



Final Year Project

Swappify

Project Team:

Mustan Ali
2112121

Rohail Rathore
2012362

Project Supervisor:

Mr. Khawaja Mohiuddin

June 25th , 2025

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science in Computer Science
in the

Faculty of Computing and Engineering Sciences

Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology University

(SZABIST University) Karachi Campus

Declaration of Authorship

We, the undersigned, hereby declare that the project titled "*Swappify*" is our original work and has been completed in accordance with the academic standards of SZABIST University under the supervision of **Mr. Khawaja Mohiuddin**.

We confirm that all sources and references used during the research and development of this project have been properly cited and acknowledged. The content of this report is original and has not been copied or plagiarized from any external source.

We further declare that this project has not been previously submitted for the award of any degree, diploma, or other qualification at any other university or institution.

Signed:

Mustan Ali (2112121)

Rohail Rathore (2012362)

Date: June 25th , 2025

Project Description

Swappify is a modern digital platform that reimagines barter trading for the online world. In an age where sustainability and conscious consumption are increasingly important, Swappify offers a secure, user-friendly space for individuals to exchange goods and services without using money. The platform fosters a circular economy by encouraging the reuse, repurposing, and recycling of items that might otherwise go to waste.

Unlike traditional marketplaces that focus solely on monetary transactions, Swappify emphasizes trust, community, and value-based trade. Core features include secure user authentication, personalized profiles, advanced item search, structured trade proposals, and a post-trade rating system to enhance user credibility. The platform simplifies the barter process, making it efficient, transparent, and accessible to a broad audience.

By enabling people to trade directly and safely, Swappify promotes sustainability while creating meaningful, cost-effective exchanges driven by utility rather than price.

Tech Stack:

- Frontend: React.js
- Backend: Node.js, Express.js
- Database: MongoDB (with Mongoose)
- Authentication: JWT, Bcrypt.js
- File Uploads: Multer
- Real-time Features: Socket.io
- Email Service: Nodemailer
- API Communication: Axios

Acknowledgement

We would like to express our heartfelt gratitude to our respected supervisor, **Mr. Khawaja Mohiuddin**, from the Computer Science faculty at Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology, for his continuous support, valuable guidance, and encouragement throughout this project. His mentorship was instrumental in the successful completion of our work.

We also extend our sincere thanks to all our teachers at SZABIST, whose knowledge, support, and dedication have greatly contributed to our learning and development during this journey.

Plagiarism-Free Certificate

This is to certify that the project titled "**Swappify**" has been successfully completed and submitted by the undersigned students. The project was carried out under the supervision of **Mr. Khawaja Mohiuddin** at **SZABIST University**

We hereby affirm that the content of this report is entirely original and has not been copied or reproduced from any external source. All references and materials used in the research and development of this project have been properly cited and acknowledged.

This project fully complies with the academic and ethical standards of **SZABIST University**.

Furthermore, we declare that this work is our own and has not been submitted elsewhere for the award of any degree or qualification.

Name: Mustan Ali

Registration #: 2112121

Signature: _____

Name: Rohail Rathore

Registration #: 2012362

Signature: _____

Supervisor: Mr. Khawaja Mohiuddin

Signature: _____

Turnitin Originality Report

Processed on: 24-Jun-2025 11:52 PKT

ID: 2696132406

Word Count: 31822

Submitted: 6

Swappify - Documentation.pdf

Similarity Index	Similarity by Source
5%	Internet Sources: 4% Publications: 2% Student Papers: 4%

1% match (Internet from 09-Mar-2016) http://louisdl.louislibraries.org/cdm/compoundobject/collection/p120701coll25/id/130769/rec/12
1% match (Internet from 20-Jun-2025) https://www.gavinoscott.com/uploads/5/7/4/0/57406539/srs.pdf
1% match (student papers from 10-May-2019) Submitted to Higher Education Commission Pakistan on 2019-05-10
1% match (Internet from 26-Aug-2024) https://digibuo.uniovi.es/dspace/bitstream/handle/10651/74035/TFG_SilviaRodriguezBares.pdf?isAllowed=y&sequence=7
1% match (Coulin, Chad Raymond. "A Situational Approach and Intelligent Tool for Collaborative Requirements Elicitation", University of Technology Sydney (Australia), 2024) Coulin, Chad Raymond. "A Situational Approach and Intelligent Tool for Collaborative Requirements Elicitation", University of Technology Sydney (Australia), 2024
1% match (student papers from 15-Jan-2024) Submitted to De Montfort University on 2024-01-15
1% match (student papers from 26-Nov-2023) Submitted to University of Greenwich on 2023-11-26

Table of Contents

Declaration of Authorship	i
Project Description	ii
Acknowledgement	iii
Plagiarism-Free Certificate	iv
Project Proposal	1
1. Introduction	2
2. Objective	2
3. Problem Description	2
4. Target Industry	2
5. Methodology	2
6. Project Scope	3
7. Feasibility Study	3
8. Solution Application Areas	4
9. Tools/Technology	4
10. Expertise of the Team Members	4
11. Milestones	4
12. Project Schedule	5
13. Work Breakdown Structure	5
Software Requirement Specification	8
1. Introduction	9
1.1 Purpose	9
1.2 Document Conventions	9
1.3 Intended Audience and Reading Suggestions	9
1.4 Product Scope	9
2. Overall Description	9
2.1 Product Perspective	9
2.2 Product Functions	10
2.3 User Classes and Characteristics	10
2.4 Operating Environment	10
2.5 Design and Implementation Constraints	11
2.6 User Documentation	11
2.7 Assumptions and Dependencies	11
3. External Interface Requirements	12
3.1 User Interfaces	12
3.2 Hardware Interfaces	24
3.3 Software Interfaces	24
3.4 Communications Interfaces	24
4. System Features	24
4.1 Sign Up	24
4.2 Login	26
4.3 Forget Password	27
4.4 Profile Management	28
4.5 List Item	30

4.6 Edit Item	31
4.7 Delete Item	32
4.8 View Item Listings	33
4.9 Search and Filter Item Listings	34
4.10 Add to Wishlist	35
4.11 View Wishlist	36
4.12 Item Detail	37
4.13 Trade Proposals	38
4.14 Accept Trade	39
4.15 Reject Trade	40
4.16 Cancel Trade	41
4.17 Advanced Trade Management	43
4.18 Manage Active trades	44
4.19 Ban User Accounts	45
4.20 Delete List Items	46
4.21 Rate and Review System	47
4.22 Email Notification	48
4.23 Trade History	49
4.24 Send Message	50
4.25 Remove from Wishlist	51
5. Other Nonfunctional Requirements	52
5.1 Performance Requirements	52
5.2 Safety Requirements	52
5.3 Security Requirements	52
5.4 Software Quality Attributes	52
5.5 Business Rules	52
6. Other Requirements	53
Software Design Specification	54
1. Introduction	55
1.1. Document Purpose	55
1.2. Scope of the Development Project	55
1.3. Definitions, Acronyms, and Abbreviations	55
1.4. Overview of Document	55
2. System Architecture Description	56
2.1. Section Overview	56
2.2. General Constraints	56
2.3. Data Design	56
2.4. Program Structure	56
2.5. Alternatives Considered	56
3. Detailed Description of Components	56
3.1. Section Overview	56
3.2. Component and Detail	57
3.2.1. Login	57
3.2.2. SignUp	57
3.2.3. Forgot Password	58

3.2.4. Profile Management	58
3.2.5. List Item	59
3.2.6. Edit Item	59
3.2.7. Delete Item	60
3.2.8. View Item Listing	60
3.2.9. Search & Filter	61
3.2.10. Add to Wishlist	61
3.2.11. View Wishlist	62
3.2.12. Item Detail	62
3.2.13. Trade Proposal	63
3.2.14. Accept Trade	63
3.2.15. Reject Trade	64
3.2.16. Cancel Trade	64
3.2.17. Advanced Trade Management	65
3.2.18. Manage Active Trades	66
3.2.19. Ban User Account	66
3.2.20. Delete List Item	67
3.2.21. Rate & Review System	68
3.2.22. Email Notification	68
3.2.23. Trade History	69
3.2.24. Send Message	69
3.2.25. Remove from Wishlist	70
4. User Interface Design	70
4.1. Section Overview	70
4.2. Interface Design Rules	71
4.3. GUI Components	71
4.4. Detailed Description	72
5. Reuse & Relationships to Other Products	84
6. Design Decisions & Tradeoffs	84
7. Pseudocode for Components	84
8. Appendices	88
8.1. Class Diagram	88
8.2. State Chart Diagram	89
8.3. Component Diagram	90
8.4. Use Case Diagram	91
8.5. Entity Relationship Diagram	92
8.6. Activity Diagram	93
8.7. Sequence Diagram	111
Test Cases	124
1. Login	125
2. SignUp	125
3. Forgot Password	126
4. Profile Management	127
5. List Item	128
6. Edit Item	129

7. Delete Item	130
8. View Item Listing	130
9. Search & Filter	131
10. Add to Wishlist	132
11. View Wishlist	133
12. Item Detail	133
13. Trade Proposal	134
14. Accept Trade	135
15. Reject Trade	135
16. Cancel Trade	136
17. Manage Active Trade	136
18. Remove from Wishlist	137
19. Advanced Trade Management	138
20. Ban User Accounts	138
21. Delete List Items	139
22. Rate and Review	140
23. Email Notification	141
24. Trade History	142
25. Send Message	142
User Manual	144
1. General Information	146
1.8. Purpose of this Document	146
1.9. Authorized Use Permission	146
1.10. System Summary	146
1.11. System Overview	146
2. Features	147
2.1. Login	147
2.2. SignUp	147
2.3. Forgot Password	148
2.4. Profile Management	149
2.4. List Item	150
2.5. Edit Item	151
2.6. Delete Item	152
2.7. View Item Listing	153
2.8. Search & Filter	154
2.9. Add to Wishlist	155
2.10. View Wishlist	155
2.11. Item Detail	156
2.12. Trade Proposal	156
2.13. Accept Trade	157
2.14. Reject Trade	158
2.15. Cancel Trade	158
2.16. Manage Active Trade	159
2.17. Remove from Wishlist	159
2.18. Advanced Trade Management	160

2.19. Ban User Accounts	160
2.20. Delete List Items	161
2.21. Rate and Review	162
2.22. Email Notification	163
2.23. Trade History	163
2.24. Send Message	164
References	144

Table of Figures

Figure 1: Work Breakdown Structure	5
Figure 2: FYP-1 Gantt Chart	6
Figure 3: FYP-2 Gantt Chart	7
Figure 4: Product Perspective Diagram	10
Figure 5: Home Page	12
Figure 6: Sign Up Page	13
Figure 7: Login Page	13
Figure 8: Forget Password	14
Figure 9: Profile Page	14
Figure 10: Update Profile	15
Figure 11: Add Item Page	15
Figure 12: Item Detail Page	16
Figure 13: Edit Item Page	16
Figure 14: User Item Page	17
Figure 15: Wishlist Page	17
Figure 16: Trade Management: Received Request Page	18
Figure 17: Trade Management: Sent Request Page	18
Figure 18: Advance Trade Management: Received Request Page	19
Figure 19: Advance Trade Management: Sent Request Page	19
Figure 20: Advance Trade: Sent Offer	20
Figure 21: Trade History	20
Figure 22: Trade Detail	21
Figure 23: Listed Items Page	21
Figure 24: Trade Proposal Page	22
Figure 25: Message System	22
Figure 26: Admin Login	23
Figure 27: Admin Dashboard	23
Figure 28: Admin Dashboard: Users List	23
Figure 29: Admin Dashboard: Items List	24
Figure 30: Admin Dashboard: Banned Users List	24
Figure 31: Home Page	72
Figure 32: Sign Up Page	73
Figure 33: Login Page	73
Figure 34: Forget Password	74
Figure 35: Profile Page	74

Figure 36: Update Profile	75
Figure 37: Add Item Page	75
Figure 38: Item Detail Page	76
Figure 39: Edit Item Page	76
Figure 40: User Item Page	77
Figure 41: Wishlist Page	77
Figure 42: Trade Management: Received Request Page	78
Figure 43: Trade Management: Sent Request Page	78
Figure 44: Advance Trade Management: Received Request Page	79
Figure 45: Advance Trade Management: Sent Request Page	79
Figure 46: Advance Trade: Sent Offer	80
Figure 47: Trade History	80
Figure 48: Trade Detail	81
Figure 49: Listed Items Page	81
Figure 50: Trade Proposal Page	82
Figure 51: Message System	82
Figure 52: Admin Login	83
Figure 53: Admin Dashboard	83
Figure 54: Admin Dashboard: Users List	83
Figure 55: Admin Dashboard: Items List	84
Figure 56: Admin Dashboard: Banned Users List	84
Figure 57: Class Diagram	88
Figure 58: State Chart Diagram	89
Figure 59: Component Diagram (User)	90
Figure 60: Component Diagram (Admin)	90
Figure 61: Use Case Diagram	91
Figure 62: Entity Relationship Diagram	92
Figure 63: Activity Diagram (Accept Trade)	93
Figure 64: Activity Diagram (Add Item)	94
Figure 65: Activity Diagram (Add to Wishlist)	94
Figure 66: Activity Diagram (Advance Trade Management)	95
Figure 67: Activity Diagram (Ban User Account)	95
Figure 68: Activity Diagram (Delete Item)	96
Figure 69: Activity Diagram (Delete List Item)	97
Figure 70: Activity Diagram (Edit Item)	98
Figure 71: Activity Diagram (Email Notification)	99
Figure 72: Activity Diagram (Forget Password)	99
Figure 73: Activity Diagram (Login)	100
Figure 74: Activity Diagram (Manage Active Trade)	101
Figure 75: Activity Diagram (Profile Management)	102
Figure 76: Activity Diagram (Rating & Review)	103
Figure 77: Activity Diagram (Reject Trade)	104
Figure 78: Activity Diagram (Remove from Wishlist)	104
Figure 79: Activity Diagram (Search & Filter)	105
Figure 80: Activity Diagram (Send Message)	105

Figure 81: Activity Diagram (Sign Up)	106
Figure 82: Activity Diagram (Trade History)	107
Figure 83: Activity Diagram (Trade Proposal)	108
Figure 84: Activity Diagram (View Listing)	109
Figure 85: Activity Diagram (Edit Item)	110
Figure 86: Sequence Diagram (Accept Trade)	111
Figure 87: Sequence Diagram (Add Item)	112
Figure 88: Sequence Diagram (Add to Wishlist)	113
Figure 89: Sequence Diagram (Advance Trade Management)	114
Figure 90: Sequence Diagram (Ban User Account)	114
Figure 91: Sequence Diagram (Delete Item)	115
Figure 92: Sequence Diagram (Delete Listed Item)	115
Figure 93: Sequence Diagram (Edit Item)	116
Figure 94: Sequence Diagram (Email Notification)	116
Figure 95: Sequence Diagram (Forget Password)	117
Figure 96: Sequence Diagram (Login)	117
Figure 97: Sequence Diagram (Manage Active Trade)	118
Figure 98: Sequence Diagram (Profile Management)	118
Figure 99: Sequence Diagram (Rating & Review)	119
Figure 100: Sequence Diagram (Reject Trade)	119
Figure 101: Sequence Diagram (Remove from Wishlist)	120
Figure 102: Sequence Diagram (Search & Filter)	120
Figure 103: Sequence Diagram (Send Message)	121
Figure 104: Sequence Diagram (Sign Up)	121
Figure 105: Sequence Diagram (Trade History)	122
Figure 106: Sequence Diagram (Trade Proposal)	122
Figure 107: Sequence Diagram (View Listing)	123
Figure 108. Login Page	146
Figure 109. Sign Up Page	147
Figure 110. Forget Password Page	148
Figure 111. Profile Page	149
Figure 112. Update Profile Page	149
Figure 113. Add Item Page	150
Figure 114. Edit Item Page	151
Figure 115. User Item Page	152
Figure 116. Listed Items Page	152
Figure 117. Listed Items Page	153
Figure 118. Wishlist Page	154
Figure 119. Item Detail Page	155
Figure 120. Trade Proposal Page	156
Figure 121. Trade Management: Received Request Page	157
Figure 122. Trade Management: Sent Request Page	158
Figure 123. Advance Trade: Sent Offer	159
Figure 124. Admin Dashboard	160
Figure 125. Admin Dashboard: Users List	160

Figure 126. Admin Dashboard: Items List	161
Figure 127. Trade Review Page	161
Figure 128. Trade History	162
Figure 129. Message System	163

Table of Tables

Table 1: Milestone	4
Table 2: Project Timeline	5
Table 3: FYP-1 Timeline	6
Table 4: FYP-2 Timeline	6
Table 5: User Sign-Up Stimulus/Response Sequences	25
Table 6: User Sign-Up Functional Requirements	25
Table 7: User Login Stimulus/Response Sequences	26
Table 8: User Login Functional Requirements	27
Table 9: User Forgot Password Stimulus/Response Sequences	27
Table 10: User Forgot Password Functional Requirements	28
Table 11: Profile Management Stimulus/Response Sequences	28
Table 12: Profile Management Functional Requirements	29
Table 13: List Item Stimulus/Response Sequences	30
Table 14: List Item Functional Requirements	31
Table 15: Edit Item Stimulus/Response Sequences	31
Table 16: Edit Item Functional Requirements	32
Table 17: Delete Item Stimulus/Response Sequences	32
Table 18: Delete Item Functional Requirements	33
Table 19: View Item Listings Stimulus/Response Sequences	33
Table 20: View Item Listings Functional Requirements	34
Table 21: Search and Filter Item Listings Stimulus/Response Sequences	34
Table 22: Search and Filter Item Listings Functional Requirements	35
Table 23: Add to Wishlist Stimulus/Response Sequences	35
Table 24: Add to Wishlist Functional Requirements	36
Table 25: View Wishlist Stimulus/Response Sequences	36
Table 26: View Wishlist Functional Requirements	37
Table 27: Item Detail Stimulus/Response Sequences	37
Table 28: Item Detail Functional Requirements	38
Table 29: Trade Proposals Stimulus/Response Sequences	38
Table 30: Trade Proposals Functional Requirements	39
Table 31: Accept Trade Stimulus/Response Sequences	39
Table 32: Accept Trade Functional Requirements	40
Table 33: Reject Trade Stimulus/Response Sequences	41
Table 34: Reject Trade Functional Requirements	41
Table 35: Cancel Trade Stimulus/Response Sequences	42
Table 36: Cancel Trade Functional Requirements	42
Table 37: Advanced Trade Management Stimulus/Response Sequences	43
Table 38: Advanced Trade Management Offer Functional Requirements	44

Table 39: Manage Active Trades Stimulus/Response Sequences	44
Table 40: Manage Active Trades Functional Requirements	45
Table 41: Ban User Accounts Stimulus/Response Sequences	45
Table 42: Ban User Accounts Functional Requirements	46
Table 43: Delete List Items Stimulus/Response Sequences	46
Table 44: Delete List Items Functional Requirements	47
Table 45: Rate and Review System Stimulus/Response Sequences	47
Table 46: Rate and Review System Functional Requirements	48
Table 47: Email Notification Stimulus/Response Sequences	48
Table 48: Email Notification Functional Requirements	49
Table 49: Trade History Stimulus/Response Sequences	49
Table 50: Trade History Functional Requirements	50
Table 51: Send Messages Stimulus/Response Sequences	50
Table 52: Send Messages Functional Requirements	51
Table 53: Remove from Wishlist Stimulus/Response Sequences	51
Table 54: Remove from Wishlist Functional Requirements	52
Table 55: Login Component Detail	57
Table 56: SignUp Component Detail	57
Table 57: Forgot Password Component Detail	58
Table 58: Profile Management Component Detail	58
Table 59: List Item Component Detail	59
Table 60: Edit Item Component Detail	59
Table 61: Delete Item Component Detail	60
Table 62: View Item Listing Component Detail	60
Table 63: Search & Filter Component Detail	61
Table 64: Add to Wishlist Component Detail	61
Table 65: View Wishlist Component Detail	62
Table 66: Item Detail Component Detail	62
Table 67: Trade Proposal Component Detail	63
Table 68: Accept Trade Component Detail	63
Table 69: Reject Trade Component Detail	64
Table 70: Cancel Trade Component Detail	64
Table 71: Advanced Trade Management Component Detail	65
Table 72: Manage Active Trades Component Detail	66
Table 73: Ban User Account Component Detail	66
Table 74: Delete List Item Component Detail	67
Table 75: Rate & Review System Component Detail	68
Table 76: Email Notification Component Detail	69
Table 77: Trade History Component Detail	69
Table 78: Send Message Component Detail	69
Table 79: Remove from Wishlist Component Detail	70

Project Proposal

1. Introduction

Swappify is a digital platform designed to facilitate a seamless and efficient barter trading experience for users in the online environment. In an era where people seek sustainable living and meaningful community engagement, Swappify provides a platform that enables users to exchange goods directly, without the need for monetary transactions. This platform fosters a sustainable ecosystem by encouraging the reuse and recycling of items that might otherwise go to waste, thereby contributing to the circular economy. Swappify emphasizes effective trade management, and community trust-building through its trading mechanisms and rating systems. Swappify aims to redefine how people perceive and engage in barter transactions, creating an alternative marketplace that focuses on utility and value rather than financial exchange.

2. Objective

To develop a user-friendly web application for barter trading that features secure user authentication, efficient profile management, advanced search, trade proposals, and user ratings. The goal is to create a streamlined and trading experience that enhances user engagement and fosters a trustworthy community for exchanging goods and services without money, promoting sustainability and repurposing of items.

3. Problem Description

In today's fast-paced world, the need for sustainable living and reducing waste has become more pressing. Traditional online marketplaces focus primarily on monetary transactions, often overlooking the value of direct exchange or bartering, which can be both cost-effective and environmentally friendly. Swappify addresses this gap by providing a dedicated platform for barter trading, where users can trade goods and services without involving money. The problem arises from the lack of platforms that facilitate secure and efficient barter transactions, as existing marketplaces do not prioritize features like robust user authentication, trust-building through ratings, and transparent trade management systems.

Without a streamlined barter system, users face challenges in finding suitable trade partners, assessing the value of items, and ensuring safe and fair exchanges. Swappify solves these issues by incorporating advanced search functionalities, detailed user profiles, trade proposal systems, and a post-trade rating mechanism to build community trust. Moreover, the platform emphasizes user security through secure authentication and transparent profiles, addressing concerns around privacy and fraud.

By focusing on these areas, Swappify not only promotes sustainability by encouraging the reuse and recycling of goods but also builds a trustworthy community where users feel confident engaging in barter transactions. The platform's user-friendly design and comprehensive features redefine barter trading in the digital age, creating an efficient, secure, and valuable alternative to traditional, money-driven marketplaces.

4. Target Industry

Swappify targets the E-commerce and Online Marketplaces industry by offering a barter system for item exchanges without money.

5. Methodology

To develop Swappify, we will be using Agile methodology, a flexible and iterative approach to software development that emphasizes collaboration and customer feedback.

Agile Methodology: Overview

This is a structured approach that segments projects into manageable phases, focusing on continuous improvement. It is an iterative process that involves planning, execution, and evaluation.^[1]

Benefits of Agile Methodology:

Agile methodology provides immediate feedback from users, enabling quick, incremental improvements and rapid adaptation to changing requirements. It enhances time efficiency by speeding up feature delivery and reducing time to market, while its flexibility allows for seamless integration of frequent changes and alignment with customer expectations.

Limitations of Agile Methodology:

Agile's focus on working software often leads to minimal documentation and requires experienced team members for quick decision-making. The flexible approach can result in scope creep and may lack the predictability in timelines and deliverables found in more structured methods.

6. Project 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, Swappify 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.

7. Feasibility Study

Given the defined scope of the project, we have performed a feasibility study to evaluate its viability and identify potential challenges. We believe the project is achievable with effective planning.

i. Risks Involved:

Technical Challenges: The development of certain features, such as advanced search filters and secure authentication, may pose technical challenges. We will address these risks by performing comprehensive research and consulting with experts.

Disputes Between Users: There could be disagreements or disputes regarding trades between users. We will include a chat system to handle such issues efficiently.

ii. Resource Requirement:

- Operating System: Windows 10 (64-bit)
- Processor: Core i7 - 8th Generation
- RAM: 16gb
- Storage: 40gb

8. Solution Application Areas

Our project holds significant value in markets where users prefer non-monetary transactions or are keen to participate in a sustainable and community-driven trading economy. Potential target users include local communities, environmentally conscious consumers, and college students looking for budget-friendly alternatives to traditional buying and selling.

9. Tools/Technology

Hardware:

For developing and testing applications, the machines should be equipped with sufficient processing power, ample memory, and adequate storage capacity. These specifications ensure smooth performance, quick data processing, and efficient handling of resource intensive tasks.

Software:

- Visual Studio Code
- MongoDB Compass
- Postman
- React JS
- Node JS
- Express JS
- MongoDB

10. Expertise of the Team Members

We are well-equipped with the necessary knowledge and skills to successfully complete this project. We have previously collaborated on similar projects, demonstrating our ability to work effectively together. Rohail Rathore specializes in frontend development, bringing expertise in designing intuitive user interfaces and ensuring a seamless user experience. Mustan Ali specializes in backend development, with skills in creating server-side logic, managing databases.

11. Milestones

1	User Authentication & Profile Management
2	Item Listings & Search and Filters
3	Wishlist
4	Trade Proposal
5	Trade Management
6	Admin Portal
7	Advance Trade Management
8	User Ratings and Reviews
9	Email Notification
10	Trade History
11	Messaging System

Table 1: Milestone

12. Project Schedule

No	Task	Start Date	End Date	Duration
1	User Authentication & Profile Management	01 Oct 24	20 Oct 24	3 Weeks
2	Item Listings & Search and Filters	21 Oct 24	10 Nov 24	3 Weeks
3	Wishlist	11 Nov 24	17 Nov 24	1 Week
4	Trade Proposal	18 Nov 24	08 Dec 24	3 Weeks
5	Trade Management	09 Dec 24	5 Jan 25	4 Weeks
6	Admin Portal	10 Feb 25	23 Feb 25	2 Weeks
7	Advance Trade Management	24 Feb 25	16 Mar 25	3 Weeks
8	User Ratings and Reviews	17 Mar 25	30 Mar 25	2 Weeks
9	Email Notification	31 Mar 25	13 Apr 25	2 Weeks
10	Trade History	14 Apr 25	27 Apr 25	2 Weeks
11	Messaging System	28 Apr 25	31 May 25	5 Weeks

Table 2: Project Timeline

13. Work Breakdown Structure

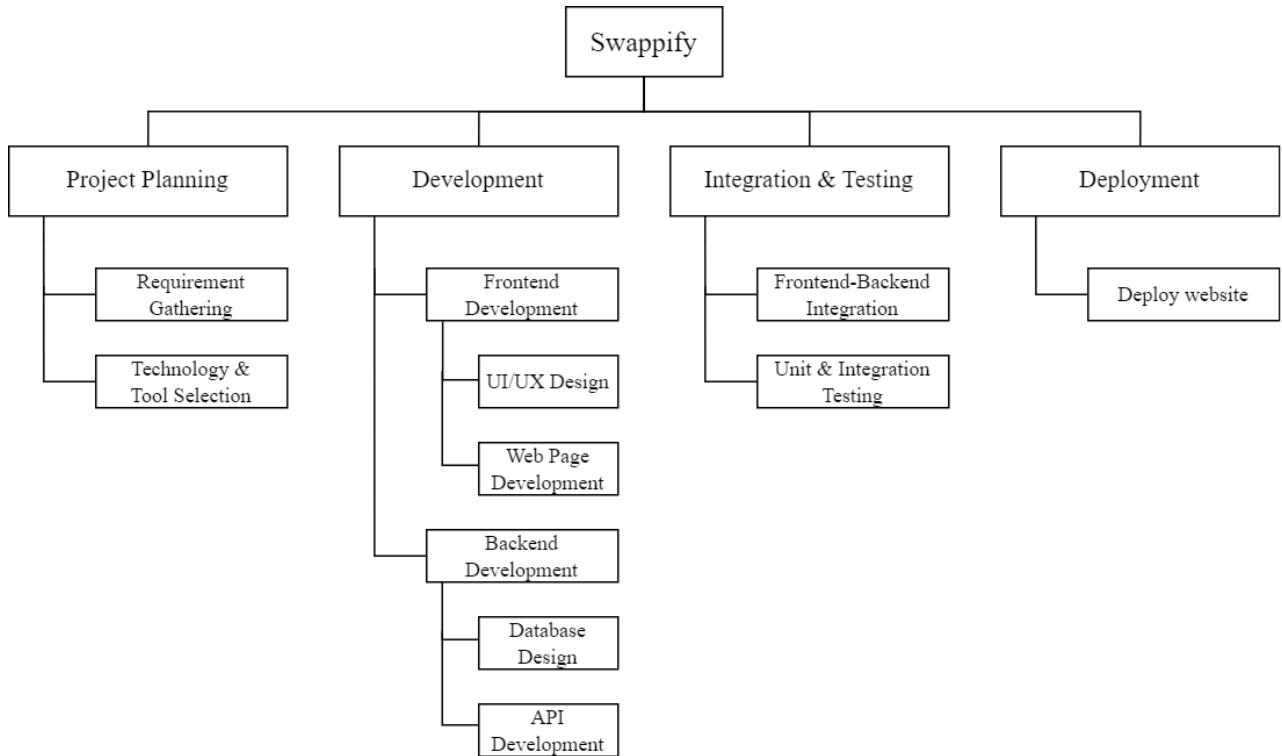


Figure 1: Work Breakdown Structure

FYP - 1:

Course	Duration	Start Date	End Date
FYP 1	14 Weeks (97 Days)	01 Oct 24	05 Jan 25

No	Task	Start Date	End Date	Duration
1	User Authentication & Profile Management	01 Oct 24	20 Oct 24	3 Weeks
2	Item Listings & Search and Filters	21 Oct 24	10 Nov 24	3 Weeks
3	Wishlist	11 Nov 24	17 Nov 24	1 Week
4	Trade Proposal (Offer a Trade)	18 Nov 24	08 Dec 24	3 Weeks
5	Trade Management	09 Dec 24	5 Jan 25	4 Weeks

Table 3: FYP-1 Timeline

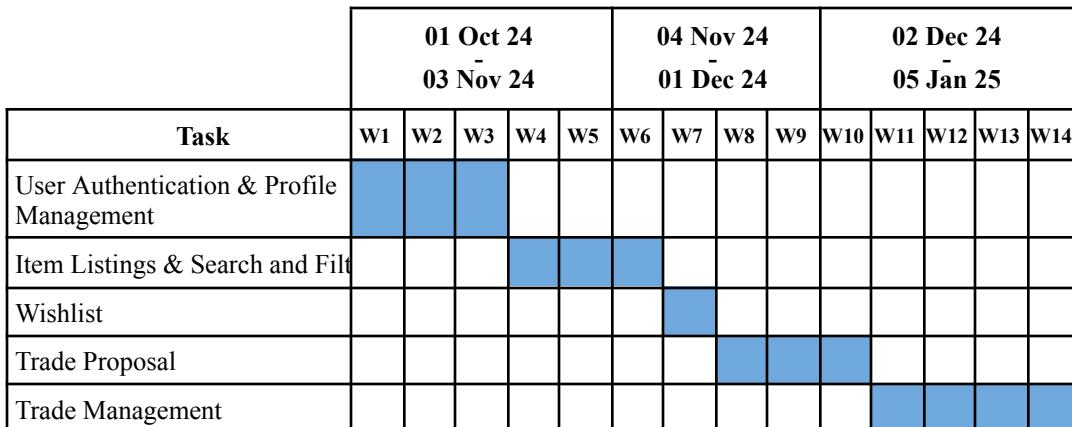
FYP - 1: Gantt Chart

Figure 2: FYP-1 Gantt Chart

FYP - 2:

Course	Duration	Start Date	End Date
FYP 2	16 Weeks (111 Days)	10 Feb 25	31 May 25

No	Task	Start Date	End Date	Duration
1	Admin Portal	10 Feb 25	23 Feb 25	2 Weeks
2	Advance Trade Management	24 Feb 25	16 Mar 25	3 Weeks
3	User Ratings and Reviews	17 Mar 25	30 Mar 25	2 Weeks
4	Email Notification	31 Mar 25	13 Apr 25	2 Weeks
5	Trade History	14 Apr 25	27 Apr 25	2 Weeks
6	Messaging System	28 Apr 25	31 May 25	5 Weeks

Table 4: FYP-2 Timeline

FYP - 2: Gantt Chart

Task	10 Feb 25 02 Mar 25			03 Mar 25 06 Apr 25				07 Apr 25 04 May 25				05 May 25 01 Jun 25				
	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15	W16
Admin Portal																
Advance Trade Management																
User Ratings and Reviews																
Email Notification																
Trade History																
Messaging System																

Figure 3: FYP-2 Gantt Chart

Software Requirement Specification

1. Introduction

1.1 Purpose

The purpose of Swappify is to revolutionize the barter trading process by providing a digital platform that efficiently connects users for direct exchanges of goods. Focusing on sustainability, Swappify 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 Swappify 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, Swappify creates an alternative marketplace centered on utility, value, and sustainability.

1.2 Document Conventions

The document is created using Google Docs and has the font "Times New Roman". The fixed font size that has been used for the body text is 10pt. For the primary headings, we use font size 16, and for the subheadings, font size 13 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, Swappify 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.

2. Overall Description

2.1 Product Perspective

Swappify 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.

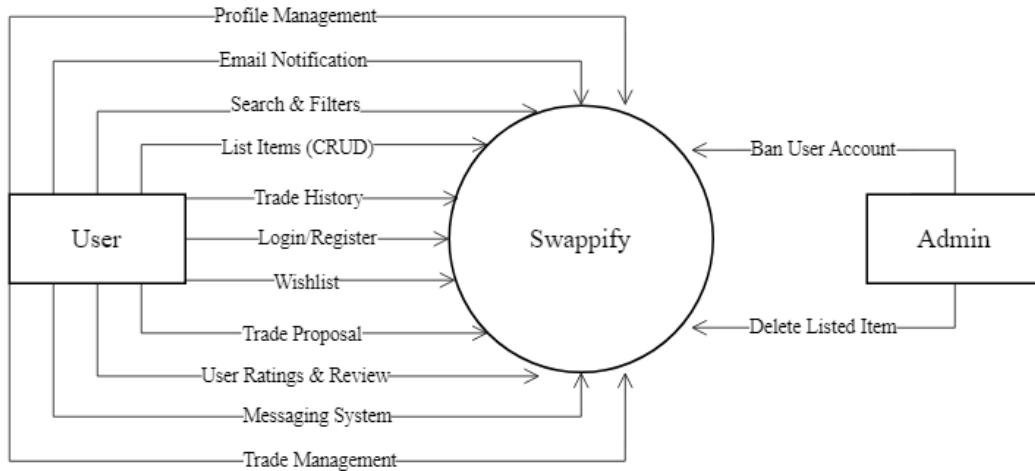


Figure 4: Product Perspective Diagram

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.
- **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.

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:** Swappify 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:** Swappify 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

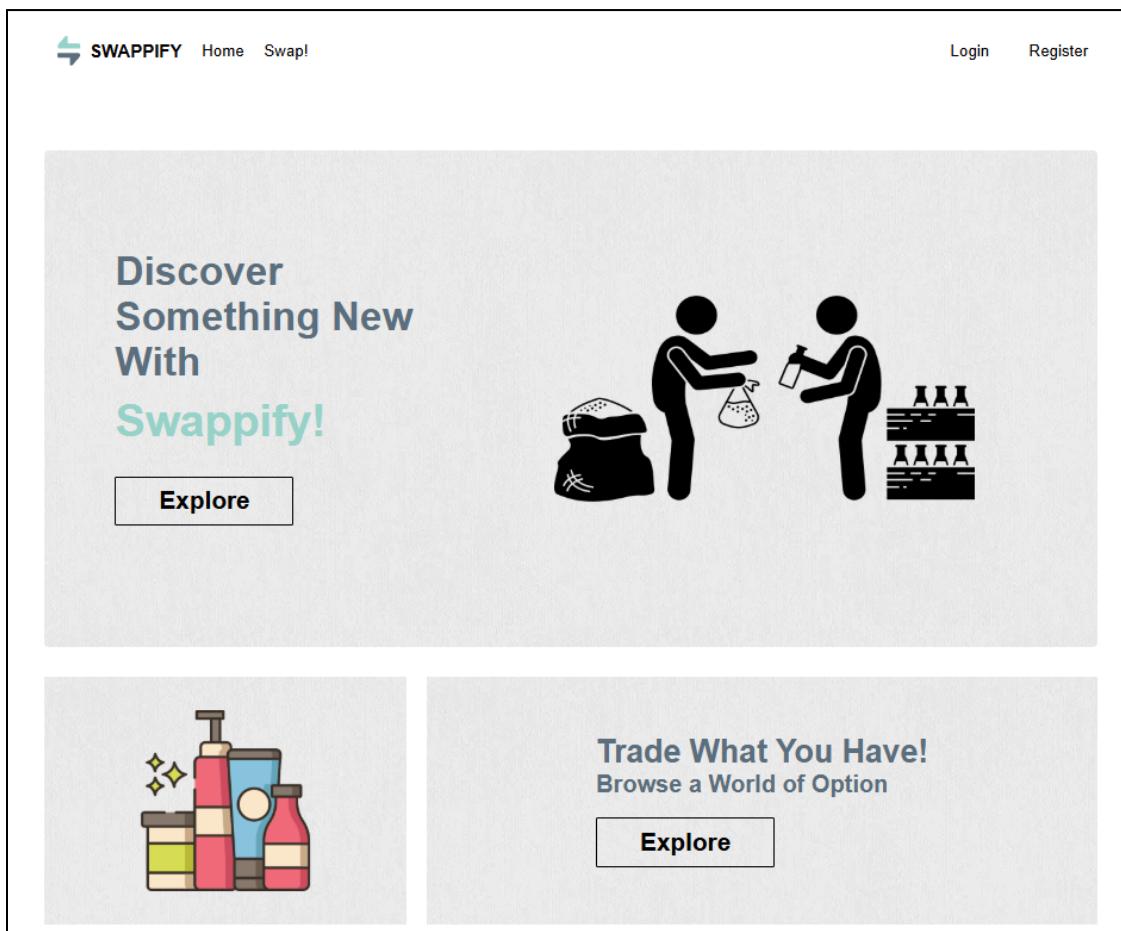


Figure 5. Home Page

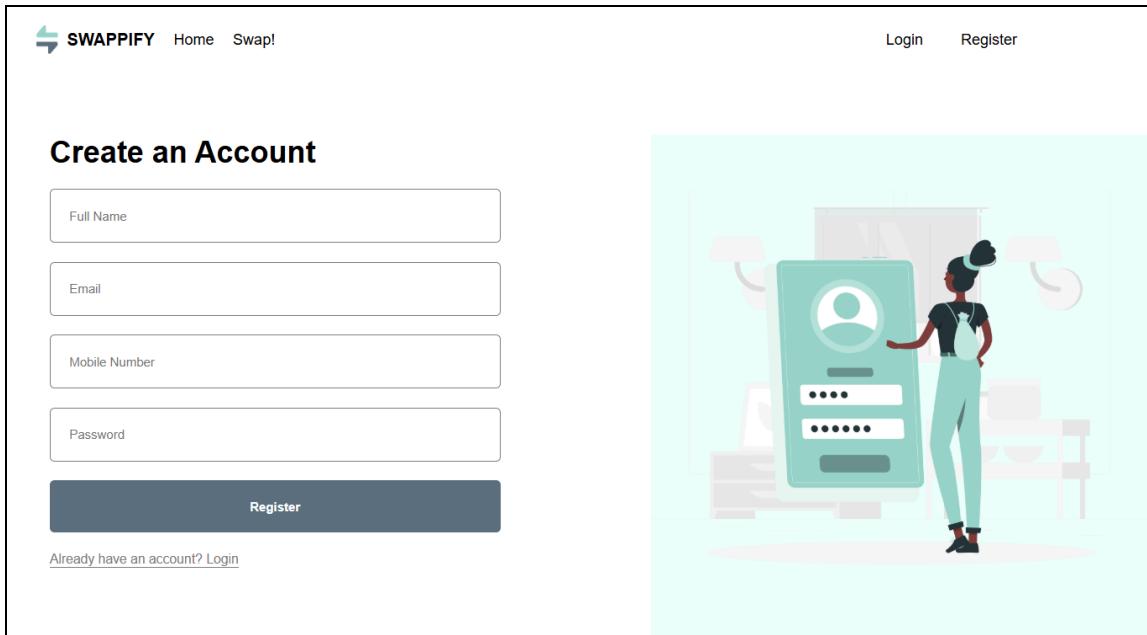


Figure 6. Sign Up Page

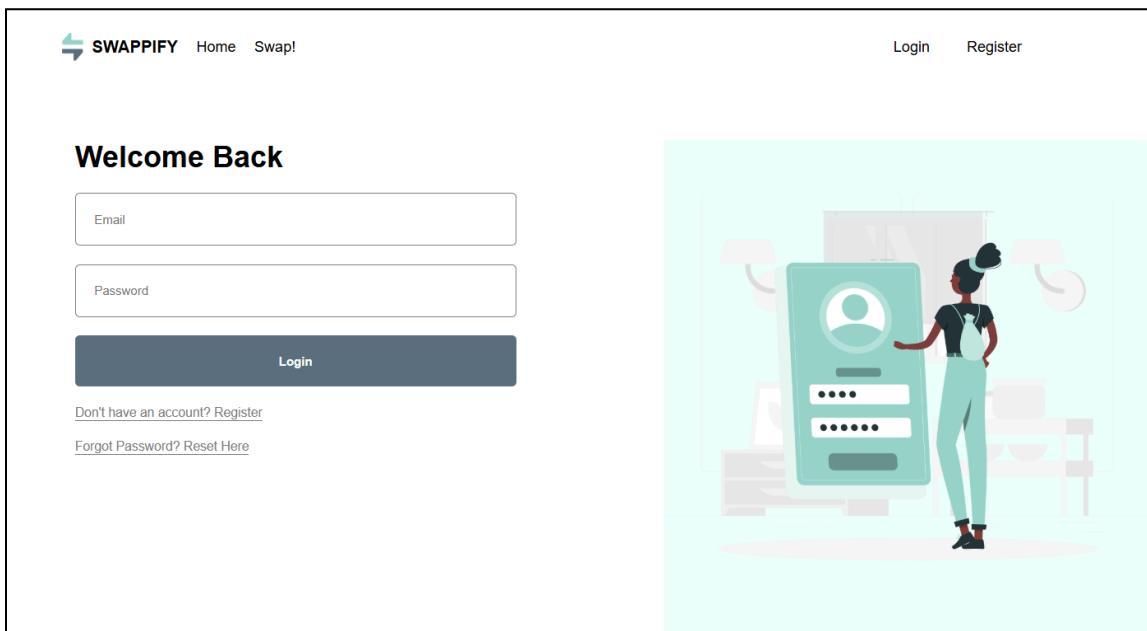


Figure 7. Login Page

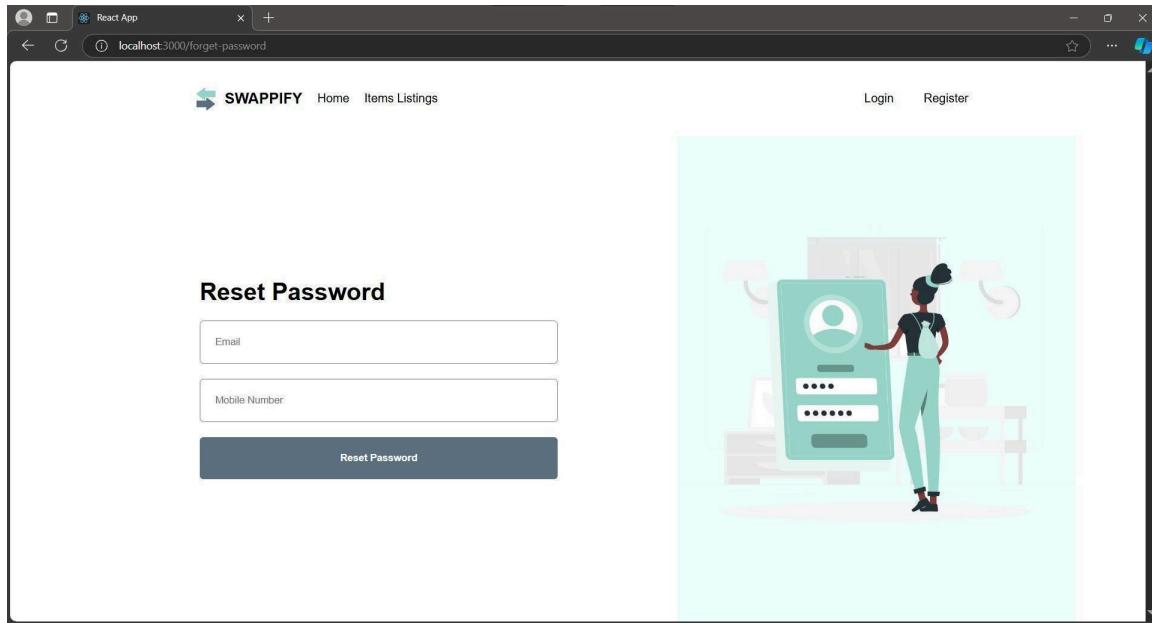


Figure 8. Forget Password Page

A screenshot of the Swappify application's User Profile page. The header includes the Swappify logo, 'Home', 'Swap!', 'Profile' (highlighted in blue), and 'Logout' links. On the left, there is a sidebar with navigation links: 'User Profile', 'MyList', 'WishList', and 'Trade Management'. The main content area is titled 'User Profile' and features a 'User Information' card. The card displays a placeholder profile picture, the full name 'Mustan Ali', the email 'MustanAli@example.com', the mobile number '03234567891', and a green 'Update Profile' button.

