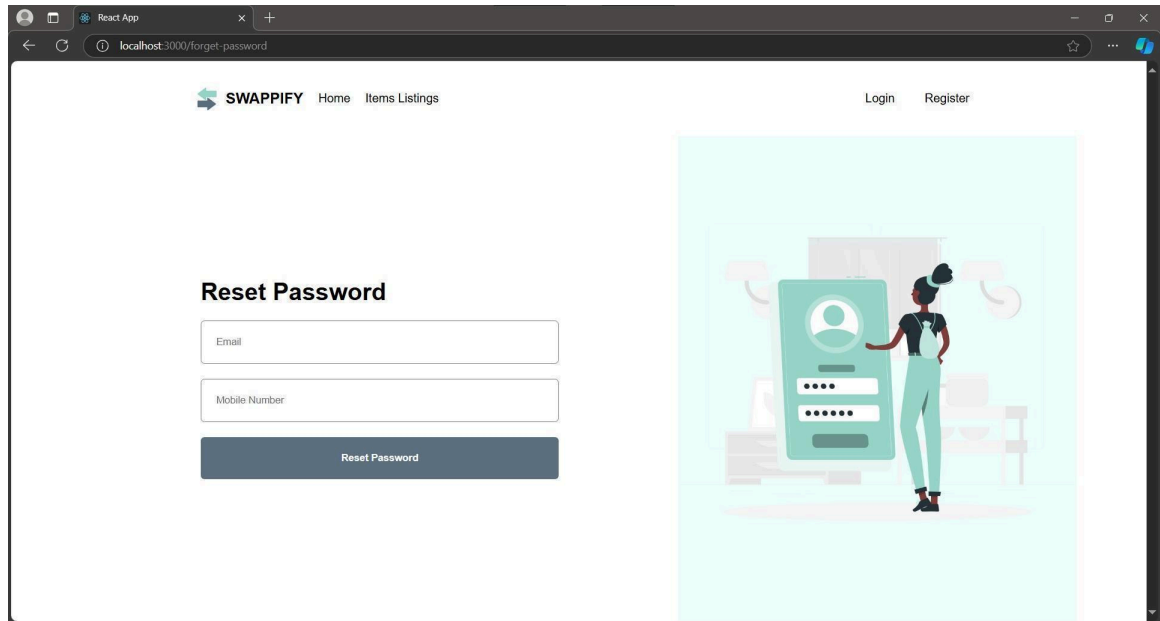


Figure 5. Forget Password Page

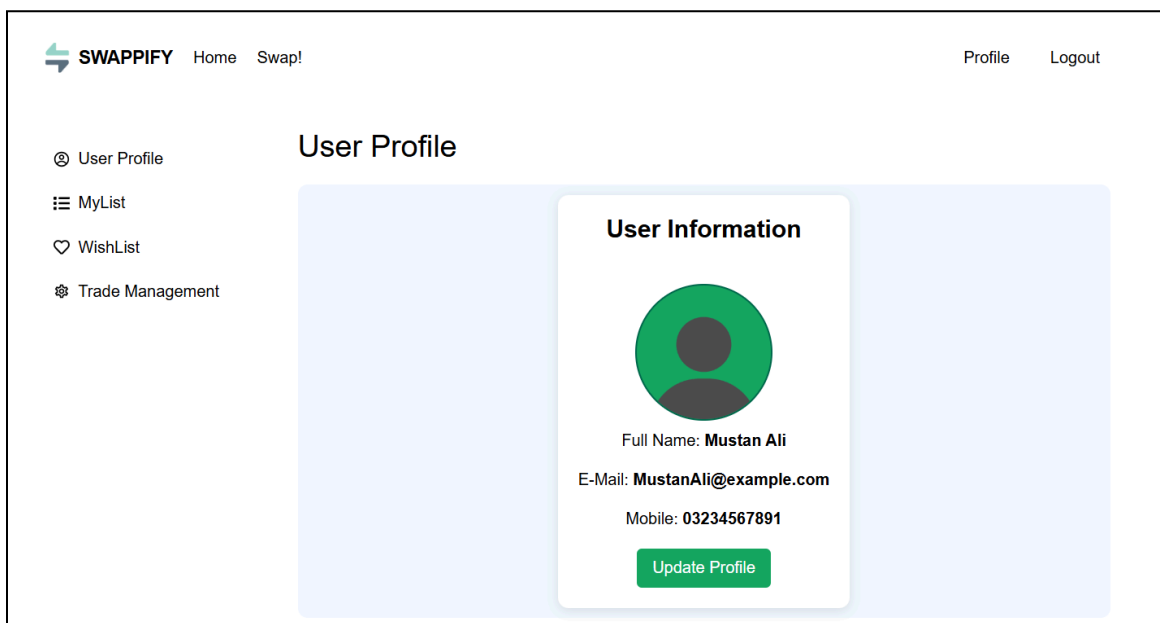


Figure 6. Profile Page

The screenshot shows a web browser window with the URL `localhost:3000/update-profile`. The page features a header with the 'SWAPPIFY' logo and navigation links for 'Home' and 'Items Listings'. On the right side of the header, there are links for 'Profile' and 'Logout'. The main content area is titled 'Update Profile' and contains a form with four input fields: 'Rohail Rathore', 'rohailrathore10@gmail.com', '03123456789', and 'Password'. Below these fields is a teal 'Update' button. To the right of the form is a large illustration of a person standing next to a large server rack.

Figure 7. Update Profile Page

The screenshot shows a web browser window with the URL `localhost:3000/add-item`. The page features a header with the 'SWAPPIFY' logo and navigation links for 'Home' and 'Items Listings'. On the right side of the header, there are links for 'Profile' and 'Logout'. The main content area is titled 'Add New Item' and contains a form with several input fields: 'Item Name', 'Item Description', 'Condition', 'Category', 'Location', 'Minimum Price', and 'Maximum Price'. Below these fields is a file upload section with a 'Choose File' button and the text 'No file chosen'. At the bottom of the form is a dark blue 'Add Item' button.