Figure 9. Profile Page

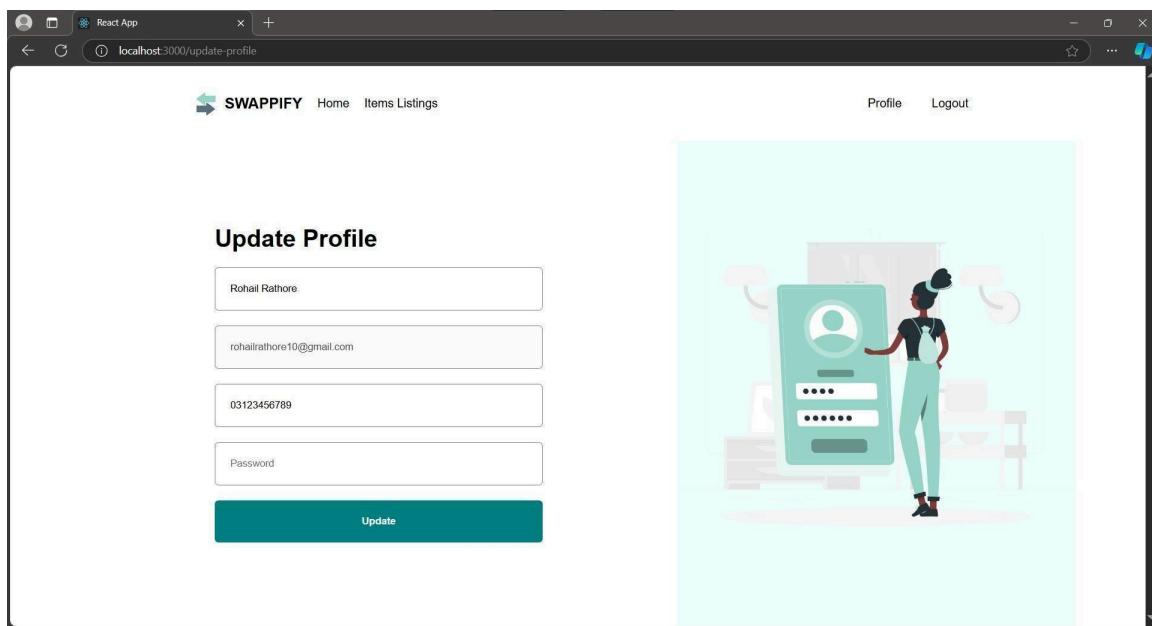


Figure 10. Update Profile Page

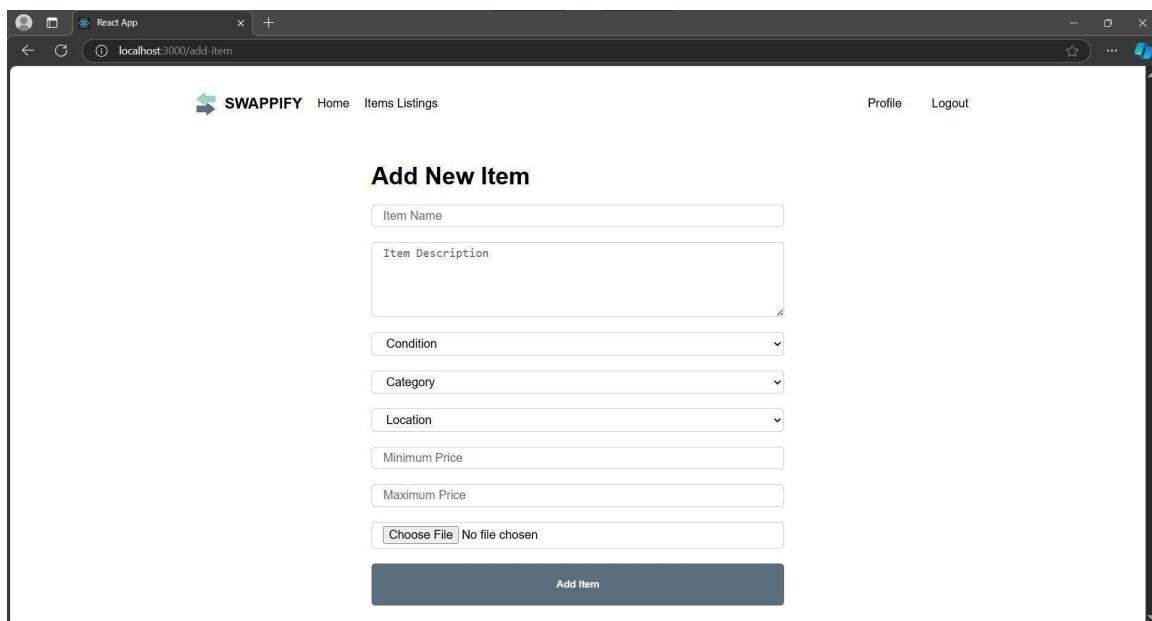


Figure 11. Add Item Page

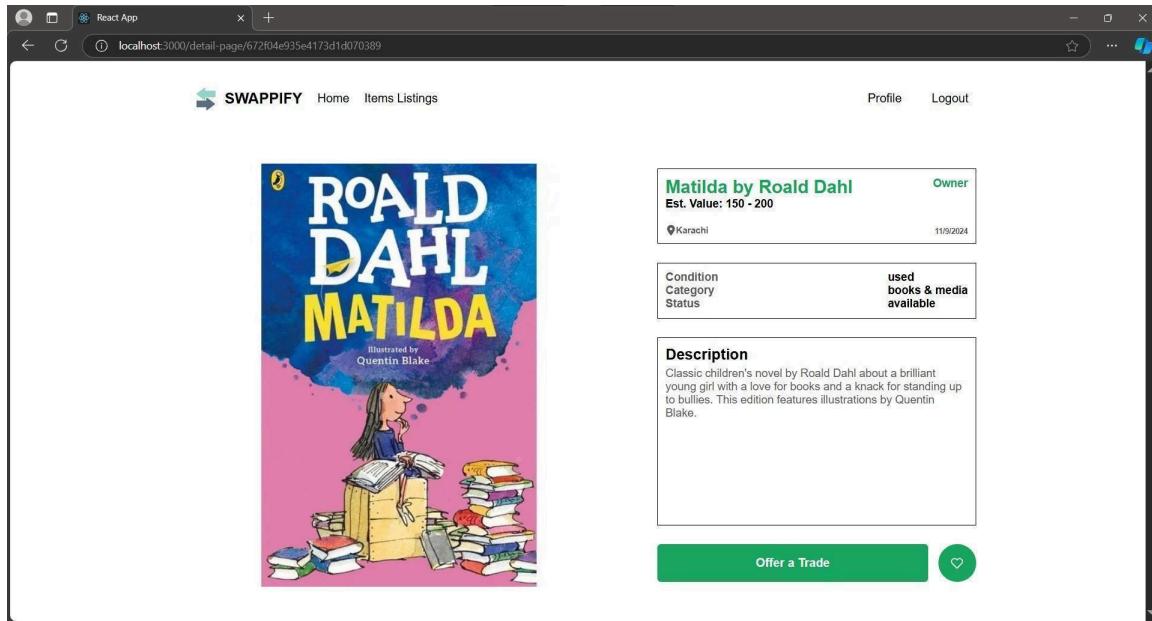


Figure 12. Item Detail Page

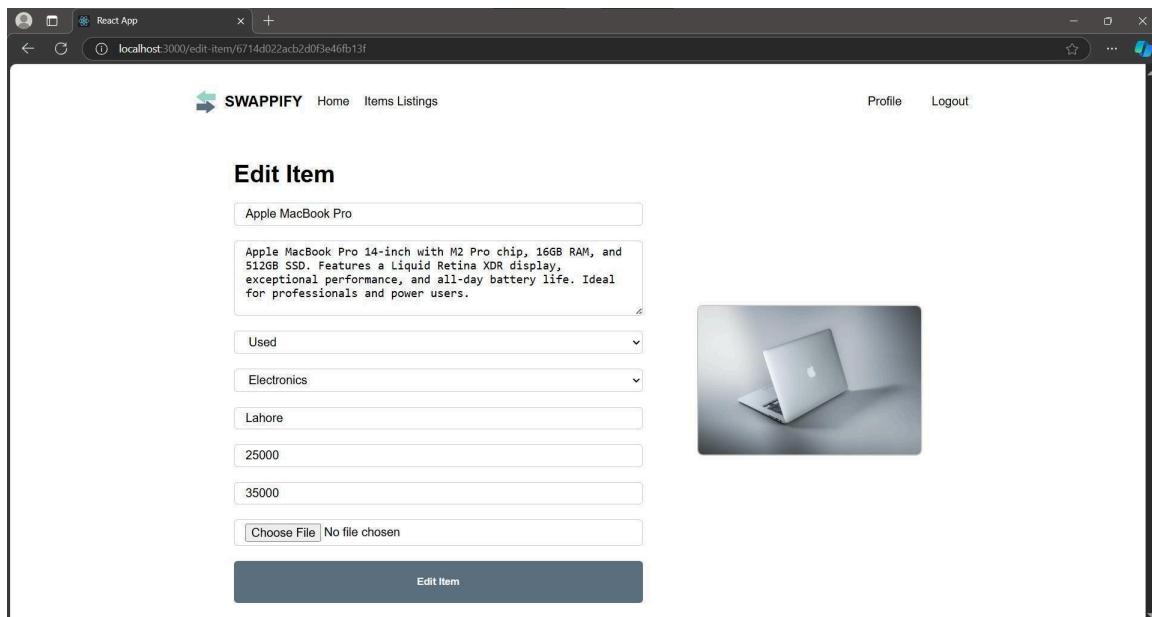
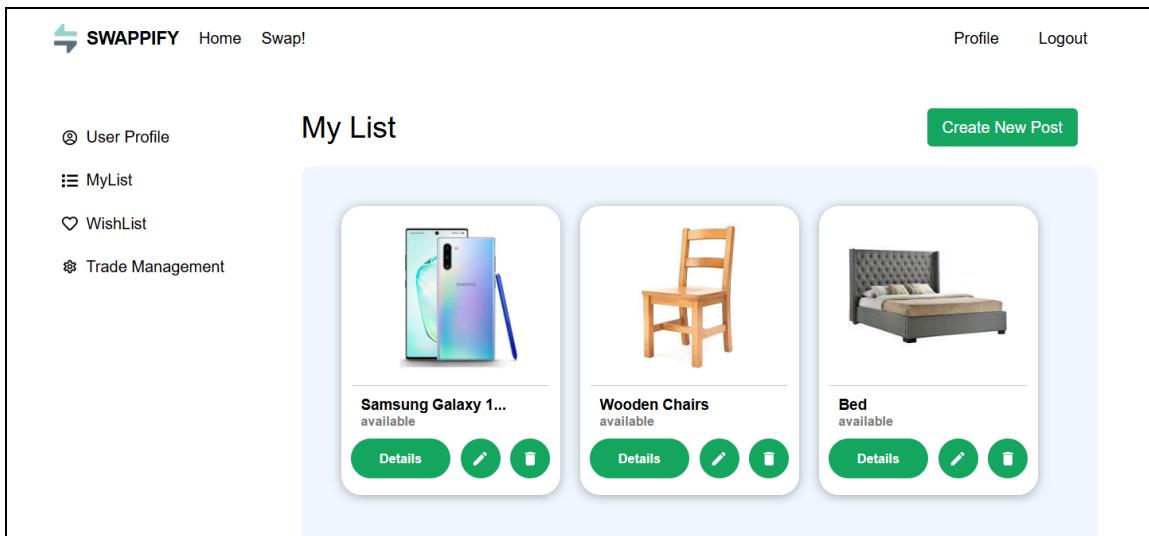


Figure 13. Edit Item Page

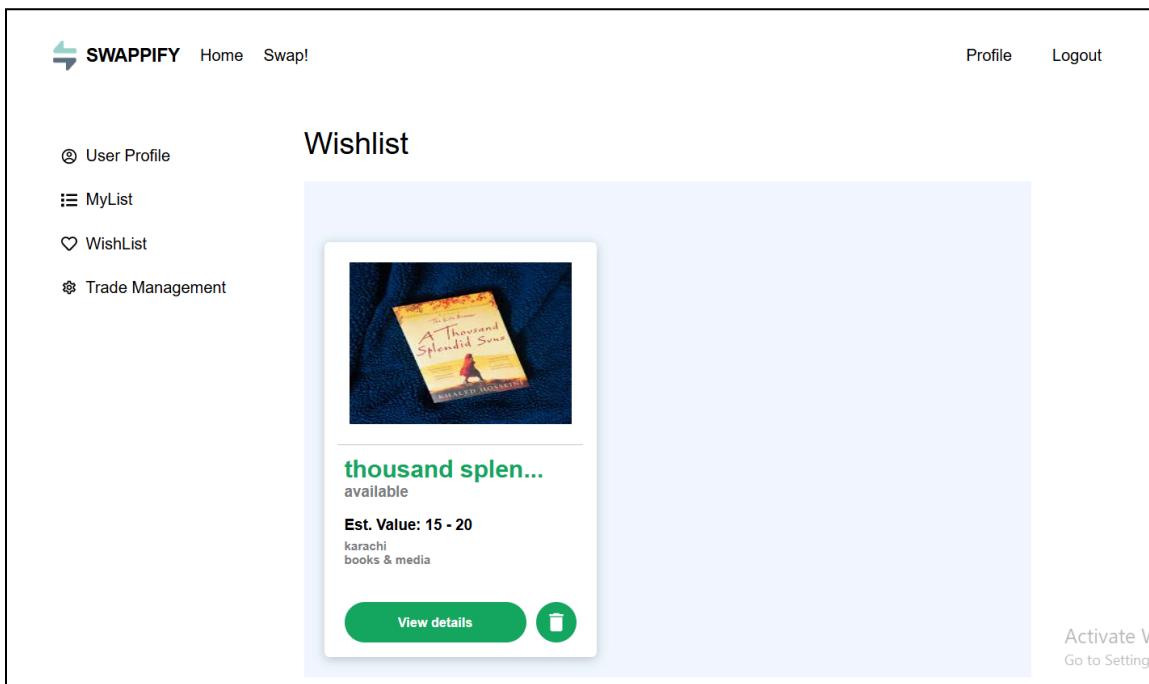


The screenshot shows the 'My List' page of the SWAPPIFY app. At the top, there's a navigation bar with the SWAPPIFY logo, 'Home', 'Swap!', 'Profile', and 'Logout'. On the left, a sidebar menu includes 'User Profile', 'MyList' (selected), 'WishList', and 'Trade Management'. The main area is titled 'My List' and features three items in cards:

- Samsung Galaxy 1...** available
Details | Edit | Delete
- Wooden Chairs** available
Details | Edit | Delete
- Bed** available
Details | Edit | Delete

A green 'Create New Post' button is located in the top right corner.

Figure 14. User Item Page



The screenshot shows the 'Wishlist' page of the SWAPPIFY app. At the top, there's a navigation bar with the SWAPPIFY logo, 'Home', 'Swap!', 'Profile', and 'Logout'. On the left, a sidebar menu includes 'User Profile', 'MyList', 'WishList' (selected), and 'Trade Management'. The main area is titled 'Wishlist' and features one item in a card:

- thousand splen...** available
Est. Value: 15 - 20
karachi
books & media
View details | Delete

A green 'Activate W' and 'Go to Settings' button is located in the bottom right corner.

Figure 15. Wishlist Page

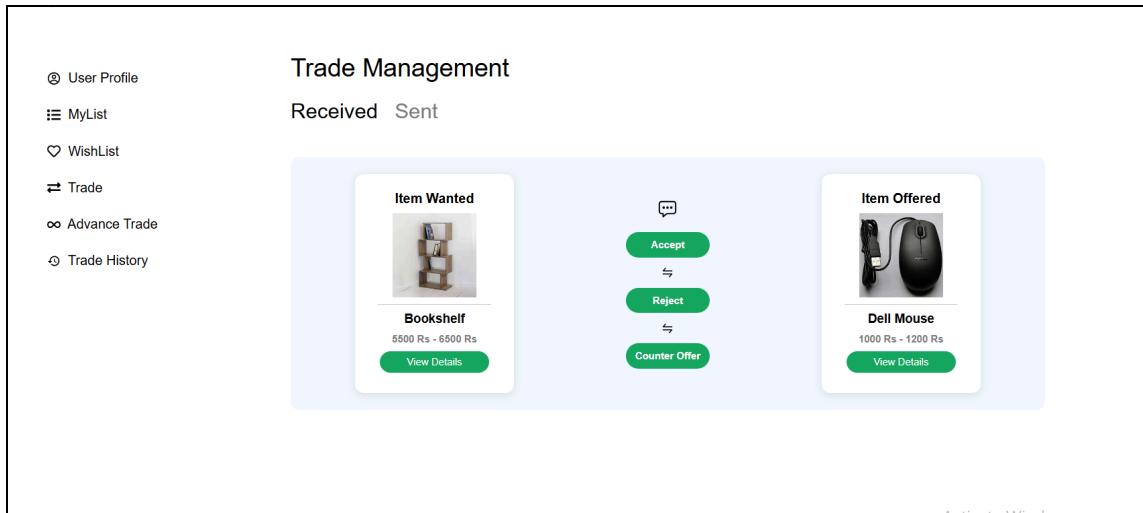


Figure 16. Trade Management: Received Request Page

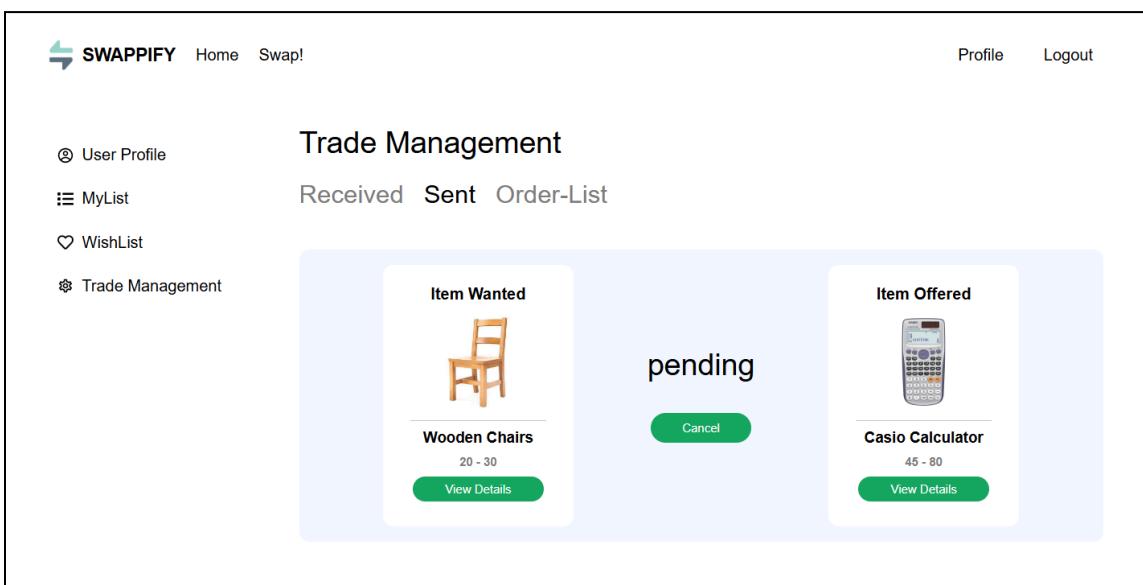


Figure 17. Trade Management: Sent Request Page

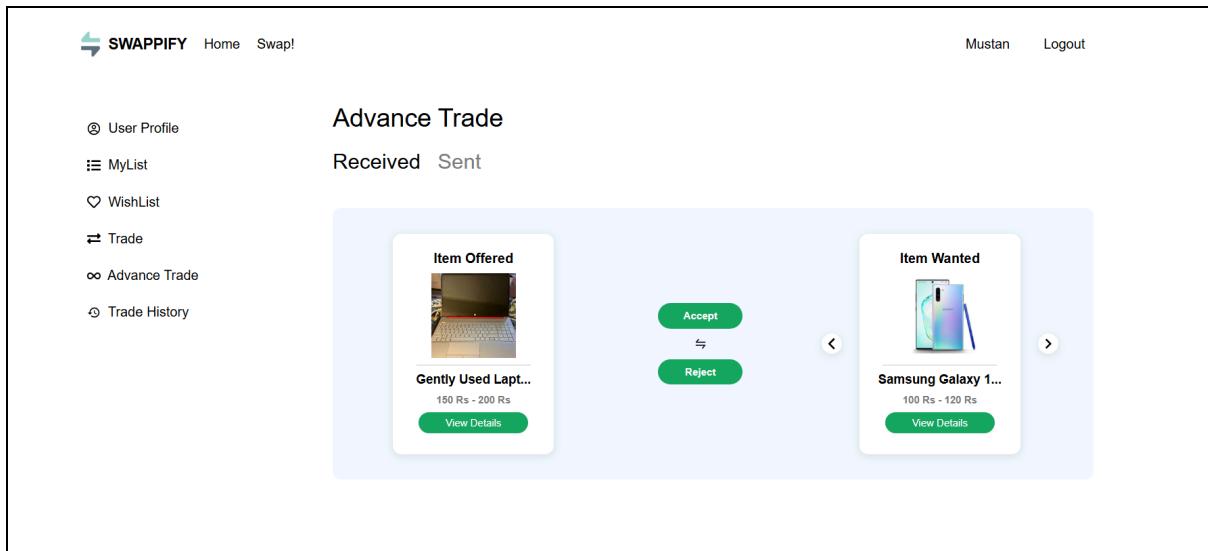


Figure 18. Advance Trade Management: Received Request Page

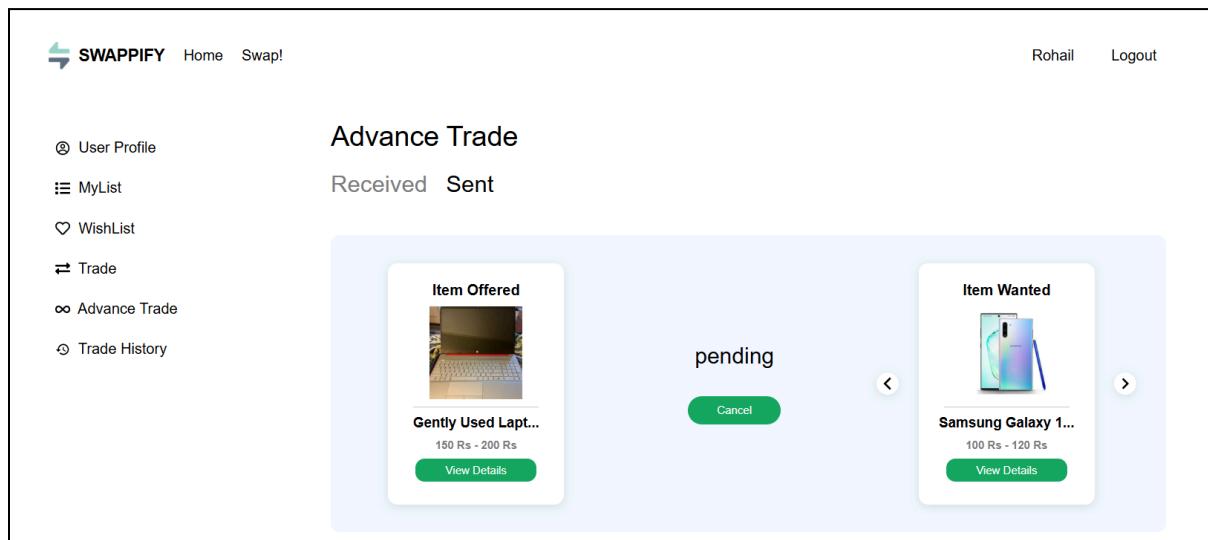


Figure 19. Advance Trade Management: Sent Request Page

Gently Used Laptop
150 Rs - 200 Rs
[View Details](#)

Counter Trade

Not a good trade. The price difference is significant.

Samsung Galaxy...
100 Rs - 120 Rs
[Remove](#)

Electric Gui...
250 Rs - 300 Rs
[Remove](#)

User List

Samsung Galaxy 1... available
[Details](#) [Deselect](#)

Dell Inspiron 15... available
[Details](#) [Select](#)

Electric Guitar available
[Details](#) [Deselect](#)

Home Workout Equ... available
[Details](#) [Select](#)

Bookshelf available
[Details](#) [Select](#)

Figure 20. Advance Trade: Sent Offer

User Profile

MyList

WishList

Trade

Advance Trade

Trade History

TradeHistory

Handmade Wooden ... 4/6/2025

Rohail Rathore

Request Type: Sent
Trade type: Normal
Status: cancelled
Price Range: 70 Rs - 90 Rs

[Item details](#)

[Trade details](#)

Wooden Chairs 4/6/2025

Mustan Ali

Request Type: Received
Trade type: Counter
Status: accepted
Price Range: 20 Rs - 30 Rs

[Item details](#)

[Trade details](#)

Figure 21. Trade History



Figure 22. Trade Detail

Figure 23. Listed Items Page

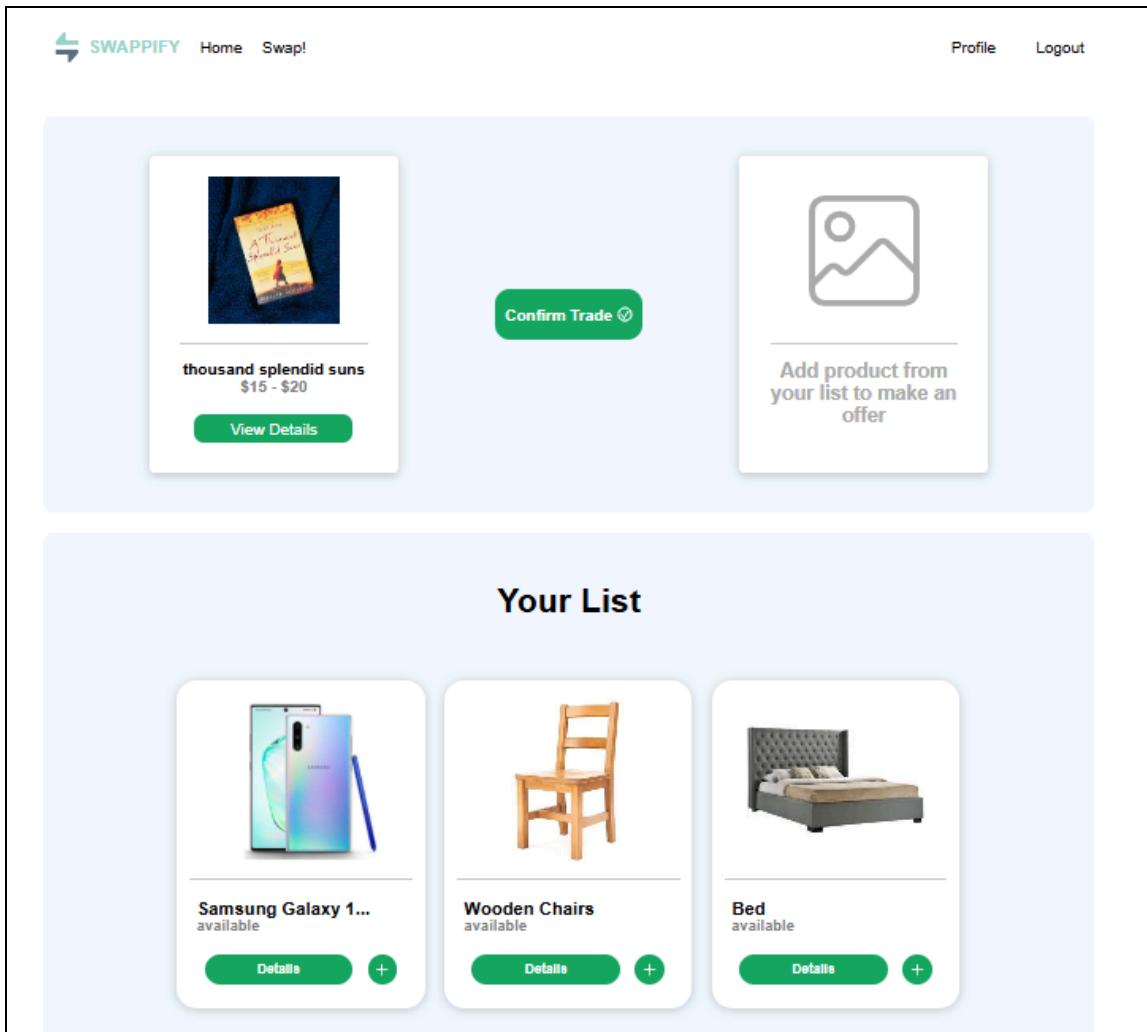


Figure 24. Trade Proposal Page

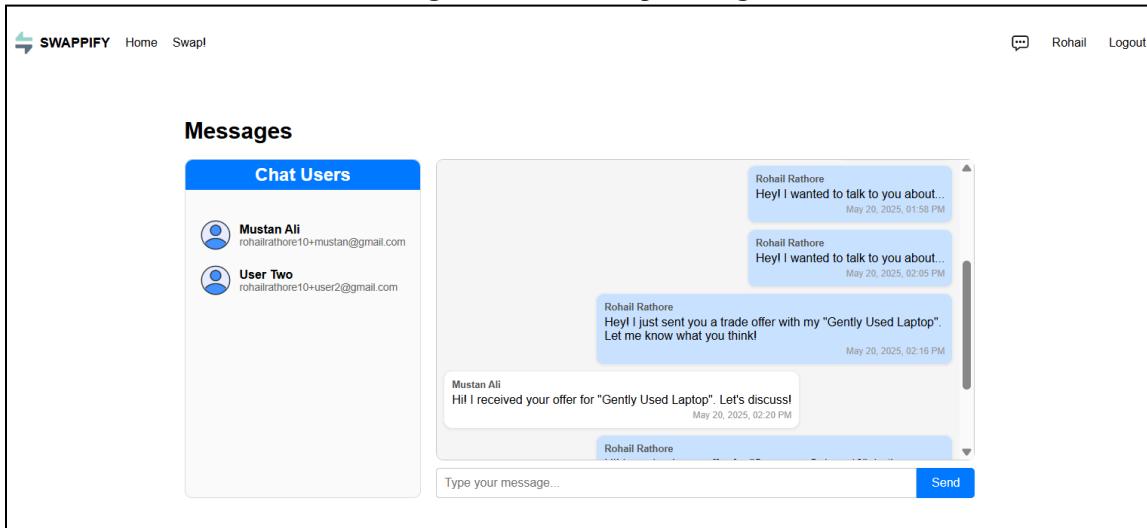


Figure 25. Message System

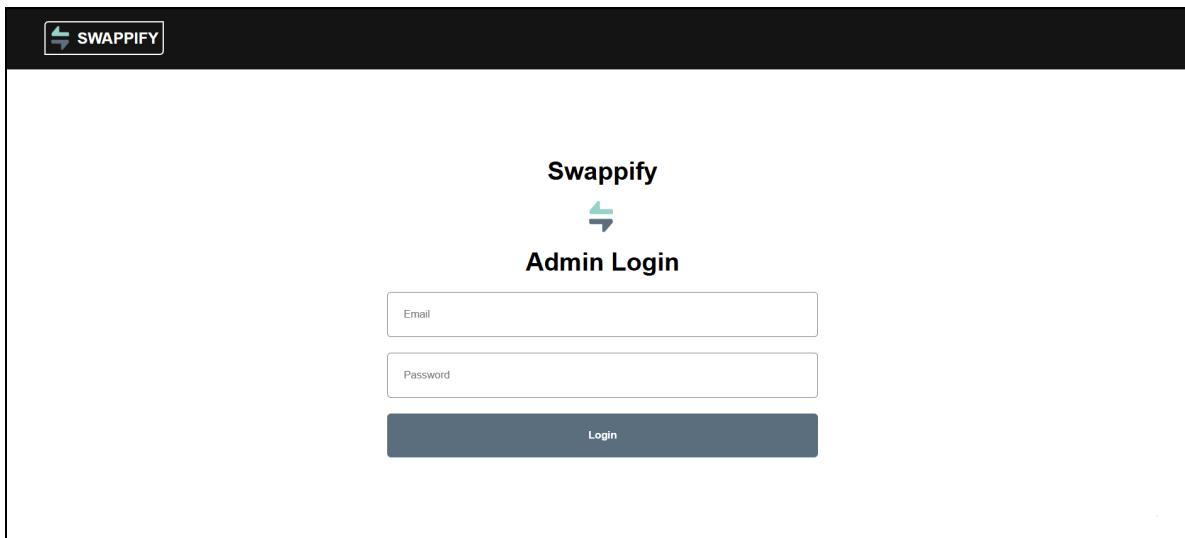


Figure 26. Admin Login

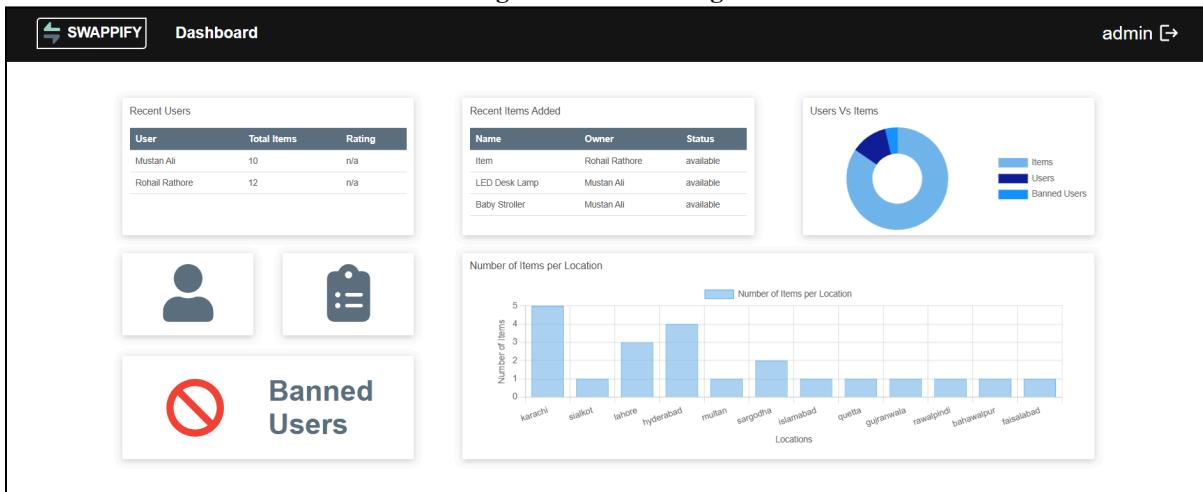


Figure 27. Admin Dashboard

Name	Phone	Email	Total Items	Action
Rohail Rathore	03123456789	rohailrathore10@gmail.com	12	∅
Mustan Ali	03234567891	rohailrathore10+mustan@gmail.com	10	∅

Figure 28. Admin Dashboard: Users List

Name	Owner	Condition	Category	Location	Price (PKR)	Status	Date Added	Action
Samsung Galaxy 10		used	electronics	karachi	100 - 120	available	11/18/2024	
Wooden Chairs		new	furniture	sialkot	20 - 30	traded	11/18/2024	
Bed		new	furniture	lahore	200 - 250	traded	11/18/2024	
Dell Inspiron 15 Laptop		used	electronics	hyderabad	200 - 250	available	11/18/2024	
thousand splendid suns		used	books & media	karachi	15 - 20	available	11/18/2024	
Casio Calculator		new	office supplies	lahore	45 - 80	traded	11/18/2024	
Dell Mouse		new	electronics	multan	15 - 25	available	11/18/2024	
Iphone 13		new	electronics	karachi	120 - 230	traded	12/27/2024	
Art		new	collectibles & antiques	karachi	50 - 75	traded	1/27/2025	

Figure 29. Admin Dashboard: Items List

Name	Phone	Email	Total Items	Action
user1	03123456788	rohailrathore10@hotmail.com	0	

Figure 30. 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:

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

Table 5: 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	High	Complete

	are filled.		
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 6: 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:

Use Case Name: Login
Description: User can login to an account on the platform.
Actors: User (Trader)
Goal: Authenticate a user to access their account.
Pre-Conditions: The user must already have an account on the platform.
Basic Course of Events /Main Flow
<p>1. User Action (Step 1): The user navigates to the home page and clicks on the "Login" button. System Response: The system presents the user with a login form that includes fields for the email and password.</p>
<p>2. User Action (Step 2): The user fills in the form with their registered email and password, then clicks on the "Submit" button. System Response: The system validates the input data: <ul style="list-style-type: none"> • Ensures the email is in a valid format. • Verifies that the fields are not empty. • If the email is not registered: The system shows an error message. • If the email is registered but the password is incorrect: The system shows an error message. </p>
<p>3. User Action (Step 3): The user submits the correct email and password. System Response: The system authenticates the user, logs them into their account, and redirects them to the Home page.</p>
Alternate Flow (Invalid Inputs)

User Action: The user submits the form with invalid inputs (e.g. incorrect email format or an empty password field).
System Response: The system displays error messages.
Exception Flow (System Errors)
System Failure: If the system encounters an error during processing of request. System Response: The system shows an error message.

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

Table 7: User Login Stimulus/Response Sequences

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 8: 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:

Use Case Name: Forget Password
Description: Users can reset their password if they forget it by receiving a reset link via email.
Actors: User (Trader)
Goal: Allow users to securely reset their password.
Pre-Conditions: The user must have a registered email address.
Basic Course of Events /Main Flow
<ol style="list-style-type: none"> User Action (Step 1): The user navigates to the login page and clicks on "Forgot Password." System Response: The system displays a form where the user enters their registered email address and mobile number. User Action (Step 2): The user submits their email address by clicking the "Submit" button. System Response: The system checks if the email exists in the database:

<ul style="list-style-type: none"> If the email exists: 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." If the email does not exist: The system shows an error message: "Email not found. Please check and try again." <p>3. User Action (Step 3): The user login with a new password. System Response: The system authenticates the user, logs them into their account, and redirects them to Home page.</p>
Alternate Flow (Invalid Inputs)
<p>User Action: The user submits the form with invalid inputs (e.g. incorrect email format or an empty password field). System Response: The system displays error messages.</p>
Exception Flow (System Errors)

Table 9: 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 10: 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:

Use Case Name: Profile Management
Description: Users can modify their account information on the platform.
Actors: User (Trader)

Goal: To allow users to modify their account information
Pre-Conditions: User must be logged in with a valid account.
Basic Course of Events /Main Flow
<p>1. User Action (Step 1): The user navigates to their profile page by clicking on the "Profile" button. System Response: The system presents the user with their current profile information, displaying fields like name, email, phone number.</p> <p>2. User Action (Step 2): The user selects the "Edit" option to update their profile information. System Response: The system allows the user to edit the available fields, such as name, email, mobile.</p> <p>3. User Action (Step 3): The user updates the desired fields and clicks the "Save" button. System Response: The system validates the updated information: <ul style="list-style-type: none"> • Ensures mandatory fields are not empty. Checks for valid email and mobile format. </p> <p>4. User Action (Step 4): The user submits valid updated profile information. System Response: The system saves the updated profile information in the database and displays new information.</p>
Alternate Flow (Invalid Inputs)
<p>User Action: The user submits the form with invalid data. System Response: The system displays appropriate error messages</p>
Exception Flow (System Errors)
<p>System Failure: If the system encounters an error during processing of request. System Response: The system shows an error message.</p>
Post Conditions: User's profile information is updated successfully.

Table 11: 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 12: 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:

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

Post Conditions: The new item is successfully added to the user's listings.

Table 13: 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 14: 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:

Use Case Name: Edit Item
Description: Users can modify the details of an item they have listed for trade.
Actors: Users (Traders).
Goal: Allow users to update the information of items they have listed for trade.
Pre-Conditions: The user must be logged into their account. The user must have at least one item listed for trade.
Basic Course of Events /Main Flow
<ol style="list-style-type: none"> User Action (Step 1): The user navigates to their item listings and selects an item to edit. System Response: The system displays the current details of the selected item in an editable form. User Action (Step 2): The user updates the desired fields and clicks the "Save" button. System Response: The system validates the updated information: <ul style="list-style-type: none"> Ensures mandatory fields are not empty. User Action (Step 3):

<p>The user submits the form with valid updates.</p> <p>System Response:</p> <p>The system saves the updated item details in the database and displays a confirmation message.</p>
<p>Alternate Flow (Invalid Inputs)</p> <p>User Action:</p> <p>The user submits the form with invalid data (e.g., leaving mandatory fields empty).</p> <p>System Response:</p> <p>The system displays appropriate error messages.</p>
<p>Exception Flow (System Errors)</p> <p>System Failure:</p> <p>If the system encounters an error during processing of request.</p> <p>System Response:</p> <p>The system shows an error message.</p>
<p>Post Conditions: The item details are successfully updated.</p>

Table 15: 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 fields are filled.	High	Complete
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 16: 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:

Use Case Name: Delete Item
Description: Users can remove an item they have listed for trade from the platform.
Actors: Users (Traders).
Goal: Allow users to delete an item they no longer want to offer for trade.
Pre-Conditions: The user must be logged into their account. The user must have at least one item listed for trade.
Basic Course of Events /Main Flow
<ol style="list-style-type: none"> User Action (Step 1): The user navigates to their item listings and selects an item to delete.

System Response: The system removes the item from the listings and updates the database. A confirmation message is displayed: "Item successfully deleted."
Alternate Flow: None
Exception Flow (System Errors) System Failure: If the system encounters an error during processing of request. System Response: The system shows an error message.
Post Conditions: The item is successfully deleted from the user's listings and the database.

Table 17: 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 18: 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:

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

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

Table 19: 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 20: 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:

Use Case Name: Search and Filter Item Listings
Description: Users can search for specific items and apply filters to narrow down the results.
Actors: Users (Traders).
Goal: Allow users to quickly find relevant items by searching keywords or applying filters.
Pre-Conditions: The user must be on the item listings page.
Basic Course of Events /Main Flow
<ol style="list-style-type: none"> User Action (Step 1): The user enters a keyword in the search bar or selects filter options (e.g., category, condition). System Response: The system retrieves and displays a list of items that match the search term or selected filters.
<ol style="list-style-type: none"> User Action (Step 2): The user views the filtered results and selects an item for more details. System Response: The system displays the selected item's detailed information, including description, images, and trade options.
Alternate Flow (No Results Found))

System Response: If no items match the search or filters, the system displays a message.
Exception Flow (System Errors)
System Failure: If the system encounters an error during processing of request.
System Response: The system shows an error message.

Post Conditions: The user successfully views filtered item listings and can select an item for more details.

Table 21: 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 22: 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:

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

System Failure: If the system encounters an error during processing of request.
System Response: The system shows an error message.
Post Conditions: The user successfully adds & views items from their wishlist.

Table 23: 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 24: 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:

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

Exception Flow (System Errors)
System Failure: If the system encounters an error during the retrieval of wishlist data.
System Response: The system displays an error message: "Unable to load wishlist. Please try again later."
Post Conditions: The user successfully views the items saved in their wishlist and can interact with them as needed.

Table 25: 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 26: 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:

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

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

Table 27: 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 28: 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:

Use Case Name: Trade Proposals
Description: Users can propose a trade by offering an item in exchange for another user's listed item.
Actors: Users (Traders).
Goal: Allow users to propose trades for listed items.
Pre-Conditions: The user must be logged into their account. The user must have an item listed for trade.
Basic Course of Events /Main Flow

<p>1. User Action (Step 1): The user navigates to an item listing they want to trade for and clicks "Propose Trade." System Response: The system displays a form where the user selects their own item to offer.</p> <p>2. User Action (Step 2): The user submits the trade proposal by clicking the "Submit" button. System Response: 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."</p>
Alternate Flow (User Cancels Proposal)
<p>User Action: The user cancels the proposal before submission. System Response: The system aborts the trade proposal, and no data is saved.</p>
Exception Flow (System Errors)
<p>System Failure: If the system encounters an error during processing of request. System Response: The system shows an error message.</p>
Post Conditions: The trade proposal is successfully submitted and the other user is notified.

Table 29: 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 30: 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:

Use Case Name: Accept Trade
Description: Users can accept a trade proposal to finalize the exchange of items.
Actors: Users (Traders).

Goal: Allow users to accept a trade proposal.
Pre-Conditions: The user must have received a trade proposal.
Basic Course of Events / Main Flow:
<p>1. User Action (Step 1): The user navigates to their trade proposals and selects a pending trade to review. System Response: The system displays the trade details, including the item offered.</p> <p>2. User Action (Step 2): The user clicks the "Accept" button to confirm the trade. System Response: 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."</p>
Alternate Flow (User Rejects Trade)
<p>User Action: The user clicks "Reject" instead of "Accept." System Response: The system updates the trade status and notifies the proposing user.</p>
Exception Flow (System Errors)
<p>System Failure: If the system encounters an error during processing of request. System Response: The system shows an error message.</p>
Post Conditions: The trade is successfully accepted, and both users are notified.

Table 31: 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 32: 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:

Use Case Name: Reject Trade
Description: Users can reject a trade proposal they are not interested in.
Actors: Users (Traders).
Goal: Allow users to decline a trade proposal.
Pre-Conditions: The user must have received a trade proposal.
Basic Course of Events /Main Flow
<p>1. User Action (Step 1): The user navigates to their trade proposals and selects a pending trade to review. System Response: The system displays the trade details, including the item offered and any terms of the trade.</p> <p>2. User Action (Step 2): The user clicks the "Reject" button to decline the trade. System Response: The system updates the trade status in the database and notifies the other user of the rejection. A message is displayed: "Trade rejected."</p>
Alternate Flow User Accepts Trade))
<p>User Action: The user clicks "Accept" instead of "Reject." System Response: The system updates the trade status and notifies the other user.</p>
Exception Flow (System Errors)
<p>System Failure: If the system encounters an error during processing of request. System Response: The system shows an error message.</p>
Post Conditions: The trade is successfully rejected, and both users are notified.

Table 33: 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 34: 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:

Use Case Name: Cancel Trade
Description: Users can cancel an ongoing trade proposal that they initiated.
Actors: Users (Traders).
Goal: Allow users to cancel their trade proposals before they are accepted.
Pre-Conditions: The user must be logged into their account and have an active trade proposal.
Basic Course of Events /Main Flow
<p>1. User Action (Step 1): The user navigates to the "Sent Requests" section and selects an active trade they want to cancel. System Response: The system fetches the details of the selected trade and displays a "Cancel Trade" button</p> <p>2. User Action (Step 2): The user clicks on the "Cancel Trade" and confirms the cancellation. System Response: 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."</p>
Alternate Flow: If the user declines to confirm the cancellation: System Response: The system aborts the cancellation and keeps the trade active.
Exception Flow (System Errors) System Failure: If the system encounters an error while processing the cancellation. System Response: The system displays an error message: "Unable to cancel the trade. Please try again later."
Post Conditions: The trade is successfully cancelled, and the other party is notified.

Table 35: 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 36: 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:

Use Case Name: Counter Offer
Description: Users can create a counter trade offer by selecting an item from the sender's list while their item remains pre-selected.
Actors: Users (Traders).
Goal: Allow users to negotiate trades by making counter offers.
Pre-Conditions: The user must have received a trade proposal and must be logged into their account.
Basic Course of Events /Main Flow
<ol style="list-style-type: none"> User Action (Step 1): The user navigates to the "Active Trades" section and clicks on the "Counter Offer" button for a specific trade. System Response: The system opens a trade proposal page with the recipient's item pre-selected. User Action (Step 2): The user selects an item from the sender's listed items to include in the counter proposal. System Response: The system updates the trade proposal with the selected item from the sender. User Action (Step 3): The user clicks "Send Counter Offer". System Response: If no item is selected from the sender's list, the system displays an error message: "Please select an item." If an item is selected: <ul style="list-style-type: none"> The system saves the counter proposal in the database. Notifies the sender about the counter proposal. Displays confirmation: "Counter offer sent successfully."
Alternate Flow: If the user cancels the counter offer: System Response: The system discards the changes and navigates back to the "Active Trades" section.
Exception Flow (System Errors) System Failure: If the system encounters an error while saving the counter proposal. System Response: The system displays an error message: "Unable to send the counter offer. Please try again later."
Post Conditions: The counter offer is successfully sent, and the sender is notified.

Table 37: 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 38: 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:

Use Case Name: Manage Active trades
Description: Users can view, accept, or reject active trade proposals.
Actors: Users (Traders).
Goal: Allow users to manage their active trade proposals.
Pre-Conditions: The user must have at least one active trade proposal.
Basic Course of Events /Main Flow
<ol style="list-style-type: none"> User Action (Step 1): The user navigates to the "Active Trades" section. System Response: The system displays a list of all active trade proposals, including the trade details and status. User Action (Step 2): The user selects a trade to manage and either accepts or rejects it. System Response: 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."
Alternate Flow (No Active Trades)
System Response: If there are no active trades, the system displays a message: "No active trades at the moment."
Exception Flow (System Errors)

System Failure: If the system encounters an error during processing of request.
System Response: The system shows an error message.
Post Conditions: The trade status is successfully updated, and both users are notified.

Table 39: 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 40: 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:

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

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

Table 41: 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 42: 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
Description: Admin can remove any items listed by users on the platform.
Actors: Admin.
Goal: Allow admin to remove user-listed items, ensuring compliance with platform policies.
Pre-Conditions: The admin must be logged into the admin portal with the required permissions.
Basic Course of Events /Main Flow
<ol style="list-style-type: none"> 1. User Action (Step 1): The admin navigates to the "Listed Items" section.

<p>System Response: The system displays a list of all items listed by users, including item details such as title, description, category, and status.</p> <p>2. User Action (Step 2): The admin selects an item to view or remove.</p> <p>System Response: The system processes the admin's action:</p> <ul style="list-style-type: none"> • If the item is removed, it is deleted from the listings and the database. • A confirmation message is displayed: "Item updated/removed successfully."
Alternate Flow (No Items to Manage)
<p>System Response: If there are no items requiring management, the system displays a message: "No items to manage at this time."</p>
<p>Exception Flow (System Errors)</p> <p>System Failure: If the system encounters an error during processing of request.</p> <p>System Response: The system shows an error message.</p>
Post Conditions: The item is successfully removed, and the action is logged.

Table 43: 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 44: 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:

Use Case Name: Rate and Review System
Description: Users can rate and review trading partners after completing a trade to provide feedback.
Actors: Users (Traders).
Goal: Allow users to rate and review their trade partners, fostering trust within the community.
Pre-Conditions: The user must have completed a trade.
Basic Course of Events /Main Flow

<p>1. User Action (Step 1): The user navigates to the "Rate Trade" section after completing a trade.</p> <p>System Response: The system displays a form allowing the user to give a rating (e.g., 1-5 stars) and write a review.</p>
<p>2. User Action (Step 2): The user submits the rating and review by clicking the "Submit" button.</p> <p>System Response: The system validates the input and saves the rating and review to the database. A confirmation message is displayed: "Rating and review submitted successfully."</p>
Alternate Flow: None
<p>Exception Flow (System Errors)</p> <p>System Failure: If the system encounters an error during processing of request.</p> <p>System Response: The system shows an error message.</p>

Table 45: 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	Complete
Req 2	The system shall save the rating and review to the database and update the partner's profile.	High	Complete

Table 46: 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:

Use Case Name: Email Notification
Description: The system sends email notifications to users regarding important updates such as trade proposals advance trade or account update.
Actors: Users (Traders).
Goal: Notify users about important updates via email.
Pre-Conditions: The user must have a registered and verified email address. The user must have relevant activities triggering notifications.
Basic Course of Events /Main Flow

1. System Action (Step 1): A relevant event occurs (e.g., new trade proposal, trade accepted, account update). System Response: The system generates an email with the details of the event and sends it to the user's registered email address.
Alternate Flow: None
Exception Flow (System Errors) System Failure: If the system encounters an error during processing of request. System Response: The system shows an error message.
Post Conditions: The user is notified of important updates via email.

Table 47: 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 48: 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:

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

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

Table 49: 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 50: Trade History Functional Requirements

4.24 Send Message

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:

Use Case Name: Send Messages
Description: Users can send messages to other users to negotiate trades or communicate regarding items.
Actors: Users (Traders).
Goal: Allow users to communicate with each other through the platform's messaging system.
Pre-Conditions: The user must be logged into their account. The user must have an active trade or user they want to message.
Basic Course of Events /Main Flow
<ol style="list-style-type: none"> User Action (Step 1): The user navigates to a trade or selects a user to message. System Response: The system displays a messaging interface where the user can compose a message. User Action (Step 2):

The user types the message and clicks "Send." System Response: The system delivers the message to the recipient.
Alternate Flow: None
Exception Flow (System Errors) System Failure: If the system encounters an error during processing of request. System Response: The system shows an error message.
Post Conditions: The message is successfully sent, and the recipient receives it.

Table 51: 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	Complete
Req 2	The system shall deliver messages in real-time to the recipient.	High	Complete

Table 52: 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:

Use Case Name: Send Messages
Description: Users can send messages to other users to negotiate trades or communicate regarding items.
Actors: Users (Traders).
Goal: Allow users to communicate with each other through the platform's messaging system.
Pre-Conditions: The user must be logged into their account. The user must have an active trade or user they want to message.
Basic Course of Events /Main Flow
<p>3. User Action (Step 1): The user navigates to a trade or selects a user to message. System Response: The system displays a messaging interface where the user can compose a message.</p> <p>4. User Action (Step 2): The user types the message and clicks "Send." System Response: The system delivers the message to the recipient.</p>

Alternate Flow: None
Exception Flow (System Errors)
System Failure: If the system encounters an error during processing of request. System Response: The system shows an error message.
Post Conditions: The message is successfully sent, and the recipient receives it.

Table 53: 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 54: 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 systems must be implemented to ensure 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.

Software Design Specification

1. Introduction

1.1. Document Purpose

The purpose of this document is to provide a detailed description of the system, including its functionality and underlying assumptions. It is designed to give the intended audience comprehensive insights into these aspects. Additionally, the document outlines the system's software architecture and explains the sequence or workflow for specific tasks. This information is intended to facilitate understanding and ease of use for both stakeholders and system developers.

1.2. Scope of the Development Project

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 & negotiate trade in real-time.

By focusing on these key areas, Swappify 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.

1.3. Definitions, Acronyms, and Abbreviations

- **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:** The 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.

1.4. Overview of Document

- **Section 1.0:** An introduction to the document, explaining its purpose, providing definitions, acronyms, abbreviations, references, and a brief overview of each section within the document.
- **Section 2.0:** A comprehensive description of the system architecture, including general constraints, data design, program structure, and alternative models considered.
- **Section 3.0:** Detailed descriptions of each system component, covering their purpose, dependencies, interfaces, resources, processing logic, and data structures, to guide the implementation process.
- **Section 4.0:** A focused discussion on user interface design, detailing the interface design rules, GUI components, and a visual or descriptive representation of the user interface.
- **Section 5.0:** An exploration of reuse strategies, including leveraging existing modules, tools, and open-source resources, and discussing their role in the product's design and implementation.
- **Section 6.0:** Documentation of key design decisions and trade-offs, providing insight into the reasoning behind the chosen design and explaining ideas that were not implemented.
- **Section 7.0:** Pseudocode for components, offering clear and structured logic for system functionality to aid in development and testing.

- **Section 8.0:** Appendices containing supplementary materials, including class diagrams, sequence diagrams, and other visual aids that provide additional context and clarity for the system design.

2. System Architecture Description

2.1. Section Overview

This section outlines the components and subsystems of Swappify, including the system's data design. It covers the system requirements, hardware and software environments, limitations, an ERD diagram, the architectural model with its major components, and alternatives to the chosen architectural model.

2.2. General Constraints

The goal is to develop an application capable of handling large datasets while ensuring a seamless user experience for both sending and receiving responses. The application is designed to be compatible with all major web browsers, focusing on accessibility and responsiveness. Additionally a key constraint is the use of cloud deployment for the application. This approach is intended to optimize performance and ensure efficient handling of data without compromising the user experience.

2.3. Data Design

The system uses MongoDB as its NoSQL database, where data is stored in collections consisting of documents. Each collection contains related documents, with each document representing a specific data entity. These collections are designed to store various types of data and allow for flexible, scalable storage of unstructured information. External files in CSV and JSON formats are used for data import and export. For a detailed diagram of the database structure, refer to the **ERD** in **Section 8** of the appendices.

2.4. Program Structure

The system follows a client-server architecture design. The client-side is built using React JS, providing a dynamic interface that interacts with the server side via API requests. The server side is powered by Node JS and Express JS, handling business logic and database interactions with MongoDB. The system utilizes several key classes that represent the major components and entities in the application. Each class is responsible for handling specific functionality and interactions within the system. These classes work together to ensure proper data handling and application functionality. For a detailed visual representation, refer to the **Class-Diagram** in **Section 8** of the appendices.

2.5. Alternatives Considered

No alternative architectural models were considered. The client server architecture design was chosen for its scalability, maintainability, and efficient data management. This model ensures a clear separation between the user interface and server side logic, providing flexibility and long term performance.

3. Detailed Description of Components

3.1. Section Overview

This section outlines the foundation for implementing the system. In Section 3.2, we provide a comprehensive description of the components, including their details, data members, methods, and a structured template for each component.

3.2. Component and Detail

3.2.1. Login

Identification	Login
Type	Module
Purpose	Its purpose is to authenticate users by validating their email and password, ensuring that only authorized users can access their accounts.
Function	The Login module processes user inputs (email and password), validates the email format, and checks for empty fields. It authenticates users by comparing credentials with the database.
Subordinates	The module consists of the login form, authentication logic, and error handling, fulfilling requirements for displaying forms, validating credentials, and showing error messages.
Dependencies	It depends on the user database & email validation.
Interfaces	The module interacts with the user through the login form and communicates with the database for authentication. Error messages are displayed for invalid inputs.
Resources	The Login module requires server resources, a database for checking user credentials, and hashing services for password verification.
Processing	The module validates the email and password, compares them with database records, and either authenticates the user or shows an error message.
Data	The module processes user email and password for authentication, ensuring proper formatting and verification against stored credentials.

Table 55: Login Component Detail

3.2.2. SignUp

Identification	Signup
Type	Module
Purpose	It allows new users to register by submitting their name, email, password, and other details to create an account.
Function	The system validates the input, checks if the email is already registered, saves user data, and sends a verification email. Once the user clicks the verification link, the account is activated, and they are redirected to the login page.
Subordinates	The module includes the sign-up form, input validation, database storage, and email verification.
Dependencies	It depends on the user database, email validation & account activation.
Interfaces	The module interacts with the user via the sign-up form, with the database for storing information, and the package to send verification links.
Resources	It requires server resources, a database for checking user credentials, and hashing services for password verification.
Processing	The system validates inputs, checks for existing emails, stores data, sends a verification email, and activates the account after verification.
Data	The system processes the user's name, email, password, and mobile number, storing them in the database if the email is not already registered.

Table 56: SignUp Component Detail

3.2.3. Forgot Password

Identification	Forgot Password
Type	Module
Purpose	It enables users to securely reset their password by receiving a generated password via their registered email, ensuring account recovery through email verification and link validation.
Function	The system provides a form to enter the registered email, checks if the email exists, sends a new password if valid, and authenticates the user after the password reset.
Subordinates	The module includes email validation, reset link generation, password updating, and user authentication.
Dependencies	It depends on the user database for email validation, an email service to send reset links, and a hashing service for secure password storage.
Interfaces	The module interacts with users through the "Forgot Password" form, with the database for email verification, and the email service to deliver reset links.
Resources	It requires server resources to process requests, a database for validating registered emails, and an email service to send reset links.
Processing	The system verifies the user's email, generates a secure reset password link, sends it to the user's registered email, and updates the password in the database upon reset.
Data	The system processes the user's registered email and stores the new password securely after resetting, ensuring data integrity and account security.

Table 57: Forgot Password Component Detail

3.2.4. Profile Management

Identification	Profile Management
Type	Module
Purpose	The purpose is to enable users to modify their profile details, ensuring their information is up-to-date for effective communication and accurate platform interactions.
Function	The user navigates to their profile, edits fields like name, email, and phone number, and saves the changes. The system validates the updated information, saves it, and displays the new details. If invalid data is entered, the system shows error messages.
Subordinates	The module includes the profile page, edit functionality, and input validation.
Dependencies	It depends on the user account database to store and retrieve profile information.
Interfaces	The system interacts with the user via the profile page and the "Edit" functionality to update information.
Resources	The module requires server resources for updating and storing profile data.
Processing	The system allows users to update their profile, validates the input, and saves the changes to the database.
Data	The system processes and stores updated user information, such as name, email, and phone number.

Table 58: Profile Management Component Detail

3.2.5. List Item

Identification	List Item
Type	Module
Purpose	It enables users to list new items for trade by providing essential details, expanding item inventory and fostering trade opportunities.
Function	The system allows users to fill out a form with item details, validates the inputs, and saves the item to the database. It provides feedback for successful listing or errors for invalid inputs.
Subordinates	The module includes the item listing form, input validation, database storage, and feedback messages.
Dependencies	It relies on the database for storing item details, file validation for images, and user authentication for account access.
Interfaces	The module interacts with the user through the "Add New Item" form and connects with the database for storing validated item details.
Resources	It requires server resources for processing item details, a database to store listings, and validation tools for verifying input formats.
Processing	The system verifies the input data, validates mandatory fields and file formats, stores the new item in the database, and displays a confirmation or error message based on the outcome.
Data	The system processes and stores item details provided by the user, including item title, description, category, and image files.

Table 59: List Item Component Detail

3.2.6. Edit Item

Identification	Edit Item
Type	Module
Purpose	It allows users to modify the details of an existing item they have listed for trade, ensuring that item listings are accurate and up-to-date..
Function	The system displays the current item details in an editable form, validates the updated information, saves the changes to the database, and provides confirmation upon successful updates.
Subordinates	The module includes the item editing form, input validation checks, and database storage for updated item details.
Dependencies	It depends on the user account system for access, the item database for storing and retrieving item details, and validation services for checking mandatory fields and data accuracy.
Interfaces	The module interacts with users via the item editing form, the database for saving updates, and error handling for invalid inputs.
Resources	It requires server resources to process the edits, a database to store the updated item details, and validation services for ensuring the data is correct.
Processing	The system processes the updated item details, validates the input, saves the new information to the database, and displays a confirmation message after successful updates.
Data	The system processes the item's updated details such as title, description, price, and images, ensuring they meet the required formats and validations before saving them in the database.

Table 60: Edit Item Component Detail

3.2.7. Delete Item

Identification	Delete Item
Type	Module
Purpose	It allows users to remove an existing item they have listed for trade, ensuring the inventory is kept accurate and updated by enabling users to manage their listings effectively.
Function	The system displays a button to delete an item, removes the item from the listings upon user confirmation, updates the database, and provides a confirmation message.
Subordinates	The module includes the delete button, the action to remove the item from the database, and the confirmation message displayed to the user.
Dependencies	It depends on the user account system for access, the item database for deleting and retrieving item details, and confirmation services for ensuring the item is successfully removed.
Interfaces	The module interacts with users via the item deletion button, the database for removing the item, and error handling for system failures.
Resources	It requires server resources to process the item removal, a database to update listings, and validation services to ensure the correct item is deleted.
Processing	The system processes the request to delete an item, removes the item from the user's listings, updates the database, and displays a confirmation message to the user.
Data	The system processes the item's details and deletes the selected item from the database after confirming the action.

Table 61: Delete Item Component Detail

3.2.8. View Item Listing

Identification	View Item Listing
Type	Module
Purpose	The purpose is to enable users to view all items available for trade, helping them make informed trading decisions and enhancing their engagement with the platform.
Function	The user navigates to the item listings page, and the system displays all available items with basic details. The user can select an item to view its full details, including descriptions, images, and trade options. If no items are available, the system displays a message.
Subordinates	The module includes item listing display and detailed view functionality.
Dependencies	It depends on the item database to retrieve and display available items.
Interfaces	The system interacts with the user through the item listings page and detailed item view.
Resources	The module requires server resources to retrieve and display item data.
Processing	The system processes the user's request to view available items and their details.
Data	The system handles data related to item listings, including basic details and full descriptions.

Table 62: View Item Listing Component Detail

3.2.9. Search & Filter

Identification	Search & Filter
Type	Module
Purpose	Its purpose is to enhance the user experience by enabling users to easily find relevant items based on search terms or selected filters (e.g., category, condition).
Function	The user enters a keyword or selects filter options, and the system retrieves and displays matching items. If no items match, the system displays a message. Users can then select an item for detailed information.
Subordinates	The module includes the search bar, filter options, item listings display, and message display for no results.
Dependencies	It depends on the item database to retrieve and filter item listings.
Interfaces	The system interacts with the user through the search bar and filter options and displays filtered results on the item listings page.
Resources	The module requires server resources to process searches, filter data, and display results.
Processing	The system processes the user's search or filter inputs, retrieves matching items, and displays them. If no results are found, it shows a message.
Data	The system processes search keywords and filter criteria to query and display relevant item listings.

Table 63: Search & Filter Component Detail

3.2.10. Add to Wishlist

Identification	Add to Wishlist
Type	Module
Purpose	It allows users to save items of interest to their wishlist for future reference, enabling easier access to desired items without repeated searching.
Function	The system allows users to add items to their wishlist, displays a confirmation message, and shows the saved items when the user navigates to the wishlist section.
Subordinates	The module includes the "Add to Wishlist" button, the wishlist section in the user profile, and the process of adding and removing items from the wishlist.
Dependencies	It depends on the user account system for item tracking, the wishlist database for saving and retrieving items, and the confirmation service for notifying the user.
Interfaces	The module interacts with users via the "Add to Wishlist" button and the wishlist section, with the database for storing and retrieving wishlist items, and error handling for system failures.
Resources	It requires server resources for processing wishlist actions, a database for storing wishlist items, and the user interface for displaying the wishlist.
Processing	The system processes the user's request to add an item to the wishlist, updates the wishlist in the database, and displays the user's wishlist when accessed.
Data	The system processes the item details, storing them in the user's wishlist for future access and removing items as requested by the user.

Table 64: Add to Wishlist Component Detail

3.2.11. View Wishlist

Identification	View Wishlist
Type	Module
Purpose	It allows users to access and browse the items they have previously added to their wishlist, enabling easy review and interaction with saved items.
Function	The system retrieves and displays the saved wishlist items, provides details of each item, and allows users to view full item details or initiate trades. If the wishlist is empty, it displays a message informing the user.
Subordinates	The module includes the wishlist display, the retrieval of item details from the database, and the interaction with the items (viewing full details or initiating trade).
Dependencies	It depends on the user account system, the wishlist database for storing saved items, and the error handling service for managing retrieval issues.
Interfaces	The module interacts with users through the wishlist section in the profile, with the database for retrieving item data, and displays error messages when necessary.
Resources	It requires server resources for fetching and displaying wishlist items, a database for storing and retrieving saved items, and the user interface for browsing and managing the wishlist.
Processing	The system processes the user's request to view the wishlist, retrieves the saved items, displays the items with relevant details, and handles cases such as an empty wishlist.
Data	The system processes the user's wishlist data, including item names, images, and descriptions, and presents them for user interaction.

Table 65: View Wishlist Component Detail

3.2.12. Item Detail

Identification	Item Detail
Type	Module
Purpose	It allows users to view detailed information about a specific item, including its name, description, condition, category, images, and owner details, to help them make informed decisions about the item.
Function	The system fetches and displays comprehensive details about an item, including available actions such as initiating a trade or adding to a wishlist. If the item is unavailable, a message is displayed to notify the user.
Subordinates	The module includes item detail display, actions such as "Propose Trade" or "Add to Wishlist," and error handling for unavailable items or system failures.
Dependencies	It depends on the item database for retrieving item details, the user interface for presenting the details, and the action services for managing trade proposals and wishlist additions.
Interfaces	The module interacts with users through the item detail page, with the database for retrieving item data.
Resources	It requires server resources for fetching and displaying item details, a database for storing item information, and the interface for presenting data and interacting with users.

Processing	The system processes the request to display item details, retrieves data from the database, displays the item with its details, and handles actions such as trade proposals or adding items to the wishlist.
Data	The system processes the item's name, description, condition, category, images, and owner information to present them on the item detail page and allow user interaction.

Table 66: Item Detail Component Detail

3.2.13. Trade Proposal

Identification	Trade Proposal
Type	Module
Purpose	This feature enables users to propose exchanges of items, facilitating seamless and transparent trading by specifying desired items and offering their own for trade.
Function	The user selects an item they wish to trade for, proposes a trade by offering their own item, and submits the proposal. The system validates the inputs, saves the proposal, and notifies the other user.
Subordinates	The feature includes the proposal form and trade validation process.
Dependencies	The feature depends on the user's login status and having an item listed for trade.
Interfaces	The system interacts with the user via the trade proposal form and notifies the recipient of the proposal.
Resources	The system uses the database to store trade proposals.
Processing	The system processes trade proposals, validates inputs, saves data, and sends notifications to the recipient.
Data	The system handles trade proposal data, including item selection and notifications.

Table 67: Trade Proposal Component Detail

3.2.14. Accept Trade

Identification	Accept Trade
Type	Module
Purpose	It allows users to confirm and finalize a trade proposal by accepting the offered terms, ensuring successful transactions and item exchanges between users on the platform.
Function	The system displays the trade proposal details, and upon user confirmation, it updates the trade status in the database and notifies the other user of the acceptance, ensuring the exchange is finalized.
Subordinates	The module includes displaying trade proposal details, the accept/reject actions, updating trade status, and notifying the other user of the trade decision.
Dependencies	It depends on the trade database for tracking proposals, the user interface for displaying trade details, and notification services for informing users about the trade status.

Interfaces	The module interacts with users through the trade proposal page, with the database for managing trade status, and with notification systems for alerting the proposing user upon acceptance.
Resources	It requires server resources to process trade proposals, a database to store trade statuses, and notification systems to communicate the result of the trade acceptance.
Processing	The system retrieves and displays the trade details, updates the trade status upon user acceptance or rejection, and ensures both users are notified of the decision.
Data	The system processes trade proposal information, including item details, user identities, and trade status updates, to facilitate the exchange and notify both users about the acceptance or rejection.

Table 68: Accept Trade Component Detail**3.2.15. Reject Trade**

Identification	Reject Trade
Type	Module
Purpose	It allows users to decline unwanted trade proposals, ensuring they maintain control over their trade interactions and only engage in desired exchanges.
Function	The system displays the trade proposal details, and upon rejection by the user, it updates the trade status in the database and notifies the proposing user of the rejection.
Subordinates	The module includes displaying trade proposal details, the reject action, updating trade status, and notifying the proposing user about the rejection.
Dependencies	It depends on the trade database for tracking proposals, the user interface for displaying trade details, and notification systems for informing users about the trade status.
Interfaces	The module interacts with users through the trade proposal page, with the database for managing trade statuses, and with notification systems to alert the proposing user about the rejection.
Resources	It requires server resources to process trade proposals, a database to store trade statuses, and notification systems to communicate the rejection outcome to the proposing user.
Processing	The system retrieves and displays the trade proposal details, updates the trade status upon user rejection, and ensures the proposing user is notified about the decision.
Data	The system processes trade proposal information, including item details, user identities, and trade status updates, to facilitate the rejection and notify the proposing user.

Table 69: Reject Trade Component Detail**3.2.16. Cancel Trade**

Identification	Cancel Trade
Type	Module
Purpose	It allows users to cancel an ongoing trade proposal they initiated, ensuring they maintain control over active trades and can withdraw proposals before acceptance.

Function	The system displays the active trade, prompts the user for confirmation to cancel, updates the trade status in the database to "Cancelled," and notifies the other user of the cancellation.
Subordinates	The module includes retrieving the active trade details, the cancel action, updating the trade status, and notifying the other party about the cancellation.
Dependencies	It depends on the trade database to track active trade proposals, the user interface to present the cancel option, and the notification system to inform the other party about the cancellation.
Interfaces	The module interacts with the user through the trade proposal section, the database to update trade status, and the notification system to inform the other party about the cancellation.
Resources	It requires server resources to process the cancellation, a database to update the trade status, and notification systems to alert the other user of the cancellation.
Processing	The system retrieves the active trade, prompts for confirmation to cancel, updates the trade status, and ensures the other user is notified of the cancellation.
Data	The system processes trade proposal details, user identities, and cancellation status to update the trade status and notify the other party.

Table 70: Cancel Trade Component Detail

3.2.17. Advanced Trade Management

Identification	Advanced Trade Management
Type	Module
Purpose	The purpose of the Advanced Trade Management feature, specifically the Counter Offer, is to allow users to negotiate trade terms by selecting an item from the sender's list while keeping their previously proposed item selected. It fosters negotiation and enhances the trade experience by allowing users to propose alternate terms.
Function	This feature enables users to create counter trade offers by selecting items from the sender's list, while keeping their own previously proposed item. It supports seamless trade negotiation and notifies the sender when a counter offer is sent.
Subordinates	The feature includes opening a trade proposal page, validating item selection, saving the counter proposal, sending notifications, and error handling for failed submissions.
Dependencies	This feature depends on active trade data (items and trade offers), a system to track user selections, and a notification system to inform the sender of the counter proposal.
Interfaces	The system interacts with the user through the trade interface, communicates with the database to save the counter offer, and uses the notification system to inform the sender.
Resources	The feature requires server resources to handle counter offer submissions, database resources for tracking trade changes, and a notification system for informing users of the counter offer.
Processing	The system fetches the trade proposal, allows the user to select an item from the sender's list, validates the selection, saves the counter offer, and notifies the sender. If the user cancels, the system discards the changes.
Data	The system processes the trade details, user selections, and trade status to ensure that the counter offer is properly handled and that the sender is notified. It also manages error handling and displays confirmation or error messages as necessary.

Table 71: Advanced Trade Management Component Detail

3.2.18. Manage Active Trades

Identification	Manage Active Trades
Type	Module
Purpose	The Manage Active Trades feature enables users to view, accept, or reject ongoing trade proposals. This functionality ensures that users can actively engage in trading and take control of the status of their trades, making the platform more interactive and user-driven.
Function	Users can view all active trade proposals, manage their status by either accepting or rejecting the trade, and receive notifications upon updating trade statuses. This ensures seamless interaction between users and improves trading efficiency.
Subordinates	The feature includes the display of active trade proposals, the ability to accept or reject trades, notifications to both parties involved in the trade, and error handling for system failures or issues when processing requests.
Dependencies	This feature relies on an active trade database to fetch trade proposals, a user notification system, and a backend system that tracks and updates the trade status based on user actions.
Interfaces	The system interacts with the trade interface for displaying active trade proposals, the database for updating trade status, and the notification system for informing users of trade updates.
Resources	The feature requires server resources for fetching active trades, updating trade statuses, and sending notifications. Additionally, it requires database resources for storing trade information and status.
Processing	The system processes the user's action of accepting or rejecting a trade, updates the trade status in the database, and notifies the other user involved in the trade. If there are no active trades, the system displays an appropriate message.
Data	The system processes data regarding active trade proposals, their details, trade status, and user actions. The system also manages the notification data to inform both users about the trade status updates.

Table 72: Manage Active Trades Component Detail

3.2.19. Ban User Account

Identification	Ban User Account
Type	Module
Purpose	The Ban User Accounts feature enables administrators to manage user behavior by banning accounts that violate platform policies. This ensures a safe and secure environment by allowing administrators to take action against disruptive or rule-breaking users.
Function	Administrators can view all user accounts, select specific users, and ban them from the platform if necessary. Banned users will no longer be able to access or interact with the platform.
Subordinates	The feature includes user account listing, selection for banning, system updates to account status, and error handling for issues during the banning process. Additionally, it includes a notification system to inform the admin of successful actions or errors.
Dependencies	This feature depends on an admin interface for managing user accounts, a backend database to store and update user statuses, and proper permission settings to ensure that only authorized admins can ban accounts.

Interfaces	The system interfaces with the admin dashboard for account management, the database for retrieving and updating user account statuses, and the notification system for informing administrators of successful or failed actions.
Resources	The feature requires database resources to update user account statuses, server resources to process admin requests, and access to the admin portal for administrators to manage accounts.
Processing	The system processes the admin's request to ban a user by updating the account status in the database. If the action is successful, a confirmation message is shown; if not, an error message is displayed.
Data	The system handles user account data, including user profiles, account statuses, and permissions. It also manages data related to the admin's action, such as the user selected for banning and confirmation messages.

Table 73: Ban User Account Component Detail

3.2.20. Delete List Item

Identification	Delete List Item
Type	Module
Purpose	The Delete List Items feature enables administrators to remove items listed by users on the platform. This ensures the integrity of the marketplace by enforcing platform policies and guidelines, preventing the sale or display of prohibited or inappropriate items.
Function	Administrators can view a list of all items listed by users, select items to review or remove, and delete those items from the marketplace when necessary.
Subordinates	The feature includes the listing of user-submitted items, item selection for removal, deletion of items from the database, and error handling in case of issues. It also includes confirmation or error messages for administrators.
Dependencies	This feature depends on the admin portal for managing listings, a backend database for storing and removing item listings, and proper permissions to ensure only authorized admins can delete items.
Interfaces	The system interfaces with the admin dashboard for viewing listed items, the database for retrieving and deleting items, and a notification system to confirm actions or alert administrators of any errors.
Resources	The feature requires database resources to store, retrieve, and delete item listings, server resources for processing the deletion request, and access to the admin portal for authorized users to manage items.
Processing	When an admin deletes an item, the system removes the item from the listings and the database. A confirmation message is displayed to the admin to verify the successful removal. If an error occurs during the deletion process, an error message is displayed.
Data	The system handles data related to the items listed by users, including item descriptions, categories, and statuses. It also manages data related to the admin's actions, such as which items are selected for removal and logs of those actions.

Table 74: Delete List Item Component Detail

3.2.21. Rate & Review System

Identification	Rate & Review System
Type	Module
Purpose	The Rate and Review System enables users to provide feedback on their trading partners after completing a trade. This feature helps enhance trust within the platform's community by allowing users to share their experiences, both positive and negative, and assist future traders in making informed decisions.
Function	The system allows users to submit a rating (typically in the form of stars or numerical value) and a written review after completing a trade. These ratings and reviews are saved in the database and can be viewed on the rated user's profile.
Subordinates	The feature includes the display of the rating and review form, input validation for ratings and reviews, storing feedback in the database, and updating the user profile to reflect the new ratings/reviews.
Dependencies	The feature depends on the user's completion of a trade, access to the "Rate Trade" section, the backend database for storing ratings and reviews, and user profiles that will display the ratings/reviews.
Interfaces	The system interfaces with the user interface for submitting ratings and reviews, the backend database for saving and retrieving the reviews, and the user profile pages to display feedback for future reference.
Resources	The feature requires access to the database for storing ratings and reviews, as well as system resources to validate and save the feedback. It also requires an interface for users to submit their ratings.
Processing	When the user submits their rating and review, the system validates the input, ensures the rating is within a valid range (e.g., 1-5 stars), and saves the review to the database. A confirmation message is displayed to the user once the process is complete. If an error occurs, the system displays an error message.
Data	The system manages data related to the ratings and reviews submitted by users. This includes the trade partner being rated, the rating value, the review text, and the user submitting the feedback. It also manages any error messages if issues occur during the submission process.

Table 75: Rate & Review System Component Detail

3.2.22. Email Notification

Identification	Email Notification
Type	Module
Purpose	The feature ensures users are notified of significant events like trade proposals, acceptances, rejections, or cancellations to help them make informed decisions.
Function	The system sends automated email notifications to users based on relevant trade events and updates to keep them informed.
Subordinates	The feature depends on event triggers, email content generation, and email delivery to the correct recipients.

Dependencies	It relies on verified user email addresses, system event triggers, and an email service to send notifications.
Interfaces	It interacts with the database for user email details and with the email service to send notifications.
Resources	The feature requires access to a database for user email addresses and an email service for delivery.
Processing	When an event occurs, the system generates and sends an email notification to the user's registered email address.
Data	The system processes user email addresses and event details to ensure notifications are sent accurately and timely.

Table 76: Email Notification Component Detail

3.2.23. Trade History

Identification	Trade History
Type	Module
Purpose	It helps users review and analyze completed trades for record-keeping. This feature enhances transparency.
Function	The feature retrieves and shows a list of completed trades, providing details about each. Users can click to view trade-specific information.
Subordinates	The system relies on the user's trade data and account information to display history. It integrates with the database for trade records.
Dependencies	It requires a user account with completed trades and access to stored trade data. The system must be able to retrieve these records.
Interfaces	The feature interfaces with the user account system and trade database to present historical trade data.
Resources	The system uses the user's account and trade data to generate the history. It also requires database access to store trade records.
Processing	Upon request, the system fetches the user's trade data from the database and displays the details of completed trades.
Data	The system processes trade-related data, including item details, trade dates, and partner information, to generate accurate trade histories.

Table 77: Trade History Component Detail

3.2.24. Send Message

Identification	Send Message
Type	Module
Purpose	The purpose is to allow users to send messages to others, fostering communication and collaboration within the platform.

Function	The user selects a trade or another user to message, types a message, and clicks "Send." The system delivers the message in real-time to the recipient. If there is an error, the system displays an error message.
Subordinates	The module includes the messaging interface for composing and sending messages.
Dependencies	It depends on the user's account and active trades to initiate messages.
Interfaces	The system interacts with the user through the messaging interface, allowing users to compose and send messages.
Resources	The module requires server resources for real-time message delivery and storage.
Processing	The system processes the user's message and delivers it to the intended recipient.
Data	The system handles message data, including the content of the message and delivery status.

Table 78: Send Message Component Detail**3.2.25. Remove from Wishlist**

Identification	Remove from Wishlist
Type	Module
Purpose	It allows users to declutter their wishlist, keeping only relevant items. This ensures users can track their desired items effectively.
Function	Users can navigate to their wishlist and select an item to remove. The system updates the wishlist after removing the selected item.
Subordinates	The system works with the user's wishlist data and profile. It integrates with the user interface to display and modify wishlist items.
Dependencies	It depends on the user's account and the wishlist data being available. The system must be able to modify the wishlist entries in the database.
Interfaces	The feature interfaces with the user's profile and wishlist storage. It allows the removal of items from the user's saved list.
Resources	The system accesses the user's profile and wishlist data. It also requires database access to update the wishlist content.
Processing	Upon removal, the system updates the wishlist by removing the selected item and ensures the change is reflected across the platform.
Data	The system processes data related to the wishlist, including item names, descriptions, and user preferences to update the list accordingly.

Table 79: Remove from Wishlist Component Detail**4. User Interface Design****4.1. Section Overview**

This section outlines the design and structure of the user interface for the system. It covers the principles, standards, and conventions used to create an intuitive and user friendly experience. The section includes the key UI components, layout, navigation, and any external libraries or frameworks used for building the interface.

Additionally, the design of various screens and user interactions will be discussed to ensure the system is visually appealing and easy to use.

4.2. Interface Design Rules

- **Consistency:** The design maintains a consistent layout, color scheme, and behavior across different pages and screens to provide a unified experience for the user.
- **Usability:** The interface is designed to be intuitive, with clear and easily recognizable controls, buttons, and labels that guide users through their tasks.
- **Responsiveness:** The design adapts seamlessly to different screen sizes and resolutions, ensuring a smooth experience across various devices, from desktops to mobile phones.
- **User Experience:** The interface is optimized for ease of navigation, minimizing user inputs while maximizing functionality and efficiency. This ensures a user-friendly environment, enhancing the overall experience.