Figure 8. Add Item Page

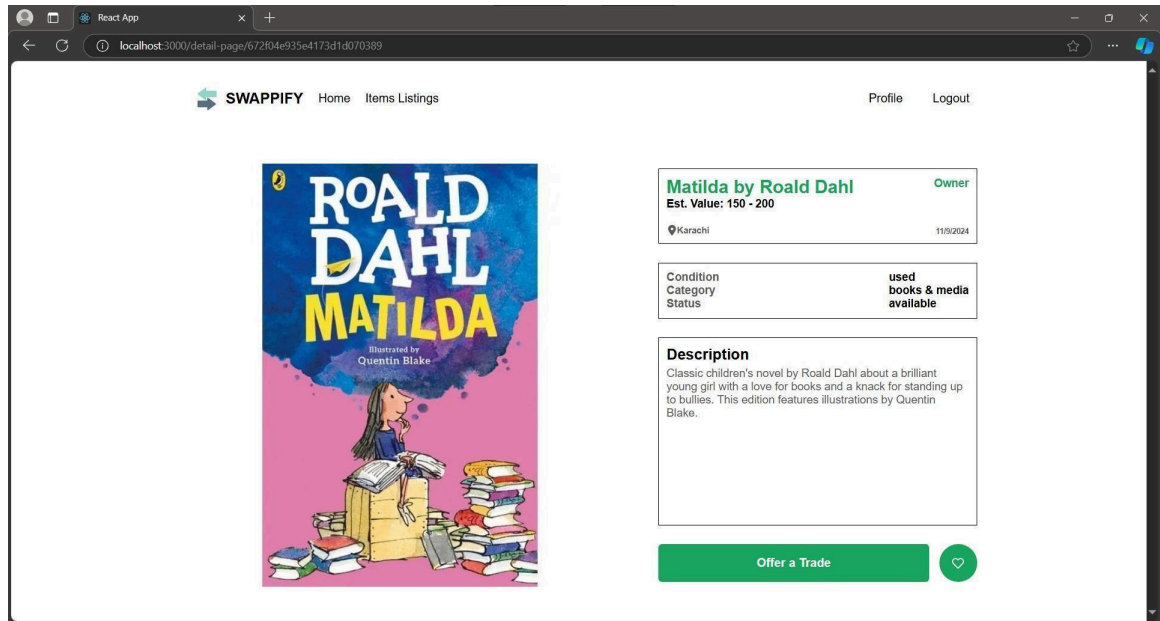


Figure 9. Item Detail Page

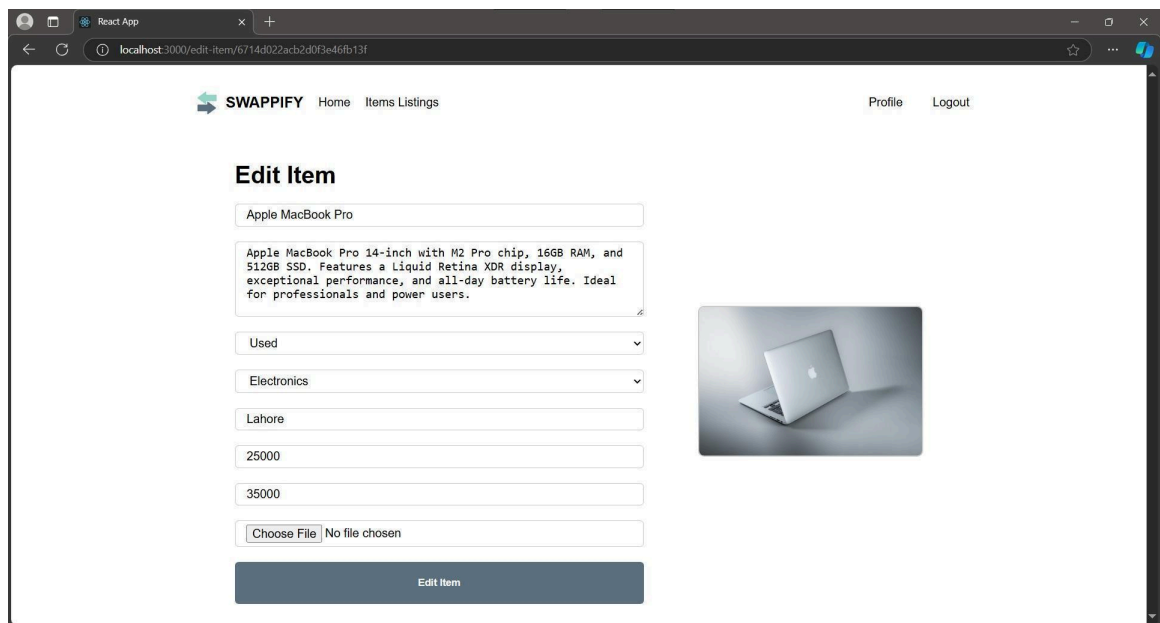


Figure 10. Edit Item Page

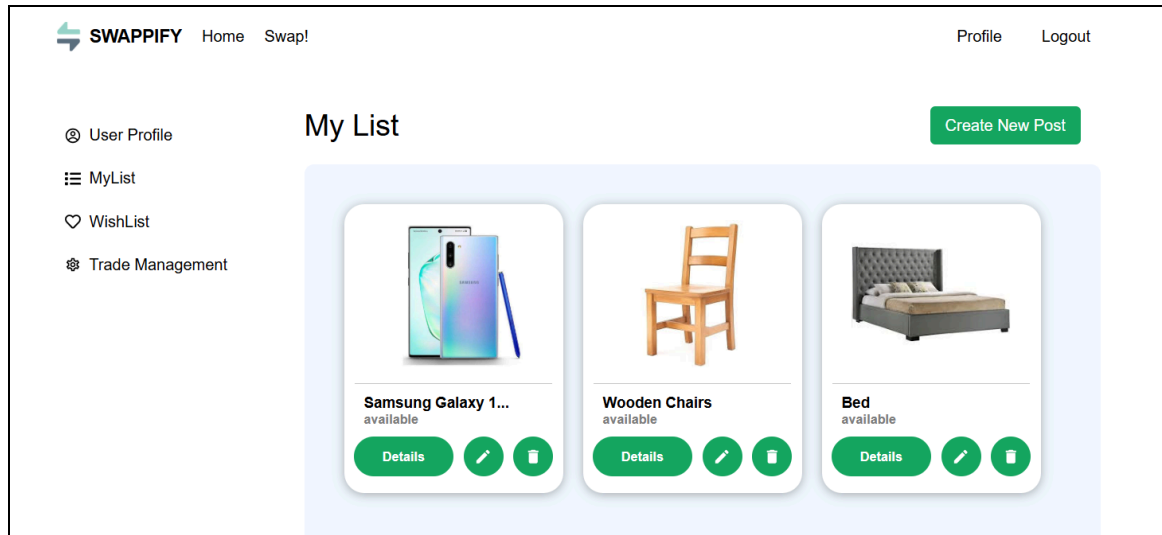


Figure 11. User Item Page

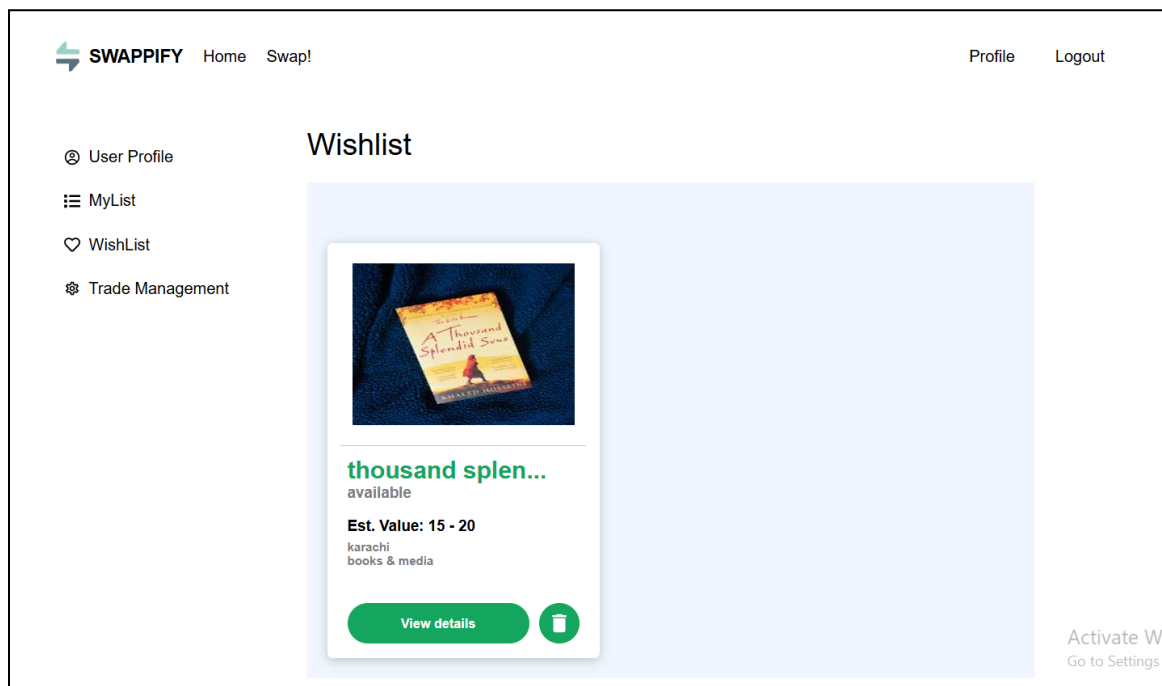


Figure 12. Wishlist Page

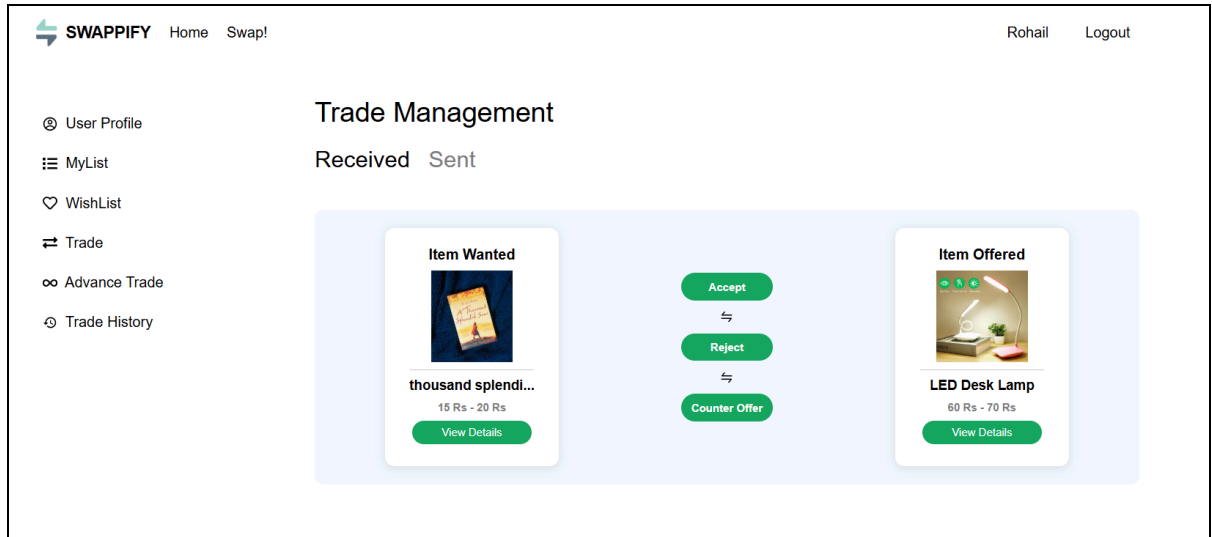


Figure 13. Trade Management: Received Request Page

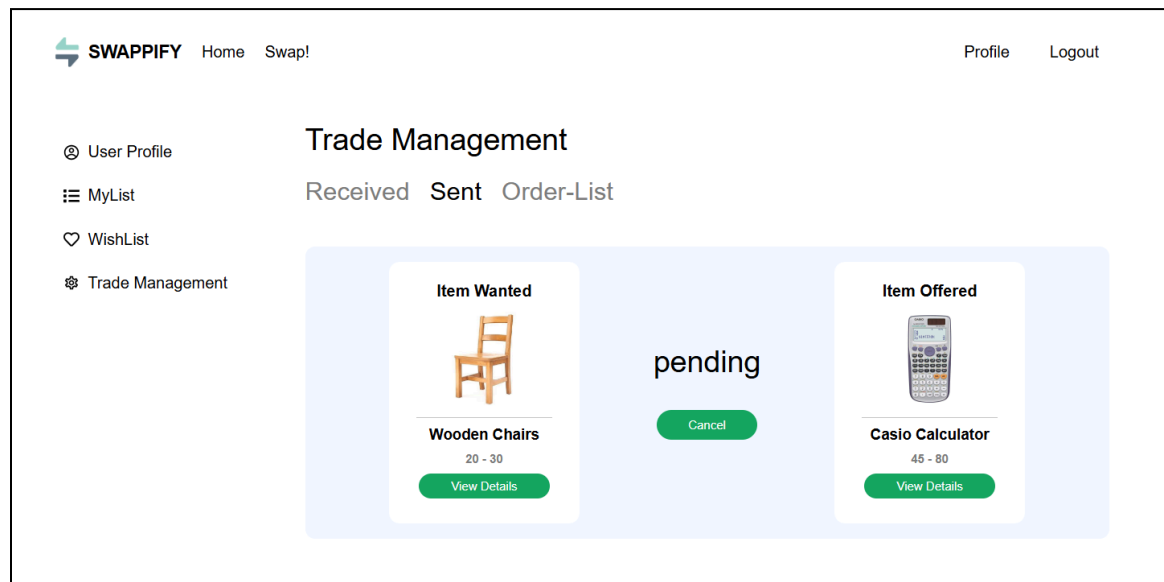


Figure 14. Trade Management: Sent Request Page

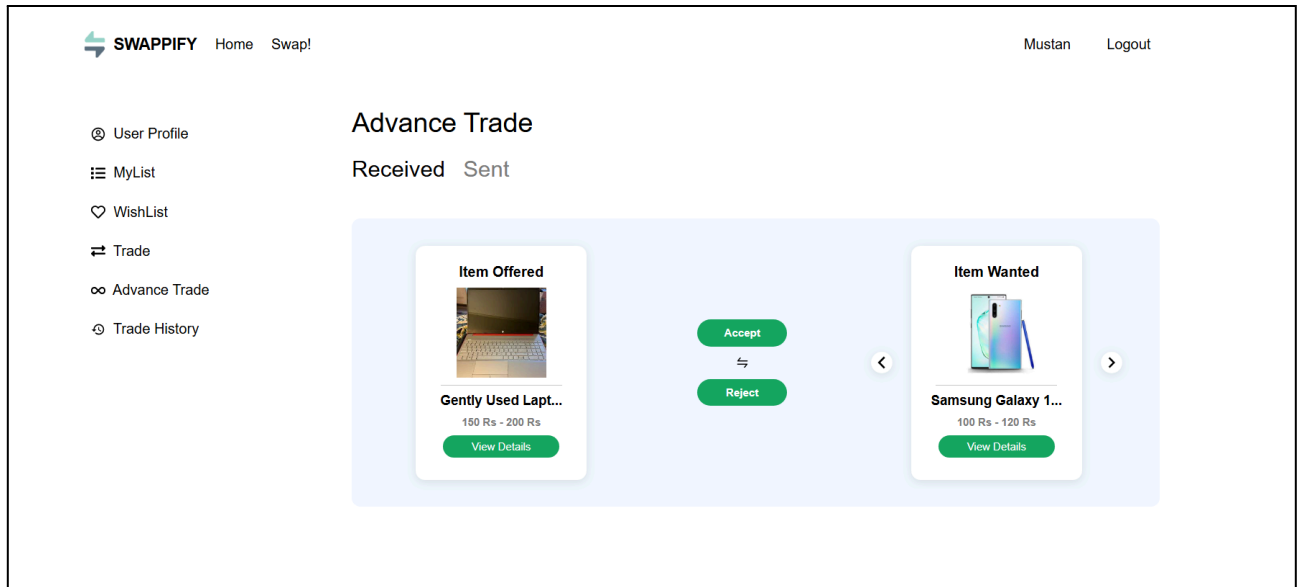


Figure 15. Advance Trade Management: Received Request Page

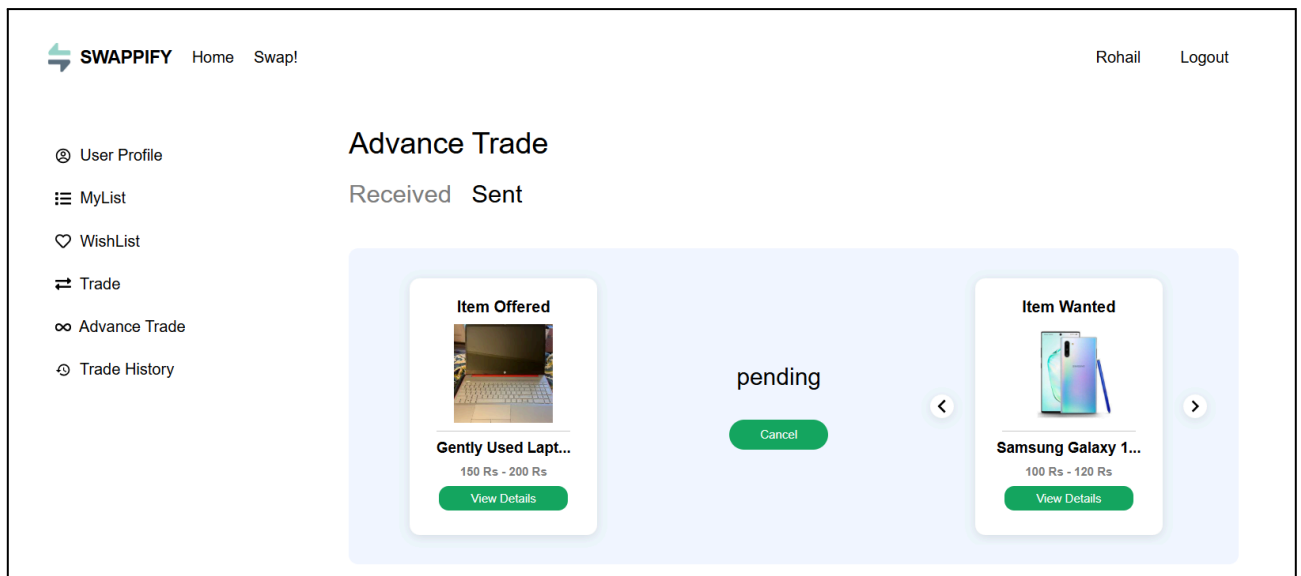


Figure 16. Advance Trade Management: Sent Request Page



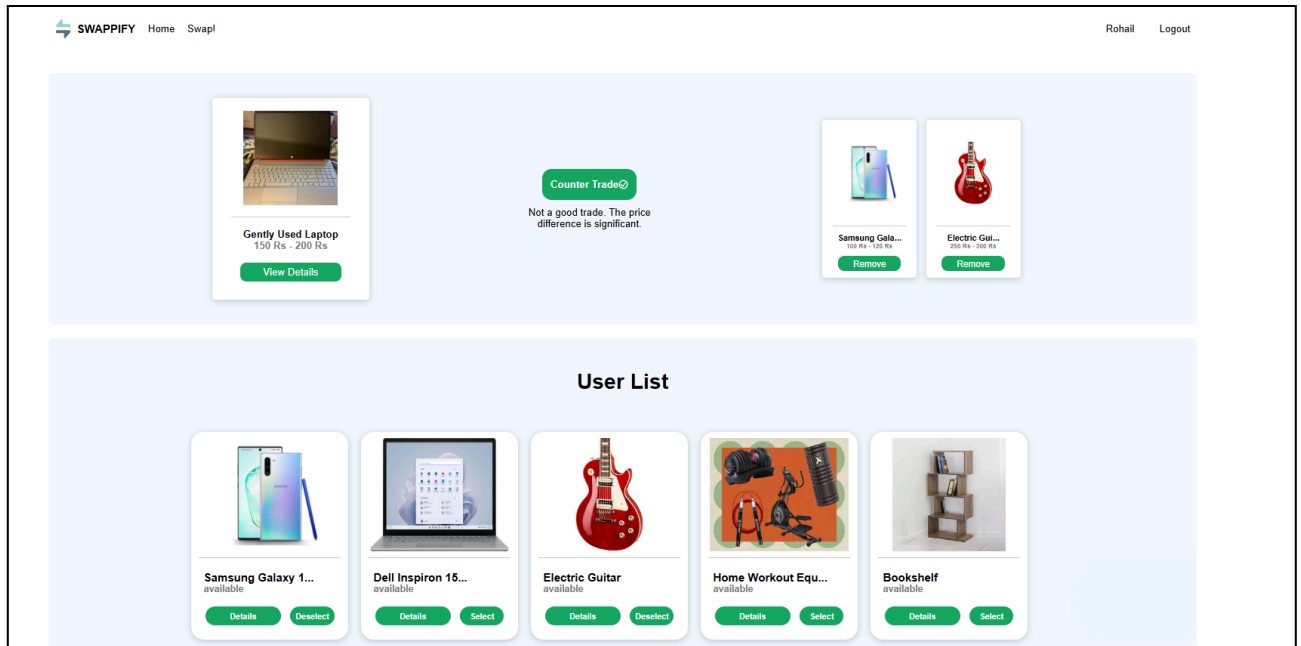


Figure 17. Advance Trade: Sent Offer

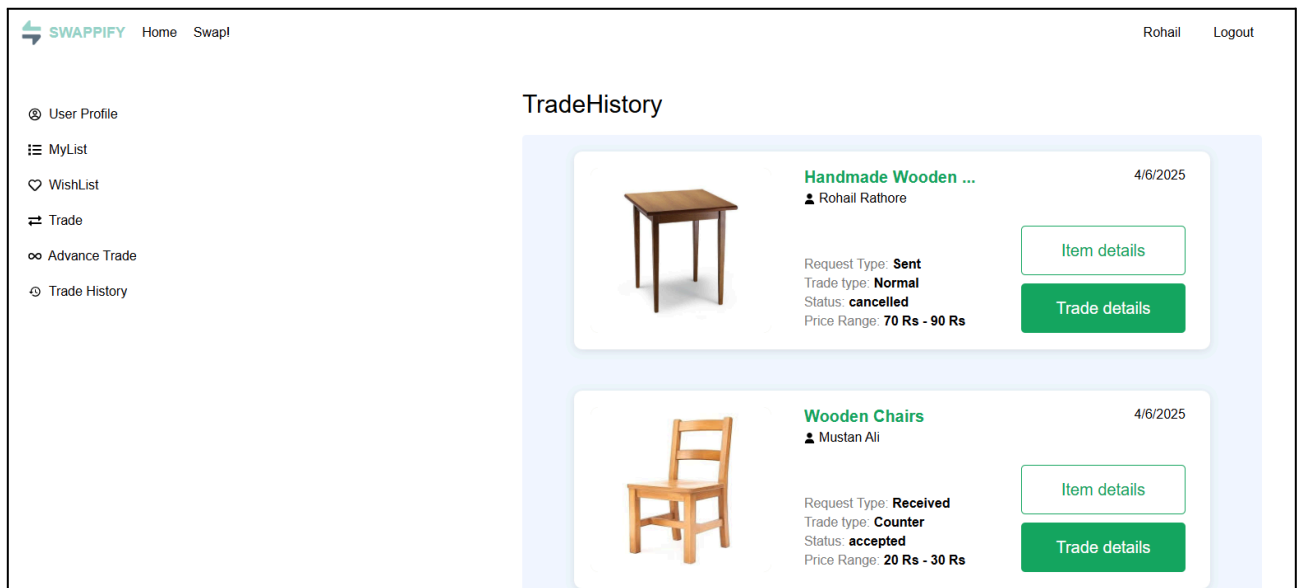


Figure 18. Trade History

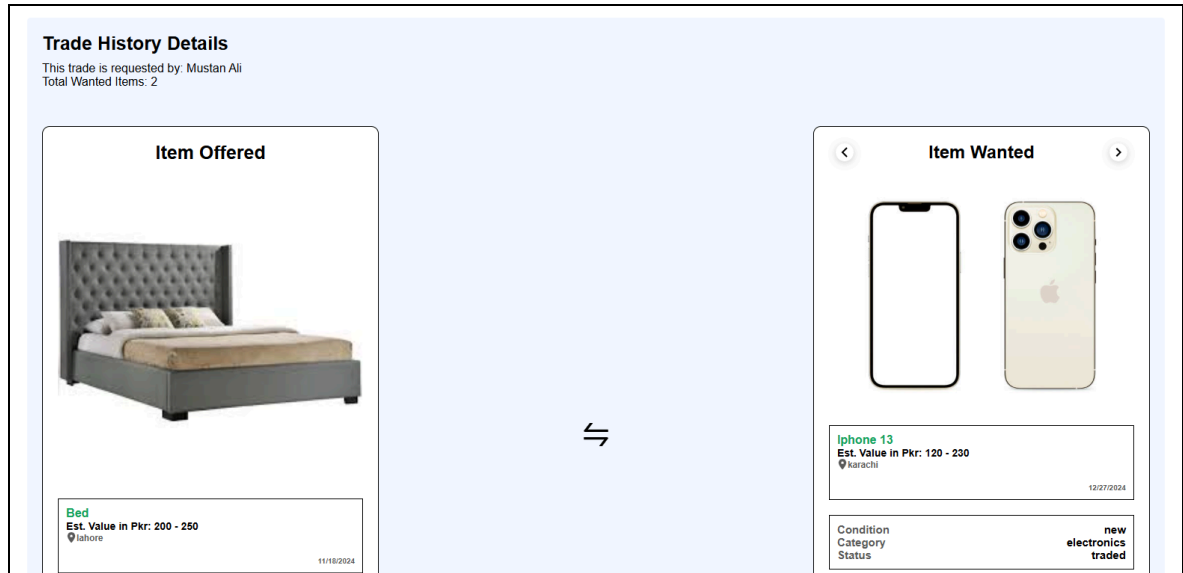


Figure 19. Trade Detail

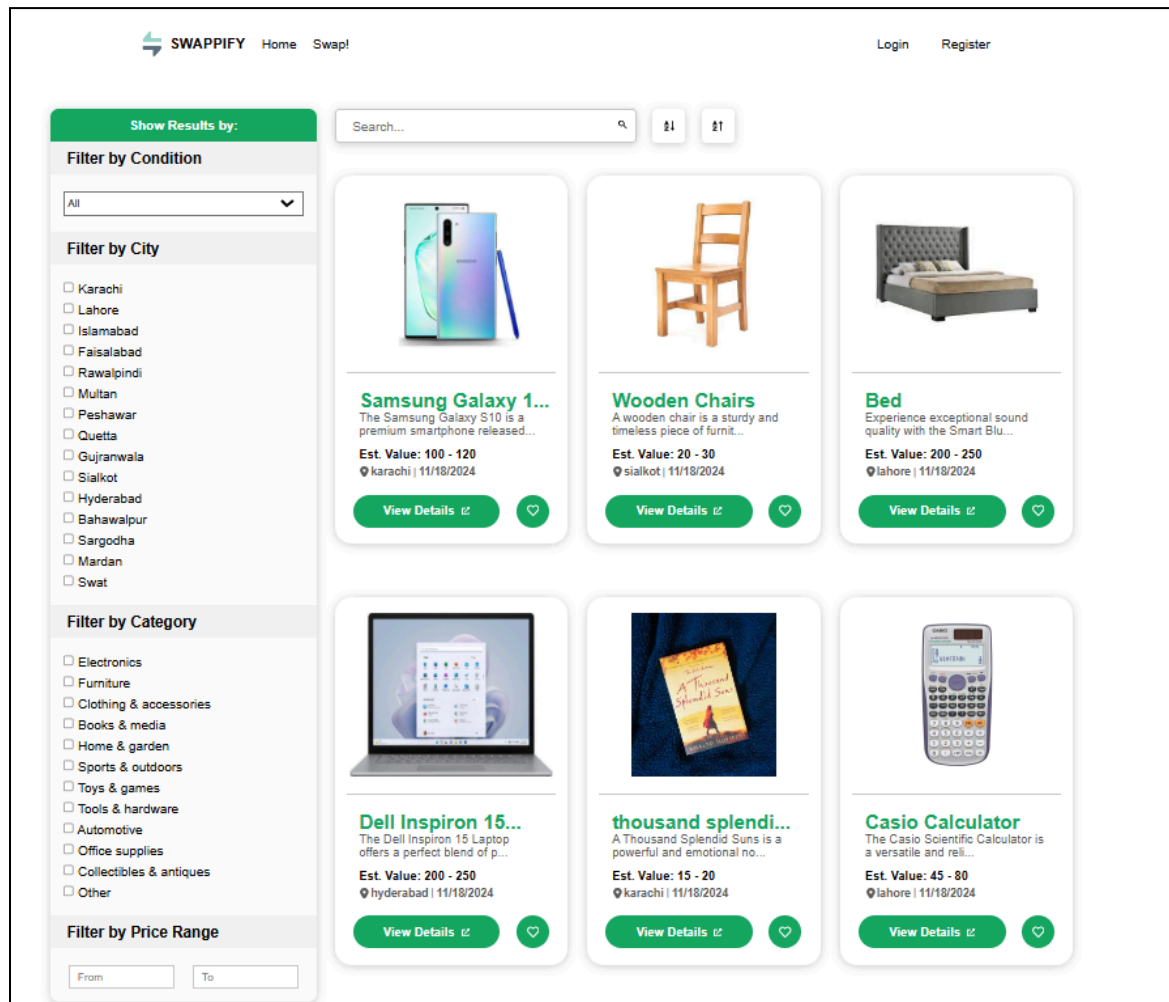


Figure 20. Listed Items Page

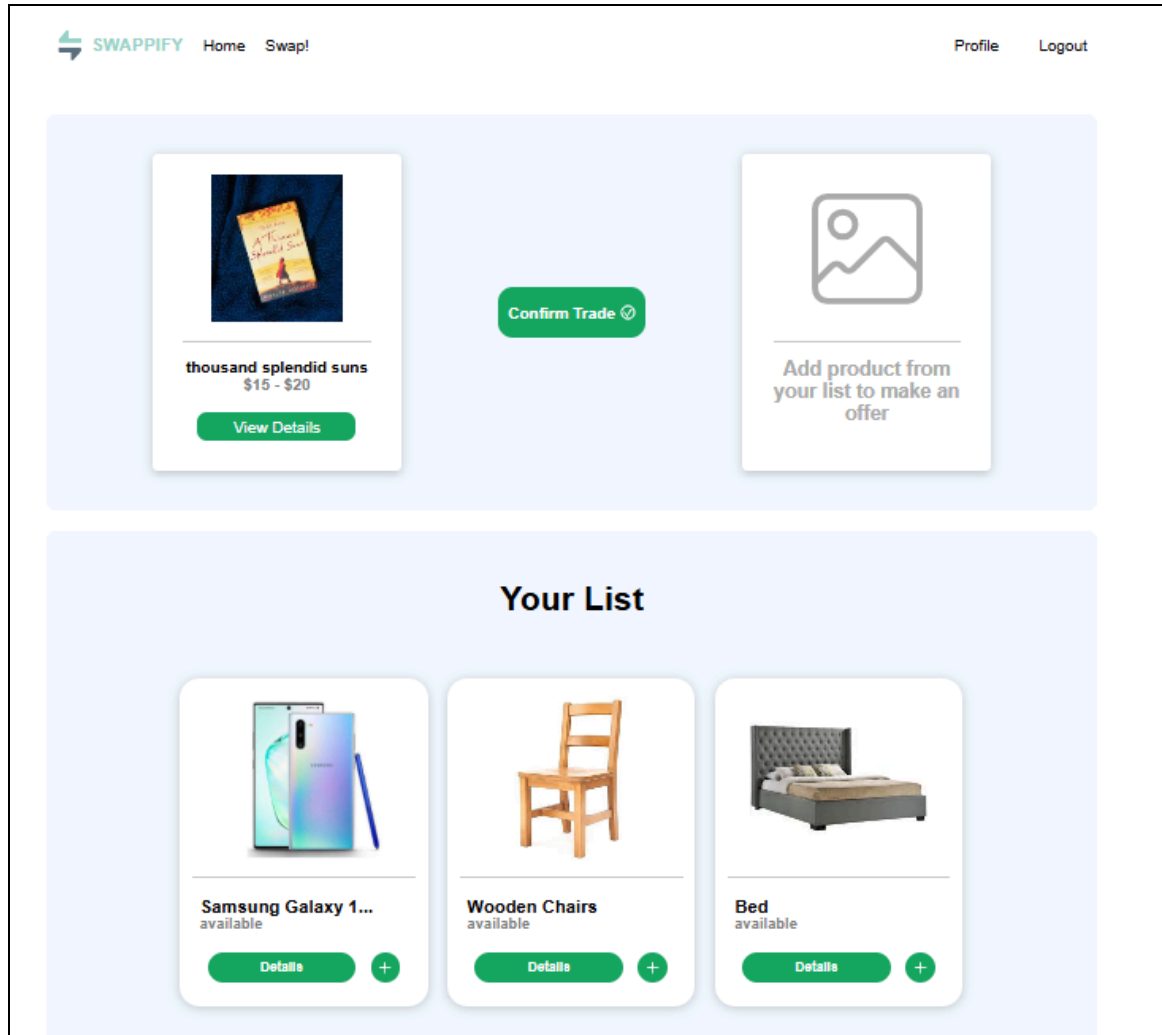


Figure 21. Trade Proposal Page

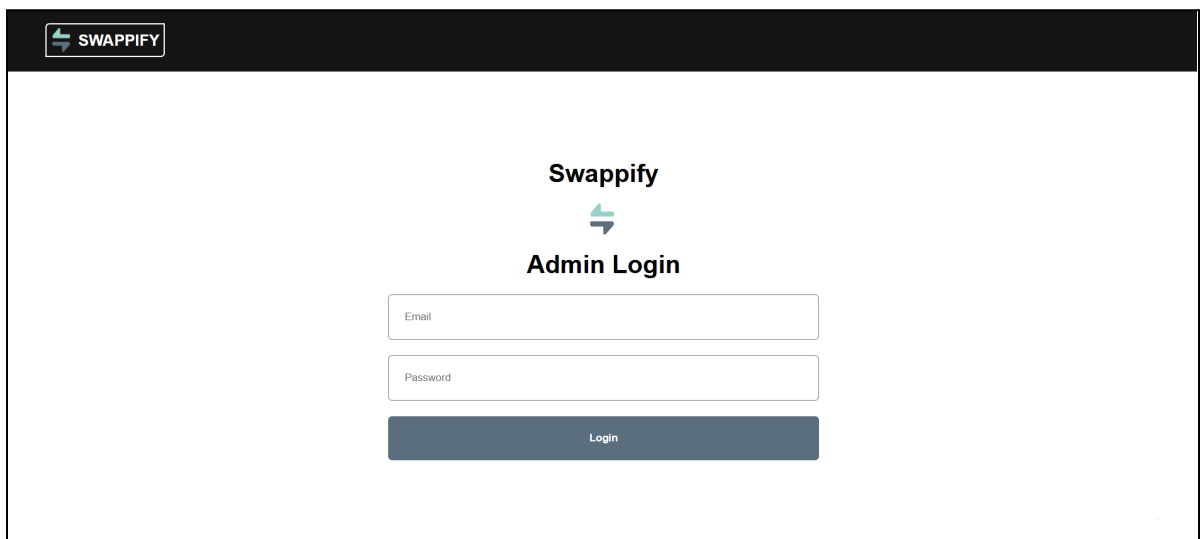


Figure 22. Admin Login

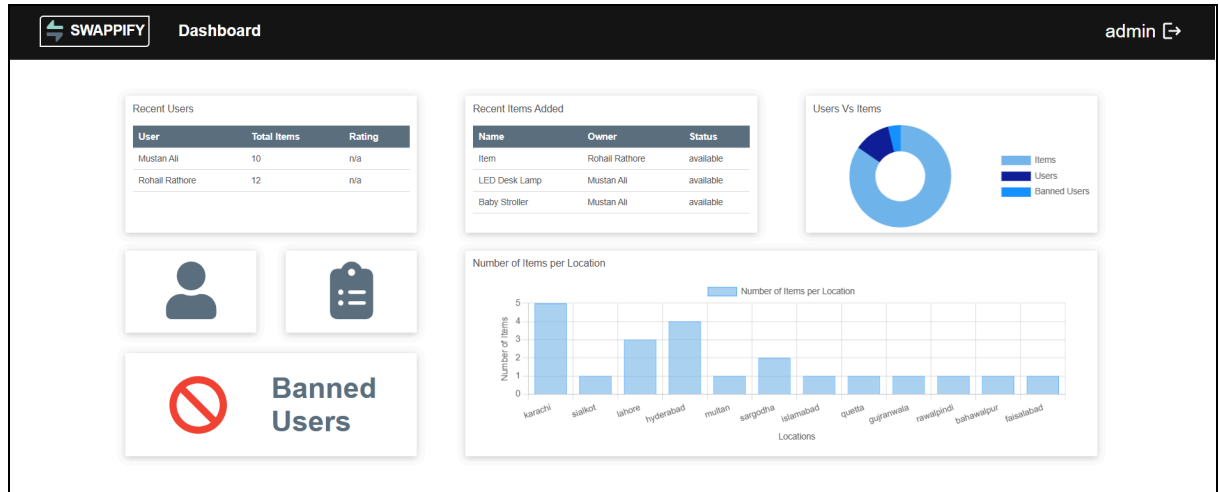


Figure 23. Admin Dashboard

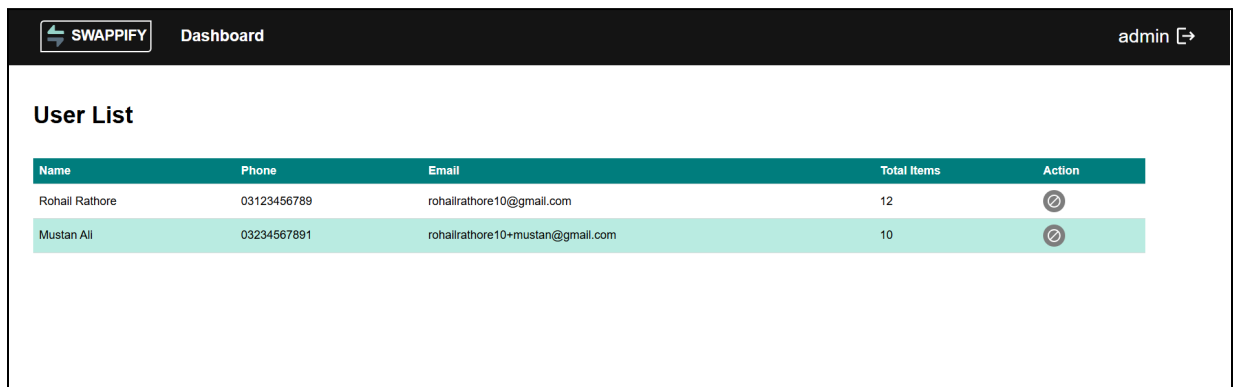


Figure 24. Admin Dashboard: Users List

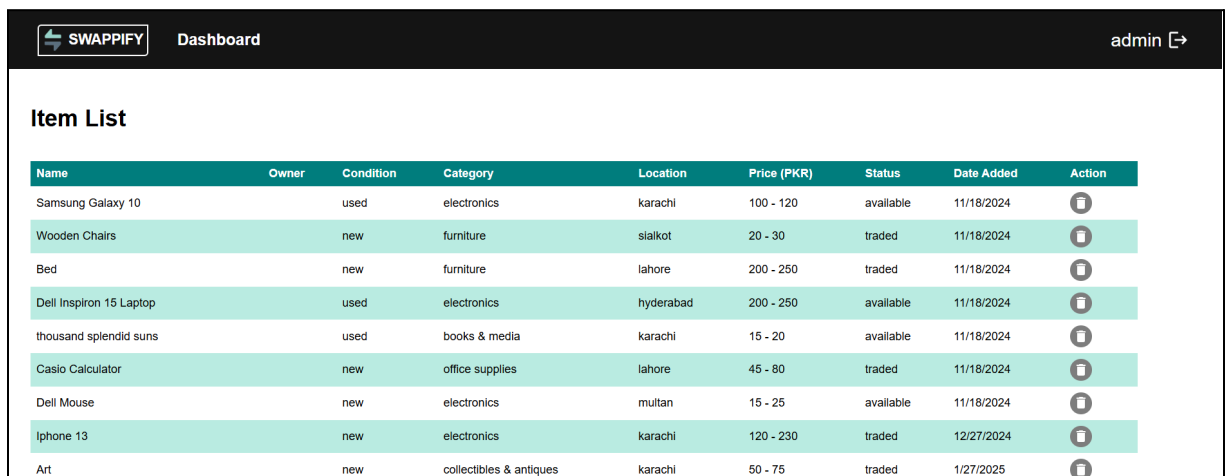


Figure 25. Admin Dashboard: Items List

SWAPPIFY Dashboard <span>admin ↗</span>				
User List				
Name	Phone	Email	Total Items	Action
user1	03123456788	rohallrathore10@hotmail.com	0	

Figure 26. Admin Dashboard: Banned Users List

## 3.2 Hardware Interfaces

- Device with stable internet connectivity, such as a desktop, laptop or tablet.
- Minimum 4GB of RAM for smooth web performance.

## 3.3 Software Interfaces

- Any modern, up-to-date web browser (e.g. Google Chrome, Firefox, Microsoft Edge).
- An active internet connection on the device.

## 3.4 Communications Interfaces

We will use HTTP communication protocols to connect with the internet, as they are widely adopted and provide a secure method for data transmission.

# 4. System Features

## 4.1 Sign Up

### 4.1.1. Description:

User sign-up feature allows new users to create an account by providing required details such as their name, email, password, and other optional information. This is the first step for users to access the platform.

### 4.1.2. Stimulus/Response Sequences:

<b>Use Case Name:</b> Sign Up
<b>Description:</b> User can sign up for an account on the platform.
<b>Actors:</b> User (Trader)
<b>Goal:</b> Create a new user account on the platform.
<b>Pre-Conditions:</b> User must not already have an account and must have an email address.

<b>Basic Course of Events /Main Flow</b>  <ol style="list-style-type: none"> <li> <b>User Action (Step 1):</b>  The user navigates to the home page and clicks on the "Sign-Up" button.  <b>System Response:</b>  The system presents the user with a sign-up form that includes fields for name, email, password, and mobile. </li> <li> <b>User Action (Step 2)</b>  The user fills in the form and clicks on the "Submit" button.  <b>System Response:</b>  The system validates the input data: <ul style="list-style-type: none"> <li>Checks for valid email format &amp; mobile.</li> <li>Ensures mandatory fields are not empty.</li> <li>The system checks if the provided email is already in use. If the email is already registered: The system shows an error message.</li> <li>If the email is not registered then the system saves the new user's data and sends a verification email.</li> </ul> </li> <li> <b>User Action (Step 3)</b>  The user opens the email and clicks on the verification link.  <b>System Response:</b>  The system verifies the link and verifies the user account. The user is then redirected to the login page with a success message. </li> </ol>			
<b>Alternate Flow (Invalid Inputs)</b> <b>User Action:</b> The user submits the form with invalid data (e.g., incorrect email format). <b>System Response:</b> The system displays appropriate error messages..			
<b>Exception Flow (System Errors)</b> <b>System Failure:</b> If the system encounters an error during processing of request. <b>System Response:</b> The system shows an error message.			
<b>Post Conditions:</b> User is redirected to Home page.			

Table 1: User Sign-Up Stimulus/Response Sequences

#### 4.1.3. Functional Requirements:

Req.ID	Description	Priority	Status
Req 1	The system shall display a sign-up form with fields for name, email, password, and mobile.	High	Complete
Req 2	The system shall validate the email, mobile format and ensure required fields are filled.	High	Complete

Req 3	The system shall verify if the provided email address is already registered in the database.	High	Complete
Req 4	The system shall store the user's information in the database if the email is not already in use.	High	Complete