4.3. GUI Components

The system utilizes several key libraries and dependencies to build the user interface, ensuring a dynamic and responsive experience. These components include:

- **React:** A JavaScript library for building user interfaces, enabling the creation of dynamic, responsive, and interactive web pages through reusable components.
- **Axios:** A promise-based HTTP client used for making API requests, allowing the application to fetch and send data to the back-end server.
- **React-Router-Dom:** A routing library used for handling client-side navigation, enabling smooth transitions between pages or views without full page reloads.
- **React-Icons:** A library providing a collection of customizable icons that enhance the visual appeal and usability of the user interface.

These libraries and components work together to deliver a robust, efficient, and user-friendly interface for the application.

4.4. Detailed Description

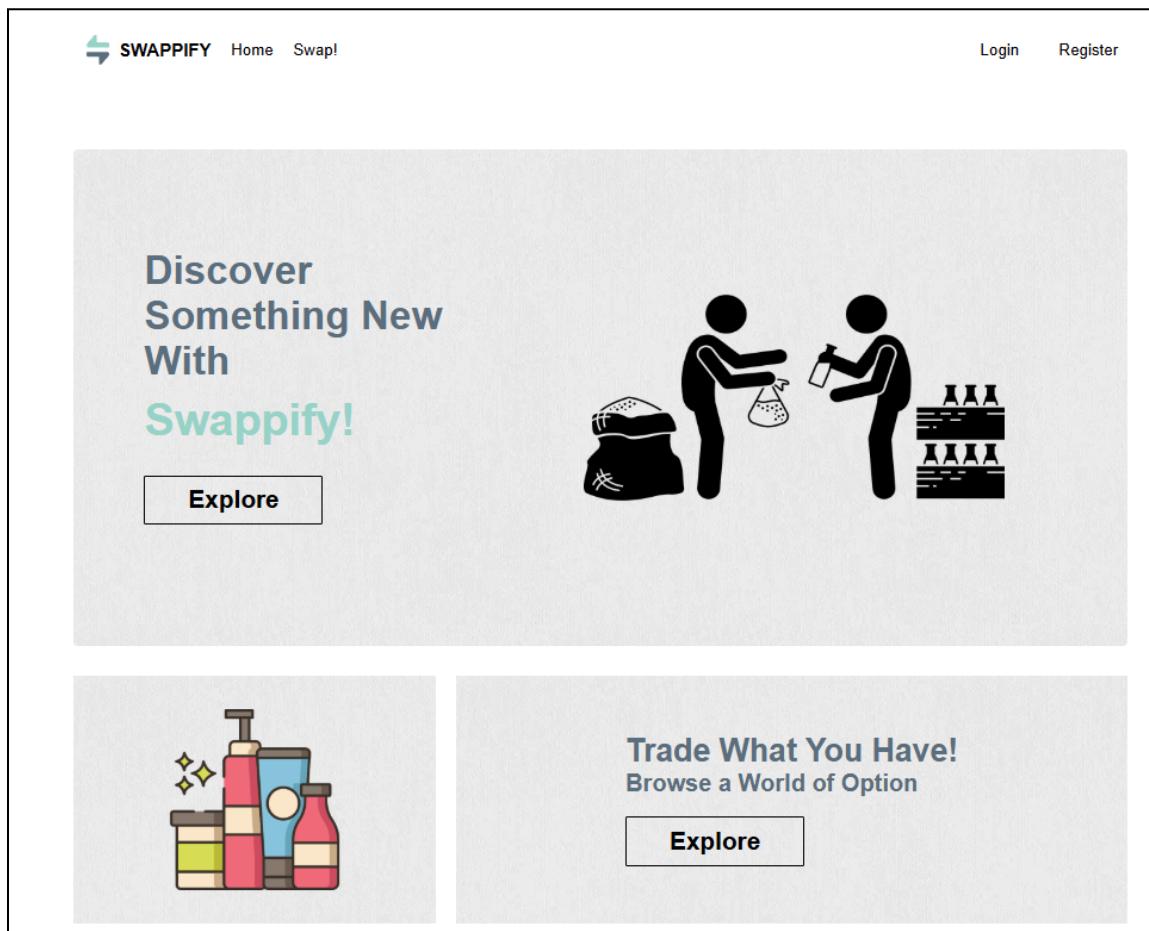


Figure 31. Home Page

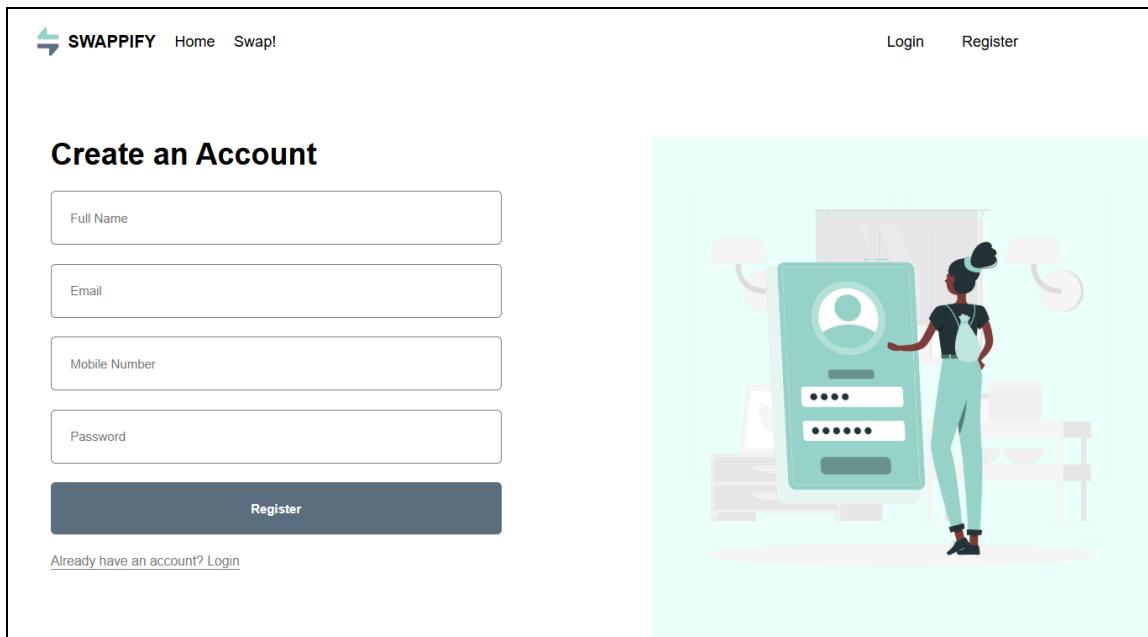


Figure 32. Sign Up Page

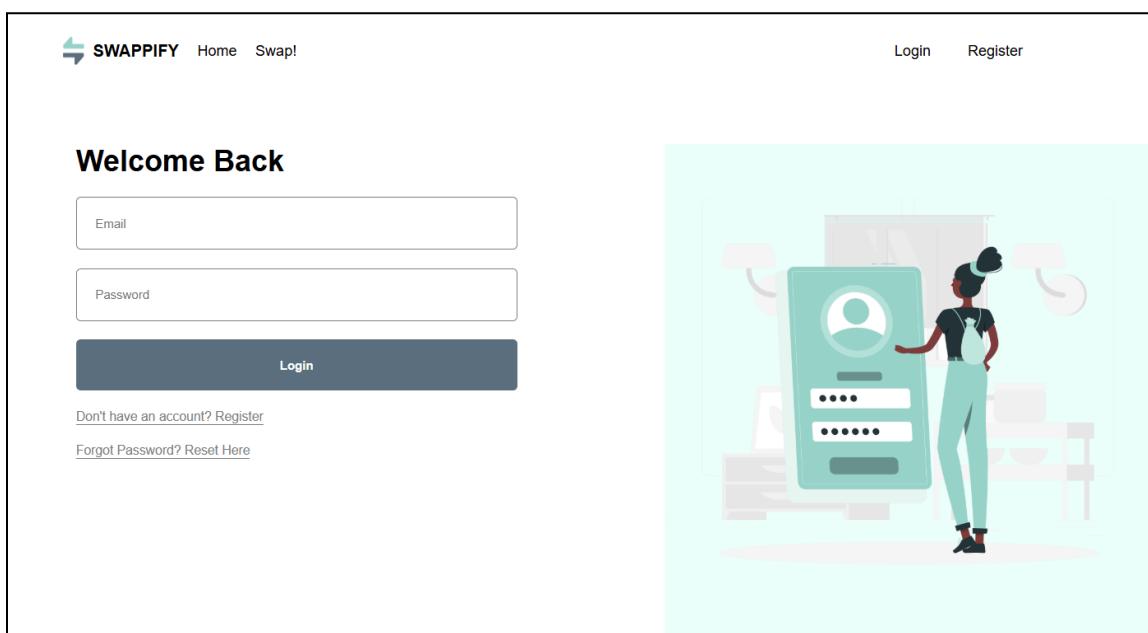


Figure 33. Login Page

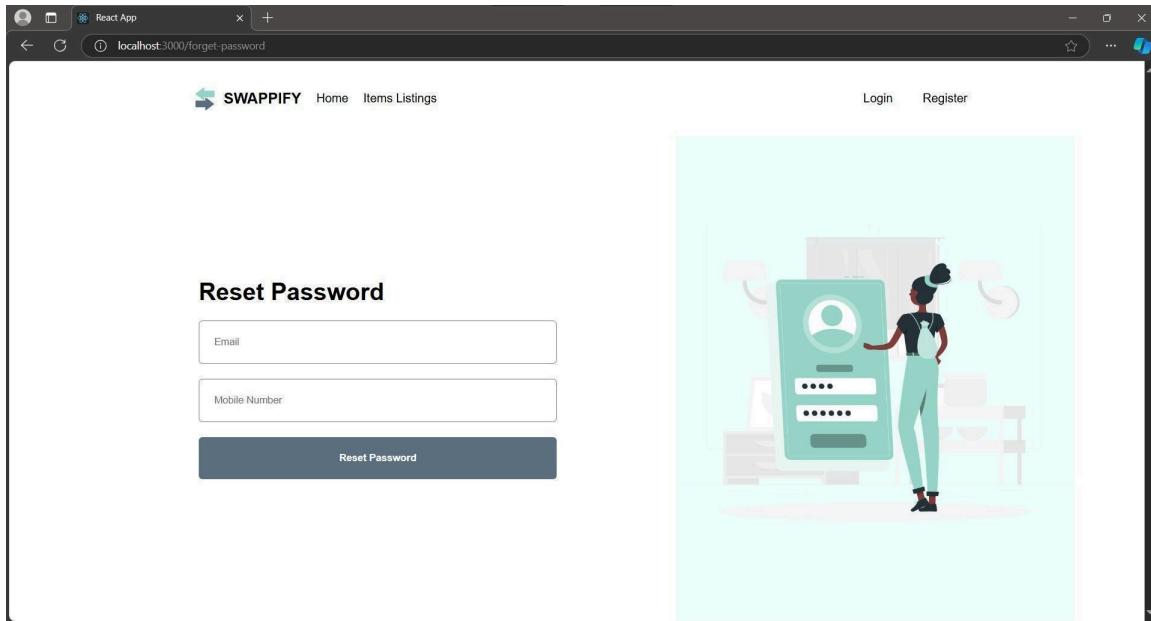


Figure 34. Forget Password Page

A screenshot of the SWAPPIFY User Profile page. The header includes the SWAPPIFY logo, 'Home', 'Swap!', 'Profile' (highlighted in blue), and 'Logout' links. On the left, there's a sidebar with navigation links: 'User Profile', 'MyList', 'WishList', and 'Trade Management'. The main content area is titled 'User Profile' and features a 'User Information' card. This card contains a circular profile picture placeholder, the text 'Full Name: Mustan Ali', 'E-Mail: MustanAli@example.com', 'Mobile: 03234567891', and a green 'Update Profile' button.

Figure 35. Profile Page

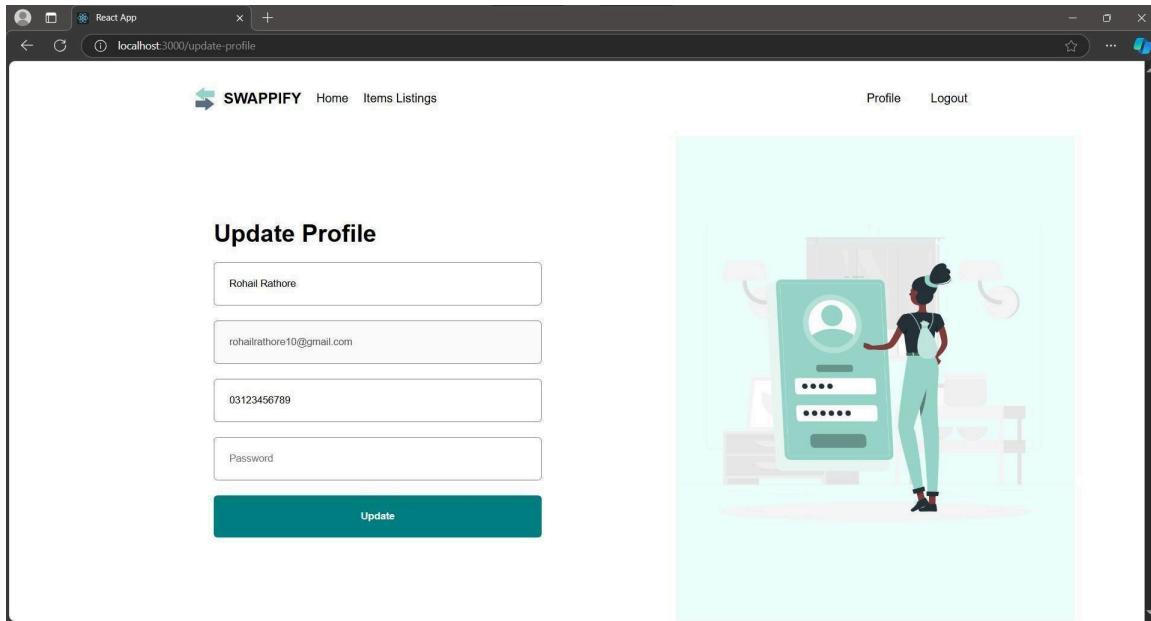


Figure 36. Update Profile Page

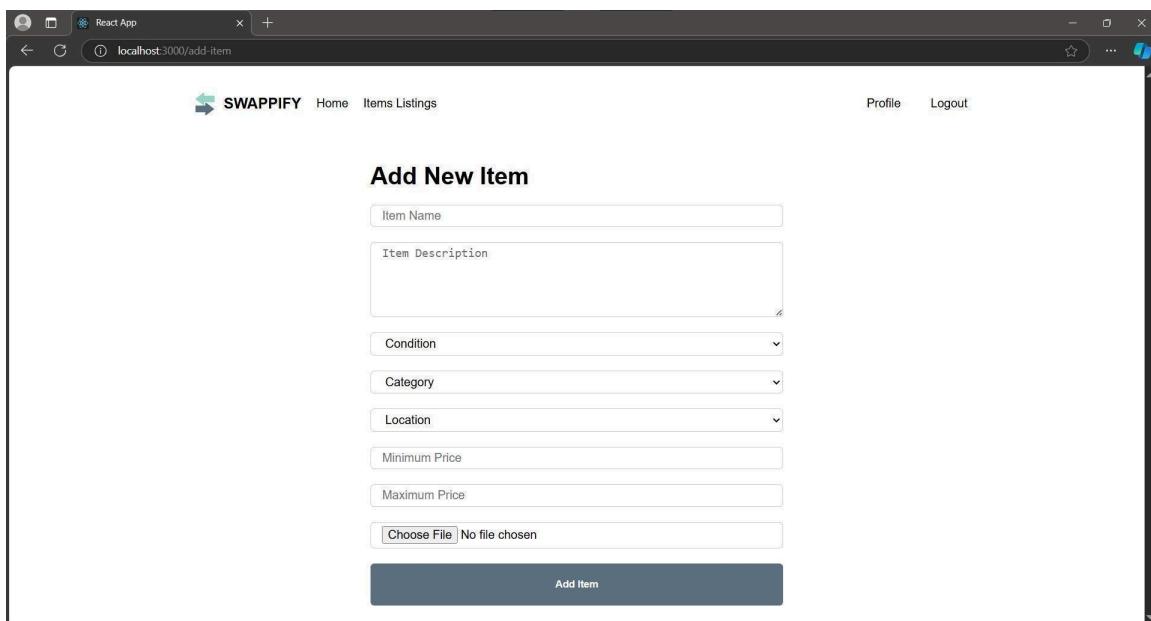


Figure 37. Add Item Page

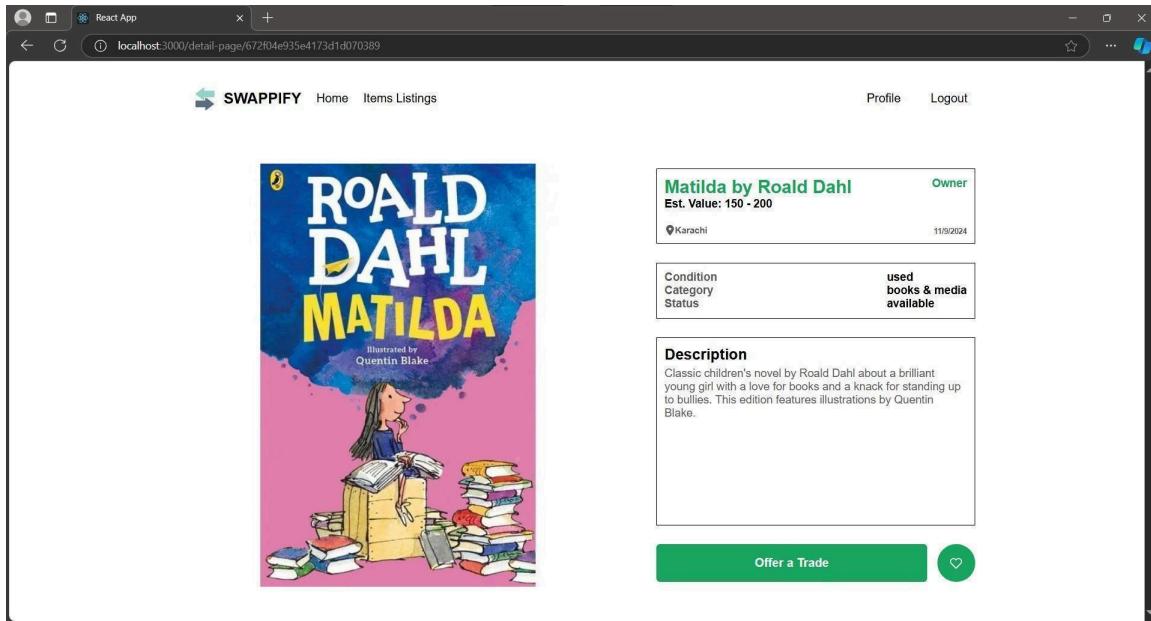


Figure 38. Item Detail Page

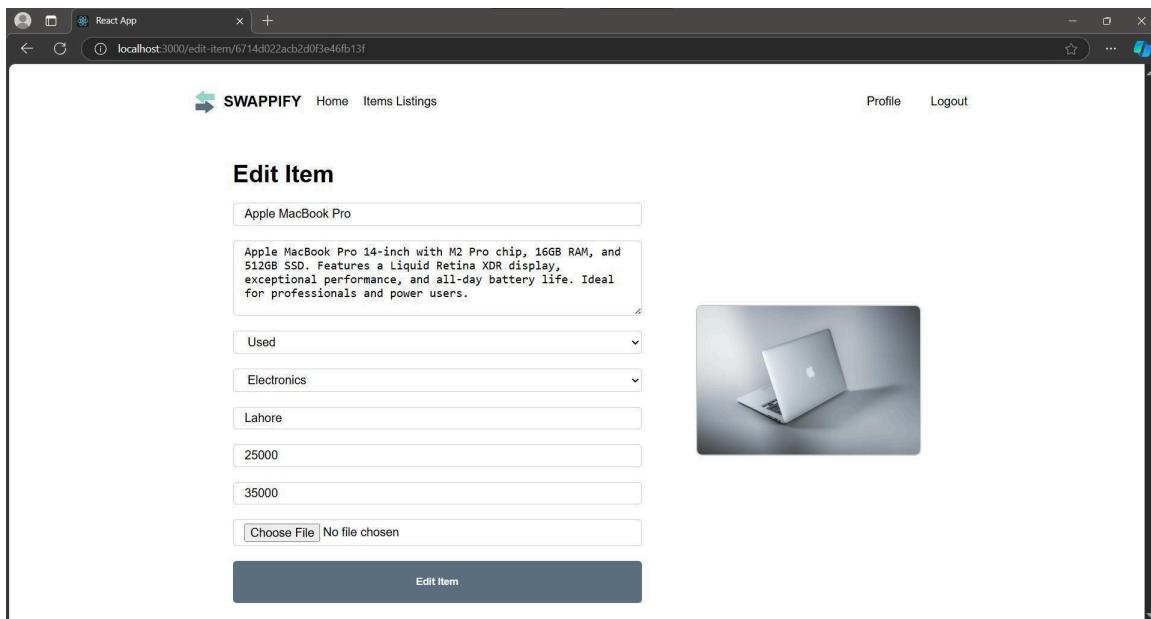


Figure 39. Edit Item Page

SWAPPIFY Home Swap!

User Profile | Profile Logout

My List

Create New Post

- User Profile
- MyList
- WishList
- Trade Management

Samsung Galaxy 1... available

Details

Wooden Chairs available

Details

Bed available

Details

Figure 40. User Item Page

SWAPPIFY Home Swap!

User Profile | Profile Logout

Wishlist

- User Profile
- MyList
- WishList
- Trade Management

thousand splen... available

Est. Value: 15 - 20
karachi books & media

[View details](#)

Activate W
Go to Settings

Figure 41. Wishlist Page

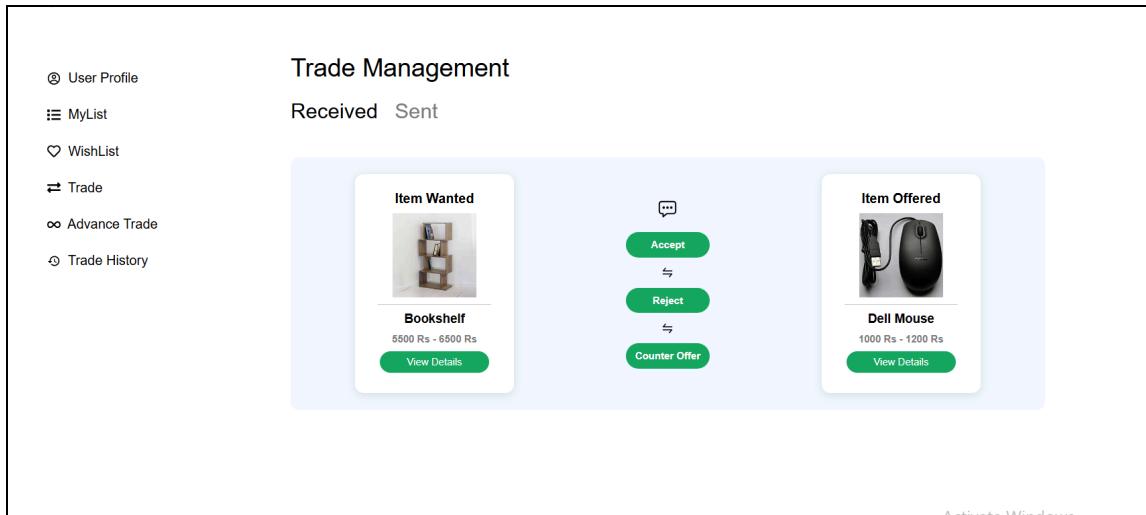


Figure 42. Trade Management: Received Request Page

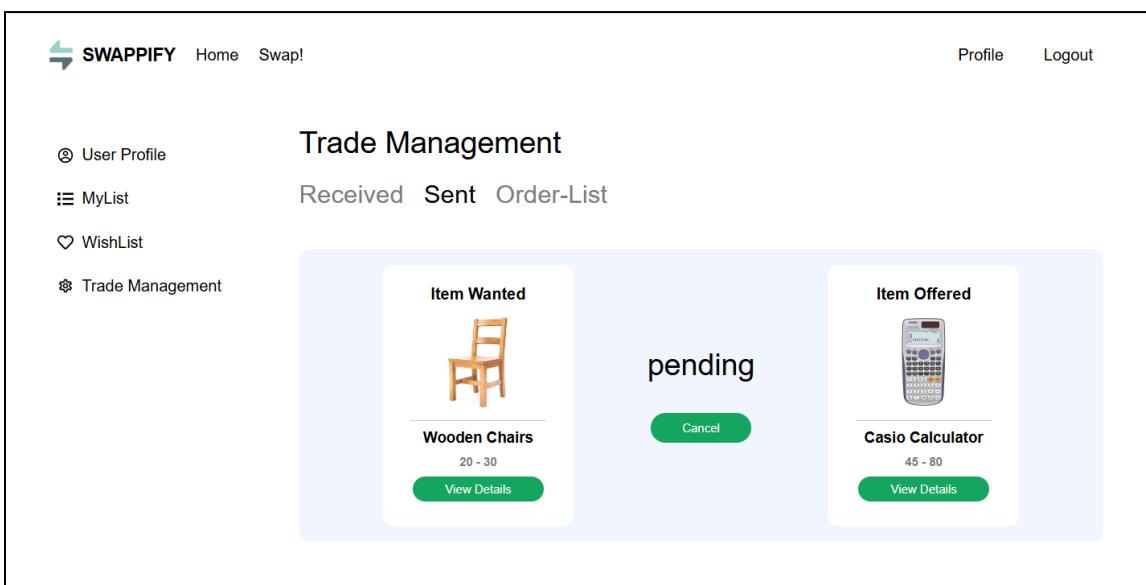


Figure 43. Trade Management: Sent Request Page

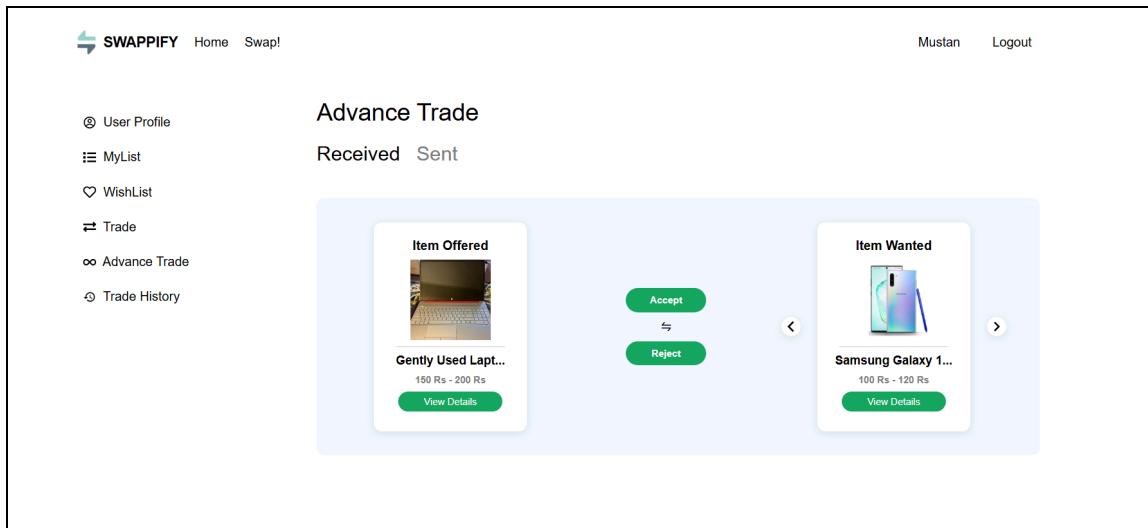


Figure 44. Advance Trade Management: Received Request Page

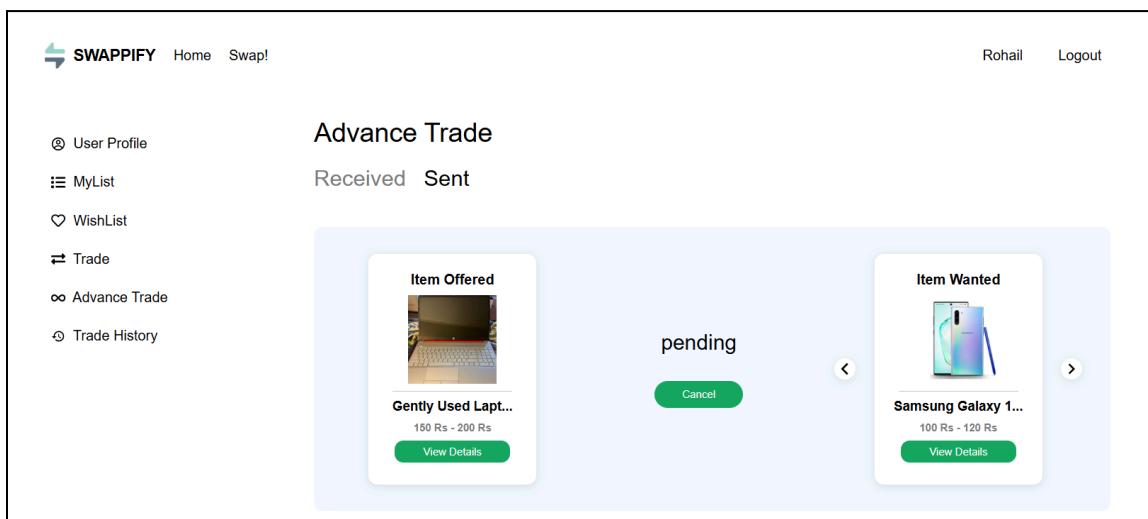


Figure 45. Advance Trade Management: Sent Request Page

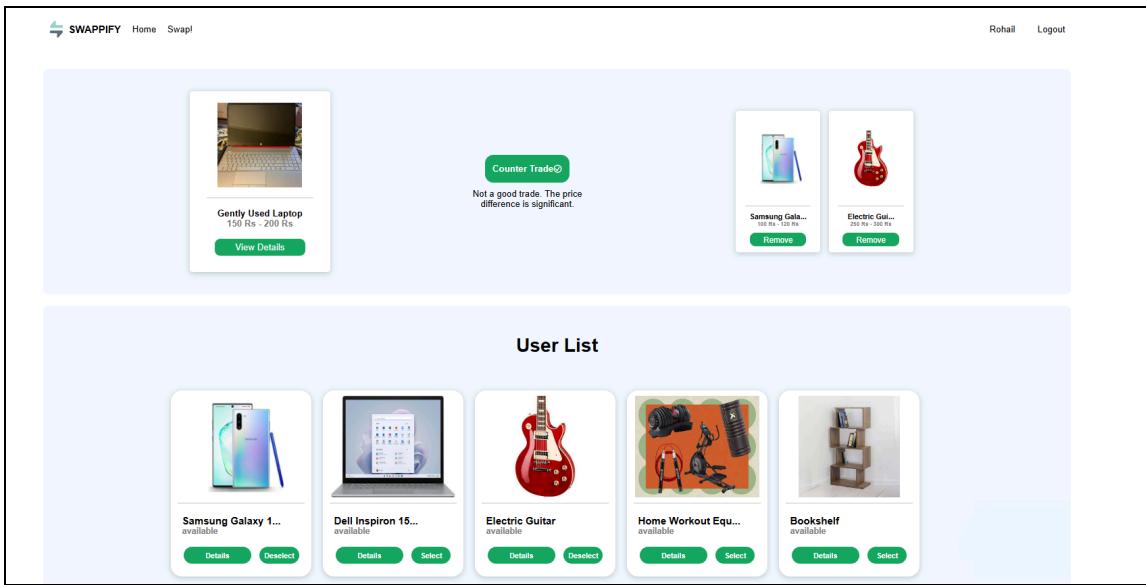


Figure 46. Advance Trade: Sent Offer

Figure 47. Trade History



Figure 48. Trade Detail

SWAPPIFY Home Swap!

Login Register

Show Results by:

Filter by Condition

All

Filter by City

- Karachi
- Lahore
- Islamabad
- Faisalabad
- Rawalpindi
- Multan
- Peshawar
- Quetta
- Gujranwala
- Sialkot
- Hyderabad
- Bahawalpur
- Sargodha
- Mardan
- Swat

Filter by Category

- Electronics
- Furniture
- Clothing & accessories
- Books & media
- Home & garden
- Sports & outdoors
- Toys & games
- Tools & hardware
- Automotive
- Office supplies
- Collectibles & antiques
- Other

Filter by Price Range

From _____ To _____

Search...

Samsung Galaxy 1...
The Samsung Galaxy S10 is a premium smartphone released...
Est. Value: 100 - 120
📍 karachi | 11/18/2024

[View Details](#)

Wooden Chairs
A wooden chair is a sturdy and timeless piece of furnit...
Est. Value: 20 - 30
📍 sialkot | 11/18/2024

[View Details](#)

Bed
Experience exceptional sound quality with the Smart Blu...
Est. Value: 200 - 250
📍 lahore | 11/18/2024

[View Details](#)

Dell Inspiron 15...
The Dell Inspiron 15 Laptop offers a perfect blend of...
Est. Value: 200 - 250
📍 hyderabad | 11/18/2024

[View Details](#)

thousand splendi...
A Thousand Splendid Suns is a powerful and emotional no...
Est. Value: 15 - 20
📍 karachi | 11/18/2024

[View Details](#)

Casio Calculator
The Casio Scientific Calculator is a versatile and reli...

Est. Value: 45 - 80
📍 lahore | 11/18/2024

[View Details](#)

Figure 49. Listed Items Page

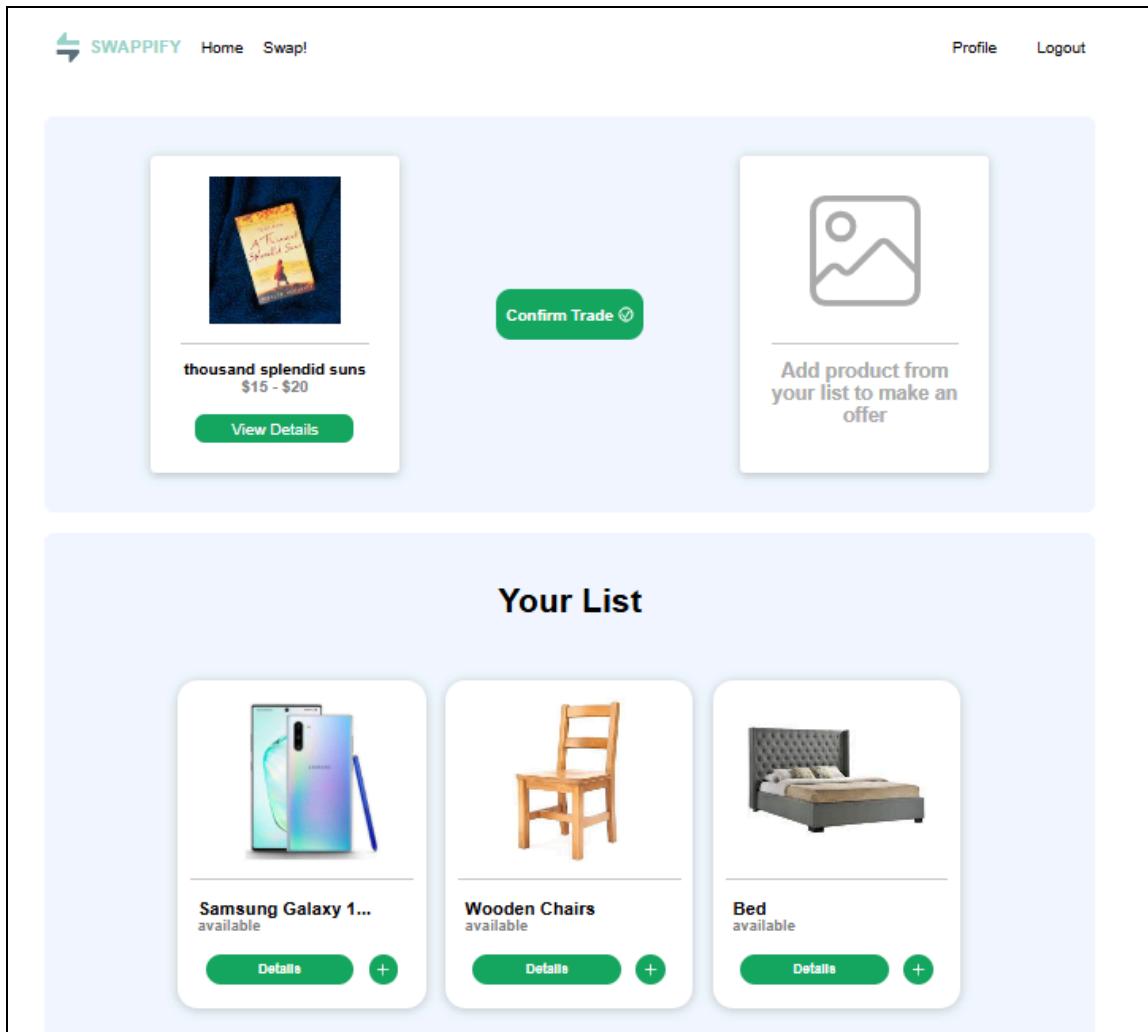


Figure 50. Trade Proposal Page

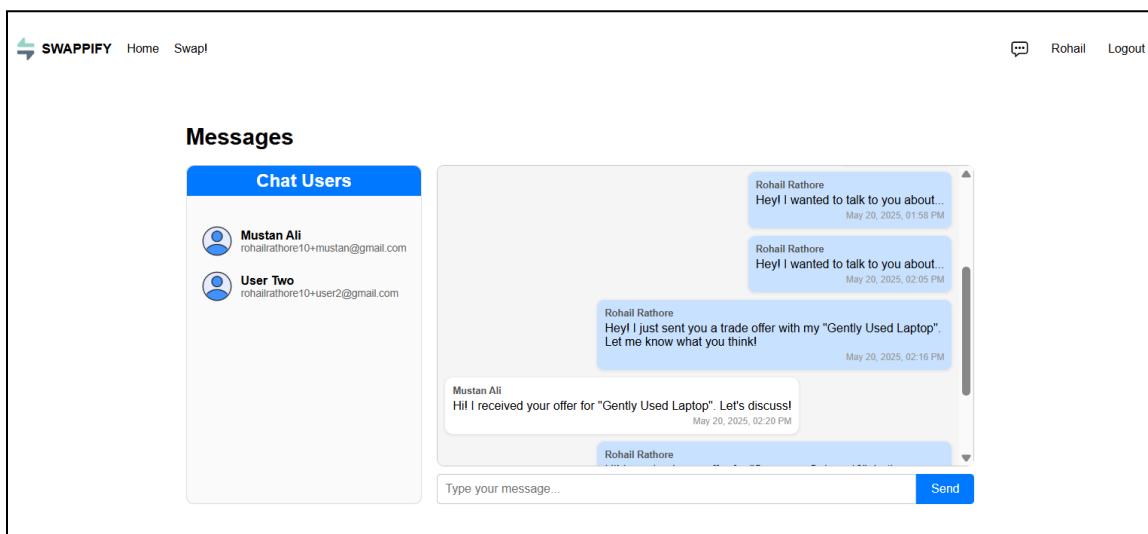


Figure 51. Message System

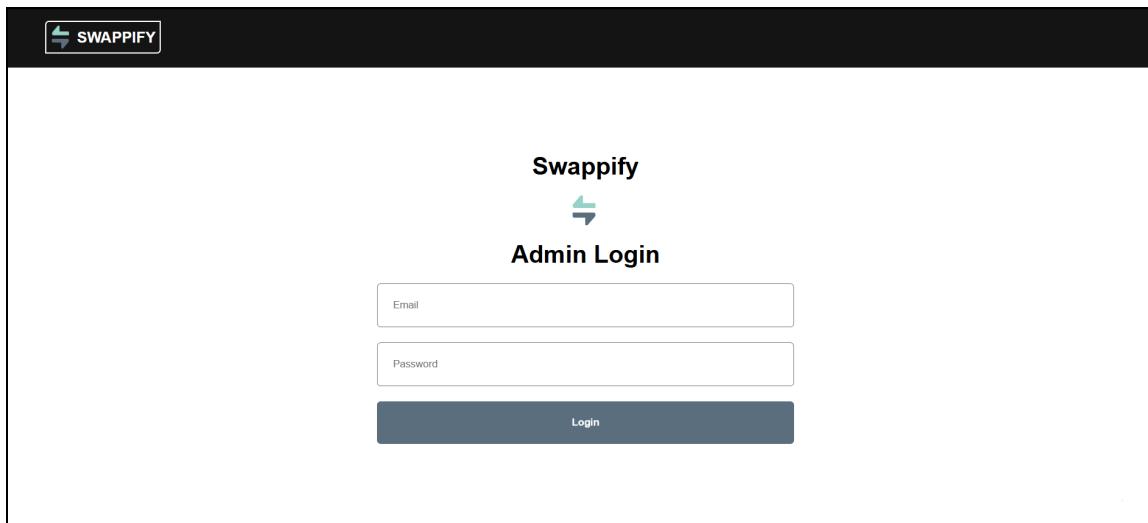


Figure 52. Admin Login

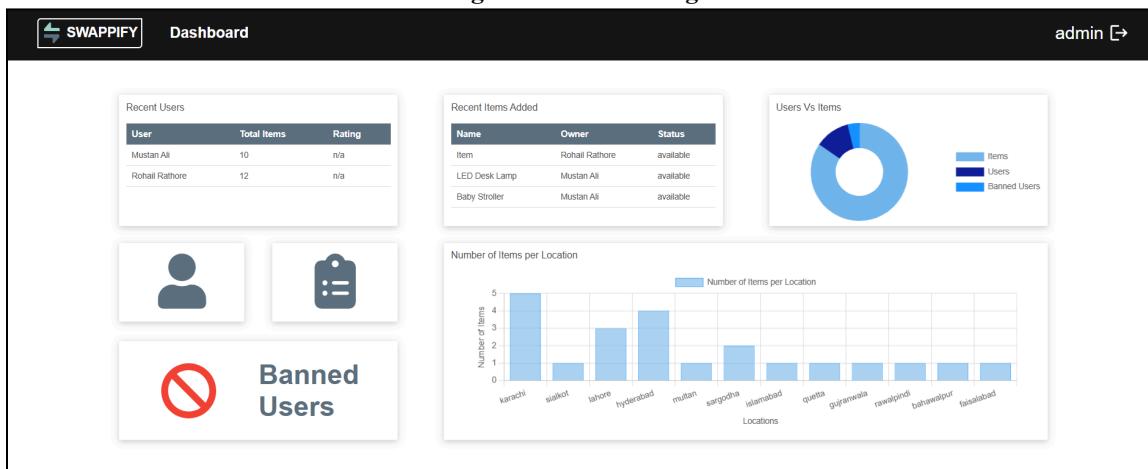


Figure 53. Admin Dashboard

Name	Phone	Email	Total Items	Action
Rohail Rathore	03123456789	rohailrathore10@gmail.com	12	∅
Mustan Ali	03234567891	rohailrathore10+mustan@gmail.com	10	∅

Figure 54. Admin Dashboard: Users List

Name	Owner	Condition	Category	Location	Price (PKR)	Status	Date Added	Action
Samsung Galaxy 10		used	electronics	karachi	100 - 120	available	11/18/2024	
Wooden Chairs		new	furniture	sialkot	20 - 30	traded	11/18/2024	
Bed		new	furniture	lahore	200 - 250	traded	11/18/2024	
Dell Inspiron 15 Laptop		used	electronics	hyderabad	200 - 250	available	11/18/2024	
thousand splendid suns		used	books & media	karachi	15 - 20	available	11/18/2024	
Casio Calculator		new	office supplies	lahore	45 - 80	traded	11/18/2024	
Dell Mouse		new	electronics	multan	15 - 25	available	11/18/2024	
Iphone 13		new	electronics	karachi	120 - 230	traded	12/27/2024	
Art		new	collectibles & antiques	karachi	50 - 75	traded	1/27/2025	

Figure 55. Admin Dashboard: Items List

Name	Phone	Email	Total Items	Action
user1	03123456788	rohailrathore10@hotmail.com	0	

Figure 56. Admin Dashboard: Banned Users List

5. Reuse & Relationships to Other Products

This project leverages established frameworks and design patterns to ensure efficiency and consistency, using tools like ReactJS and Axios for their reliability and scalability, which allowed the focus to remain on core functionalities. Open-source libraries such as React-Router-Dom and Mongoose were integrated to reduce development time and maintain a reliable and efficient system. Although some external modules were considered, they were discarded due to compatibility issues, performance concerns, or their inability to meet the project's specific requirements, leading to the decision to develop custom solutions.