Req 5	The system shall send a verification email after successful registration.	High	Complete
Req 6	The system shall activate the user's account once the verification link is clicked.	High	Complete

Table 2: User Sign-Up Functional Requirements

## 4.2 Login

### 4.2.1. Description:

The Login feature allows users to access their account by entering their email and password. This feature is crucial for user authentication, ensuring that only authorized users can access personalized functionalities.

### 4.2.2. Stimulus/Response Sequences:

<b>Use Case Name:</b> Login
<b>Description:</b> User can login to an account on the platform.
<b>Actors:</b> User (Trader)
<b>Goal:</b> Authenticate a user to access their account.
<b>Pre-Conditions:</b> User must already have an account on the platform.
<b>Basic Course of Events /Main Flow</b> <ol style="list-style-type: none"> <li> <b>User Action (Step 1):</b>  The user navigates to the home page and clicks on the "Login" button.  <b>System Response:</b>  The system presents the user with a login form that includes fields for the email and password. </li> <li> <b>User Action (Step 2):</b>  The user fills in the form with their registered email and password, then clicks on the "Submit" button.  <b>System Response:</b>  The system validates the input data: </li> </ol>

<ul style="list-style-type: none"> <li>Ensures the email is in a valid format.</li> <li>Verifies that the fields are not empty.</li> <li>If the email is not registered: The system shows an error message.</li> <li>If the email is registered but the password is incorrect: The system shows an error message.</li> </ul> <p><b>3. User Action (Step 3):</b> The user submits the correct email and password.</p> <p><b>System Response:</b> The system authenticates the user, logs them into their account, and redirects them to Home page.</p>
<p><b>Alternate Flow (Invalid Inputs)</b></p> <p><b>User Action:</b> The user submits the form with invalid inputs (e.g. incorrect email format or an empty password field).</p> <p><b>System Response:</b> The system displays error messages.</p>
<p><b>Exception Flow (System Errors)</b></p> <p><b>System Failure:</b> If the system encounters an error during processing of request.</p> <p><b>System Response:</b> The system shows an error message.</p>
<p><b>Post Conditions:</b> The user is logged in and redirected to Home page.</p>

Table 3: User Login Stimulus/Response Sequences

i.