6. Design Decisions & Tradeoffs

In designing the system, we chose ReactJS for its component based architecture and fast rendering, despite considering alternatives like Vue.js and Angular. For the backend, Node.js with Express.js was selected over frameworks like Django for its lightweight nature and seamless integration with JavaScript. MongoDB was chosen over PostgreSQL for its flexibility in handling unstructured data, which suited our needs better. We decided to use WebSockets for real time communication, ensuring instant updates, which were crucial for enhancing the user experience. These decisions prioritize simplicity, scalability, and efficiency while meeting the project's immediate needs.

7. Pseudocode for Components

Login:

1. Enter email and password
2. If email and password are correct then
3. Authenticate based on the role which is Trader/Admin.
4. Redirect to profile

SignUp:

1. Enter email, password, full name & mobile number.
2. If validation of details is successful then
3. Send verification email.

Forgot Password:

1. Enter registered email & mobile number.
2. If email & mobile exists & belong to same user then:
3. Send a new password to the email.

Profile Management:

1. Navigate to the profile page.
2. View current profile information.
3. Select "Edit" to update profile fields.
4. If updated fields are valid then:
5. Save the updated information in the database.
6. Display the updated profile information.
7. If inputs are invalid, display appropriate error messages.

List Item:

1. Navigate to the "Add New Item" section.
2. Fill out the item details in the provided form.
3. If all fields are filled & data formats are valid then:
4. Save the item in the database.
5. If inputs are invalid, display appropriate error messages.

Edit Item:

1. Navigate to my list & select an item to edit.
2. View the current details of the selected item in an editable form.
3. Update the desired fields and click the "Save" button.
4. If all fields are filled & data is valid then:
5. Save the updated item details in the database.
6. Display "Item successfully updated."
7. If inputs are invalid, display appropriate error messages.

Delete Item:

1. Navigate to my list select an item to delete.
2. If the item is selected, then:
3. Ask for confirmation, if confirmation is given then
4. Remove the item from the listings.
5. Update the database.
6. If a system error occurs, display an error message.

View Item Listings:

1. Navigate to the item listings page.
2. If items are available then:
3. Retrieve & display a list of all available items with essential details.
4. Allow the user to select an item to view more details
5. If no items are available, display "No items available."
6. If a system error occurs, display an error message.

Search & Filters:

1. Enter a keyword in the search bar or select filter options
2. If items match the search term or filters then:
 3. Display a list of filtered items.
 4. Allow the user to select an item to view more details
 5. If no results are found, display "No items found."
 6. If a system error occurs, display an error message.

Add to Wishlist:

1. Navigate to an item & click "Add to Wishlist."
2. Add the item to the user's wishlist.
3. Display "Item added to wishlist."
4. If a system error occurs, display an error message.

View Wishlist:

1. Navigate to the wishlist section through the profile.
2. If the wishlist is not empty, then:
 3. Retrieve & display the list of saved items with details such as item name, image & description.
 4. If the user clicks on an item, display the full details & trade options for that item.
 5. If the wishlist is empty, display: "Your wishlist is empty"
 6. If a system error occurs display an error message

Item Detail:

1. Click on an item from the item listing page or wishlist.
2. Retrieve & display item details such as name, description, condition, category, images, and owner information..
3. Display available actions.
4. If the user selects an action, perform the action.
5. If a system error occurs display an error message

Trade Proposal:

1. Navigate to an item listing you want to trade for and click "Propose Trade."
2. Select your own item to offer in exchange.
3. Submit the trade proposal by clicking the "Submit" button.
4. If the proposal is submitted:
 5. Save the proposal in the database.
 6. Notify the other user about the trade proposal.
 7. Display confirmation message: "Trade proposal sent."
8. If the user cancels the proposal before submission:
 9. Abort the trade proposal, and no data is saved.
10. If a system error occurs, display an error message..

Accept Trade:

1. Navigate to trade proposals and select a pending trade to review.
2. The system displays the trade details, including the item offered.
3. Click the "Accept" button to confirm the trade.
4. The system finalizes the trade by updating the trade status in the database & notifies the other user of the acceptance.
5. If a system error occurs, display an error message.

Reject Trade:

1. Navigate to trade proposals and select a pending trade to review.
2. The system displays the trade details, including the item offered.
3. Click the "Reject" button to decline the trade.
4. The system updates the trade status in the database.
5. If a system error occurs, display an error message.

Cancel Trade:

6. Navigate to the "Sent Requests" section.
7. The system fetches the details of the selected trade & displays a "Cancel Trade" button.
8. Click the "Cancel Trade" button.
9. The system updates the trade status to "Cancelled" in the database.
10. If a system error occurs, display an error message.

Advanced Trade Management:

1. The user goes to the "Active Trades" section & clicks the "Counter Offer" button for a specific trade.
2. The system opens a trade proposal page with the recipient's item pre-selected.
3. Select an Item for Counter Offer:
4. The user selects an item from the sender's listed items to include in the counter proposal.
5. The system updates the trade proposal with the selected item from the sender.
6. The user clicks the "Send Counter Offer" button.
7. If no item is selected from the sender's list, the system displays an error message.
8. If an item is selected:
9. The system saves the counter proposal in the database.
10. The system notifies the sender about the counter proposal.
11. A confirmation message is displayed: "Counter offer sent successfully."
12. If a system error occurs, display an error message.

Manage Active Trades:

1. The user navigates to the "Active Trades" section.
2. The system displays a list of all active trade proposals, including trade details & status.
3. The user selects a trade to manage and either accepts or rejects it.
4. The system updates the trade status based on the user's action and notifies the other party.
5. If a system error occurs, display an error message.

Ban User Accounts:

1. The admin navigates to the "User Accounts" section.
2. The system displays a list of all user accounts with details such as account status & profile information.
3. The admin selects a user account to ban.
4. The system processes the admin's action, updates the user's account status in the database.
5. If a system error occurs, display an error message.

Delete List Items:

1. The admin navigates to the "Listed Items" section.
2. The system displays a list of all items listed by users, including item details such as title, description, category, and status.
3. The admin selects an item to view or remove.
4. The system processes the admin's action.
5. If the item is removed, it is deleted from the listings and the database.
6. If a system error occurs, display an error message.

Rating and Review:

1. The user navigates to the "Rate Trade" section after completing a trade.
2. The system displays a form allowing the user to provide a rating & write a review.
3. The user submits the rating and review by clicking the "Submit" button.
4. The system validates the input and saves the rating and review to the database.
5. If a system error occurs, display an error message.

Email Notification:

1. A relevant event occurs (e.g., new trade proposal, trade accepted, trade rejection).
2. The system generates an email with the details of the event.
3. The system sends the email to the user's registered email address.

Trade History:

1. The user navigates to the "Trade History" section.
2. The system retrieves & displays a list of all the user's completed trades, including details such as the traded items, trade partners, and trade dates.
3. The user selects a specific trade to view more details.
4. The system displays the full details of the selected trade, including items traded, counterparties, and the status of the trade.

Send Messages:

1. User selects a user to message
2. Display a messaging interface with a text input field.
3. The user types a message and clicks "Send".
4. Message is saved in the database.
5. The recipient receives the message in real-time.

Remove from Wishlist:

1. The user navigates to their wishlist.
2. The system displays a list of all items currently in the user's wishlist.
3. The user selects an item to remove and clicks the "Remove" button.
4. The system removes the item from the wishlist and updates the database.

8. Appendices

8.1. Class Diagram

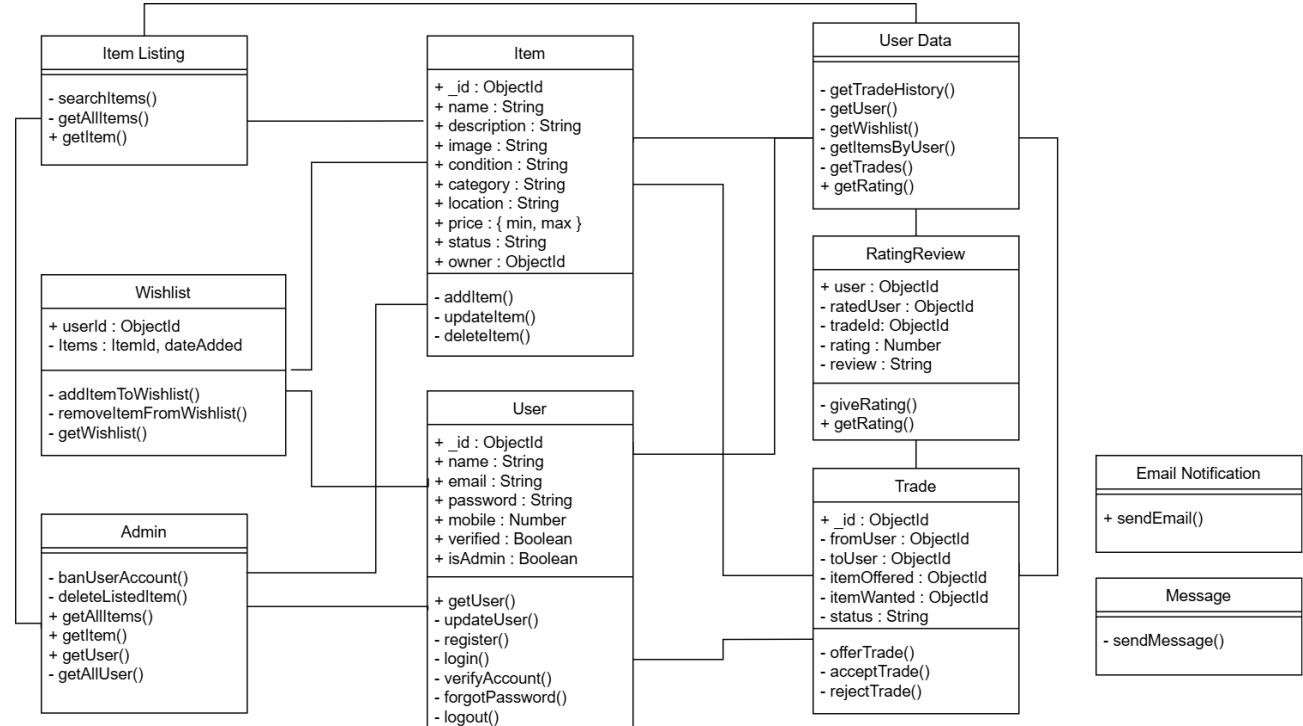


Figure 57: Class Diagram

8.2. State Chart Diagram

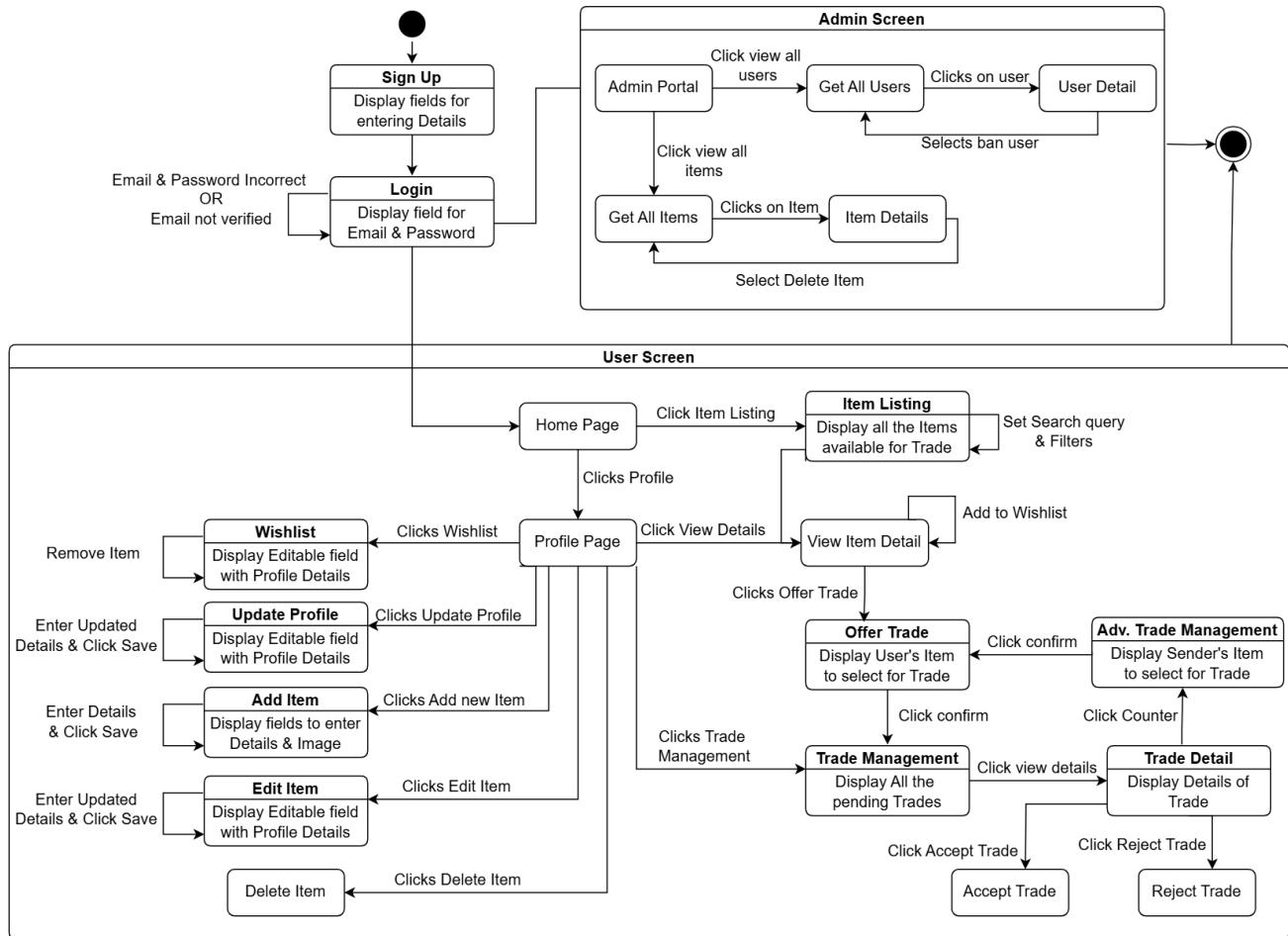


Figure 58: State Chart Diagram

8.3. Component Diagram

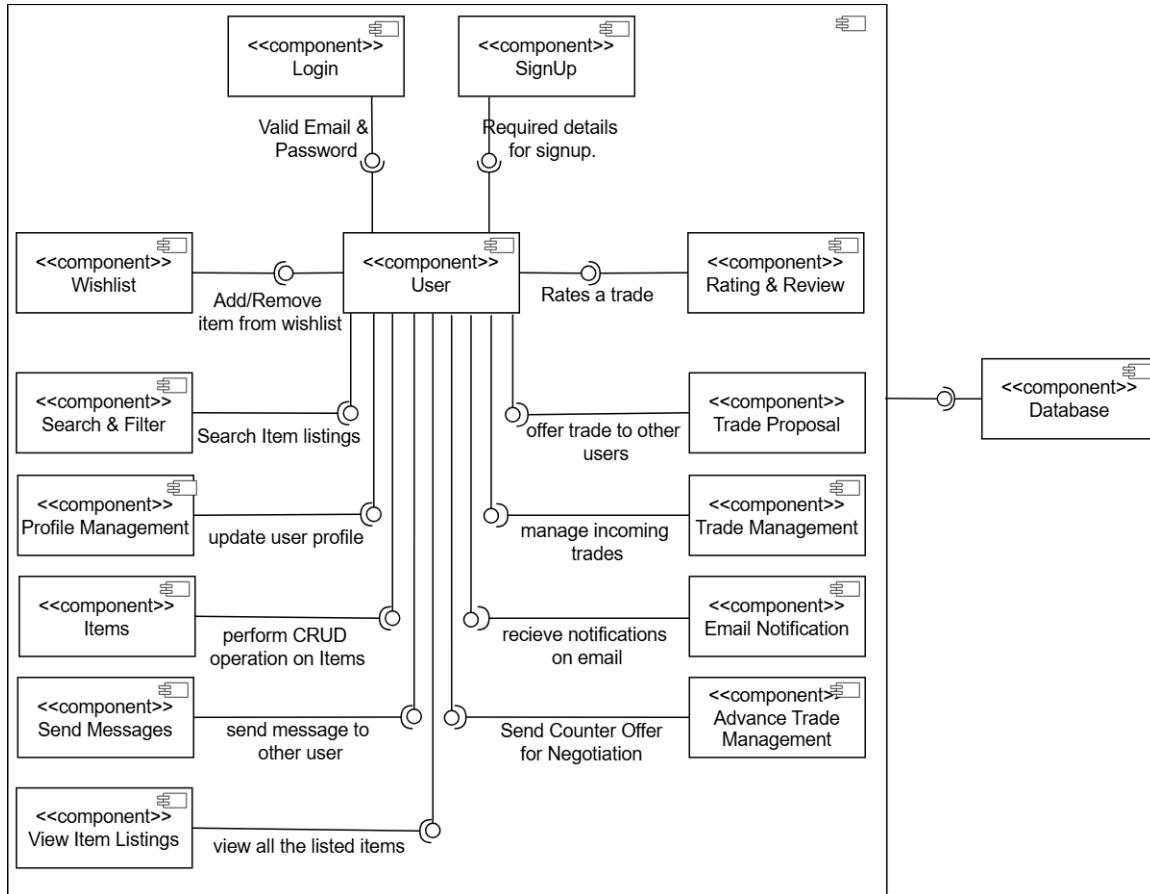


Figure 59: Component Diagram (User)

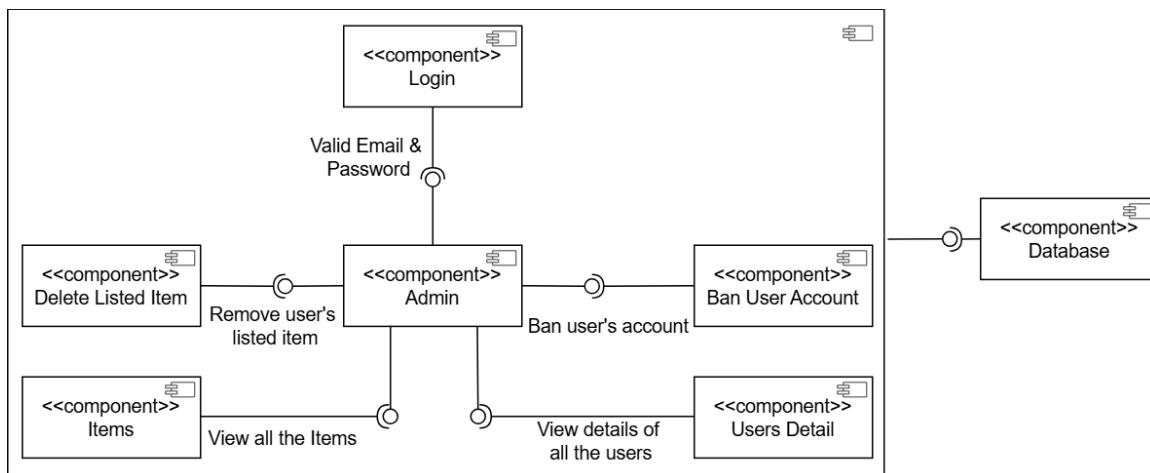


Figure 60: Component Diagram (Admin)

8.4. Use Case Diagram

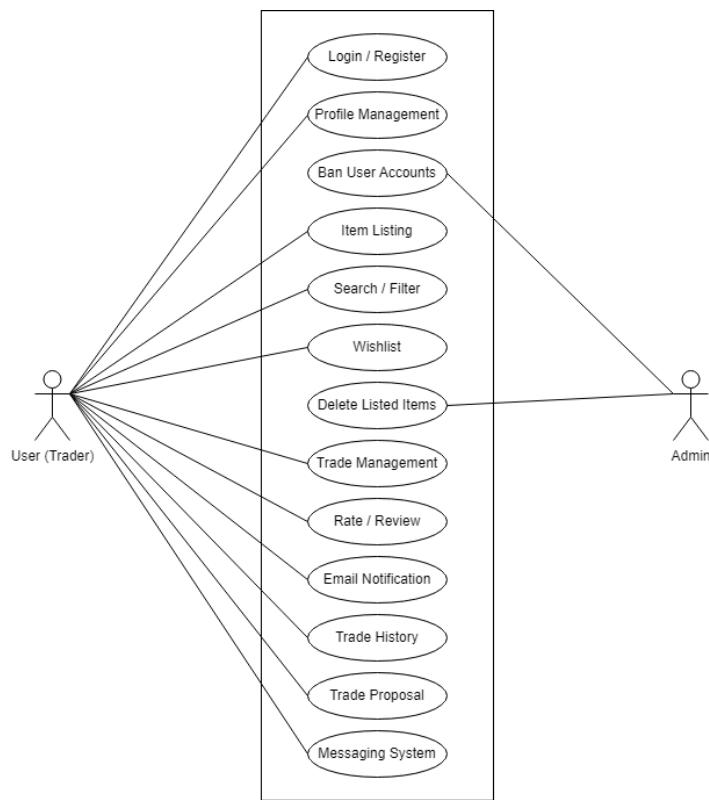


Figure 61: Use Case Diagram

8.5. Entity Relationship Diagram

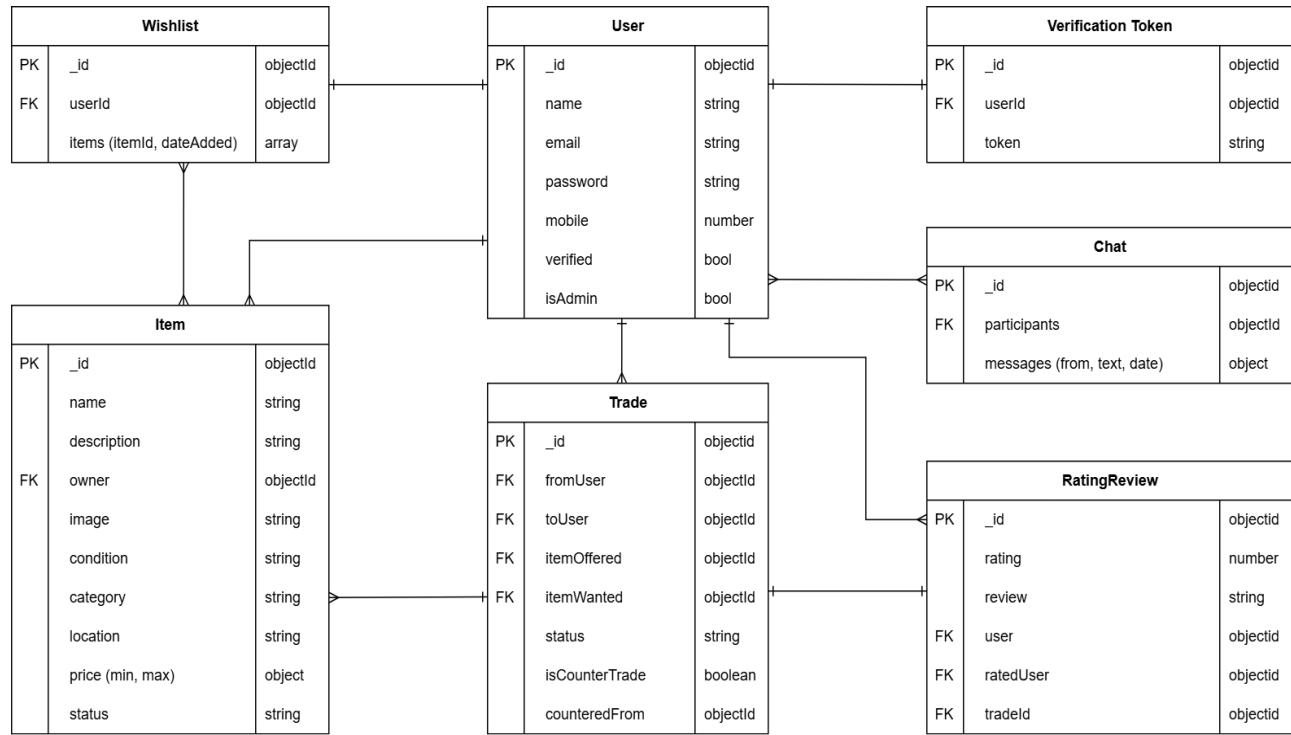


Figure 62: Entity Relationship Diagram

8.6. Activity Diagram

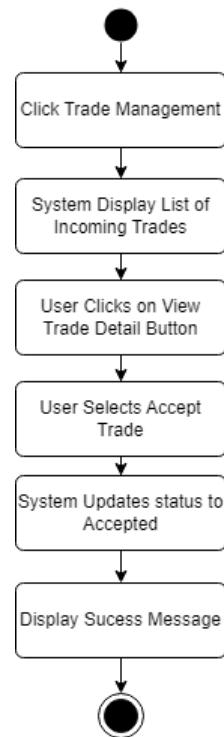


Figure 63: Activity Diagram (Accept Trade)

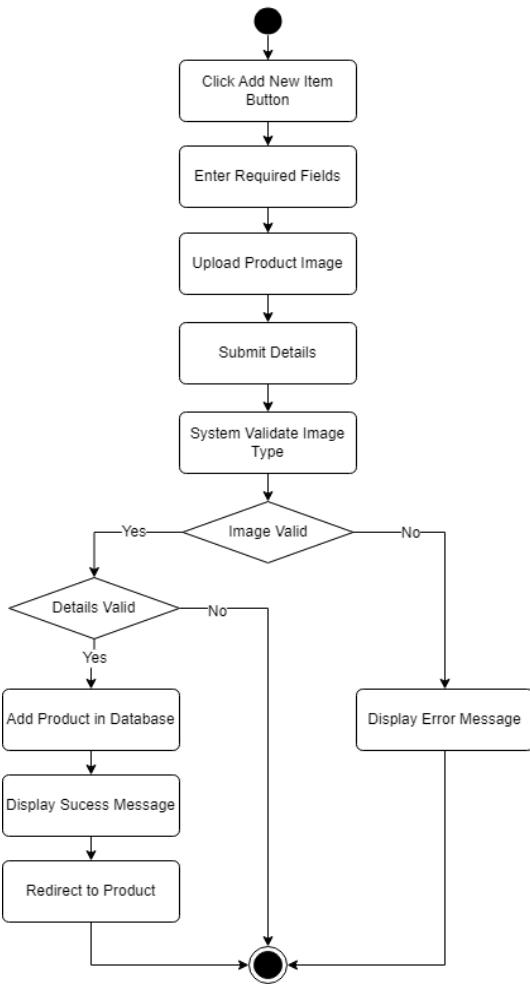


Figure 64: Activity Diagram (Add Item)

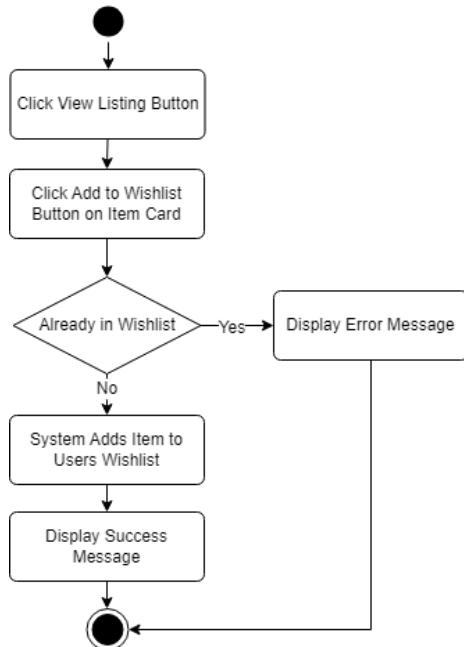
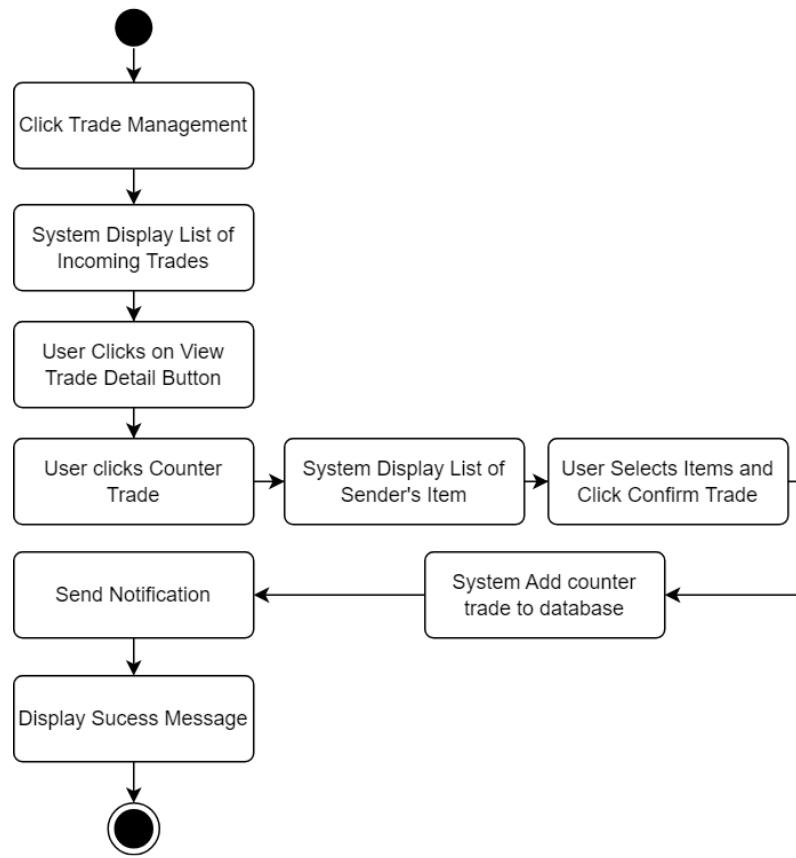
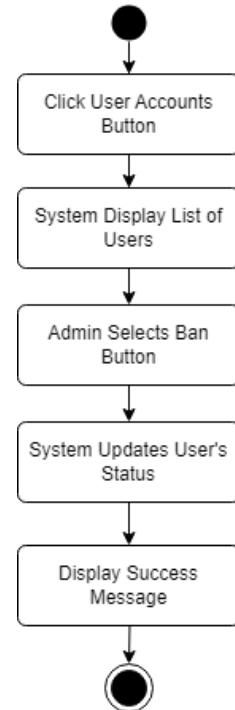


Figure 65: Activity Diagram (Add to Wishlist)

**Figure 66: Activity Diagram (Advance Trade Management)****Figure 67: Activity Diagram (Ban User Account)**

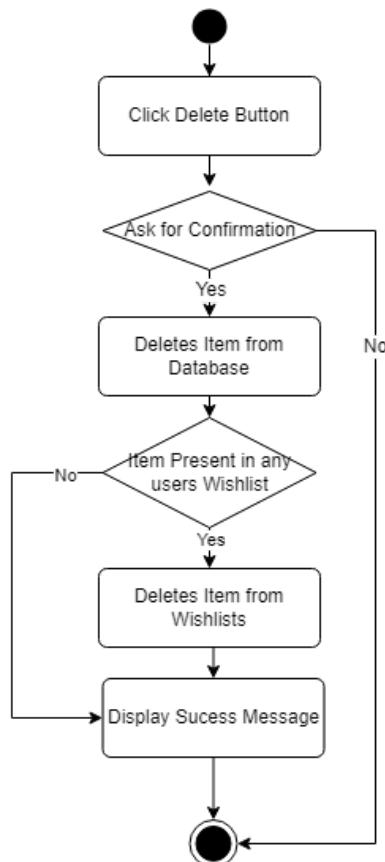


Figure 68: Activity Diagram (Delete Item)

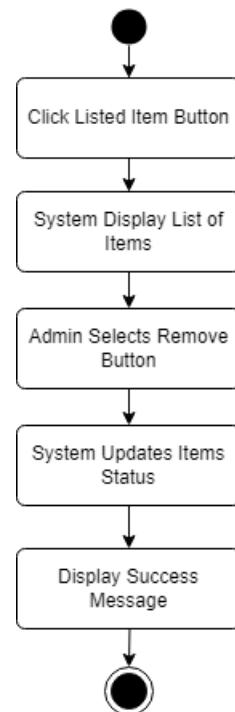


Figure 69: Activity Diagram (Delete List Item)

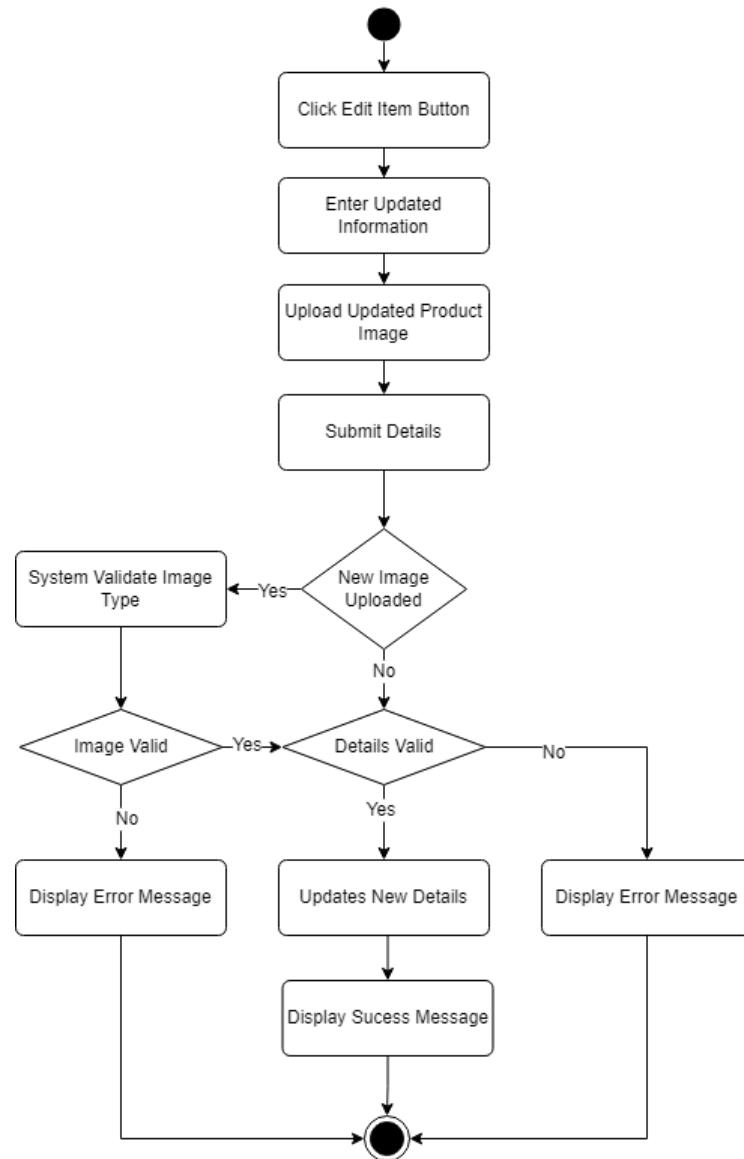


Figure 70: Activity Diagram (Edit Item)

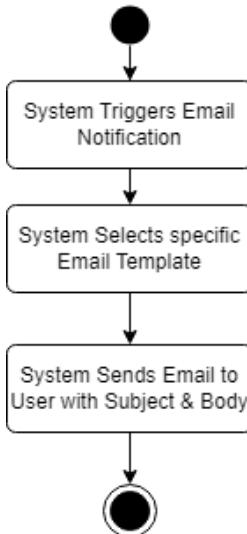


Figure 71: Activity Diagram (Email Notification)

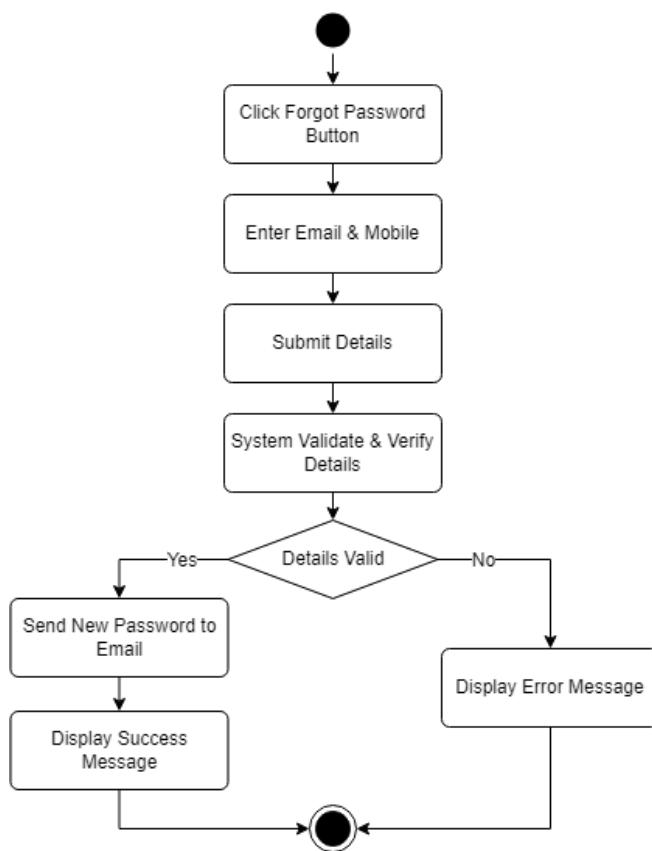


Figure 72: Activity Diagram (Forget Password)

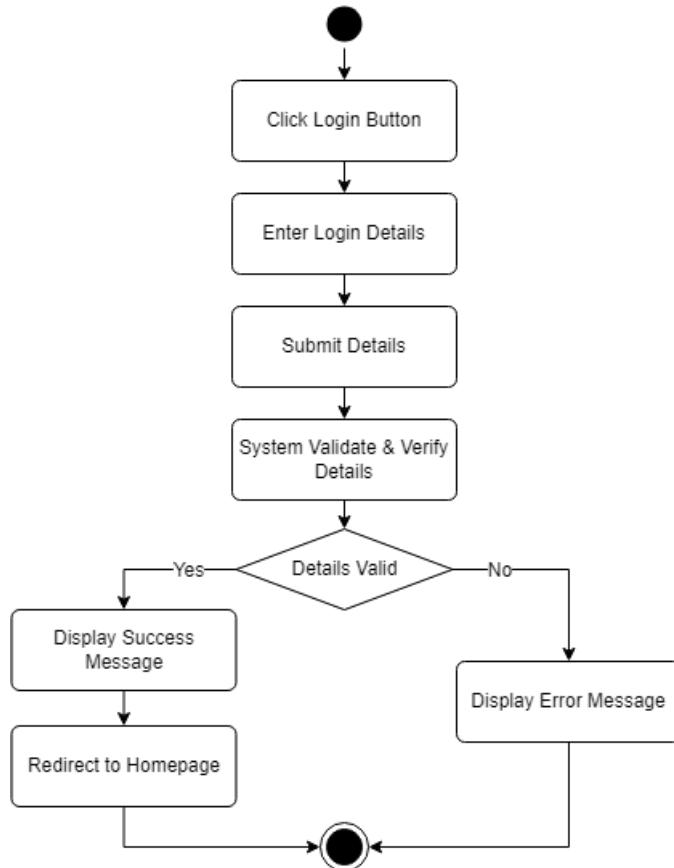


Figure 73: Activity Diagram (Login)

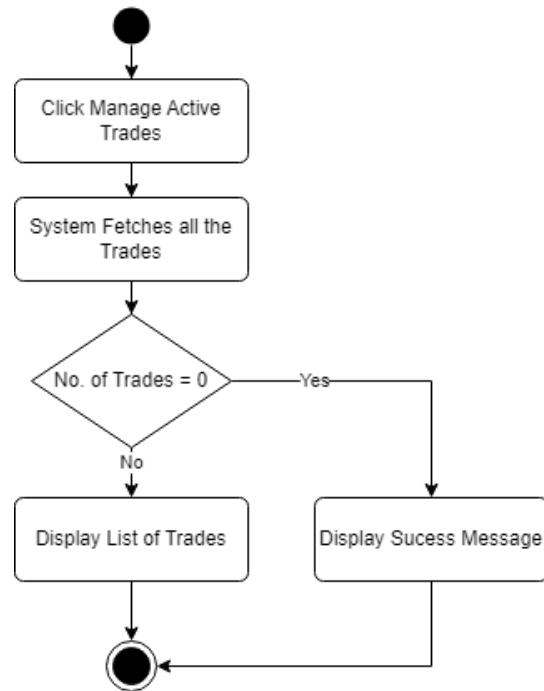


Figure 74: Activity Diagram (Manage Active Trade)

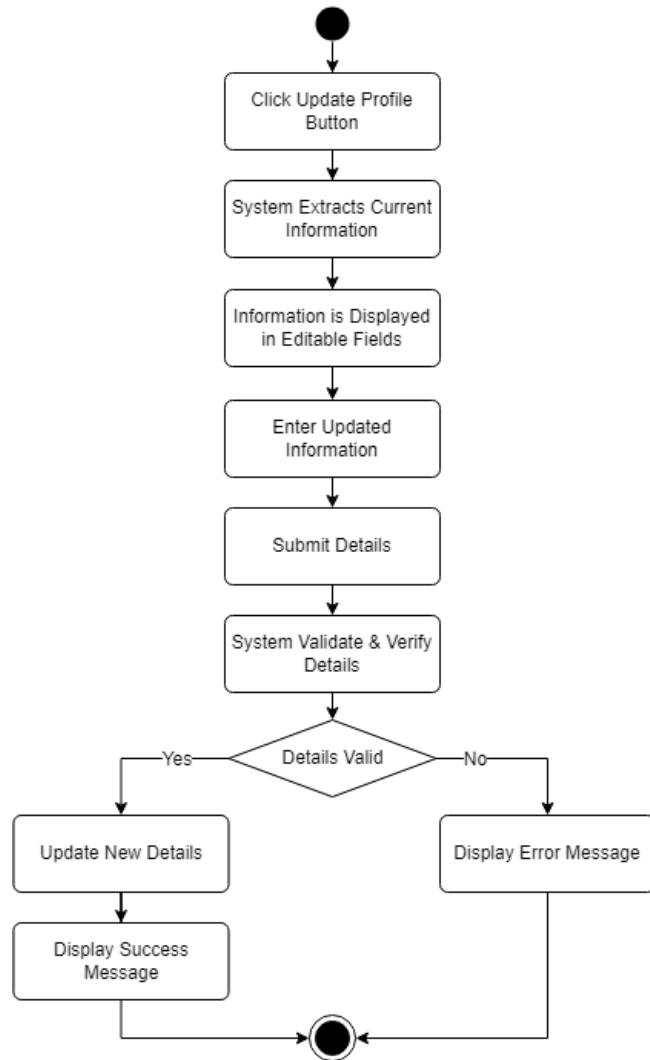


Figure 75: Activity Diagram (Profile Management)

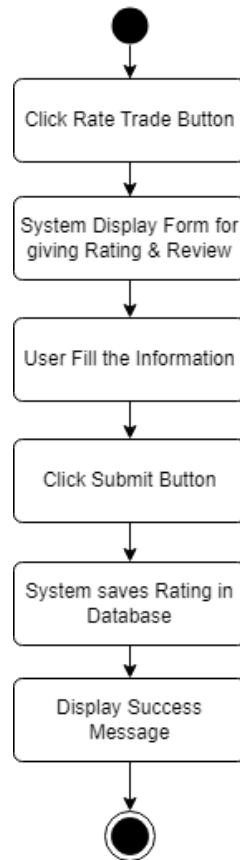


Figure 76: Activity Diagram (Rating & Review)

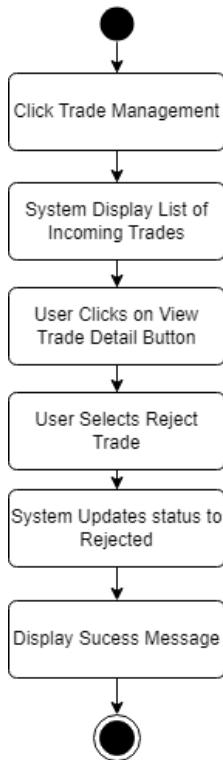


Figure 77: Activity Diagram (Reject Trade)

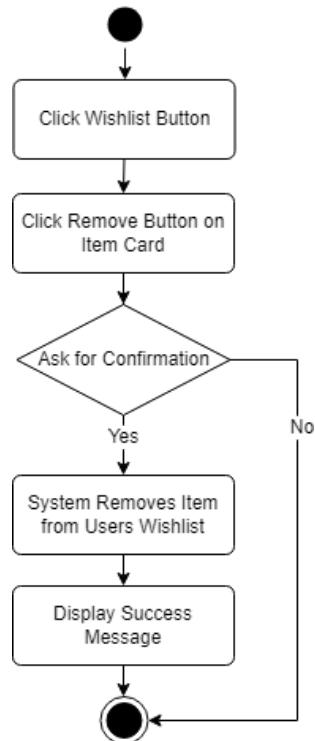


Figure 78: Activity Diagram (Remove from Wishlist)

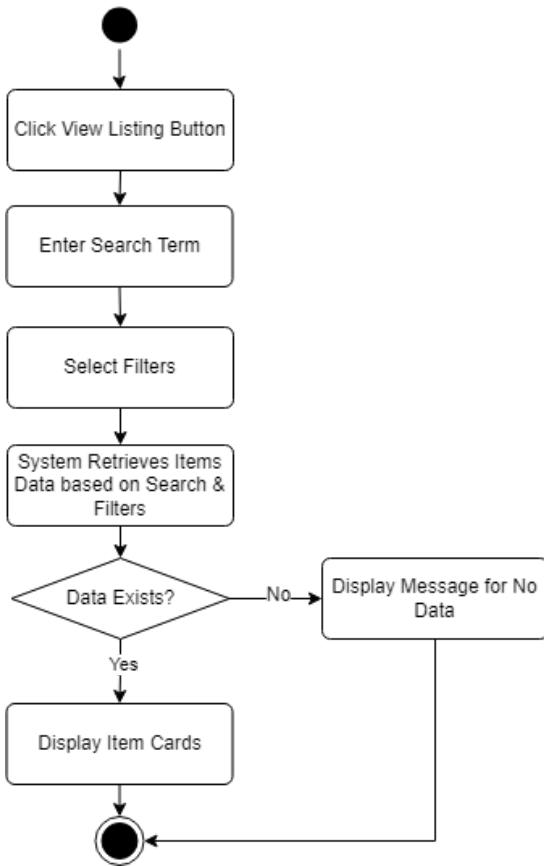


Figure 79: Activity Diagram (Search & Filter)

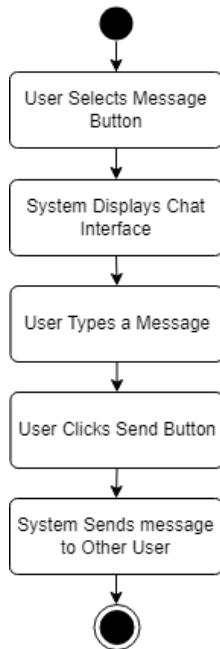


Figure 80: Activity Diagram (Send Message)

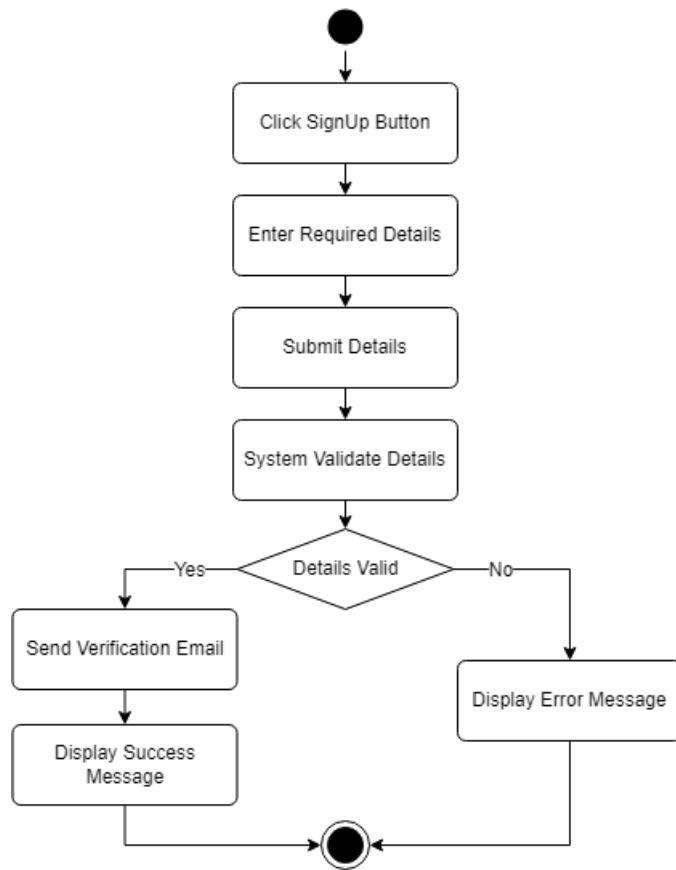


Figure 81: Activity Diagram (Sign Up)

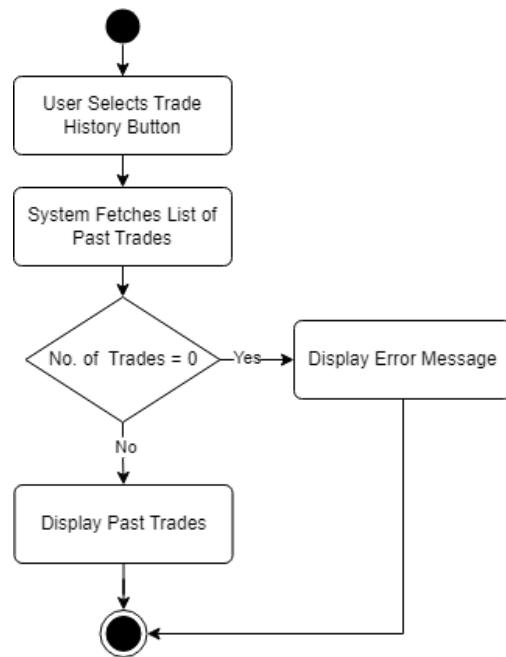


Figure 82: Activity Diagram (Trade History)

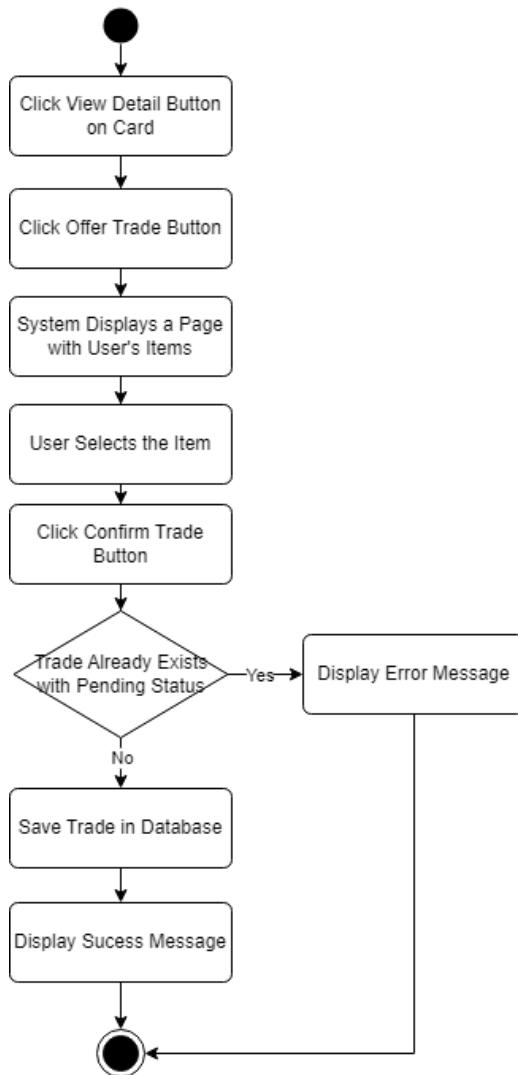


Figure 83: Activity Diagram (Trade Proposal)

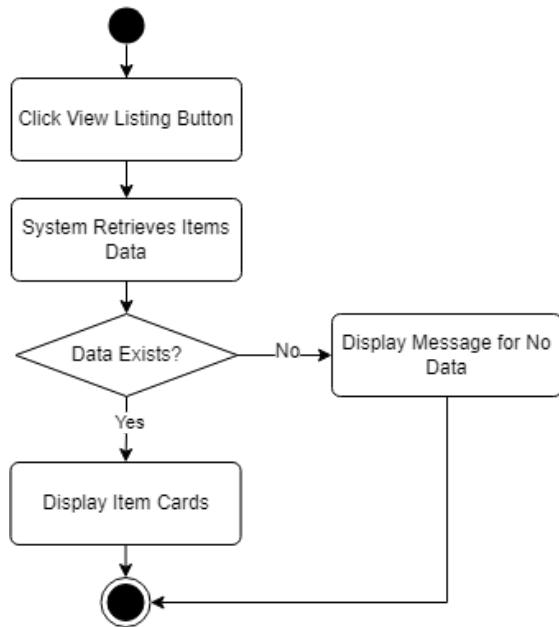


Figure 84: Activity Diagram (View Listing)

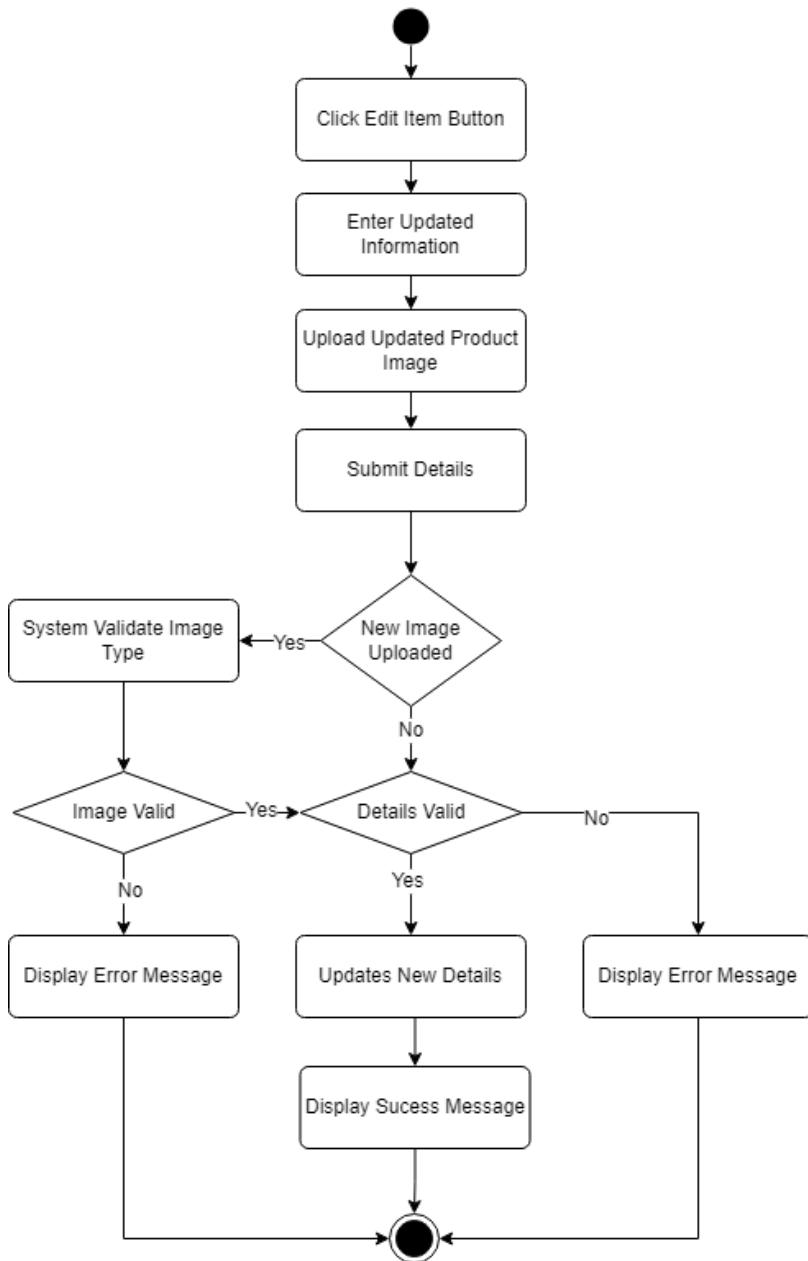


Figure 85: Activity Diagram (Edit Item)

8.7. Sequence Diagram

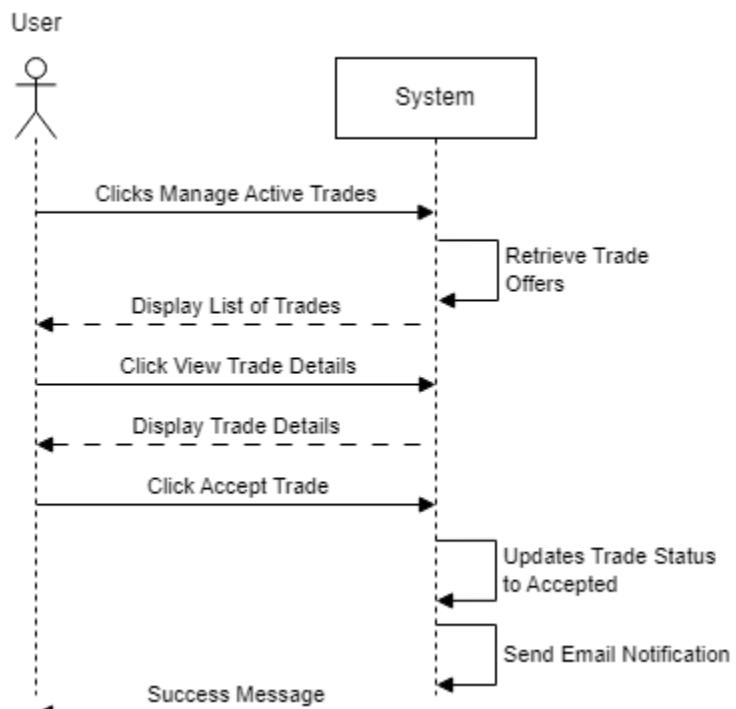


Figure 86: Sequence Diagram (Accept Trade)

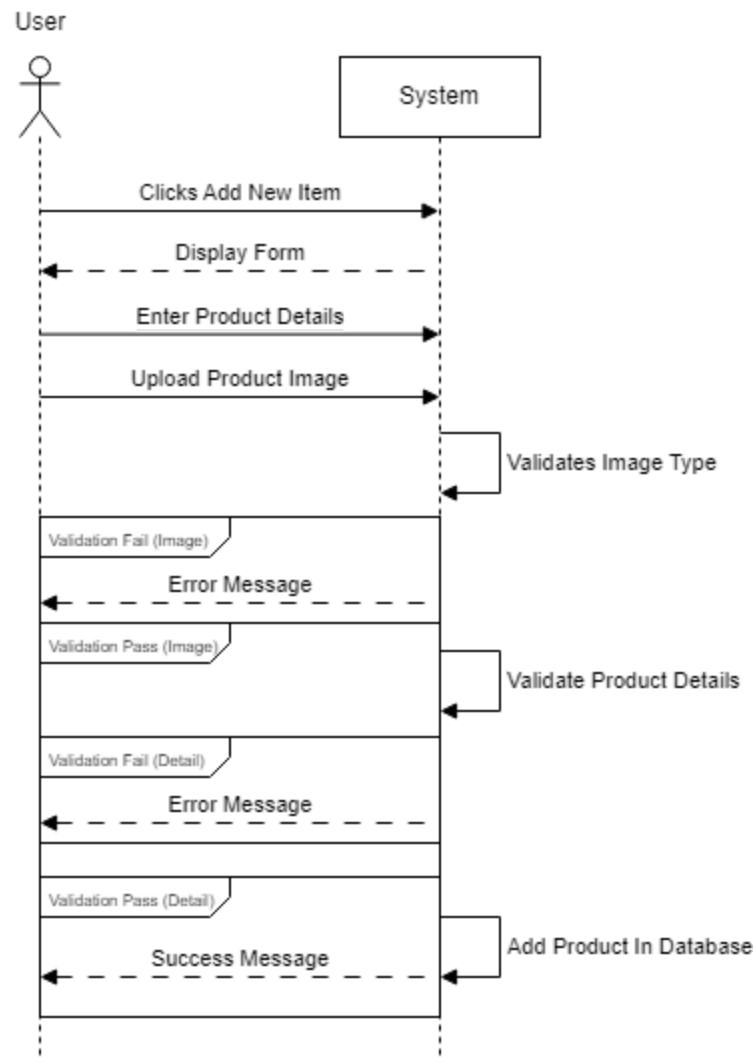


Figure 87: Sequence Diagram (Add Item)

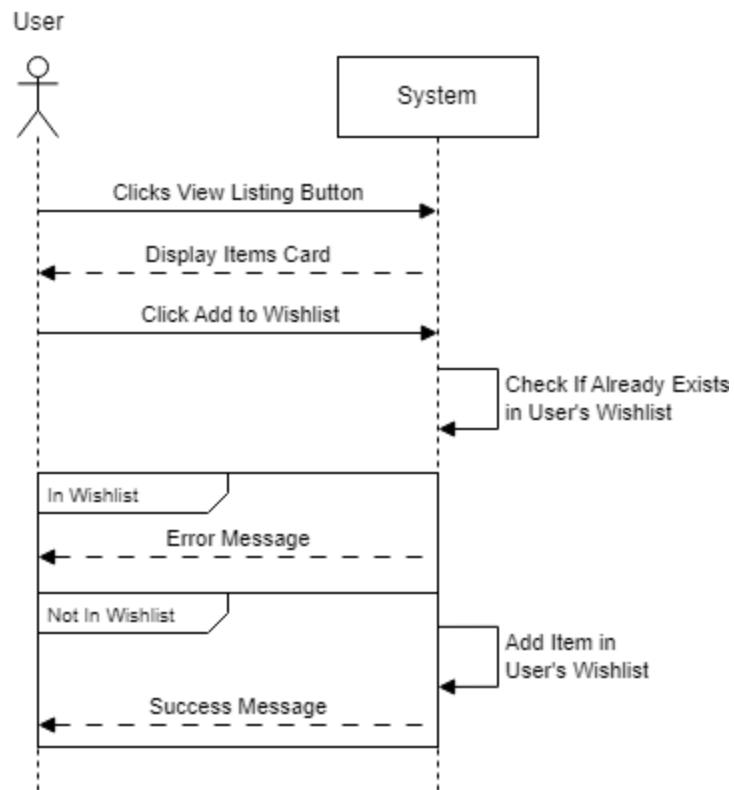


Figure 88: Sequence Diagram (Add to Wishlist)

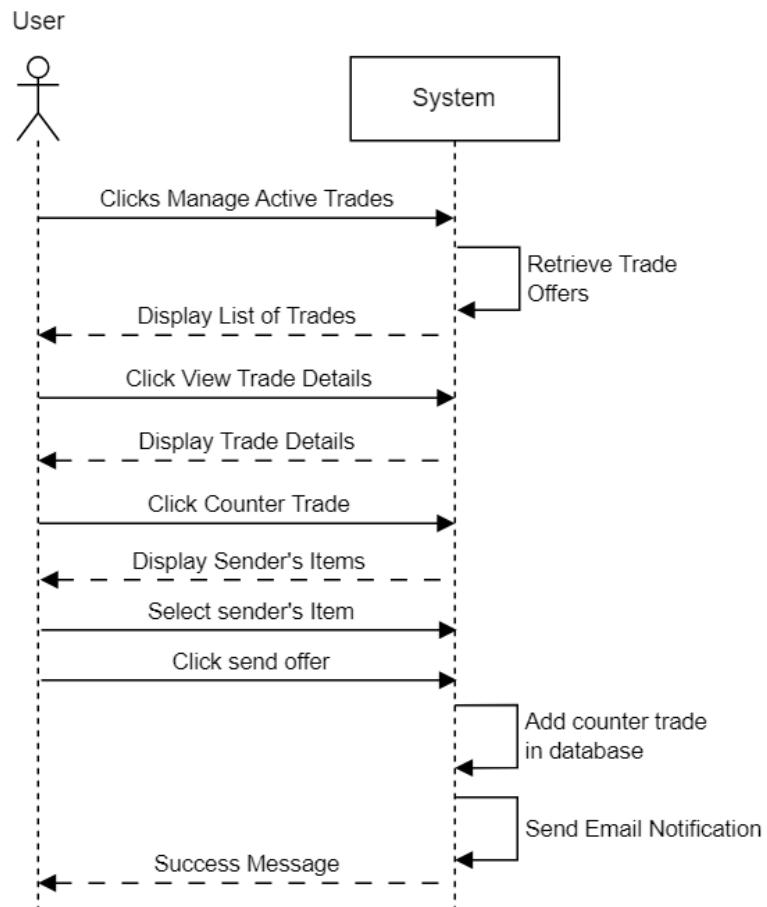


Figure 89: Sequence Diagram (Advance Trade Management)

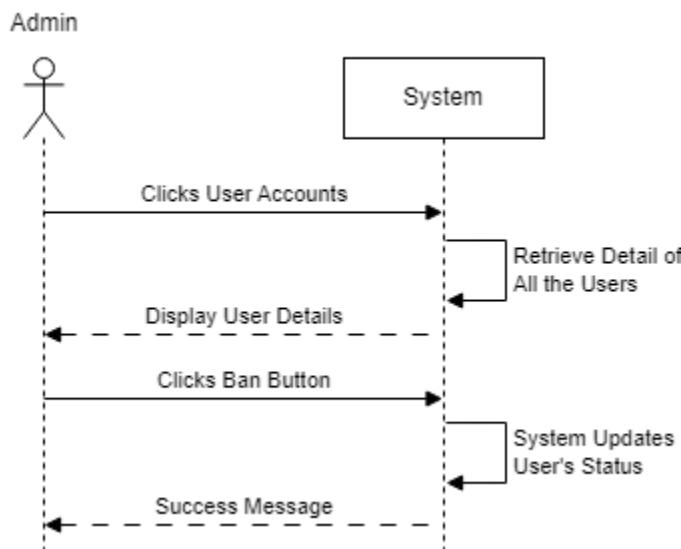


Figure 90: Sequence Diagram (Ban User Account)

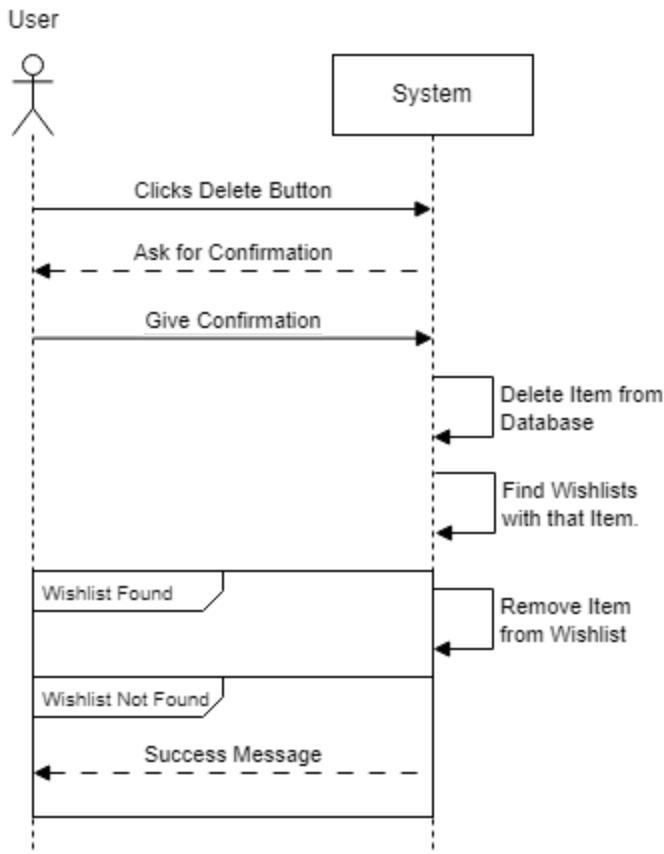


Figure 91: Sequence Diagram (Delete Item)

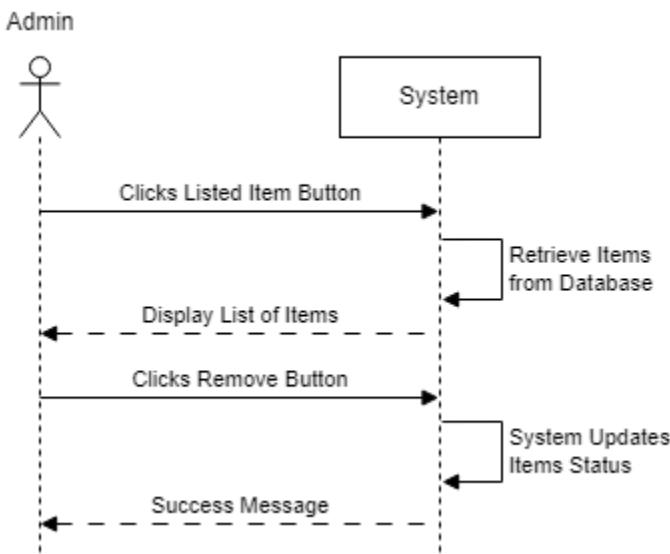


Figure 92: Sequence Diagram (Delete Listed Item)

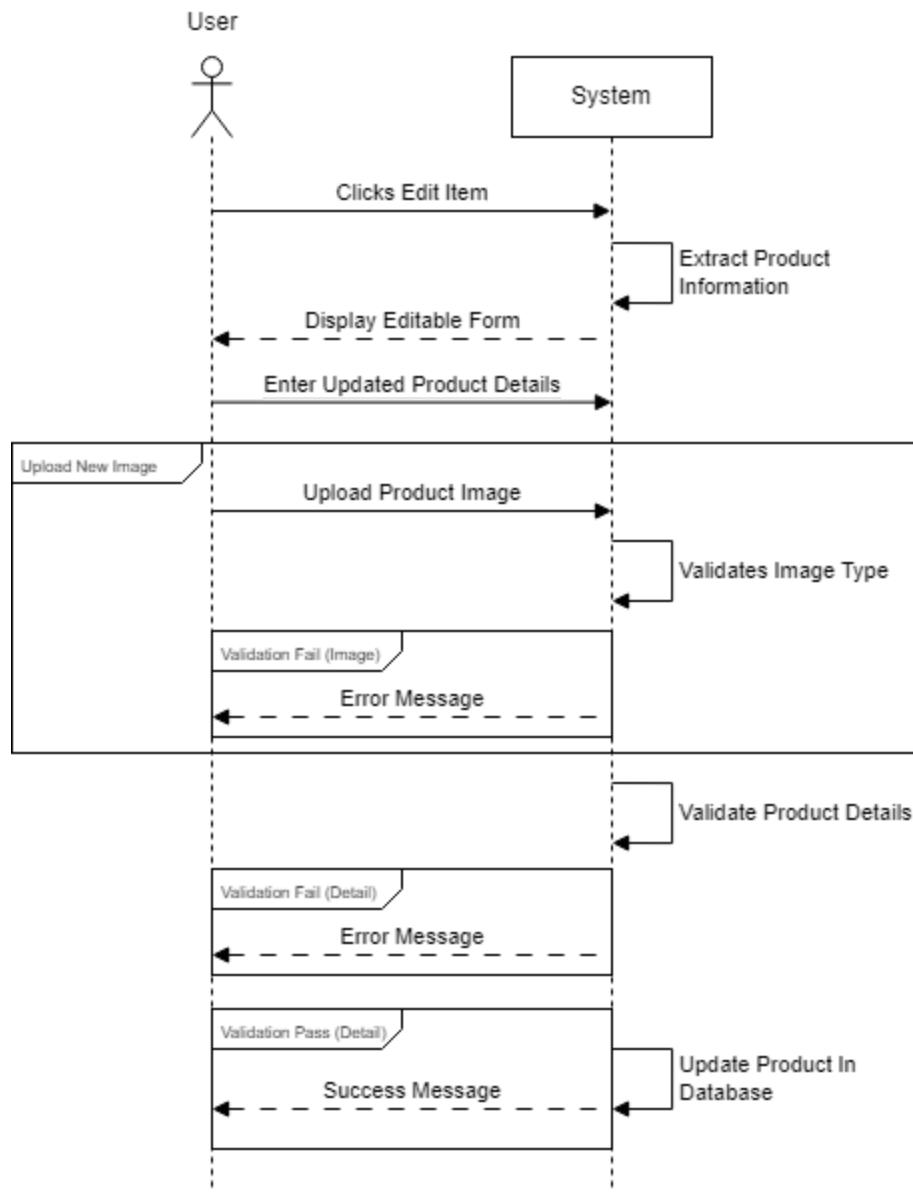


Figure 93: Sequence Diagram (Edit Item)

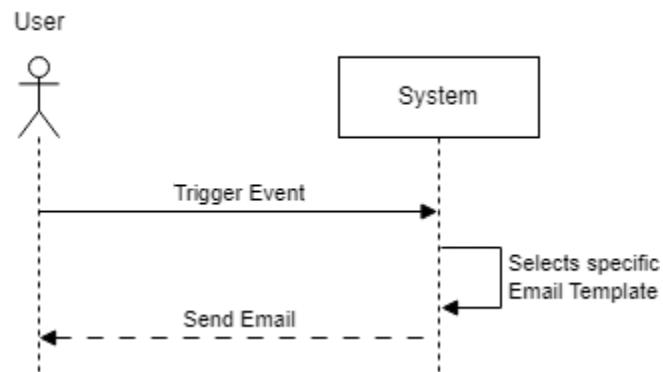


Figure 94: Sequence Diagram (Email Notification)

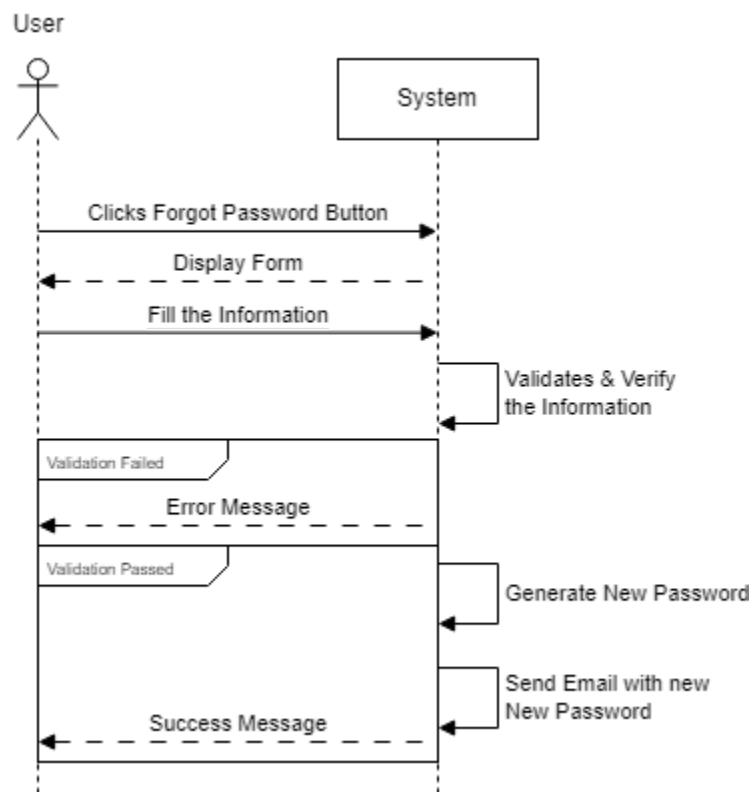


Figure 95: Sequence Diagram (Forget Password)

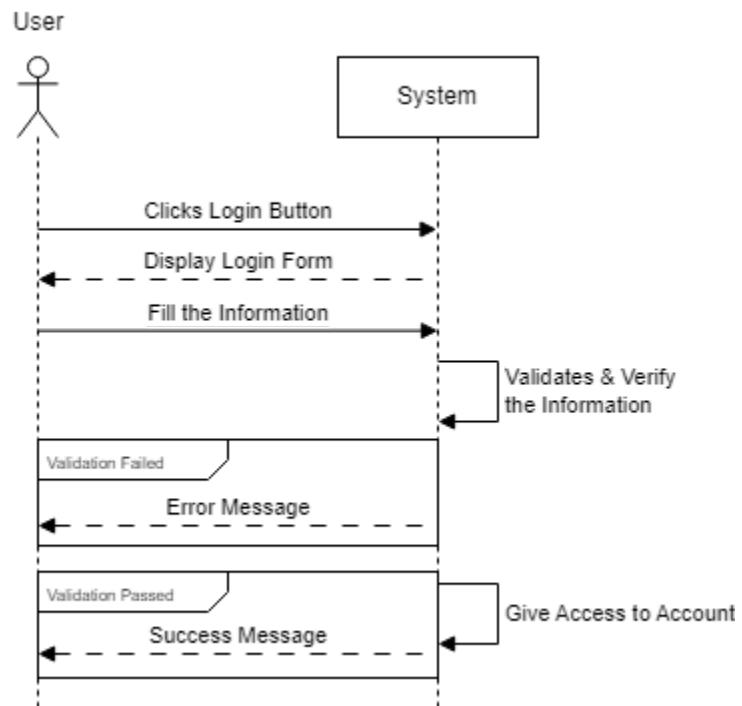


Figure 96: Sequence Diagram (Login)

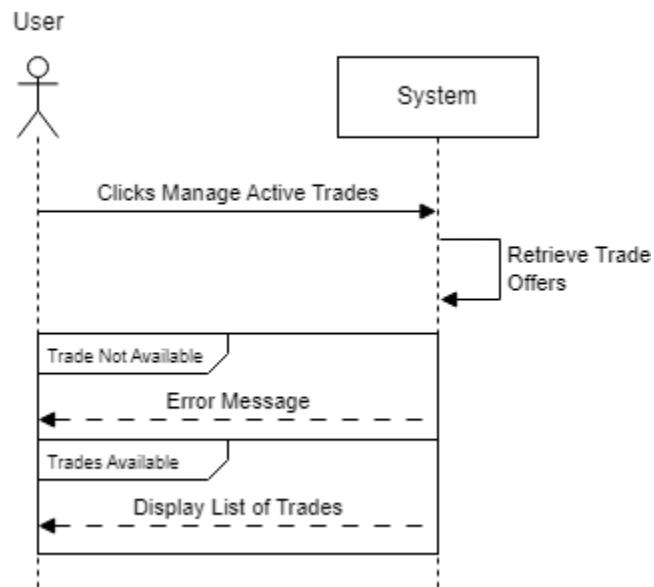


Figure 97: Sequence Diagram (Manage Active Trade)

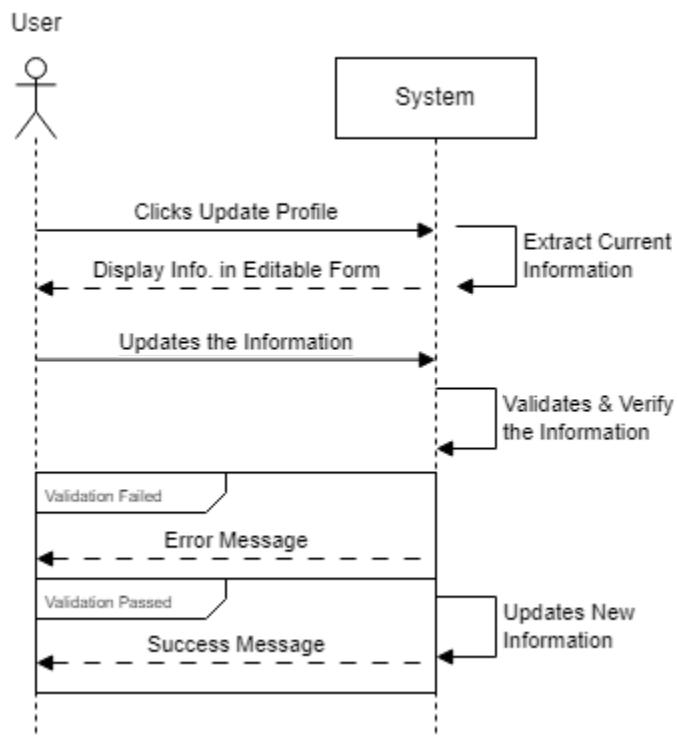


Figure 98: Sequence Diagram (Profile Management)

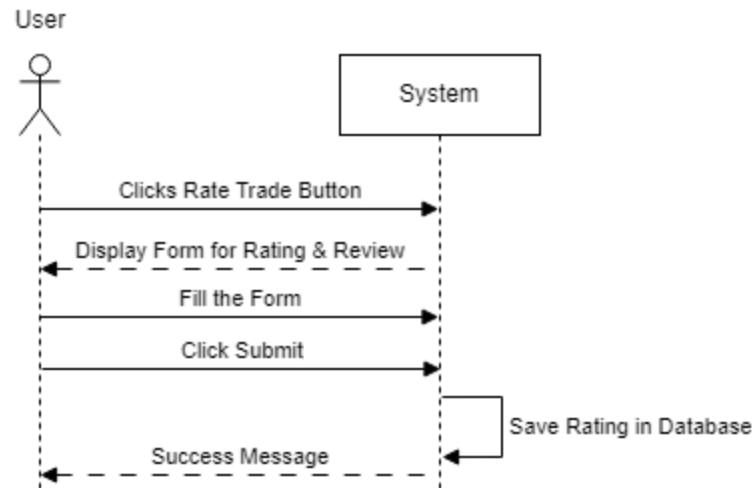


Figure 99: Sequence Diagram (Rating & Review)

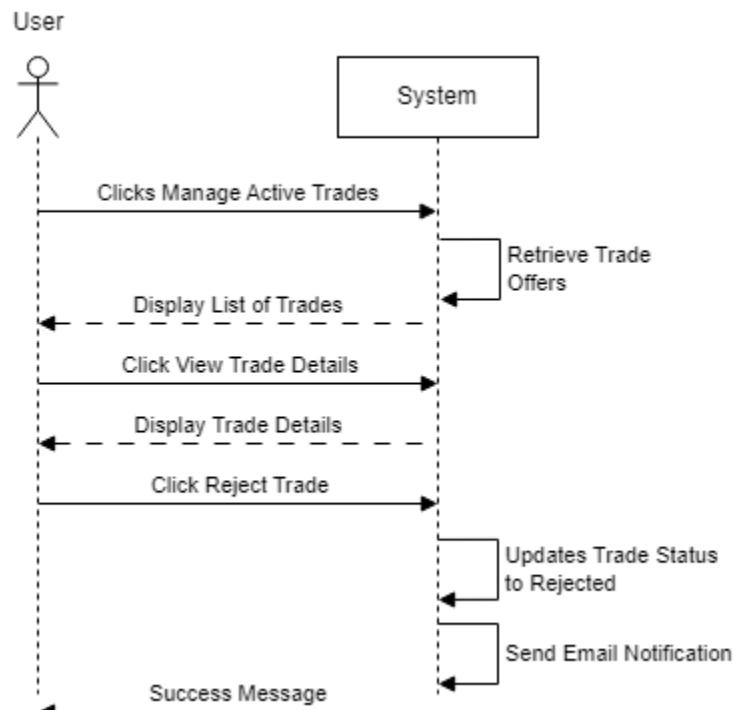


Figure 100: Sequence Diagram (Reject Trade)

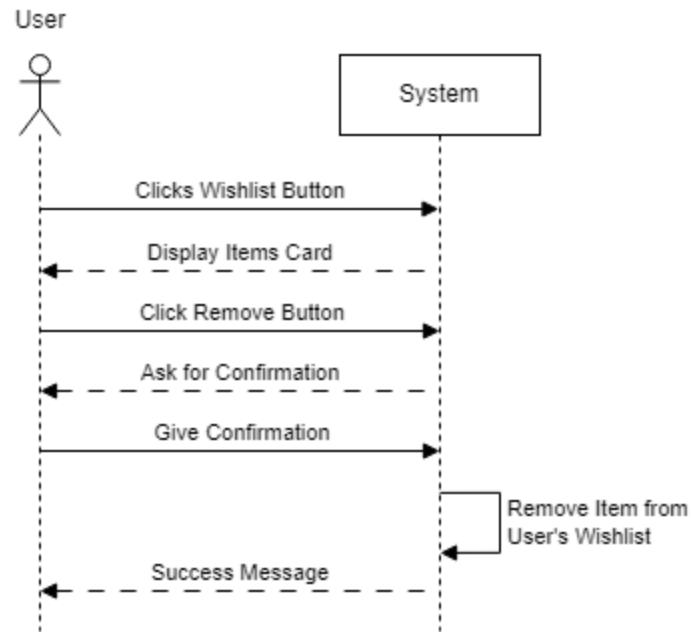


Figure 101: Sequence Diagram (Remove from Wishlist)