#### 4.2.3. Functional Requirements:

Req.ID	Description	Priority	Status
Req 1	The system shall display a Login form with fields for email & password.	High	Complete
Req 2	The system shall authenticate if the email and password are valid.	High	Complete
Req 3	The system shall display error messages for any invalid input.	High	Complete

Table 4: User Login Functional Requirements

## 4.3 Forget Password

### 4.3.1 Description:

The Forgot Password feature enables users to securely reset their password by receiving a new generated password via their registered email. This process allows users to regain account access quickly and safely, ensuring account security through email verification and link validation.



### 4.3.2 Stimulus/Response Sequences:

<b>Use Case Name:</b> Forget Password
<b>Description:</b> Users can reset their password if they forget it by receiving a reset link via email.
<b>Actors:</b> User (Trader)
<b>Goal:</b> Allow users to securely reset their password.
<b>Pre-Conditions:</b> The user must have a registered email address.
<b>Basic Course of Events /Main Flow</b> <ol style="list-style-type: none"> <li> <b>1. User Action (Step 1):</b>  The user navigates to the login page and clicks on "Forgot Password."  <b>System Response:</b>  The system displays a form where the user enters their registered email address and mobile number. </li> <li> <b>2. User Action (Step 2):</b>  The user submits their email address by clicking the "Submit" button.  <b>System Response:</b>  The system checks if the email exists in the database: <ul style="list-style-type: none"> <li><b>If the email exists:</b> The system sends a generated password to the user's email and displays a confirmation message: "A password reset link has been sent to your email."</li> <li><b>If the email does not exist:</b> The system shows an error message: "Email not found. Please check and try again."</li> </ul> </li> <li> <b>3. User Action (Step 3):</b>  The user login with a new password.  <b>System Response:</b>  The system authenticates the user, logs them into their account, and redirects them to Home page. </li> </ol>
<b>Alternate Flow (Invalid Inputs)</b> <p><b>User Action:</b>  The user submits the form with invalid inputs (e.g. incorrect email format or an empty password field).</p> <p><b>System Response:</b>  The system displays error messages.</p>
<b>Exception Flow (System Errors)</b> <p><b>System Failure:</b>  If the system encounters an error during processing of request.</p> <p><b>System Response:</b>  The system shows an error message.</p>

**Post Conditions:** The user is logged in and redirected to the Home page.

**Table 5: User Forgot Password Stimulus/Response Sequences**

### 4.3.3 Functional Requirements:

Req.ID	Description	Priority	Status
Req 1	The system shall display a form for users to enter their registered email for password reset.	High	Complete
Req 2	The system shall send a password to the user's email if it is registered.	High	Complete
Req 3	The system shall display appropriate error messages for unregistered emails or invalid reset links.	High	Complete
Req 4	The system shall validate the new password before saving it.	High	Complete

**Table 6: User Forgot Password Functional Requirements**

## 4.4 Profile Management

### 4.4.1 Description:

The Profile Update feature allows users to modify their account information. This includes updating details. The feature ensures that users can keep their profile current, which is crucial for maintaining effective communication and accurate information within the platform.

### 4.4.2 Stimulus/Response Sequences:

<b>Use Case Name:</b> Profile Management
<b>Description:</b> Users can modify their account information on the platform.
<b>Actors:</b> User (Trader)
<b>Goal:</b> To allow users to modify their account information
<b>Pre-Conditions:</b> User must be logged in with a valid account.
<b>Basic Course of Events /Main Flow</b> <ol style="list-style-type: none"> <li> <b>User Action (Step 1):</b>  The user navigates to their profile page by clicking on the "Profile" button.  <b>System Response:</b>  The system presents the user with their current profile information, displaying fields like name, email, phone number. </li> <li> <b>User Action (Step 2):</b> </li> </ol>

<p>The user selects the "Edit" option to update their profile information.</p> <p><b>System Response:</b> The system allows the user to edit the available fields, such as name, email, mobile.</p> <p><b>3. User Action (Step 3):</b> The user updates the desired fields and clicks the "Save" button.</p> <p><b>System Response:</b> The system validates the updated information:</p> <ul style="list-style-type: none"> <li>Ensures mandatory fields are not empty. Checks for valid email and mobile format.</li> </ul> <p><b>4. User Action (Step 4):</b> The user submits valid updated profile information.</p> <p><b>System Response:</b> The system saves the updated profile information in the database and displays new information.</p>
<p><b>Alternate Flow (Invalid Inputs)</b></p> <p><b>User Action:</b> The user submits the form with invalid data.</p> <p><b>System Response:</b> The system displays appropriate error messages</p>
<p><b>Exception Flow (System Errors)</b></p> <p><b>System Failure:</b> If the system encounters an error during processing of request.</p> <p><b>System Response:</b> The system shows an error message.</p>
<p><b>Post Conditions:</b> User's profile information is updated successfully.</p>

Table 7: Profile Management Stimulus/Response Sequences

#### 4.4.3 Functional Requirements:

Req.ID	Description	Priority	Status
Req 1	The system shall display the user's current profile information.	High	Complete
Req 2	The system shall allow users to edit their profile information and save changes.	High	Complete
Req 3	The system shall validate the updated information, ensuring fields like email and mobile are correctly formatted.	High	Complete
Req 4	The system shall store the updated profile information in the database.	High	Complete

Req 5	The system shall display appropriate error messages for any invalid inputs.	High	Complete
-------	---	------	----------

Table 8: Profile Management Functional Requirements

## 4.5 List Item

### 4.5.1 Description:

The List Item feature enables users to list new items for trade by providing necessary details. This feature is essential for expanding the item inventory, fostering trade opportunities.

### 4.5.2 Stimulus/Response Sequences:

<b>Use Case Name:</b> List Item
<b>Description:</b> Users can list items they want to trade on the platform.
<b>Actors:</b> User (Trader)
<b>Goal:</b> Allow users to list a new item for trade on the platform.
<b>Pre-Conditions:</b> Users must be logged into their account.
<b>Basic Course of Events /Main Flow</b> <ol style="list-style-type: none"> <li> <b>User Action (Step 1):</b>  The user navigates to the "Add New Item" section.  <b>System Response:</b>  The system presents the user with a form for item details. </li> <li> <b>User Action (Step 2):</b>  The user fills out the item details and clicks the "Submit" button.  <b>System Response:</b>  The system validates the input: <ul style="list-style-type: none"> <li>Ensures mandatory fields are filled.</li> <li>Checks for valid data formats (e.g., valid image files).</li> </ul> </li> <li> <b>User Action (Step 3):</b>  The user submits valid item details.  <b>System Response:</b>  The system saves the item in the database and displays a confirmation message: "Item successfully listed." </li> </ol>
<b>Alternate Flow (Invalid Inputs)</b>

<b>User Action:</b> The user submits the form with invalid data (e.g., empty fields or incorrect file format for images). <b>System Response:</b> The system displays error messages.
<b>Exception Flow (System Errors)</b> <b>System Failure:</b> If the system encounters an error during processing of request. <b>System Response:</b> The system shows an error message.
<b>Post Conditions:</b> The new item is successfully added to the user's listings.

Table 9: List Item Stimulus/Response Sequences

### 4.5.3 Functional Requirements:

Req.ID	Description	Priority	Status
Req 1	The system shall provide a form for users to input item details, including images.	High	Complete
Req 2	The system shall validate the required fields and ensure the data formats are correct.	High	Complete
Req 3	The system shall store the new item details in the database upon successful validation.	High	Complete
Req 4	The system shall display appropriate error messages for invalid inputs or system failures.	High	Complete

Table 10: List Item Functional Requirements

## 4.6 Edit Item

### 4.6.1 Description:

The Edit Item feature allows users to modify the details of an existing item they have listed for trade. This feature is crucial for ensuring that item listings remain accurate and up-to-date, enhancing user experience and promoting effective trading.

### 4.6.2 Stimulus/Response Sequences:

<b>Use Case Name:</b> Edit Item
<b>Description:</b> Users can modify the details of an item they have listed for trade.
<b>Actors:</b> Users (Traders).

<b>Goal:</b> Allow users to update the information of items they have listed for trade.
<b>Pre-Conditions:</b> The user must be logged into their account. The user must have at least one item listed for trade.
<b>Basic Course of Events /Main Flow</b> <ol style="list-style-type: none"> <li> <b>User Action (Step 1):</b>  The user navigates to their item listings and selects an item to edit.  <b>System Response:</b>  The system displays the current details of the selected item in an editable form. </li> <li> <b>User Action (Step 2):</b>  The user updates the desired fields and clicks the "Save" button.  <b>System Response:</b>  The system validates the updated information: <ul style="list-style-type: none"> <li>Ensures mandatory fields are not empty.</li> </ul> </li> <li> <b>User Action (Step 3):</b>  The user submits the form with valid updates.  <b>System Response:</b>  The system saves the updated item details in the database and displays a confirmation message. </li> </ol>
<b>Alternate Flow (Invalid Inputs)</b> <p><b>User Action:</b> The user submits the form with invalid data (e.g., leaving mandatory fields empty).</p> <p><b>System Response:</b> The system displays appropriate error messages.</p>
<b>Exception Flow (System Errors)</b> <p><b>System Failure:</b> If the system encounters an error during processing of request.</p> <p><b>System Response:</b> The system shows an error message.</p>
<b>Post Conditions:</b> The item details are successfully updated.

Table 11: Edit Item Stimulus/Response Sequences

#### 4.6.3 Functional Requirements:

Req.ID	Description	Priority	Status
Req 1	The system shall display the current details of the item in an editable format.	High	Complete
Req 2	The system shall validate the updated fields to ensure all required	High	Complete

	fields are filled.		
Req 3	The system shall save the updated item information to the database upon validation.	High	Complete
Req 4	The system shall display error messages for invalid inputs or system errors.	High	Complete

Table 12: Edit Item Functional Requirements

## 4.7 Delete Item

### 4.7.1 Description:

The Delete Item feature allows users to remove an existing item they have listed for trade. This feature is essential for maintaining an accurate inventory, enabling users to manage their listings effectively by removing items that are no longer available for trade.

### 4.7.2 Stimulus/Response Sequences:

<b>Use Case Name:</b> Delete Item
<b>Description:</b> Users can remove an item they have listed for trade from the platform.
<b>Actors:</b> Users (Traders).
<b>Goal:</b> Allow users to delete an item they no longer want to offer for trade.
<b>Pre-Conditions:</b> The user must be logged into their account. The user must have at least one item listed for trade.
<b>Basic Course of Events /Main Flow</b>  <b>1. User Action (Step 1):</b> The user navigates to their item listings and selects an item to delete. <b>System Response:</b> The system removes the item from the listings and updates the database. A confirmation message is displayed: "Item successfully deleted."
<b>Alternate Flow:</b> None
<b>Exception Flow (System Errors)</b> <b>System Failure:</b> If the system encounters an error during processing of request. <b>System Response:</b> The system shows an error message.
<b>Post Conditions:</b> The item is successfully deleted from the user's listings and the database.

Table 13: Delete Item Stimulus/Response Sequences

### 4.7.3 Functional Requirements:

Req.ID	Description	Priority	Status
Req 1	The system shall display a button for deleting an item.	High	Complete
Req 2	The system shall remove the item from the database after the user clicks on the button.	High	Complete

Table 14: Delete Item Functional Requirements

## 4.8 View Item Listings

### 4.8.1 Description:

The View All Item Listings feature enables users to browse and view all items available for trade. Users can see essential details for each item. This feature is crucial for facilitating informed trading decisions and enhancing user engagement by providing a comprehensive overview of the items available in the marketplace.

### 4.8.2 Stimulus/Response Sequences:

<b>Use Case Name:</b> View Item Listings
<b>Description:</b> Users can browse and view all items available for trade on the platform.
<b>Actors:</b> Users (Traders).
<b>Goal:</b> Allow users to view the list of items available for trade.
<b>Pre-Conditions:</b> The user must be on the item listings page.
<b>Basic Course of Events /Main Flow</b> <ol style="list-style-type: none"> <li> <b>User Action (Step 1):</b>  The user navigates to the item listings page.  <b>System Response:</b>  The system retrieves and displays a list of all available items, including details. </li> <li> <b>User Action (Step 2):</b>  The user selects an item to view more details.  <b>System Response:</b>  The system displays the selected item's details, including a full description, images, and trade options. </li> </ol>
<b>Alternate Flow (No Items Available))</b>  <b>System Response:</b>



If no items are listed, the system displays a message.
<b>Exception Flow (System Errors)</b> <b>System Failure:</b> If the system encounters an error during processing of request. <b>System Response:</b> The system shows an error message.
<b>Post Conditions:</b> The user successfully views item listings and can select an item for more information.

Table 15: View Item Listings Stimulus/Response Sequences

### 4.8.3 Functional Requirements:

Req.ID	Description	Priority	Status
Req 1	The system shall display a list of all available items including basic details.	High	Complete
Req 2	The system shall allow users to select an item and view its detailed information.	High	Complete
Req 3	The system shall display a message when no items are available for viewing.	Medium	Complete

Table 16: View Item Listings Functional Requirements

## 4.9 Search and Filter Item Listings

### 4.9.1 Description:

The Search and Filter feature allows users to quickly find specific items by entering keywords or applying various filters. This feature is essential for improving user experience, making navigation efficient, and ensuring that users can easily access the items they are interested in.

### 4.9.2 Stimulus/Response Sequences:

<b>Use Case Name:</b> Search and Filter Item Listings
<b>Description:</b> Users can search for specific items and apply filters to narrow down the results.
<b>Actors:</b> Users (Traders).
<b>Goal:</b> Allow users to quickly find relevant items by searching keywords or applying filters.
<b>Pre-Conditions:</b> The user must be on the item listings page.
<b>Basic Course of Events /Main Flow</b>

<p><b>1. User Action (Step 1):</b> The user enters a keyword in the search bar or selects filter options (e.g., category, condition). <b>System Response:</b> The system retrieves and displays a list of items that match the search term or selected filters.</p> <p><b>2. User Action (Step 2):</b> The user views the filtered results and selects an item for more details. <b>System Response:</b> The system displays the selected item's detailed information, including description, images, and trade options.</p>
<p><b>Alternate Flow (No Results Found))</b></p> <p><b>System Response:</b> If no items match the search or filters, the system displays a message.</p>
<p><b>Exception Flow (System Errors)</b></p> <p><b>System Failure:</b> If the system encounters an error during processing of request. <b>System Response:</b> The system shows an error message.</p>
<p><b>Post Conditions:</b> The user successfully views filtered item listings and can select an item for more details.</p>

Table 17: Search and Filter Item Listings Stimulus/Response Sequences

### 4.9.3 Functional Requirements:

Req.ID	Description	Priority	Status
Req 1	The system shall provide a search bar and filters for users to search or refine item listings.	High	Complete
Req 2	The system shall display item listings based on the search or filter inputs.	High	Complete
Req 3	The system shall display a message when no items match the search or filters.	Medium	Complete

Table 18: Search and Filter Item Listings Functional Requirements

## 4.10 Add to Wishlist

### 4.10.1 Description:

The Wishlist feature allows users to save items of interest for future reference. Users can easily add items to their wishlist, enabling them to find desired items without having to search for those items repeatedly.

### 4.10.2 Stimulus/Response Sequences:

<b>Use Case Name:</b> Add to Wishlist
<b>Description:</b> Users can add items to their wishlist for future reference.
<b>Actors:</b> Users (Traders).
<b>Goal:</b> Allow users to save items to a wishlist for easier access later.
<b>Pre-Conditions:</b> The user must be logged into their account.
<b>Basic Course of Events /Main Flow</b> <ol style="list-style-type: none"> <li> <b>User Action (Step 1):</b>  The user navigates to an item listing and clicks the "Add to Wishlist" button.  <b>System Response:</b>  The system adds the item to the user's wishlist and displays a confirmation message: "Item added to wishlist." </li> <li> <b>User Action (Step 2):</b>  The user views their wishlist by navigating to their profile or the wishlist section.  <b>System Response:</b>  The system displays all items saved in the user's wishlist. </li> </ol>
<b>Alternate Flow:</b> None
<b>Exception Flow (System Errors)</b> <b>System Failure:</b> If the system encounters an error during processing of request. <b>System Response:</b> The system shows an error message.
<b>Post Conditions:</b> The user successfully adds & views items from their wishlist.

Table 19: Add to Wishlist Stimulus/Response Sequences

#### 4.10.3 Functional Requirements:

Req.ID	Description	Priority	Status
Req 1	The system shall allow users to add items to their wishlist from item listings.	High	Complete
Req 2	The system shall display the user's wishlist in a separate section of the profile.	High	Complete
Req 3	The system shall allow users to remove items from their wishlist.	High	Complete

Table 20: Add to Wishlist Functional Requirements

## 4.11 View Wishlist

### 4.11.1 Description:

The View Wishlist feature allows users to access and browse items they have previously added to their wishlist. This functionality enables users to easily find and review their saved items without needing to search for them again.

### 4.11.2 Stimulus/Response Sequences:

<b>Use Case Name:</b> View Wishlist
<b>Description:</b> Users can view and manage items saved in their wishlist.
<b>Actors:</b> Users (Traders).
<b>Goal:</b> Allow users to browse and interact with their saved wishlist items.
<b>Pre-Conditions:</b> The user must be logged into their account and must have at least one item saved in their wishlist.
<b>Basic Course of Events /Main Flow</b> <ol style="list-style-type: none"> <li> <b>2. User Action (Step 1):</b>            The user navigates to their wishlist section through the profile or wishlist menu.  <b>System Response:</b>            The system fetches the list of saved wishlist items from the database and displays them, including details such as item name, image, and description.         </li> <li> <b>3. User Action (Step 2):</b>            The user clicks on an item in the wishlist to view its full details or initiate a trade.  <b>System Response:</b>            The system displays the full details of the selected item, including owner information and trade options.         </li> </ol>
<b>Alternate Flow:</b> If the wishlist is empty: <b>System Response:</b> The system displays a message: "Your wishlist is empty. Start adding items to your wishlist."
<b>Exception Flow (System Errors)</b> <b>System Failure:</b> If the system encounters an error during the retrieval of wishlist data. <b>System Response:</b> The system displays an error message: "Unable to load wishlist. Please try again later."
<b>Post Conditions:</b> The user successfully views the items saved in their wishlist and can interact with them as needed.

**Table 21: View Wishlist Stimulus/Response Sequences**

### 4.11.3 Functional Requirements:

Req.ID	Description	Priority	Status
Req 1	The system shall fetch and display the user's wishlist items with details (e.g., name, image).	High	Complete
Req 2	The system shall allow users to view full details of a wishlist item.	High	Complete
Req 3	The system shall display an appropriate message if the wishlist is empty.	Medium	Complete
Req 4	The system shall handle errors gracefully and display a message if the wishlist cannot be loaded.	Medium	Complete

**Table 22: View Wishlist Functional Requirements**

## 4.12 Item Detail

### 4.12.1 Description:

The Item Detail feature allows users to view comprehensive information about a specific item listed on the platform. This includes details such as item name, description, condition, category, images, and owner information. Users can also take actions like initiating a trade or adding the item to their wishlist.

### 4.12.2 Stimulus/Response Sequences:

<b>Use Case Name:</b> View Item Detail
<b>Description:</b> Users can view detailed information about a specific item.
<b>Actors:</b> Users (Traders).
<b>Goal:</b> Provide users with all relevant information about an item to facilitate informed decisions.
<b>Pre-Conditions:</b> The item must exist in the database, and the user must navigate to the item's detail page.
<b>Basic Course of Events /Main Flow</b> <ol style="list-style-type: none"> <li> <b>User Action (Step 1):</b>  The user clicks on an item from the item listing page or wishlist.  <b>System Response:</b>  The system fetches the item details from the database and displays them, including: <ul style="list-style-type: none"> <li>Item name</li> <li>Description</li> </ul> </li> </ol>

<ul style="list-style-type: none"> <li>• Condition</li> <li>• Category</li> <li>• Images</li> <li>• Owner information (e.g., name, rating)</li> <li>• Available actions (e.g., "Propose Trade," "Add to Wishlist").</li> </ul> <p><b>2. User Action (Step 2):</b> The user selects an available action, such as "Propose Trade" or "Add to Wishlist."</p> <p><b>System Response:</b> The system performs the selected action and displays a confirmation message.</p>
<p><b>Alternate Flow:</b> If the item is unavailable (e.g., deleted by the owner):</p> <p><b>System Response:</b> The system displays a message: "This item is no longer available."</p>
<p><b>Exception Flow (System Errors)</b></p> <p><b>System Failure:</b> If the system encounters an error while fetching item details.</p> <p><b>System Response:</b> The system displays an error message: "Unable to load item details. Please try again later."</p>
<p><b>Post Conditions:</b> The user successfully views detailed information about the item and can take appropriate actions.</p>

Table 23: Item Detail Stimulus/Response Sequences

#### 4.12.3 Functional Requirements:

Req.ID	Description	Priority	Status
Req 1	The system shall fetch and display detailed information about an item, including images.	High	Complete
Req 2	The system shall allow users to take actions like "Propose Trade" or "Add to Wishlist."	High	Complete
Req 3	The system shall display an appropriate message if the item is unavailable.	Medium	Complete
Req 4	The system shall handle errors gracefully and display a message if the item details cannot be loaded.	Medium	Complete

Table 24: Item Detail Functional Requirements

## 4.13 Trade Proposals

### 4.13.1 Description:

The Trade Proposal feature allows users to initiate a trade by proposing an exchange of items with other users. This feature enables users to communicate their interest in specific items. Users can specify which items they wish to trade. This functionality is essential for facilitating seamless and transparent trading.

### 4.13.2 Stimulus/Response Sequences:

<b>Use Case Name:</b> Trade Proposals
<b>Description:</b> Users can propose a trade by offering an item in exchange for another user's listed item.
<b>Actors:</b> Users (Traders).
<b>Goal:</b> Allow users to propose trades for listed items.
<b>Pre-Conditions:</b> The user must be logged into their account. The user must have an item listed for trade.
<b>Basic Course of Events /Main Flow</b> <ol style="list-style-type: none"> <li> <b>User Action (Step 1):</b>  The user navigates to an item listing they want to trade for and clicks "Propose Trade."  <b>System Response:</b>  The system displays a form where the user selects their own item to offer. </li> <li> <b>User Action (Step 2):</b>  The user submits the trade proposal by clicking the "Submit" button.  <b>System Response:</b>  The system saves the proposal in the database, and notifies the other user about the trade proposal. A confirmation message is displayed: "Trade proposal sent." </li> </ol>
<b>Alternate Flow (User Cancels Proposal)</b> <p><b>User Action:</b>  The user cancels the proposal before submission.</p> <p><b>System Response:</b>  The system aborts the trade proposal, and no data is saved.</p>
<b>Exception Flow (System Errors)</b> <p><b>System Failure:</b>  If the system encounters an error during processing of request.</p> <p><b>System Response:</b>  The system shows an error message.</p>
<b>Post Conditions:</b> The trade proposal is successfully submitted and the other user is notified.

**Table 25: Trade Proposals Stimulus/Response Sequences**

### 4.13.3 Functional Requirements:

Req.ID	Description	Priority	Status
Req 1	The system shall allow users to propose trades by selecting their own item for exchange.	High	Complete
Req 2	The system shall validate the trade proposal inputs before submission.	High	Complete
Req 3	The system shall notify the other user when a trade proposal is submitted.	High	Complete

**Table 26: Trade Proposals Functional Requirements**

## 4.14 Accept Trade

### 4.14.1 Description:

The Accept Trade feature allows users to finalize a trade proposal by agreeing to the offered terms. This feature is crucial for completing transactions between users, enabling the exchange of items through the platform. It ensures smooth trade management and user engagement by facilitating successful exchanges.

### 4.14.2 Stimulus/Response Sequences:

<b>Use Case Name:</b> Accept Trade
<b>Description:</b> Users can accept a trade proposal to finalize the exchange of items.
<b>Actors:</b> Users (Traders).
<b>Goal:</b> Allow users to accept a trade proposal.
<b>Pre-Conditions:</b> The user must have received a trade proposal.
<b>Basic Course of Events / Main Flow:</b> <ol style="list-style-type: none"> <li> <b>User Action (Step 1):</b>  The user navigates to their trade proposals and selects a pending trade to review.  <b>System Response:</b>  The system displays the trade details, including the item offered. </li> <li> <b>User Action (Step 2):</b>  The user clicks the "Accept" button to confirm the trade.  <b>System Response:</b>  The system finalizes the trade by updating the trade status in the database and notifying the other user of the acceptance. A success message is displayed: "Trade accepted." </li> </ol>



<b>Alternate Flow (User Rejects Trade)</b>  <b>User Action:</b> The user clicks "Reject" instead of "Accept." <b>System Response:</b> The system updates the trade status and notifies the proposing user.
<b>Exception Flow (System Errors)</b> <b>System Failure:</b> If the system encounters an error during processing of request. <b>System Response:</b> The system shows an error message.
<b>Post Conditions:</b> The trade is successfully accepted, and both users are notified.