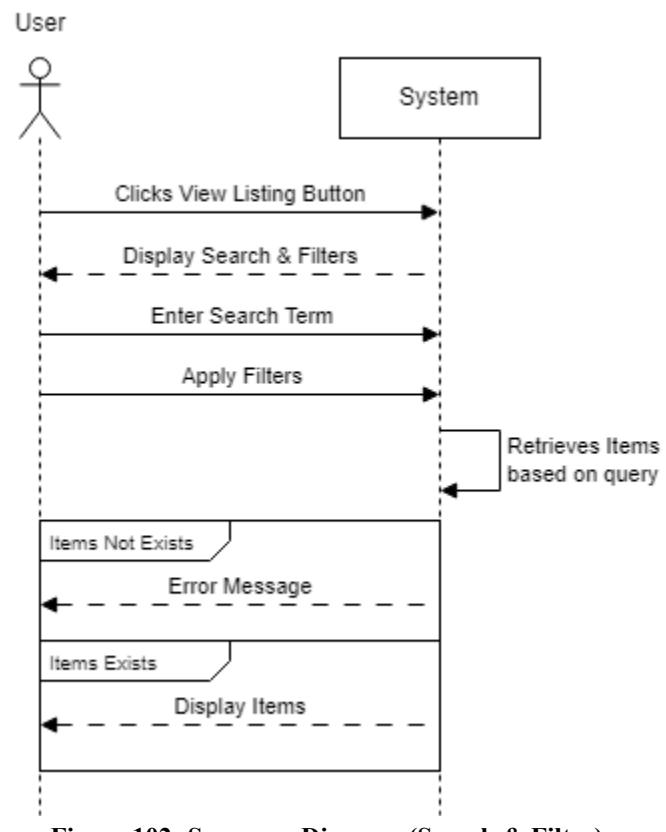


Figure 102: Sequence Diagram (Search & Filter)

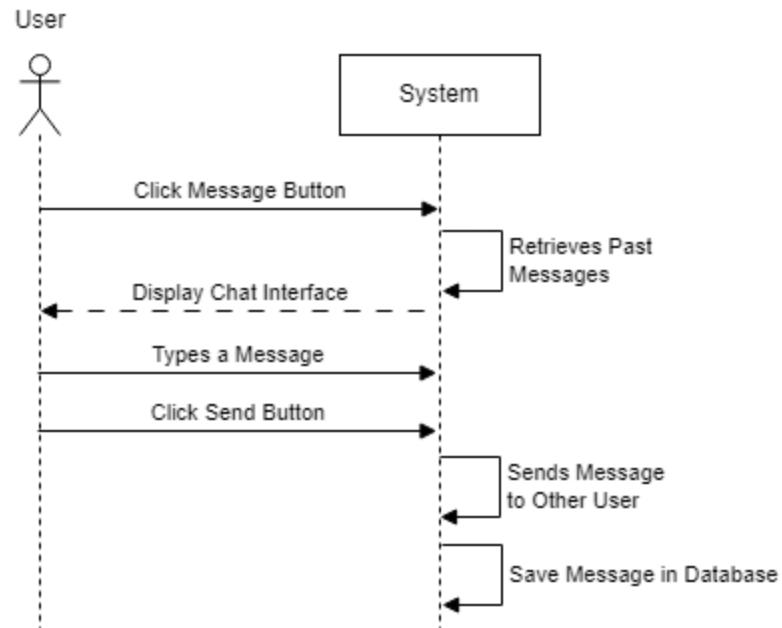


Figure 103: Sequence Diagram (Send Message)

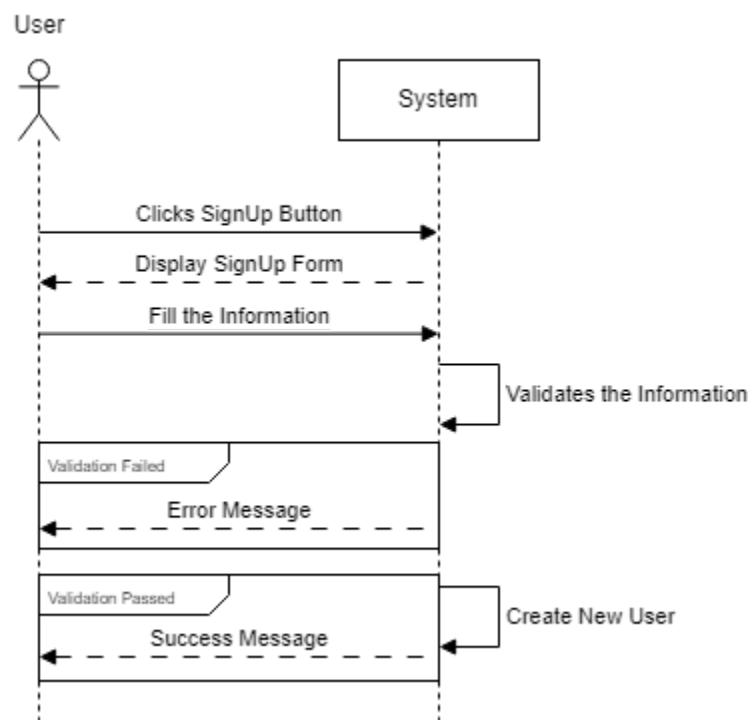


Figure 104: Sequence Diagram (Sign Up)

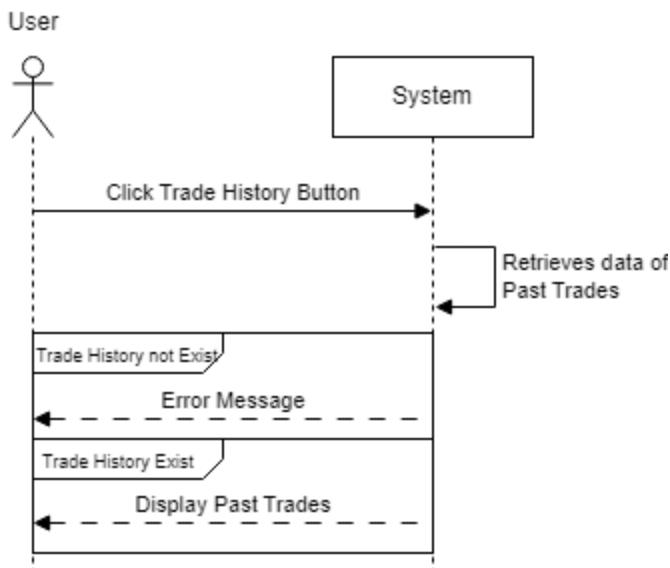


Figure 105: Sequence Diagram (Trade History)

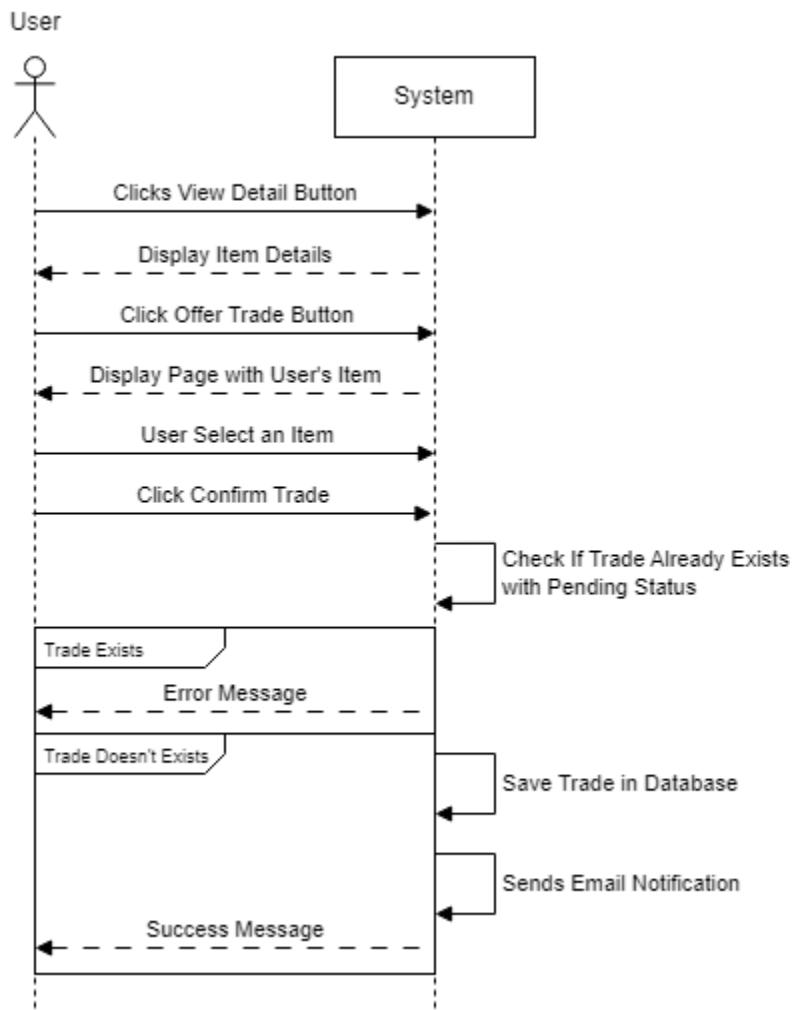


Figure 106: Sequence Diagram (Trade Proposal)

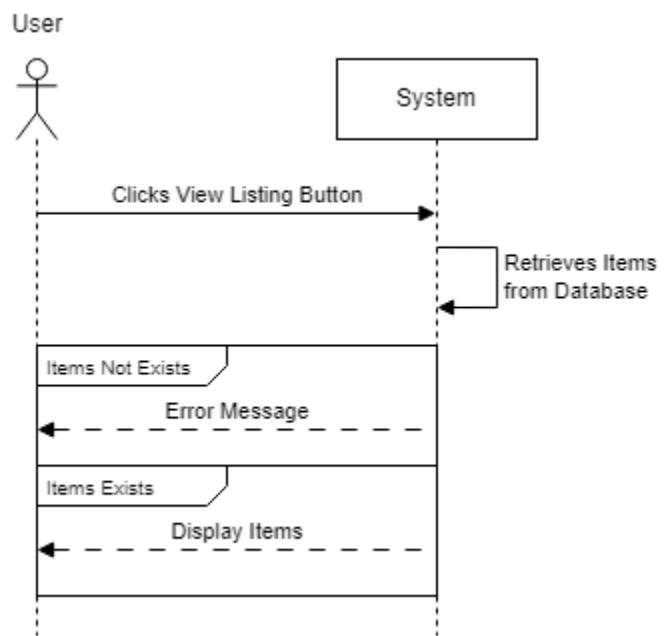


Figure 107: Sequence Diagram (View Listing)

Test Cases

1. Login

Test Case ID	TC_01				
Test Case Name	Login	Test Case Description	This test case verifies the functionality of the user login feature.		
Created By	Mustan & Rohail	Version	1.0	Date	30 Dec 2024
S.no	Prerequisites:				
1	User must already have an account.				
2	User must know their registered email and password.				
Test Scenario	Verify that a user can successfully log in with valid credentials and that appropriate error messages are shown for invalid inputs.				
Step no.	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended	
1	Navigate to the home page and click the "Login" button.	The system displays a login form with fields for email and password.	As Expected	Pass	
2	Fill in the login form with an invalid email format (e.g., "email.com") and click "Submit."	The system displays an error message: "Invalid email format."	As Expected	Pass	
3	Leave the email and password fields empty and click "Submit."	The system displays an error message: "Email and password fields cannot be empty."	As Expected	Pass	
4	Enter an unregistered email and a valid password, then click "Submit."	The system displays an error message: "Email is not registered."	As Expected	Pass	
5	Enter a registered email and an incorrect password, then click "Submit."	The system displays an error message: "Incorrect password."	As Expected	Pass	
6	Enter a registered email and the correct password, then click "Submit."	The system authenticates the user, logs them into their account, and redirects them to the Home page.	As Expected	Pass	

2. SignUp

Test Case ID	TC_02				
Test Case Name	SignUp	Test Case Description	This test case verifies the functionality of the user sign-up feature.		
Created By	Mustan & Rohail	Version	1.0	Date	30 Dec 2024

S.no	Prerequisites:			
1	User must not have an existing account.			
2	User must have a valid email address.			
Test Scenario		Verify that a user can successfully sign up by providing valid details and completing the account verification process.		
Step no.	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to the home page and click the "Sign-Up" button.	The system displays a sign-up form with fields for name, email, password, and mobile.	As Expected	Pass
2	Fill in the form with valid name, email, password, and mobile and click "Submit".	The system validates the input: mandatory fields are filled, email/mobile format is valid, and email is not already registered.	As Expected	Pass
3	Submit the form with an already-registered email address.	The system displays an error message: "Email is already in use."	As Expected	Pass
4	Submit the form with invalid email format	The system displays an error message: "Invalid email format."	As Expected	Pass
5	Successfully submit the form with valid data for a new user.	The system saves the data, sends a verification email, and displays a success message: "Check your email for verification."	As Expected	Pass
6	Open the verification email and click the verification link.	The system activates the account, redirects to the login page, and displays a success message: "Account verified successfully. Please log in."	As Expected	Pass

3. Forgot Password

Test Case ID	TC_03			
Test Case Name	Forgot Password	Test Case Description	This test case verifies the functionality of the "Forgot Password" feature, ensuring users can securely reset their password using their registered email and mobile number.	
Created By	Mustan & Rohail	Version	1.0	Date
				30 Dec 2024

S.no	Prerequisites:			
1	User must already have an account.			
Test Scenario	Verify that a user can reset their password by entering their registered email and mobile number, and receive a temporary password via email.			
Step no.	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to the login page and click on the "Forgot Password" link.	The system displays a form for entering a registered email address and mobile number.	As Expected	Pass
2	Enter an invalid email and mobile and click "Submit."	The system displays an error message: "Invalid format."	As Expected	Pass
3	Enter an unregistered email address and click "Submit."	The system displays an error message: "Email not found."	As Expected	Pass
4	Enter an unregistered email address and a valid mobile number, then click "Submit."	The system displays an error message: "Email not found."	As Expected	Pass
5	Enter a registered email address and valid mobile number, then click "Submit."	The system sends a reset link or password to the email and displays a confirmation message: "A password reset link has been sent to your email."	As Expected	Pass
6	Leave the email or mobile number fields empty and click "Submit."	The system displays an error message: "Both email and mobile number are required."	As Expected	Pass
7	Enter a valid registered email address and mobile number, then click "Submit."	The system validates the input and sends a temporary password to the registered email address. Displays a success message: "Temporary password sent."	As Expected	Pass

4. Profile Management

Test Case ID	TC_04				
Test Case Name	Profile Management	Test Case Description	This test case verifies the functionality of updating user profile information, ensuring the system validates and saves the changes.		
Created By	Mustan & Rohail	Version	1.0	Date	30 Dec 2024

S.no	Prerequisites:					
1	User must have a valid account and be logged in.					
Test Scenario	Verify that users can view, edit, and update their profile information while ensuring all inputs are validated.					
Step no.	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended		
1	Navigate to the profile page by clicking on the "Profile" button.	The system displays the current profile information, including fields like name, email, and phone number.	As Expected	Pass		
2	Click the "Edit" button to modify profile information.	The system displays editable fields for name, email, and mobile number.	As Expected	Pass		
3	Enter an invalid data format and click "Save."	The system displays an error message: "Invalid format."	As Expected	Pass		
4	Enter valid profile information and click "Save."	The system validates the inputs, saves the updated profile in the database, and displays the updated information on the profile page.	As Expected	Pass		
5	Verify that the updated profile information is correctly displayed upon returning to the profile page.	The system retrieves and displays the updated information from the database.	As Expected	Pass		

5. List Item

Test Case ID	TC_05					
Test Case Name	List Item	Test Case Description	This test case verifies the functionality of listing a new item for trade, ensuring the system validates inputs and stores item details.			
Created By	Mustan & Rohail	Version	1.0	Date 30 Dec 2024		
S.no	Prerequisites:					
1	User must have a valid account and be logged in.					
Test Scenario	Verify that users can list a new item for trade by providing valid details and that the system processes the inputs correctly.					
Step no.	Step Details	Expected Results	Actual Results	Pass / Fail / Not		

				executed / Suspended
1	Navigate to the "Add New Item" section.	The system displays a form for entering item details, including fields like name, description, category, and image upload.	As Expected	Pass
2	Submit the form with invalid data (e.g., empty mandatory fields, invalid file format, or invalid input values).	"The system displays appropriate error messages: "Fields cannot be empty", "Invalid file format." OR "Invalid input."	As Expected	Pass
3	Enter all valid item details (e.g., name, description, category) and upload a valid image file.	The system validates the inputs, saves the item details to the database, and displays a success message: "Item successfully listed."	As Expected	Pass
4	Verify that the newly listed item is displayed under the user's account.	The system retrieves the new item details from the database and displays them under the user's listings.	As Expected	Pass

6. Edit Item

Test Case ID	TC_06			
Test Case Name	Edit Item	Test Case Description	These test cases cover various scenarios for editing an item, including valid and invalid inputs, error handling, and successful updates.	
Created By	Mustan & Rohail	Version	1.0	Date 30 Dec 2024
S.no	Prerequisites:			
1	User must have a valid account and be logged in.			
2	The user must have at least one item listed for trade.			
Test Scenario	Verify that users can propose a trade, cancel proposals before submission, and handle system errors appropriately.			
Step no.	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to the my list and select an item to edit.	The system displays the current details of the selected item in an editable form.	As Expected	Pass
2	Modify valid fields (e.g., item name, description) and click the "Save" button.	The system validates the inputs, updates the item details in the database, and displays a confirmation	As Expected	Pass

		message: "Item updated successfully."		
2	Attempt to edit an item and leave mandatory fields empty before saving.	The system displays an error message indicating that mandatory fields cannot be empty.	As Expected	Pass
2	Attempt to enter invalid data formats (e.g., non-numeric values for numeric fields) and save the form.	The system displays an error message indicating invalid inputs.	As Expected	Pass

7. Delete Item

Test Case ID	TC_07				
Test Case Name	Delete Item	Test Case Description	These test cases cover various scenarios for deleting an item, including successful deletion and error handling.		
Created By	Mustan & Rohail	Version	1.0	Date	30 Dec 2024
S.no	Prerequisites:				
1	User must have a valid account and be logged in.				
2	The user must have at least one item listed for trade.				
Test Scenario	Verify that users can propose a trade, cancel proposals before submission, and handle system errors appropriately.				
Step no.	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended	
1	Navigate to my list page and select an item to delete.	The system displays a confirmation dialog for deleting the selected item.	As Expected	Pass	
2	Confirm the deletion by clicking the "Delete" button in the confirmation dialog.	The system removes the item from the listings, updates the database, and displays a confirmation message: "Item successfully deleted."	As Expected	Pass	
2	Cancel the deletion process in the confirmation dialog.	The system does not delete the item, and the item remains in the user's listings.	As Expected	Pass	

8. View Item Listing

Test Case ID	TC_08		
Test Case Name	View Item Listings	Test Case Description	This test case verifies the functionality of viewing all available items for trade and the detailed view of individual items.

Created By	Mustan & Rohail	Version	1.0	Date	30 Dec 2024
<hr/>					
S.no	Prerequisites:				
1	User must have a valid account and be logged in.				
<hr/>					
Test Scenario	Verify that users can view the list of items available for trade and detailed information for each item.				
<hr/>					
Step no.	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended	
1	Navigate to the "Item Listings" page.	The system retrieves and displays a list of all available items, including their basic details (e.g., name, description).	As Expected	Pass	
2	Select an item from the list to view more details.	The system displays the selected item's detailed information, including a full description, images, and trade options.	As Expected	Pass	
3	Verify the system response when no items are available for trade.	The system displays a message: "No items available for trade."	As Expected	Pass	

9. Search & Filter

Test Case ID	TC_09				
Test Case Name	Search & Filter	Test Case Description	This test case verifies the functionality of searching and filtering items, as well as handling scenarios where no results are found or system errors occur.		
Created By	Mustan & Rohail	Version	1.0	Date	30 Dec 2024
<hr/>					
S.no	Prerequisites:				
1	User must have a valid account and be logged in.				
2	The system must have items available in the database with searchable and filterable attributes.				
<hr/>					
Test Scenario	Verify that users can search for specific items or apply filters to refine the displayed item listings.				
<hr/>					
Step no.	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended	
1	Enter a keyword in the search bar and press	The system retrieves and displays items matching the	As Expected	Pass	

	"Search."	search keyword.		
2	Apply a filter (e.g., category, item condition, price range) and click "Apply Filters."	The system retrieves and displays items matching the selected filter criteria.	As Expected	Pass
3	Enter a keyword and apply a filter simultaneously (e.g., "Laptop" + "Condition: New").	The system retrieves and displays items matching both the search keyword and the filter criteria.	As Expected	Pass
4	Enter a keyword or apply filters that do not match any items.	The system displays a message: "No items match found."	As Expected	Pass

10. Add to Wishlist

Test Case ID	TC_10					
Test Case Name	Add to Wishlist	Test Case Description	This test case verifies the functionality of adding items to a wishlist			
Created By	Mustan & Rohail	Version	1.0	Date 30 Dec 2024		
<hr/>						
S.no	Prerequisites:					
1	User must have a valid account and be logged in.					
2	Items must be available on the item listings page.					
3	The user must have access to a wishlist section in their profile.					
<hr/>						
Test Scenario	Verify that users can search for specific items or apply filters to refine the displayed item listings.					
<hr/>						
Step no.	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended		
1	Navigate to an item listing and click the "Add to Wishlist" button.	The system adds the selected item to the user's wishlist and displays a confirmation message: "Item added to wishlist."	As Expected	Pass		
2	Navigate to the "Wishlist" section in the user profile.	The system displays all items added to the user's wishlist, with their details (e.g., name, description, image).	As Expected	Pass		
3	Attempt to add the same item to the wishlist again.	The system prevents duplicate entries and displays a message: "This item is already in your wishlist."	As Expected	Pass		

11. View Wishlist

Test Case ID	TC_11				
Test Case Name	View Wishlist	Test Case Description	These test cases cover scenarios for viewing and interacting with items saved in the wishlist, including handling empty wishlists.		
Created By	Mustan & Rohail	Version	1.0	Date	30 Dec 2024
S.no	Prerequisites:				
1	User must have a valid account and be logged in.				
2	The user must have a wishlist.				
Test Scenario	Verify that users can propose a trade, cancel proposals before submission, and handle system errors appropriately.				
Step no.	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended	
1	Navigate to the wishlist section through the profile.	The system fetches and displays the wishlist items, including details such as item name, image, and description.	As Expected	Pass	
2	Click on an item in the wishlist to view its full details or initiate a trade.	The system displays the full details of the selected item, including owner information and trade options.	As Expected	Pass	
2	Attempt to view the wishlist when it is empty.	The system displays a message: "Your wishlist is empty"	As Expected	Pass	

12. Item Detail

Test Case ID	TC_12				
Test Case Name	Item Detail	Test Case Description	This test case verifies the functionality of viewing detailed information about an item.		
Created By	Mustan & Rohail	Version	1.0	Date	30 Dec 2024
S.no	Prerequisites:				
1	User must have a valid account and be logged in.				
2	Items must exist in the database.				
Test Scenario	Verify that users can search for specific items or apply filters to refine the displayed item listings.				

Step no.	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Click on an item from the item listing page or wishlist.	The system fetches and displays detailed information about the selected item, including name, description, condition, category, images, owner info, and available actions.	As Expected	Pass
2	Attempt to access an item that has been deleted by its owner.	The system displays error message	As Expected	Pass

13. Trade Proposal

Test Case ID	TC_13			
Test Case Name	Trade Proposal	Test Case Description		This test case validates the functionality of proposing a trade by offering an item in exchange for another user's listed item.
Created By	Mustan & Rohail	Version	1.0	Date 30 Dec 2024
S.no	Prerequisites:			
1	User must have a valid account and be logged in.			
2	The user must have at least one item listed for trade.			
3	The item being proposed for trade must exist in the database.			
Test Scenario	Verify that users can propose a trade, cancel proposals before submission, and handle system errors appropriately.			
Step no.	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to an item listing and click the "Propose Trade" button.	The system displays a form where the user selects an item from their own listings to offer.	As Expected	Pass
2	Select an item for exchange and click "Submit."	The system saves the proposal in the database and displays a confirmation message: "Trade proposal sent."	As Expected	Pass
2	Attempt to access an item that has been deleted by its owner.	The system displays error message	As Expected	Pass
2	Attempt to submit the form with invalid inputs (e.g., no item selected for	The system displays error message	As Expected	Pass

	exchange).			
2	Verify that the user receives the trade proposal in their profile.	The user receives a trade with its details, including the item being offered and the item requested for trade.	As Expected	Pass

14. Accept Trade

Test Case ID	TC_14			
Test Case Name	Accept Trade	Test Case Description		These test cases cover scenarios for accepting or rejecting a trade proposal, handling valid input
Created By	Mustan & Rohail	Version	1.0	Date 30 Dec 2024
S.no	Prerequisites:			
1	User must have a valid account and be logged in.			
2	The user must have received at least one pending trade proposal.			
Test Scenario	Verify that users can propose a trade, cancel proposals before submission, and handle system errors appropriately.			
Step no.	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to the trade management section.	The system displays the list of trades with its details, including the item being offered and its description.	As Expected	Pass
2	Click the "Accept" button to confirm the trade.	The system finalizes the trade, updates the trade status in the database.	As Expected	Pass
2	Click "Reject" instead of "Accept" for a trade proposal.	The system updates the trade status as rejected.	As Expected	Pass

15. Reject Trade

Test Case ID	TC_15			
Test Case Name	Reject Trade	Test Case Description		These test cases cover scenarios for rejecting trade proposals.
Created By	Mustan & Rohail	Version	1.0	Date 30 Dec 2024
S.no	Prerequisites:			
1	User must have a valid account and be logged in.			
2	The user must have received at least one pending trade proposal.			

<u>Test Scenario</u>	Verify that users can propose a trade, cancel proposals before submission, and handle system errors appropriately.			
Step no.	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to the trade management section.	The system displays the list of trades with its details, including the item being offered and its description.	As Expected	Pass
2	Click the "Reject" button to decline the trade.	The system updates the trade status in the database	As Expected	Pass
2	Click "Accept" instead of "Reject" for a trade proposal.	The system finalizes the trade, updates the trade status in the database.	As Expected	Pass

16. Cancel Trade

Test Case ID	TC_16			
Test Case Name	Cancel Trade	Test Case Description	These test cases cover scenarios for canceling trade proposals.	
Created By	Mustan & Rohail	Version	1.0	Date 30 Dec 2024
S.no	Prerequisites:			
1	User must have a valid account and be logged in.			
2	The user must have at least one active trade proposal they initiated.			
<u>Test Scenario</u>	Verify that users can propose a trade, cancel proposals before submission, and handle system errors appropriately.			
Step no.	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to the "Sent Requests" section and select an active trade proposal to cancel.	The system displays the details of the selected trade and shows the "Cancel Trade" button.	As Expected	Pass
2	Click the "Cancel Trade" button and confirm the cancellation.	The system updates the trade status to "Cancelled" in the database.	As Expected	Pass

17. Manage Active Trade

Test Case ID	TC_17		
Test Case Name	Manage Active Trades	Test Case Description	The test will cover scenarios for viewing, accepting, rejecting trade proposals.

Created By	Mustan & Rohail	Version	1.0	Date	30 Dec 2024
<hr/>					
S.no	Prerequisites:				
1	User must have a valid account and be logged in.				
<hr/>					
Test Scenario	Verify that users can propose a trade, cancel proposals before submission, and handle system errors appropriately.				
<hr/>					
Step no.	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended	
1	Navigate to the "Sent Offers" section	The system displays a list of all the trade offers the user has sent, with trade details (items, status).	As Expected	Pass	
2	Navigate to the "Received Offers" section.	The system displays a list of all received trade offers with relevant details (items, status).	As Expected	Pass	

18. Remove from Wishlist

Test Case ID	TC_18			
Test Case Name	Remove from Wishlist	Test Case Description	The test will cover scenarios for removing individual items	
Created By	Mustan & Rohail	Version	1.0	Date
<hr/>				
S.no	Prerequisites:			
1	User must have a valid account and be logged in.			
2	The user must have at least one item saved in their wishlist.			
<hr/>				
Test Scenario	Verify that users can propose a trade, cancel proposals before submission, and handle system errors appropriately.			
<hr/>				
Step no.	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to the "Wishlist" section.	The system displays the user's wishlist, showing all saved items.	As Expected	Pass
2	Select an item from the wishlist and click "Remove from Wishlist."	The system removes the selected item from the wishlist, updates the display	As Expected	Pass
3	Verify that the item is no longer present in the wishlist after removal.	The removed item no longer appears in the wishlist, and the remaining items are displayed correctly.	As Expected	Pass

19. Advanced Trade Management

Test Case ID	TC_19			
Test Case Name	Advanced Trade Management	Test Case Description	The test will cover scenarios for creating and submitting advance trade offer.	
Created By	Mustan & Rohail	Version	1.0	Date 13 March 2025
S.no		Prerequisites:		
1		User must have a valid account and be logged in.		
2		The user must have at least one item listed for trade.		
3		User must have received a trade proposal.		
Test Scenario	Verify that the user can successfully create and submit a counter offer by selecting an item from the sender's list.			
Step no.	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to "Active Trades" and click on the "Counter Offer" button for a specific trade.	The system displays the trade proposal page with the recipient's item pre-selected.	As Expected	Pass
2	Select an item from the sender's list.	The system updates the trade proposal with the selected item.	As Expected	Pass
3	Click "Send Offer" without selecting an item.	Click "Send Counter Offer" without selecting an item. The system displays an error message: "Please select an item."	As Expected	Pass
4	Click "Send Counter Offer" after selecting an item.	The system saves the counter offer, notifies the sender, and displays a confirmation message: "Counter offer sent successfully."	As Expected	Pass
5	Attempt to send a duplicate counter offer for the same trade.	The system displays an error message: "Same offer already exists."	As Expected	Pass
6	Verify that the item is no longer present in the wishlist after removal.	The removed item no longer appears in the wishlist, and the remaining items are displayed correctly.	As Expected	Pass

20. Ban User Accounts

Test Case ID	TC_20	
---------------------	-------	--

Test Case Name	Ban User Accounts	Test Case Description	This test case verifies the functionality of banning user accounts.		
Created By	Mustan & Rohail	Version	1.0	Date	13 March 2025
S.no	Prerequisites:				
1	Admin must be logged into the admin portal.				
2	Admin must have the necessary permissions to ban users.				
Test Scenario	Verify that the admin can successfully ban a user account.				
Step no.	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended	
1	Navigate to the "User Accounts" section.	The system displays a list of all user accounts with their status.	As Expected	Pass	
2	Select a user account to ban.	The system processes the request and updates the account status.	As Expected	Pass	
3	Attempt to ban a user when there are no users to manage.	The system displays a message: "No user accounts to manage."	As Expected	Pass	
4	Click the "Unban" button.	The system updates the account status and restores access.	As Expected	Pass	

21. Delete List Items

Test Case ID	TC_21				
Test Case Name	Delete List Items	Test Case Description	This test case verifies the functionality of admin deleting user-listed items		
Created By	Mustan & Rohail	Version	1.0	Date	13 March 2025
S.no	Prerequisites:				
1	Admin must be logged into the admin portal.				
2	Admin must have the necessary permissions to ban users.				
Test Scenario	Verify that the admin can successfully delete user-listed items from the platform.				
Step no.	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended	
1	Navigate to the "Listed Items" section.	The system displays a list of all listed items with their details.	As Expected	Pass	

2	Select an item to view or delete.	The system processes the request and displays item details.	As Expected	Pass
3	Click the "Delete" button to remove the item.	The system removes the item from the listing and database.	As Expected	Pass
4	Attempt to delete an item when there are no items listed.	The system displays a message: "No items to manage at this time."	As Expected	Pass
5	Verify that the deleted item no longer appears in the listing.	The system does not display the deleted item in the "Listed Items" section.	As Expected	Pass

22. Rate and Review

Test Case ID	TC_22							
Test Case Name	Rate and Review	Test Case Description		This test case verifies the functionality of the rate and review system				
Created By	Mustan & Rohail	Version		1.0	Date 15 March 2025			
S.no	Prerequisites:							
1	The user must be logged into their account.							
2	The user must have completed at least one trade.							
Test Scenario	Verify that users can successfully rate and review their trade partners after completing a trade.							
Step no.	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended	Fixes (If Failed)			
1	Navigate to the "Rate Trade" section after completing a trade.	The system displays a form for rating (1-5 stars) and writing a review.	As Expected	Pass				
2	Enter a valid rating and review, then click "Submit."	The system validates the input, saves the rating and review, and displays a confirmation message: "Rating and review submitted successfully."	As Expected	Pass				
3	Attempts to rate a trade that was already rated.	New rating is created for the Trade which was already rated.	Pass (after fix)	Add a check for existing rating by tradeId				
		As Expected	Pass					

4	Attempt to submit a review without a rating.	The system displays an error message: "All fields are required."	As Expected	Pass	
5	Attempt to submit a rating without a review.	The system displays an error message: "All fields are required."	As Expected	Pass	
6	Sender attempts to rate the trade.	The system displays an error message: "Not authorized to rate this trade"	Rating and review were submitted successfully.	Pass (after fix)	Add a condition that ensures only the recipient of the trade can rate it.
			As Expected	Pass	
7	Verify that the submitted rating and review are saved and appear on the trade partner's profile.	The system updates the trade partner's profile with the new rating and review.	As Expected	Pass	

23. Email Notification

Test Case ID	TC_23								
Test Case Name	Email Notification System	Test Case Description		This test case verifies the functionality of the email notification system for trade updates.					
Created By	Mustan & Rohail	Version	1.0	Date	15 March 2025				
S.no	Prerequisites:								
1	The user must have a registered and verified email address.								
2	The user must have relevant activities triggering notifications (e.g., trade proposal, trade acceptance, account updates).								
Test Scenario	Verify that users can successfully rate and review their trade partners after completing a trade.								
Step no.	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended					
1	Trigger a trade-related event (e.g., new trade proposal).	The system generates and sends an email notification to the user.	As Expected	Pass					
2	Verify that the email contains the correct trade details.	The email includes all relevant trade detail	As Expected	Pass					

24. Trade History

Test Case ID	TC_24							
Test Case Name	Trade History	Test Case Description	This test case verifies the functionality of the trade history feature, ensuring users can access and review past trades.					
Created By	Mustan & Rohail	Version	1.0	Date	15 March 2025			
S.no	Prerequisites:							
1	The user must be logged into their account.							
2	The user must have completed at least one trade.							
Test Scenario	Verify that users can view a history of their past trades and access detailed trade information.							
Step no.	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended				
1	Navigate to the "Trade History" section.	The system retrieves and displays a list of all trades.	As Expected	Pass				
2	Select a specific trade from the history.	The system displays full details of the selected trade, including traded items, counterparties, and trade status.	As Expected	Pass				
3	Attempt to access trade history when no trades have been completed.	The system displays a message: "No trade history available."	As Expected	Pass				

25. Send Message

Test Case ID	TC_25							
Test Case Name	Send Message	Test Case Description	This test case verifies the functionality of the messaging feature, ensuring users can send and receive messages within the platform.					
Created By	Mustan & Rohail	Version	1.0	Date	15 March 2025			
S.no	Prerequisites:							
1	The user must be logged into their account.							
2	The user must have an active trade or a user to message.							

<u>Test Scenario</u>	Verify that users can send messages to other users and receive them in real-time.				
Step no.	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended	Fixes (If Failed)
1	Navigate to a trade and click message button.	The system displays a messaging interface.	As Expected	Pass	
2	Type a message in the messaging interface and Click the "Send" button.	The system delivers the message to the recipient.	As Expected	Pass	
3	Attempt to send an empty message.	The system doesn't deliver the message to the recipient.	As Expected	Pass	
4	Verify that the recipient receives the sent message in real-time.	The recipient's chat updates with the new message instantly.	A new chat room was being created each time due to inconsistent room id.	Fail	Sort was added when generating the room id to ensure consistency between sender and receiver.
			As Expected	Pass (after fix)	
5	Check if a message appears in the sender's message history after sending.	The sent message appears in the sender's message history.	Old Messages were not appearing in the message history	Fail	Set state was added inside useEffect to display the message history.
			As Expected	Pass (after fix)	

User Manual

1. General Information

1.1. Purpose of this Document

This user manual is designed to provide comprehensive guidance for users of the Swappify platform. It covers key instructions for getting started, managing profiles, initiating trade proposals, and using the platform's features effectively. The manual includes step-by-step explanations and best practices to help users navigate the barter trading system with ease and confidence.

1.2. Authorized Use Permission

Swappify is intended for individual users who wish to exchange goods and services without monetary transactions. Access is authorized for verified users who adhere to community guidelines and Admin. All users except Admin must complete the registration and verification process to participate in trades, post items, or communicate within the platform.

1.3. System Summary

Swappify is a digital barter trading platform aimed at promoting sustainable consumption by enabling users to exchange items without using money. The system emphasizes transparency, trust, and security through user authentication, advanced item search, direct trade proposals, and post-trade rating features. It fosters a community-driven ecosystem built around the principles of reuse and mutual value.

1.4. System Overview

The Swappify platform provides a user-friendly interface where users can list items, propose trades, and negotiate directly with others. The system includes:

- **User Authentication & Profile Management:** Ensures only legitimate users access the platform.
- **Item Listings & Filters:** Allows browsing and filtering of items by categories.
- **Trade Proposal System:** Enables users to initiate, accept, or reject barter offers.
- **Wishlist Functionality:** Lets users bookmark desired items for future reference.
- **Messaging Feature:** Provides a secure environment for real-time trade negotiation.
- **Rating & Review System:** Builds community trust through transparent feedback after trades.

Advanced features like admin controls, trade history logs, and email notifications enhance system reliability and user engagement. Swappify combines technology and sustainability to redefine the online trading experience.

2. Features

2.1. Login

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.

How to Use:

1. Navigate to the homepage and click on the "Login" button.
2. Enter your registered email and password in the login form.
3. Click the "Submit" button to proceed.
4. Wait while the system validates your credentials.
5. If the credentials are correct, you will be logged in.
6. You will be redirected to the Home page of the platform.
7. If the input is invalid, review the displayed error messages and try again.
8. If a system error occurs, wait and try again later.

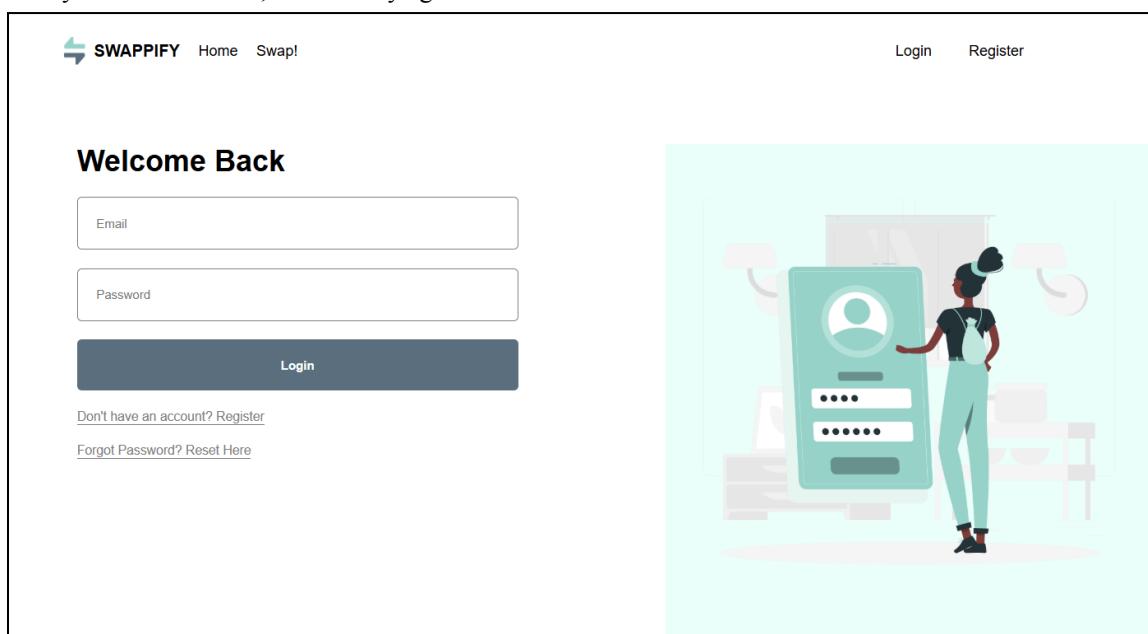


Figure 108. Login Page

2.2. SignUp

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.

How to Use:

1. Navigate to the home page of the platform.

2. Click on the "Sign-Up" button.
3. Enter your name, email, password, and mobile number in the sign-up form.
4. Click the "Submit" button.
5. If the email is already registered, an error message will be shown.
6. If the inputs are valid and the email is not in use, a verification email will be sent.
7. Open your email and click on the verification link.

Figure 109. Sign Up Page

2.3. Forgot Password

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.

How to Use:

1. Navigate to the login page of the platform.
2. Click on the "Forgot Password".
3. Enter your registered email address and mobile number in the form.
4. Click the "Submit" button.
5. If the email is registered, a password reset link will be sent to your email.
6. Open your email and follow the link to receive a new password.
7. Use the new password to log in to your account.
8. If the email is not registered or inputs are invalid, review the error message and try again

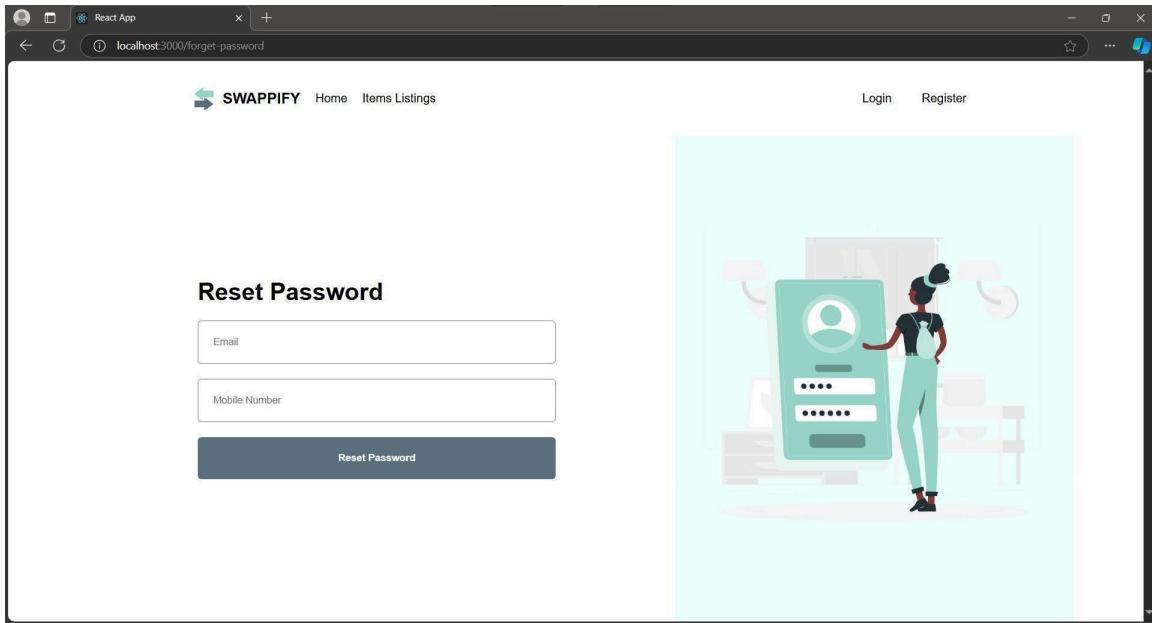


Figure 110. Forget Password Page.