Table 27: Accept Trade Stimulus/Response Sequences

#### 4.14.3 Functional Requirements:

Req.ID	Description	Priority	Status
Req 1	The system shall display the details of the trade proposal for the user to review.	High	Complete
Req 2	The system shall allow users to accept the trade proposal, updating the trade status.	High	Complete
Req 3	The system shall notify the other user upon trade acceptance.	Medium	Complete

Table 28: Accept Trade Functional Requirements

### 4.15 Reject Trade

#### 4.15.1 Description:

The Reject Trade feature allows users to decline incoming trade proposals they are not interested in. This feature is essential for maintaining user control over trade interactions, ensuring users only engage in desired trades.

#### 4.15.2 Stimulus/Response Sequences:

<b>Use Case Name:</b> Reject Trade
<b>Description:</b> Users can reject a trade proposal they are not interested in.
<b>Actors:</b> Users (Traders).
<b>Goal:</b> Allow users to decline a trade proposal.

<b>Pre-Conditions:</b> The user must have received a trade proposal.
<b>Basic Course of Events /Main Flow</b>  <b>1. User Action (Step 1):</b> The user navigates to their trade proposals and selects a pending trade to review. <b>System Response:</b> The system displays the trade details, including the item offered and any terms of the trade.  <b>2. User Action (Step 2):</b> The user clicks the "Reject" button to decline the trade. <b>System Response:</b> The system updates the trade status in the database and notifies the other user of the rejection. A message is displayed: "Trade rejected."
<b>Alternate Flow User Accepts Trade))</b>  <b>User Action:</b> The user clicks "Accept" instead of "Reject." <b>System Response:</b> The system updates the trade status and notifies the other user.
<b>Exception Flow (System Errors)</b> <b>System Failure:</b> If the system encounters an error during processing of request. <b>System Response:</b> The system shows an error message.
<b>Post Conditions:</b> The trade is successfully rejected, and both users are notified.

Table 29: Reject Trade Stimulus/Response Sequences

#### 4.15.3 Functional Requirements:

Req.ID	Description	Priority	Status
Req 1	The system shall display the details of the trade proposal for the user to review.	High	Complete
Req 2	The system shall allow users to reject the trade proposal, updating the trade status.	High	Complete

Table 30: Reject Trade Functional Requirements

## 4.16 Cancel Trade

### 4.16.1 Description:

The Cancel Trade feature allows users to cancel an ongoing trade proposal that they initiated. This feature ensures that users have control over their active trades and can withdraw proposals if needed.

### 4.16.2 Stimulus/Response Sequences:

<b>Use Case Name:</b> Cancel Trade
<b>Description:</b> Users can cancel an ongoing trade proposal that they initiated.
<b>Actors:</b> Users (Traders).
<b>Goal:</b> Allow users to cancel their trade proposals before they are accepted.
<b>Pre-Conditions:</b> The user must be logged into their account and have an active trade proposal.
<b>Basic Course of Events /Main Flow</b> <ol style="list-style-type: none"> <li> <b>1. User Action (Step 1):</b>  The user navigates to the "Sent Requests" section and selects an active trade they want to cancel.  <b>System Response:</b>  The system fetches the details of the selected trade and displays a "Cancel Trade" button </li> <li> <b>2. User Action (Step 2):</b>  The user clicks on the "Cancel Trade" and confirms the cancellation.  <b>System Response:</b>  The system updates the trade status to "Cancelled" in the database.  Notify the other party about the cancellation.  Display confirmation message: "Trade has been cancelled successfully." </li> </ol>
<b>Alternate Flow:</b> If the user declines to confirm the cancellation: <b>System Response:</b> The system aborts the cancellation and keeps the trade active.
<b>Exception Flow (System Errors)</b> <b>System Failure:</b> If the system encounters an error while processing the cancellation. <b>System Response:</b> The system displays an error message: "Unable to cancel the trade. Please try again later."
<b>Post Conditions:</b> The trade is successfully cancelled, and the other party is notified.

Table 31: Cancel Trade Stimulus/Response Sequences

### 4.16.3 Functional Requirements:

Req.ID	Description	Priority	Status
Req 1	The system shall allow users to cancel active trade proposals they initiated.	High	Complete
Req 2	The system shall ask for confirmation before cancelling a trade.	High	Complete
Req 4	The system shall handle errors gracefully and display an appropriate message if cancellation fails.	Medium	Complete

Table 32: Cancel Trade Functional Requirements

## 4.17 Advanced Trade Management

### 4.17.1 Description:

The Advanced Trade Management feature allows users to create a counter trade offer, the counter offer lets the user select an item from the sender's list while keeping their previously proposed item selected. This feature facilitates seamless negotiation between traders.

### 4.17.2 Stimulus/Response Sequences:

<b>Use Case Name: Counter Offer</b>
<b>Description:</b> Users can create a counter trade offer by selecting an item from the sender's list while their item remains pre-selected.
<b>Actors:</b> Users (Traders).
<b>Goal:</b> Allow users to negotiate trades by making counter offers.
<b>Pre-Conditions:</b> The user must have received a trade proposal and must be logged into their account.
<b>Basic Course of Events /Main Flow</b> <ol style="list-style-type: none"> <li> <b>User Action (Step 1):</b>  The user navigates to the "Active Trades" section and clicks on the "Counter Offer" button for a specific trade.  <b>System Response:</b>  The system opens a trade proposal page with the recipient's item pre-selected. </li> <li> <b>User Action (Step 2):</b>  The user selects an item from the sender's listed items to include in the counter proposal.  <b>System Response:</b>  The system updates the trade proposal with the selected item from the sender. </li> </ol>

<p><b>3. User Action (Step 3):</b> The user clicks "Send Counter Offer".</p> <p><b>System Response:</b> If no item is selected from the sender's list, the system displays an error message: "Please select an item to proceed." If an item is selected:</p> <ul style="list-style-type: none"> <li>• The system saves the counter proposal in the database.</li> <li>• Notifies the sender about the counter proposal.</li> <li>• Displays confirmation: "Counter offer sent successfully."</li> </ul>
<p><b>Alternate Flow:</b> If the user cancels the counter offer:</p> <p><b>System Response:</b> The system discards the changes and navigates back to the "Active Trades" section.</p>
<p><b>Exception Flow (System Errors)</b> <b>System Failure:</b> If the system encounters an error while saving the counter proposal.</p> <p><b>System Response:</b> The system displays an error message: "Unable to send the counter offer. Please try again later."</p>
<p><b>Post Conditions:</b> The counter offer is successfully sent, and the sender is notified.</p>

Table 33: Advanced Trade Management Stimulus/Response Sequences

#### 4.17.3 Functional Requirements:

Req.ID	Description	Priority	Status
Req 1	The system shall pre-select the recipient's item in the counter offer page.	High	Complete
Req 2	The system shall allow users to select an item from the sender's list for the counter proposal.	High	Complete
Req 3	The system shall validate that at least one item is selected before sending the counter offer.	High	Complete
Req 4	The system shall notify the sender about the counter offer upon successful submission.	Medium	Complete
Req 5	The system shall display appropriate error messages if counter offer submission fails.	Medium	Complete

Table 34: Advanced Trade Management Offer Functional Requirements

## 4.18 Manage Active trades

### 4.18.1 Description:

The Manage Active Trades feature allows users to view, accept, or reject ongoing trade proposals. This feature is crucial for facilitating user engagement and ensuring that users can actively participate in trading activities.

### 4.18.2 Stimulus/Response Sequences:

<b>Use Case Name:</b> Manage Active trades
<b>Description:</b> Users can view, accept, or reject active trade proposals.
<b>Actors:</b> Users (Traders).
<b>Goal:</b> Allow users to manage their active trade proposals.
<b>Pre-Conditions:</b> The user must have at least one active trade proposal.
<b>Basic Course of Events /Main Flow</b> <ol style="list-style-type: none"> <li> <b>User Action (Step 1):</b>  The user navigates to the "Active Trades" section.  <b>System Response:</b>  The system displays a list of all active trade proposals, including the trade details and status. </li> <li> <b>User Action (Step 2):</b>  The user selects a trade to manage and either accepts or rejects it.  <b>System Response:</b>  The system updates the trade status based on the user's action and notifies the other party. A confirmation message is displayed: "Trade updated successfully." </li> </ol>
<b>Alternate Flow (No Active Trades)</b> <p><b>System Response:</b>  If there are no active trades, the system displays a message: "No active trades at the moment."</p>
<b>Exception Flow (System Errors)</b> <p><b>System Failure:</b>  If the system encounters an error during processing of request.  <b>System Response:</b>  The system shows an error message.</p>
<b>Post Conditions:</b> The trade status is successfully updated, and both users are notified.

**Table 35: Manage Active trades Stimulus/Response Sequences**

### 4.18.3 Functional Requirements:

Req.ID	Description	Priority	Status
Req 1	The system shall display a list of all active trades with relevant details.	High	Complete
Req 2	The system shall allow users to accept or reject active trade proposals.	High	Complete
Req 3	The system shall notify the other party when a trade status is updated.	Medium	Complete

Table 36: Manage Active trades Functional Requirements

## 4.19 Ban User Accounts

### 4.19.1 Description:

The Manage User Accounts feature allows admin to view user accounts and ban users if necessary. This feature is essential for maintaining a safe and secure environment on the platform.

### 4.19.2 Stimulus/Response Sequences:

<b>Use Case Name:</b> Ban User Accounts
<b>Description:</b> Administrators can ban user accounts if required.
<b>Actors:</b> Admin.
<b>Goal:</b> Allow admin to ban users.
<b>Pre-Conditions:</b> The admin must be logged into the admin portal with the required permissions.
<b>Basic Course of Events /Main Flow</b> <ol style="list-style-type: none"> <li> <b>User Action (Step 1):</b>  The admin navigates to the "User Accounts" section.  <b>System Response:</b>  The system displays a list of all user accounts with details such as account status (active, banned) and profile information. </li> <li> <b>User Action (Step 2):</b>  The admin selects a user account to ban.  <b>System Response:</b>  The system processes the admin's action, updates the user's account status in the database. A confirmation message is displayed: "User account updated successfully." </li> </ol>
<b>Alternate Flow (No Users to Manage)</b>

<p><b>User Action (Step 1):</b> The admin navigates to the "User Accounts" section.</p> <p><b>System Response:</b> The system displays a list of all user accounts, including banned accounts.</p> <p><b>User Action (Step 2):</b> The admin selects a banned user account to unban.</p> <p><b>System Response:</b> The system processes the admin's action, updates the user's account status in the database.</p> <p><b>Alternate Flow (No Users to Manage)</b></p> <p><b>System Response:</b> If there are no user accounts requiring management, the system displays a message: "No user accounts to manage."</p>
<p><b>Exception Flow (System Errors)</b></p> <p><b>System Failure:</b> If the system encounters an error during processing of request.</p> <p><b>System Response:</b> The system shows an error message.</p>
<p><b>Post Conditions:</b> The user's account is successfully updated.</p>

Table 37: Ban User Accounts Stimulus/Response Sequences

### 4.19.3 Functional Requirements:

Req.ID	Description	Priority	Status
Req 1	The system shall display a list of all user accounts with account status and information.	High	Complete
Req 2	The system shall allow the admin to ban users.	High	Complete

Table 38: Ban User Accounts Functional Requirements

## 4.20 Delete List Items

### 4.20.1 Description:

The Manage Listed Items feature enables administrators to view, edit, or remove any items listed by users on the platform. This feature is critical for maintaining the integrity of the marketplace and ensuring compliance with platform policies.

### 4.20.2 Stimulus/Response Sequences:

**Use Case Name:** Delete List Items



<b>Description:</b> Admin can remove any items listed by users on the platform.
<b>Actors:</b> Admin.
<b>Goal:</b> Allow admin to remove user-listed items, ensuring compliance with platform policies.
<b>Pre-Conditions:</b> The admin must be logged into the admin portal with the required permissions.
<b>Basic Course of Events /Main Flow</b> <ol style="list-style-type: none"> <li> <b>User Action (Step 1):</b>  The admin navigates to the "Listed Items" section.  <b>System Response:</b>  The system displays a list of all items listed by users, including item details such as title, description, category, and status. </li> <li> <b>User Action (Step 2):</b>  The admin selects an item to view or remove.  <b>System Response:</b>  The system processes the admin's action: <ul style="list-style-type: none"> <li>If the item is removed, it is deleted from the listings and the database.</li> <li>A confirmation message is displayed: "Item updated/removed successfully."</li> </ul> </li> </ol>
<b>Alternate Flow (No Items to Manage)</b> <p><b>System Response:</b>  If there are no items requiring management, the system displays a message: "No items to manage at this time."</p>
<b>Exception Flow (System Errors)</b> <p><b>System Failure:</b>  If the system encounters an error during processing of request.</p> <p><b>System Response:</b>  The system shows an error message.</p>
<b>Post Conditions:</b> The item is successfully removed, and the action is logged.