2.4. Profile Management

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.

How to Use:

1. Log in to the platform with your valid account.
2. Click on the "Profile" button to navigate to your profile page.
3. Click the "Edit" option to enable profile editing.
4. Update the desired fields such as name, email, or mobile number.
5. Click the "Save" button to submit the changes.
6. If the input is valid, the system saves the new information and updates your profile.
7. If the input is invalid, review the error message and correct the information.

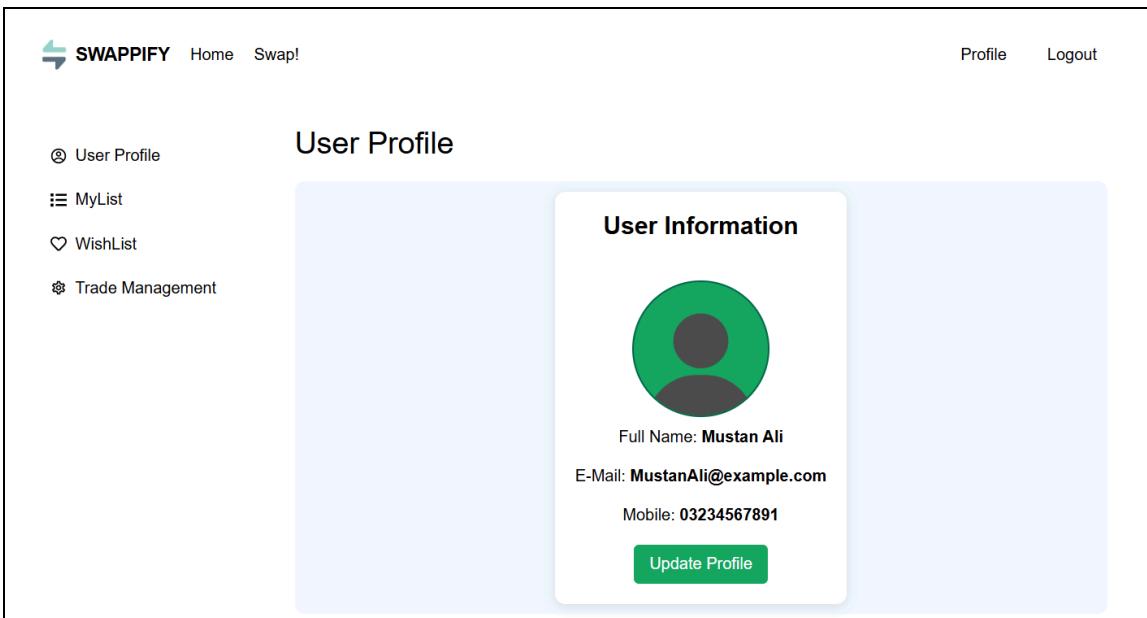


Figure 111. Profile Page

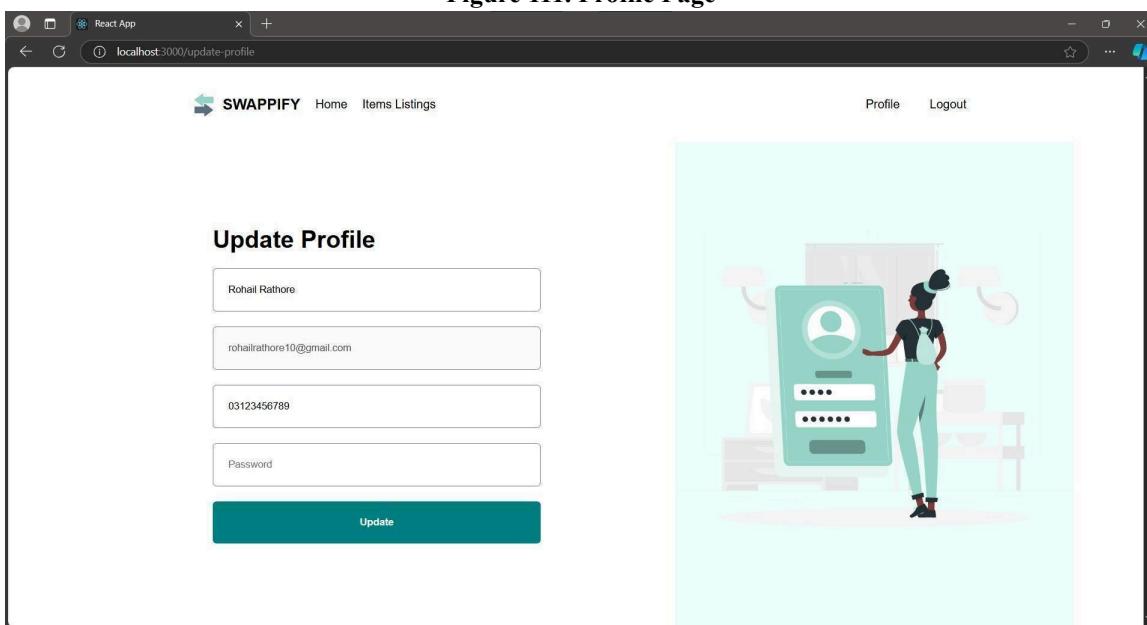


Figure 112. Update Profile Page

2.5. List Item

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.

How to Use:

1. Log in to your account on the platform.

2. Navigate to the "Add New Item" section.
3. Fill in the item details in the provided form, including uploading an image.
4. Click the "Submit" button.
5. If the input is valid, the system will save the item and display a confirmation message.
6. If the input is invalid, review the displayed error message and correct the form.

The screenshot shows a web browser window with the URL `localhost:3000/add-item`. The page has a header with the SWAPPIFY logo, a Home link, and a Logout link. Below the header, the title "Add New Item" is centered. The form consists of several input fields: "Item Name", "Item Description" (with a text area and an "x" button), "Condition", "Category", "Location", "Minimum Price", "Maximum Price", and a file input field labeled "Choose File" with the placeholder "No file chosen". At the bottom is a large blue "Add Item" button.

Figure 113. Add Item Page

2.6. Edit Item

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.

How to Use:

1. Log in to your account on the platform.
2. Navigate to your listed items section.
3. Select the item you want to edit.
4. View the item details displayed in an editable form.
5. Update the desired fields such as name, description, or image.
6. Click the "Save" button to submit the changes.
7. If the input is valid, the system will save the changes and display a confirmation message.
8. If the input is invalid, review the error message and correct the information.

The screenshot shows a web browser window titled 'React App' with the URL 'localhost:3000/edit-item/6714d022acb2d0f3e46fb13f'. The page is titled 'Edit Item' and features a form for updating an item listing. The item is identified as an 'Apple MacBook Pro' with the following details:

- Description:** Apple MacBook Pro 14-inch with M2 Pro chip, 16GB RAM, and 512GB SSD. Features a Liquid Retina XDR display, exceptional performance, and all-day battery life. Ideal for professionals and power users.
- Condition:** Used
- Category:** Electronics
- Location:** Lahore
- Original Price:** 25000
- Offer Price:** 35000
- Image:** A photograph of a silver Apple MacBook Pro laptop is displayed on the right side of the form.
- File Input:** A file input field labeled 'Choose File' with the placeholder 'No file chosen'.
- Buttons:** A large blue 'Edit Item' button at the bottom of the form.

Figure 114. Edit Item Page

2.7. Delete Item

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.

How to Use:

1. Log in to your account on the platform.
2. Navigate to your listed items section.
3. Select the item you want to delete.
4. Click the "Delete" button associated with the item.
5. Confirm the deletion if prompted.
6. If successful, a confirmation message will be displayed.
7. The item will be removed from your listings and deleted from the database.

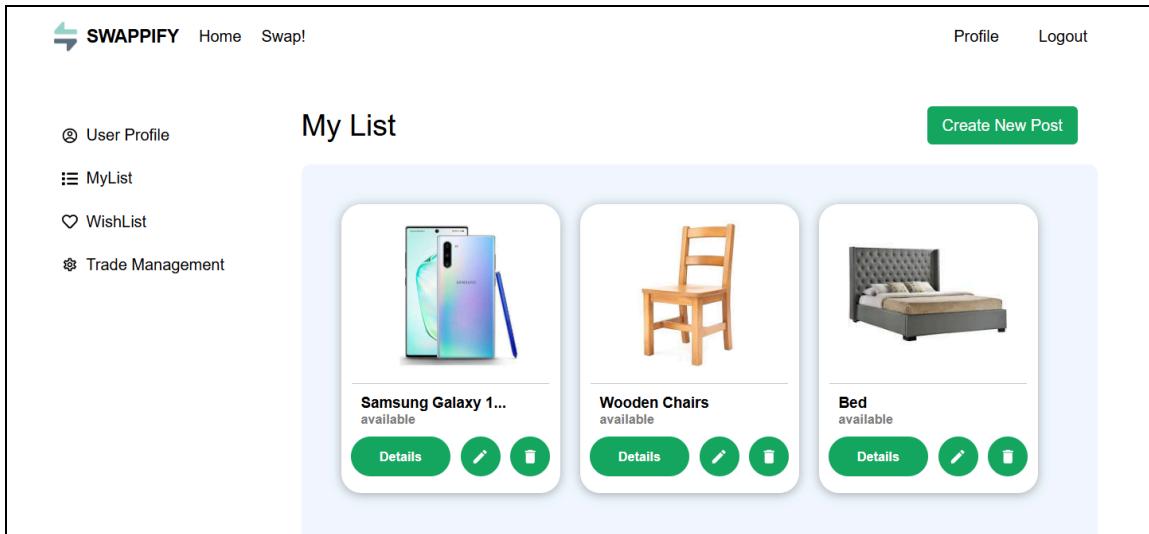


Figure 115. User Item Page

2.8. View Item Listing

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.

How to Use:

1. Navigate to the item listings page.
2. Browse through the list of available items.
3. Select the item you want to view.
4. The system will display the item's full details, including description, images, and trade options.
5. If no items are available, a message will be shown informing you.

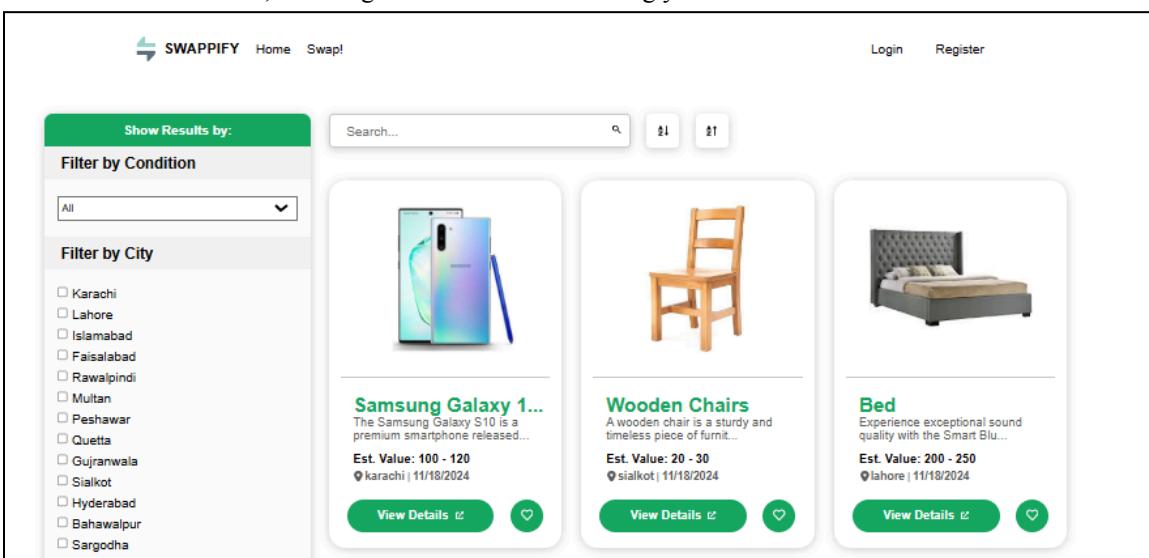


Figure 116. Listed Items Page

2.9. Search & Filter

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.

How to Use:

1. Navigate to the item listings page.
2. Enter a keyword in the search bar or select filter options such as category or condition.
3. Click the search button or apply the filters.
4. Wait while the system retrieves matching items.
5. Browse through the filtered list of items.
6. Select an item to view its detailed information.
7. If no items match your search or filters, a message will be displayed.

The screenshot shows the SWAPPIFY application's item listing page. At the top, there is a navigation bar with the SWAPPIFY logo, Home, Swap!, Login, and Register buttons. On the left side, there is a sidebar with several filter sections: 'Show Results by:' (dropdown set to 'All'), 'Filter by Condition' (dropdown set to 'All'), 'Filter by City' (checkboxes for Karachi, Lahore, Islamabad, Faisalabad, Rawalpindi, Multan, Peshawar, Quetta, Gujranwala, Sialkot, Hyderabad, Bahawalpur, Sargodha, Mardan, Swat), 'Filter by Category' (checkboxes for Electronics, Furniture, Clothing & accessories, Books & media, Home & garden, Sports & outdoors, Toys & games, Tools & hardware, Automotive, Office supplies, Collectibles & antiques, Other), and 'Filter by Price Range' (input fields for 'From' and 'To'). The main content area displays six items in a grid:

- Samsung Galaxy 1...**: A smartphone with a blue case. Description: "The Samsung Galaxy S10 is a premium smartphone released...". Est. Value: 100 - 120. Posted by karachi | 11/18/2024. Buttons: View Details, Heart.
- Wooden Chairs**: A wooden chair. Description: "A wooden chair is a sturdy and timeless piece of furniture...". Est. Value: 20 - 30. Posted by sialkot | 11/18/2024. Buttons: View Details, Heart.
- Bed**: A dark-colored bed frame with a headboard. Description: "Experience exceptional sound quality with the Smart Blu...". Est. Value: 200 - 250. Posted by lahore | 11/18/2024. Buttons: View Details, Heart.
- Dell Inspiron 15...**: A silver laptop. Description: "The Dell Inspiron 15 Laptop offers a perfect blend of p...". Est. Value: 200 - 250. Posted by hyderabad | 11/18/2024. Buttons: View Details, Heart.
- thousand splendi...**: A book cover for "A Thousand Splendid Suns". Description: "A Thousand Splendid Suns is a powerful and emotional no...". Est. Value: 15 - 20. Posted by karachi | 11/18/2024. Buttons: View Details, Heart.
- Casio Calculator**: An electronic calculator. Description: "The Casio Scientific Calculator is a versatile and reli...". Est. Value: 45 - 80. Posted by lahore | 11/18/2024. Buttons: View Details, Heart.

Figure 117. Listed Items Page

2.10. Add to Wishlist

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.

How to Use:

1. Log in to your account on the platform.
2. Navigate to the item listings page.
3. Find the item you want to save for later.
4. Click the "Add to Wishlist" button associated with the item.
5. Wait for the confirmation message: "Item added to wishlist."

2.11. View Wishlist

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.

How to Use:

1. Log in to your account on the platform.
2. Navigate to your wishlist section via your profile.
3. Wait while the system loads your saved wishlist items.
4. Browse through the list of items saved in your wishlist.
5. Click on any item to view its full details, including owner info and trade options.
6. If your wishlist is empty, a message will be displayed: "Your wishlist is empty".

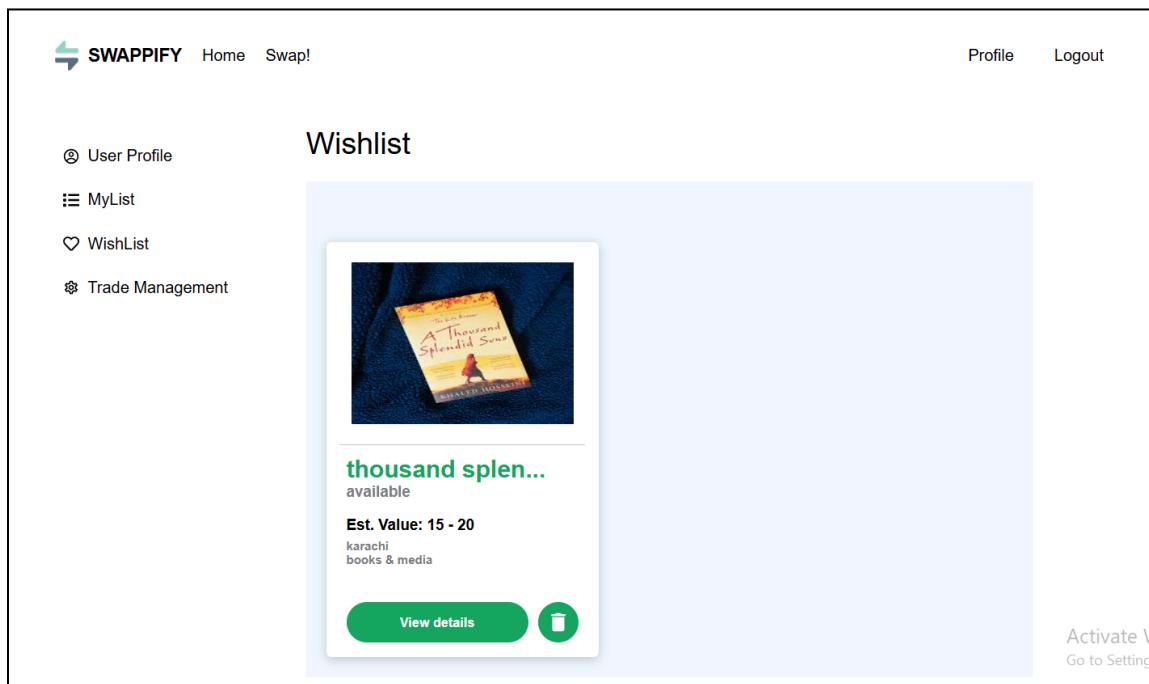


Figure 118. Wishlist Page

2.12. Item Detail

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.

How to Use:

1. Log in to your account on the platform.
2. Navigate to the item listings page or your wishlist.
3. Click on the item you want to view in detail.
4. Review the item details.

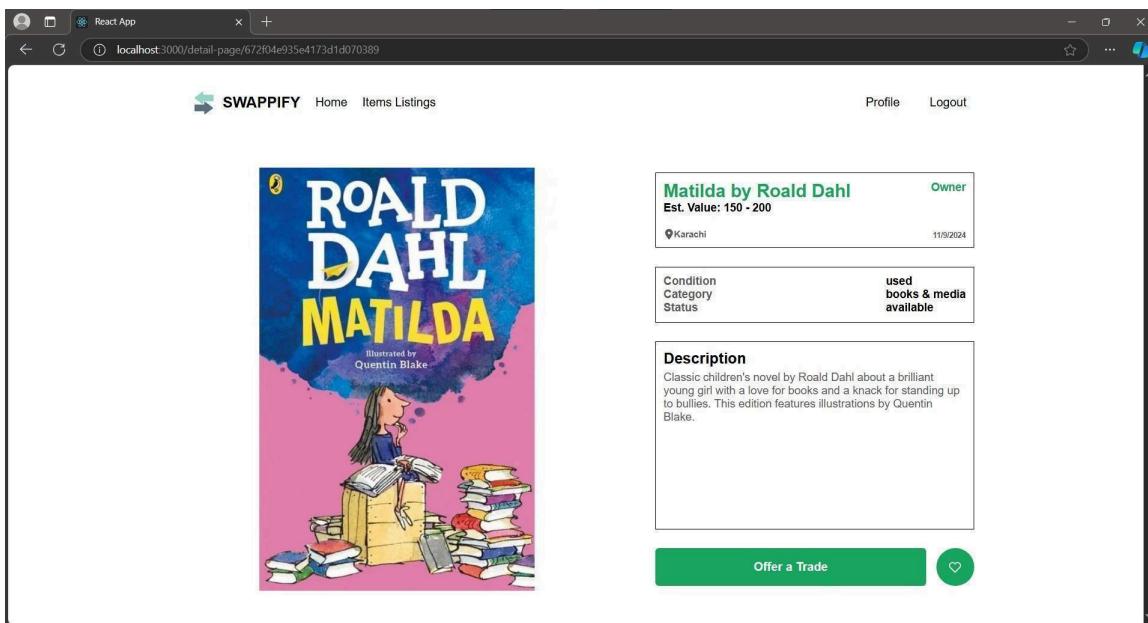


Figure 119. Item Detail Page

2.13. Trade Proposal

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.

How to Use:

1. Log in to your account on the platform.
2. Navigate to the item listing you want to trade for.
3. Click the "Offer Trade" button on the item page.
4. Select one of your own items to offer in exchange.
5. Click the "Submit" button to send the trade proposal.
6. Wait for the confirmation message: "Trade proposal sent."
7. If you change your mind, click cancel before submitting to abort the proposal.

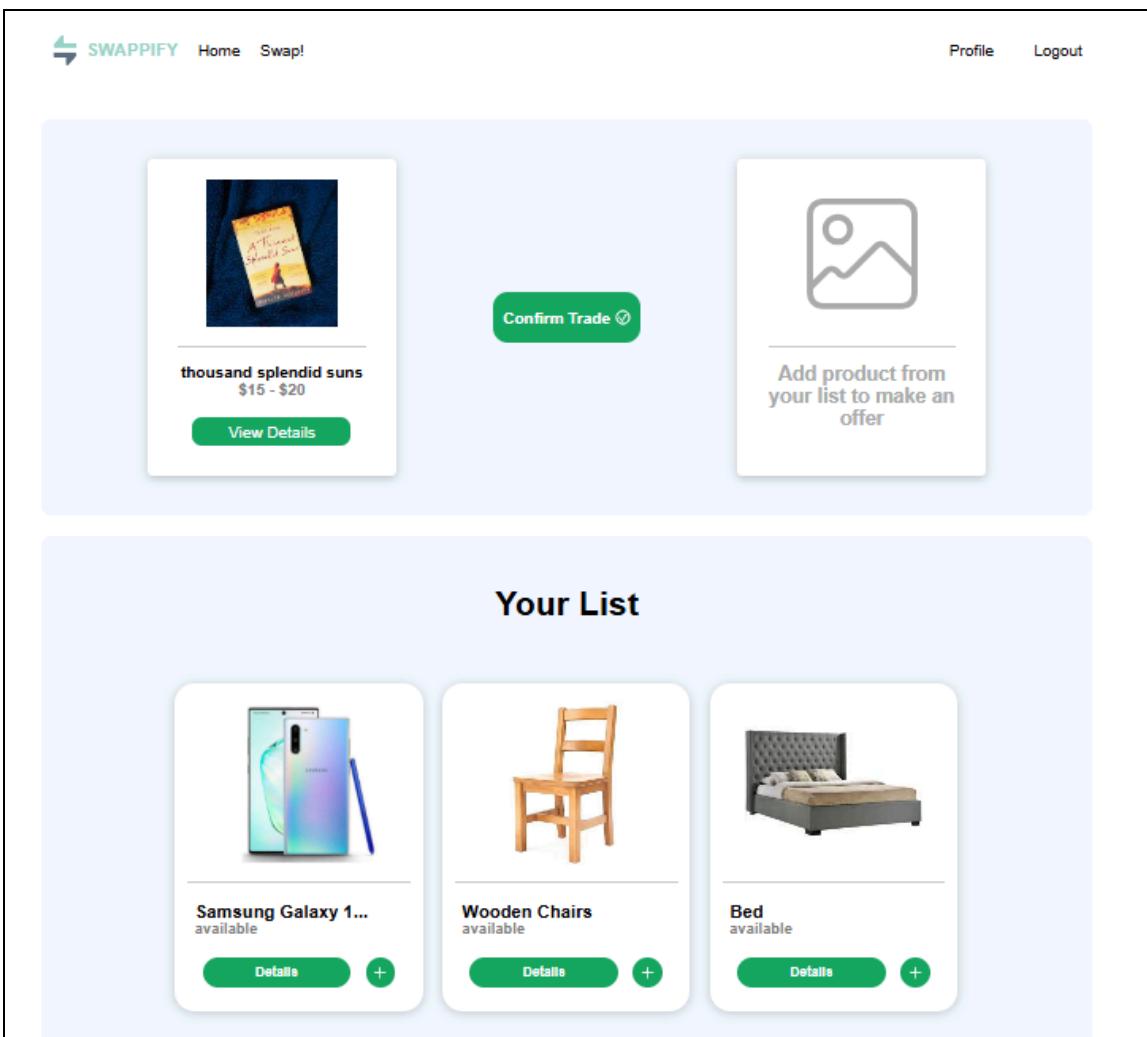


Figure 120. Trade Proposal Page

2.14. Accept Trade

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.

How to Use:

1. Log in to your account on the platform.
2. Navigate to your trade proposals section.
3. Select a pending trade proposal to review its details.
4. Click the "Accept" button to confirm the trade.
5. Wait for the confirmation message: "Trade accepted."
6. The other user will be notified of your acceptance.

2.15. Reject Trade

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.

How to Use:

1. Log in to your account on the platform.
2. Navigate to your trade proposals section.
3. Select a pending trade proposal to review its details.
4. Click the "Reject" button to decline the trade.

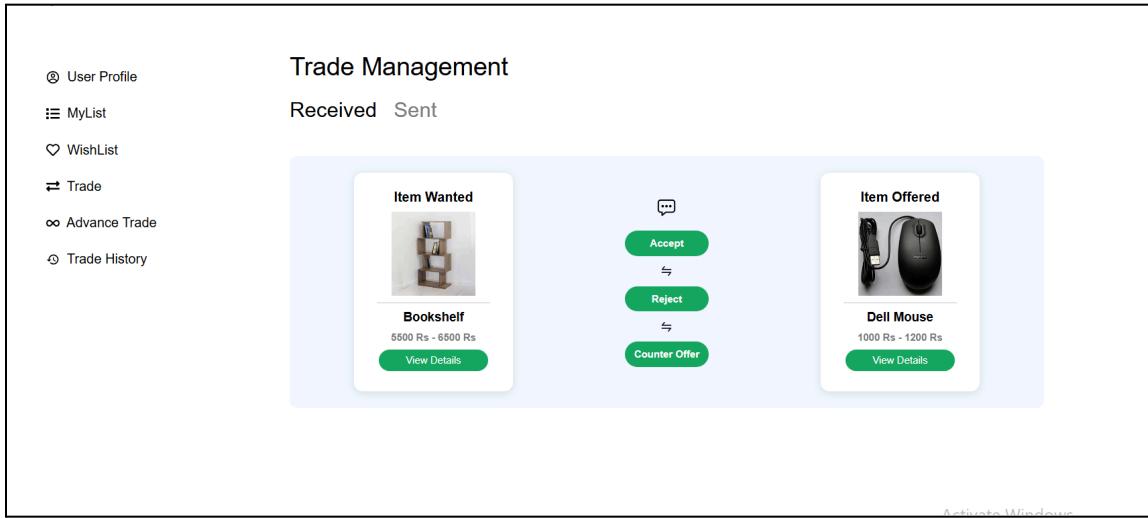


Figure 121. Trade Management: Received Request Page

2.16. Cancel Trade

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.

How to Use:

1. Log in to your account on the platform.
2. Navigate to the "Sent Requests" section in your trade proposals.
3. Select the active trade proposal you want to cancel.
4. Click the "Cancel Trade" button associated with the selected trade.
5. Confirm the cancellation when prompted.

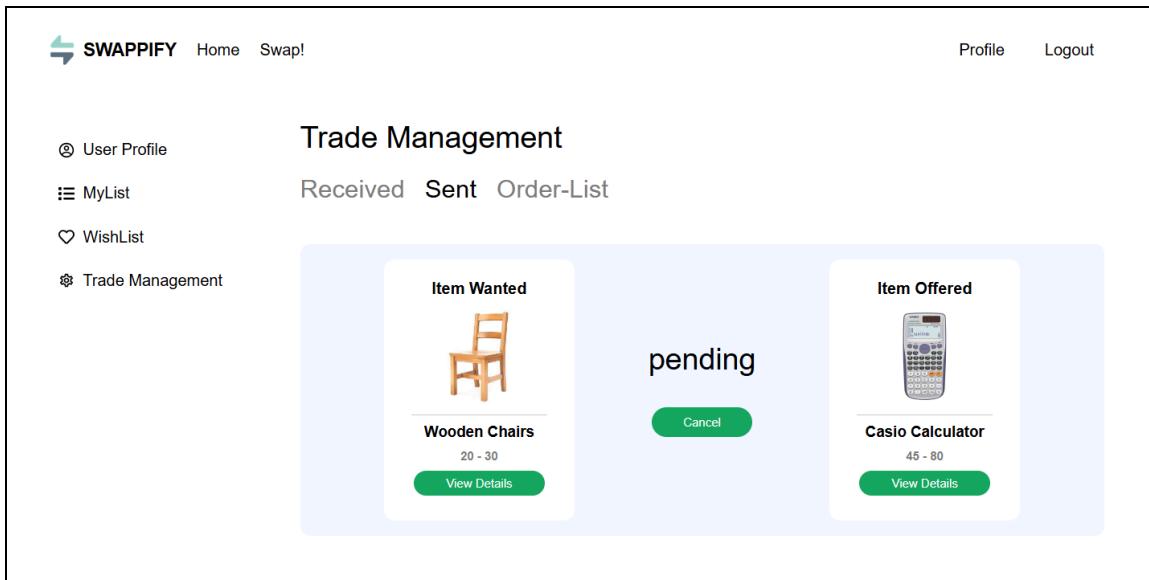


Figure 122. Trade Management: Sent Request Page

2.17. Manage Active Trade

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.

How to Use:

1. Log in to your account on the platform.
2. Navigate to the "Trades" section from your profile.
3. The system will display a list of all active trade proposals.
4. Click "Accept" to confirm the trade or "Reject" to decline it.
5. The system will update the trade status and notify the other party.
6. If there are no active trades, the system will show: "No active trades."

2.18. Remove from Wishlist

Description:

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

How to Use:

1. Log in to your account.
2. Navigate to your Wishlist by selecting it from your profile or the main menu.
3. Browse the list of saved items.
4. Locate the item you wish to remove.
5. Click the "Remove" button associated with the item.
6. The system will update your wishlist.

- The item will no longer appear in your wishlist.

2.19. Advanced Trade Management

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.

How to Use:

- Log in to your Swappify account.
- Navigate to the "Active Trades" section.
- Locate the trade proposal you wish to counter and click the "Counter Offer" button.
- The trade proposal page will open with your originally proposed item pre-selected.
- Browse the sender's item list and select items you'd like in return.
- Click "Send Counter Offer".
- Notify the sender.
- Show confirmation: "Counter offer sent successfully."

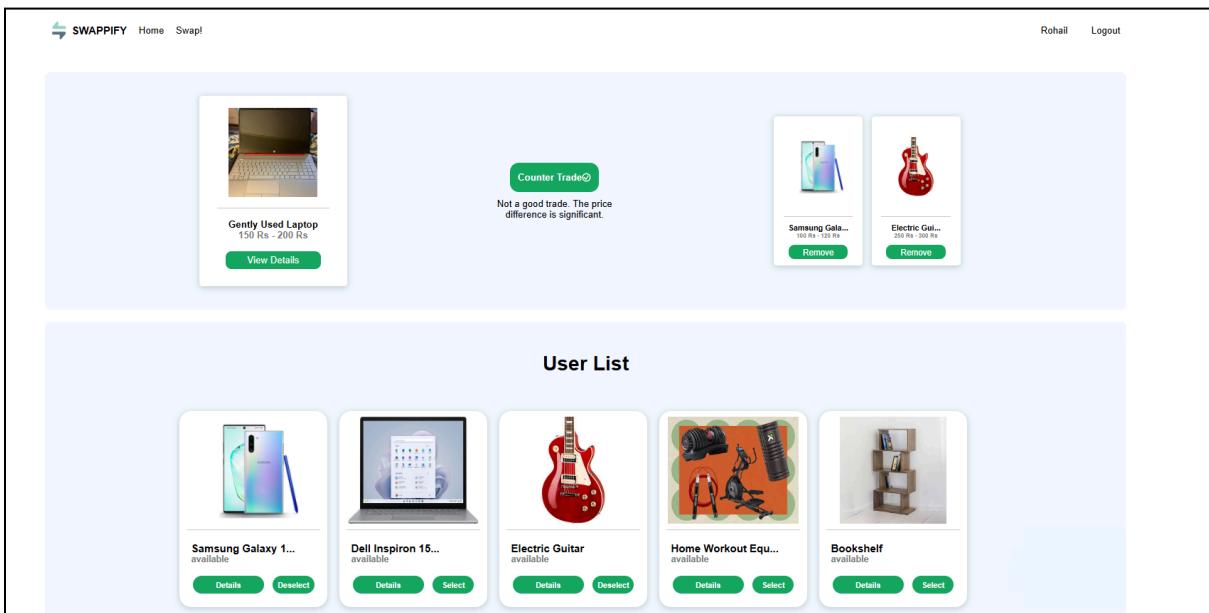


Figure 123. Advance Trade: Sent Offer

2.20. Ban User Accounts

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.

How to Use:

- Log in to the Admin Portal with the required permissions.

2. Navigate to the "User Accounts" section.
3. Review the list of all user accounts displayed with their account status and profile information.
4. Select the user account you want to ban.
5. Click the "Ban" button next to the selected account.

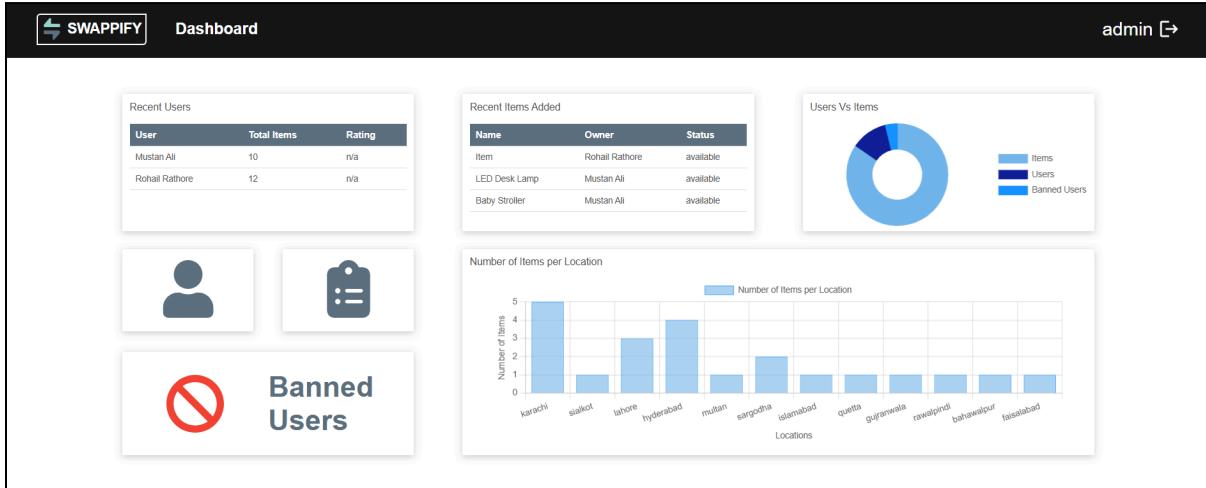


Figure 124. Admin Dashboard

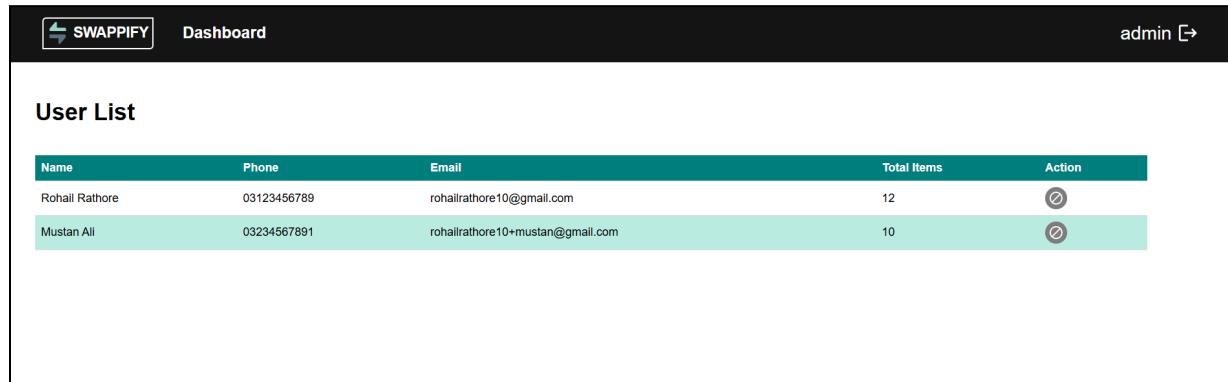


Figure 125. Admin Dashboard: Users List

2.21. Delete List Items

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.

How to Use:

1. Log in to the Admin Portal with the required permissions.
2. Navigate to the "Listed Items" section.
3. Review the list of all user-listed items with their details (title, description, category, and status).
4. Select the item you want to remove.
5. Click the "Remove" or "Delete" button for that item.
6. Wait for the system to process the deletion.

Name	Owner	Condition	Category	Location	Price (PKR)	Status	Date Added	Action
Samsung Galaxy 10		used	electronics	karachi	100 - 120	available	11/18/2024	
Wooden Chairs		new	furniture	sialkot	20 - 30	traded	11/18/2024	
Bed		new	furniture	lahore	200 - 250	traded	11/18/2024	
Dell Inspiron 15 Laptop		used	electronics	hyderabad	200 - 250	available	11/18/2024	
thousand splendid suns		used	books & media	karachi	15 - 20	available	11/18/2024	
Casio Calculator		new	office supplies	lahore	45 - 80	traded	11/18/2024	
Dell Mouse		new	electronics	multan	15 - 25	available	11/18/2024	
Iphone 13		new	electronics	karachi	120 - 230	traded	12/27/2024	
Art		new	collectibles & antiques	karachi	50 - 75	traded	1/27/2025	

Figure 126. Admin Dashboard: Items List

2.22. Rate and Review

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.

How to Use:

1. Log in to your account after completing a trade.
2. Navigate to the "Rate Trade" section from your dashboard or trade history.
3. Fill out the rating form by selecting a star rating (e.g., 1 to 5 stars).
4. Write your review about the trade experience.
5. Click the "Submit" button to send your feedback.

This trade is requested by: Mustan Ali
Total Wanted Items: 2

★★★★★

Write your review here...

Submit

Item Offered

Item Wanted

Figure 127. Trade Review Page

2.23. Email Notification

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.

How to Use:

1. Ensure your email address is registered and verified in your account settings.
2. Perform or wait for a relevant activity (receive trade proposal, trade accepted, or account update).
3. Check your email inbox for a notification from the platform.

2.24. Trade History

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.

How to Use:

1. Log into your account on the platform.
2. Navigate to the "Trade History" section from the main menu or user dashboard.
3. View the list of your trades, including item details, trade partners, and dates.
4. Select a specific trade from the list to view more detailed information about that trade.
5. Review the full trade details including items exchanged, counterparties involved, and trade status.

The screenshot shows the TradeHistory page of the SWAPPIFY platform. At the top, there's a navigation bar with a logo, 'Home', 'Swap!', 'Rohail' (user profile), and 'Logout'. On the left, a sidebar lists navigation options: User Profile, MyList, WishList, Trade, Advance Trade, and Trade History. The main area is titled 'TradeHistory' and displays two completed trades in a grid format.

Trade Details	Date	Action Buttons
 Handmade Wooden ... <i>Rohail Rathore</i> Request Type: Sent Trade type: Normal Status: cancelled Price Range: 70 Rs - 90 Rs	4/6/2025	Item details Trade details
 Wooden Chairs <i>Mustan Ali</i> Request Type: Received Trade type: Counter Status: accepted Price Range: 20 Rs - 30 Rs	4/6/2025	Item details Trade details

Figure 128. Trade History

2.25. Send Message

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.

How to Use:

1. Log into your account on the platform.
2. Navigate to an active trade or select a user you want to message.
3. Open the messaging interface where you can compose your message.
4. Type your message in the text box.
5. Click the "Send" button to deliver your message to the recipient.

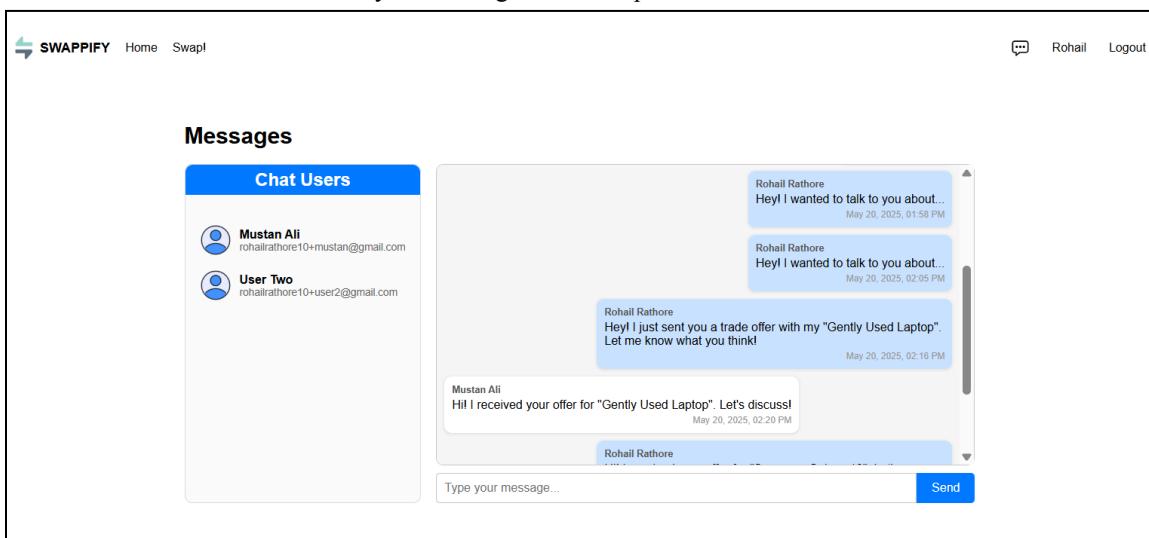


Figure 129. Message System

References

- Geeksforgeeks. <https://www.geeksforgeeks.org/what-is-agile-methodology>
- 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>
- Socket.IO Documentation <https://socket.io/docs/v4/>
- JSON Web Token (JWT) <https://www.npmjs.com/package/jsonwebtoken>
- Multer <https://www.npmjs.com/package/multer>