Table 39: Delete List Items Stimulus/Response Sequences

#### 4.20.3 Functional Requirements:

Req.ID	Description	Priority	Status
Req 1	The system shall display a list of all listed items with relevant details.	High	Complete
Req 2	The system shall allow the admin to view or remove listed items.	High	Complete
Req 3	The system shall delete items in the database based on the admin's action.	High	Complete

Table 40: Delete List Items Functional Requirements

## 4.21 Rate and Review System

### 4.21.1 Description:

The Rate Users After Trades feature enables users to provide feedback and ratings for their trading partners after completing a trade. This feature enhances trust within the community by allowing users to share their experiences.

### 4.21.2 Stimulus/Response Sequences:

<b>Use Case Name:</b> Rate and Review System
<b>Description:</b> Users can rate and review trading partners after completing a trade to provide feedback.
<b>Actors:</b> Users (Traders).
<b>Goal:</b> Allow users to rate and review their trade partners, fostering trust within the community.
<b>Pre-Conditions:</b> The user must have completed a trade.
<b>Basic Course of Events /Main Flow</b> <ol style="list-style-type: none"> <li> <b>User Action (Step 1):</b>  The user navigates to the "Rate Trade" section after completing a trade.  <b>System Response:</b>  The system displays a form allowing the user to give a rating (e.g., 1-5 stars) and write a review. </li> <li> <b>User Action (Step 2):</b>  The user submits the rating and review by clicking the "Submit" button.  <b>System Response:</b>  The system validates the input and saves the rating and review to the database. A confirmation message is displayed: "Rating and review submitted successfully." </li> </ol>
<b>Alternate Flow:</b> None
<b>Exception Flow (System Errors)</b> <b>System Failure:</b> If the system encounters an error during processing of request. <b>System Response:</b> The system shows an error message.
<b>Post Conditions:</b> The rating and review are successfully submitted.

Table 41: Rate and Review System Stimulus/Response Sequences

### 4.21.3 Functional Requirements:

Req.ID	Description	Priority	Status
Req 1	The system shall allow users to submit ratings and reviews for trade partners.	High	Pending
Req 2	The system shall save the rating and review to the database and update the partner's profile.	High	Pending

Table 42: Rate and Review System Functional Requirements

## 4.22 Email Notification

### 4.22.1 Description:

The Alert Users to Important Trade Updates feature notifies users about significant changes in trades, such as trade changes, new trade proposals, or trade cancellations. This ensures users are promptly informed to make timely decisions.

### 4.22.2 Stimulus/Response Sequences:

<b>Use Case Name:</b> Email Notification
<b>Description:</b> The system sends email notifications to users regarding important updates such as trade proposals, advance trade or account update.
<b>Actors:</b> Users (Traders).
<b>Goal:</b> Notify users about important updates via email.
<b>Pre-Conditions:</b> The user must have a registered and verified email address. The user must have relevant activities triggering notifications.
<b>Basic Course of Events /Main Flow</b> <ol style="list-style-type: none"> <li> <b>System Action (Step 1):</b>  A relevant event occurs (e.g., new trade proposal, trade accepted, account update). <b>System Response:</b>  The system generates an email with the details of the event and sends it to the user's registered email address. </li> </ol>
<b>Alternate Flow:</b> None
<b>Exception Flow (System Errors)</b> <b>System Failure:</b> If the system encounters an error during processing of request. <b>System Response:</b> The system shows an error message.

<b>Post Conditions:</b> The user is notified of important updates via email.
--

Table 43: Email Notification Stimulus/Response Sequences

### 4.22.3 Functional Requirements:

Req.ID	Description	Priority	Status
Req 1	The system shall generate and send email notifications to users for relevant platform activities.	High	Complete

Table 44: Email Notification Functional Requirements

## 4.23 Trade History

### 4.23.1 Description:

The View Past Trades feature allows users to access a history of their completed trades, enhancing transparency and record-keeping. This feature is essential for users to track their trading activities and assess their trading patterns.

### 4.23.2 Stimulus/Response Sequences:

<b>Use Case Name:</b> Trade History
<b>Description:</b> Users can view a history of their past trades on the platform.
<b>Actors:</b> Users (Traders).
<b>Goal:</b> Allow users to access a record of their past trades for reference.
<b>Pre-Conditions:</b> The user must be logged into their account. The user must have completed at least one trade.
<b>Basic Course of Events /Main Flow</b> <ol style="list-style-type: none"> <li> <b>User Action (Step 1):</b>  The user navigates to the "Trade History" section.  <b>System Response:</b>  The system retrieves and displays a list of all the user's completed trades, including details such as the traded items, trade partners, and trade dates. </li> <li> <b>User Action (Step 2):</b>  The user selects a specific trade to view more details.  <b>System Response:</b>  The system displays the full details of the selected trade, including items traded, counterparties, and the status of the trade. </li> </ol>

<b>Alternate Flow (No Trade History)</b>  <b>System Response:</b> If the user has no completed trades, the system displays a message: "No trade history available."
<b>Exception Flow (System Errors)</b> <b>System Failure:</b> If the system encounters an error during processing of request. <b>System Response:</b> The system shows an error message.
<b>Post Conditions:</b> The user successfully views their trade history.

Table 45: Trade History Stimulus/Response Sequences

### 4.23.3 Functional Requirements:

Req.ID	Description	Priority	Status
Req 1	The system shall display a list of all completed trades for the user.	High	Complete
Req 2	The system shall allow users to view detailed information about each completed trade.	High	Complete
Req 3	The system shall display a message if no trade history is available for the user.	High	Complete

Table 46: Trade History Functional Requirements

## 4.24 Send Messages

### 4.24.1 Description:

The Send Messages feature enables users to communicate with other users to negotiate trades or arrange meetups. This functionality fosters interaction and collaboration within the platform.

### 4.24.2 Stimulus/Response Sequences:

<b>Use Case Name:</b> Send Messages
<b>Description:</b> Users can send messages to other users to negotiate trades or communicate regarding items.
<b>Actors:</b> Users (Traders).
<b>Goal:</b> Allow users to communicate with each other through the platform's messaging system.
<b>Pre-Conditions:</b> The user must be logged into their account. The user must have an active trade or user they want to message.

<b>Basic Course of Events /Main Flow</b>  <b>1. User Action (Step 1):</b> The user navigates to a trade or selects a user to message. <b>System Response:</b> The system displays a messaging interface where the user can compose a message.  <b>2. User Action (Step 2):</b> The user types the message and clicks "Send." <b>System Response:</b> The system delivers the message to the recipient.
<b>Alternate Flow:</b> None
<b>Exception Flow (System Errors)</b> <b>System Failure:</b> If the system encounters an error during processing of request. <b>System Response:</b> The system shows an error message.
<b>Post Conditions:</b> The message is successfully sent, and the recipient receives it.

Table 47: Send Messages Stimulus/Response Sequences

#### 4.24.3 Functional Requirements:

Req.ID	Description	Priority	Status
Req 1	The system shall provide a messaging interface where users can type and send messages.	High	Pending
Req 2	The system shall deliver messages in real-time to the recipient.	High	Pending

Table 48: Send Messages Functional Requirements

### 4.25 Remove from Wishlist

#### 4.25.1 Description:

The Wishlist feature allows users to save items of interest for future reference. Users can easily remove items from their wishlist.

#### 4.25.2 Stimulus/Response Sequences:

<b>Use Case Name:</b> Send Messages
<b>Description:</b> Users can send messages to other users to negotiate trades or communicate regarding items.
<b>Actors:</b> Users (Traders).

<b>Goal:</b> Allow users to communicate with each other through the platform's messaging system.
<b>Pre-Conditions:</b> The user must be logged into their account. The user must have an active trade or user they want to message.
<b>Basic Course of Events /Main Flow</b>  <b>3. User Action (Step 1):</b> The user navigates to a trade or selects a user to message. <b>System Response:</b> The system displays a messaging interface where the user can compose a message.  <b>4. User Action (Step 2):</b> The user types the message and clicks "Send." <b>System Response:</b> The system delivers the message to the recipient.
<b>Alternate Flow:</b> None
<b>Exception Flow (System Errors)</b> <b>System Failure:</b> If the system encounters an error during processing of request. <b>System Response:</b> The system shows an error message.
<b>Post Conditions:</b> The message is successfully sent, and the recipient receives it.

Table 49: Remove from Wishlist Stimulus/Response Sequences

### 4.25.3 Functional Requirements:

Req.ID	Description	Priority	Status
Req 1	The system shall allow users to remove items from their wishlist.	High	Complete
Req 2	The system shall display the user's wishlist in a separate section of the profile.	High	Complete
Req 3	The system shall allow users to remove items from their wishlist.	High	Complete

Table 50: Remove from Wishlist Functional Requirements

## 5. Other Nonfunctional Requirements

### 5.1 Performance Requirements

- The system should be able to handle up to 1000 concurrent users without performance degradation.

- Page load times should not exceed 5 seconds under normal conditions.
- The server response time should be within a few seconds for all user requests.

## 5.2 Safety Requirements

- In case of system failure, the application should have a well-defined recovery plan, including disaster recovery protocols, to restore services as quickly as possible.
- The platform must comply with relevant safety standards and regulations, such as data protection laws, to safeguard user information and maintain platform security.

## 5.3 Security Requirements

- Authentication mechanisms must be implemented to ensure that only authorized users can access restricted features.
- User passwords should be stored using secure hashing algorithms.
- All user data should be encrypted during transmission.

## 5.4 Software Quality Attributes

- The application should maintain 99.9% uptime to ensure high availability.
- The interface should be user-friendly.
- The codebase should be modular and documented to facilitate easy maintenance and future updates.
- The system should be designed to handle increasing user loads as the platform grows.

## 5.5 Business Rules

- Only registered users are allowed to list items for trade.
- Admins have the right to remove any inappropriate listings or users violating community guidelines.
- Users must agree to the terms before they can engage in any trading activity on the platform.

# 6. Other Requirements

A stable internet connection is required for optimal performance and functionality of the application.

## 4. Appendix A: Glossary

- **Barter Trading:** A system of exchange in which goods or services are traded directly for other goods or services without using money.
- **HTTP:** Hypertext Transfer Protocol, the foundation of data communication on the web.
- **HTTPS:** Hypertext Transfer Protocol Secure, an extension of HTTP that uses encryption for secure data transmission.
- **Data Encryption:** Process of converting data into a coded format to prevent unauthorized access.
- **User Authentication:** The process of verifying the identity of a user trying to access the system.



## 5. Appendix B: Analysis Models

### a. Use Case Diagram

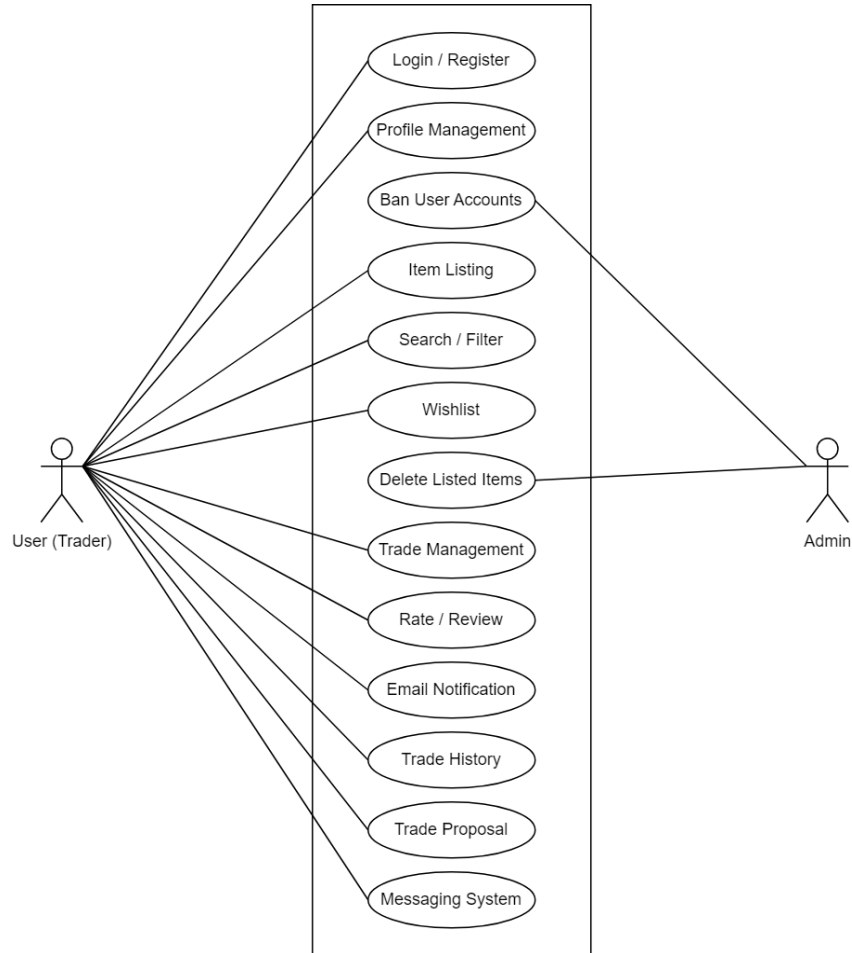


Figure 27: Use Case Diagram

## 6. Appendix C: To Be Determined List

Not Applicable