Software Requirements Specification

for

Re_Store

Version 1.0

Prepared by

Group 4 **Group Name: Divide and Conquer**

Bharatula Anirudh Srivatsa	230290	bharatula23@iitk.ac.in
Voora Rakesh	231174	vrakesh23@iitk.ac.in
Soma Koushik	231018	koushiks23@iitk.ac.in
Sanjay Raghav Vangala	230916	sanjayra23@iitk.ac.in
Yashwanth Reddy Junutula	231194	yashwanth23@iitk.ac.in
Saatvik Gundapaneni	230428	gundapanen23@iitk.ac.in
Vempati Prem Santhosh	231137	psanthosh23@iitk.ac.in
Ayush Yadav	230272	ayushydv23@iitk.ac.in
Meghana Kadari	230512	kmeghana23@iitk.ac.in
Chapati Venkata Pranaya	230324	chapative23@iitk.ac.in

Course: CS253
Mentor TA: Jeswaanth Gogula
Date of Submission: 24th January 2025

CONTENTS

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Initial Draft (v1.0)		Information about the revision. This table does not need to be filled in whenever a document is	00/00/00

Version	Primary Author(s)	Description of Version	Date Completed
Number		Touched, only when the version is being upgraded.	

1 Introduction

1.1 Product Scope

The project aims to digitalize used product sales and purchases within the campus community. The platform is set to have three main features: a straightforward marketplace for transactions such that sellers may hold an auction for multiple buyers interested in a single item, a product request where a buyer can request items that are not currently listed on the market, and a secure payment gateway. On one hand, the software serves as a simple and safe way for people within the campus to exchange goods. On the other hand, it acts as a community-driven marketplace that gives access to cheaper items while encouraging the reuse of goods.

We also aim to develop this website so that sharing personal details is completely secure. Re_Store will feature a messaging system that enables users to communicate directly within the platform to facilitate smooth transactions.

1.2 Intended Audience and Document Overview

This document is for the following audiences:

Users (Students, Faculty, and Staff): The campus community—students, faculty, and staff—needs to feel comfortable using the platform to buy, sell, and take part in auctions. Whether someone wants to sell used items, find great deals, or request specific products, the platform should feel intuitive and easy to navigate. Clear, simple instructions will help everyone make the most of its features, whether it's bidding in an auction or saving a product for later. Ultimately, the goal is for users to see how the platform can fit seamlessly into their daily lives, offering convenience and a sense of trust with secure transactions.

Marketing Team: The marketing team needs to understand what makes the platform stand out fully—its features like auctions, product listings,product requests and product delivery. This knowledge helps them craft messages that connect with the campus community. By highlighting the platform's benefits, they can create campaigns encouraging students, faculty, and staff to use it regularly, making it a go-to place for buying, selling, and finding what they need on campus.

Developers: Developers need to dive deep into the platform's design to see how all its features—like auctions, product requests, messaging system —work together smoothly. They're responsible for ensuring everything runs well, from secure payments to easy-to-use interfaces. Their job is to build a platform that meets today's needs and can grow and adapt as the campus community continues to use it, all while maintaining security, speed, and a great user experience.

Document Overview:

- **Introduction:** Describe the scope of the product, the target audience, and definitions.
- **General Description**: Summary of product functionality, constraints, and assumptions.
- **Specific Requirements**: List of external interfaces, functional needs, and use cases.
- **Non-Functional Requirements**: Performance, security, and quality attributes.
- **Other Requirements**: Additional constraints and compliance details.
- **Appendices**: Data dictionary and group activity log.

1.3 Definitions, Acronyms and Abbreviations

Refer to the following table for definitions of a few terms we used.

Term	Definition
API	Application Programming Interface
Database	Collection of all the information monitored by this system
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
IITK	Indian Institute of Technology, Kanpur
OTP	One Time Password
SRS	Software Requirements Specification
UI	User Interface
UML	Unified Modeling Language

1.4 Document Conventions

● Arial font size 12 has been maintained throughout the text.
● Single spacing is employed throughout the text, and a consistent 1’ margin is maintained across the document.
● Italics have been used for writing comments and highlighting.

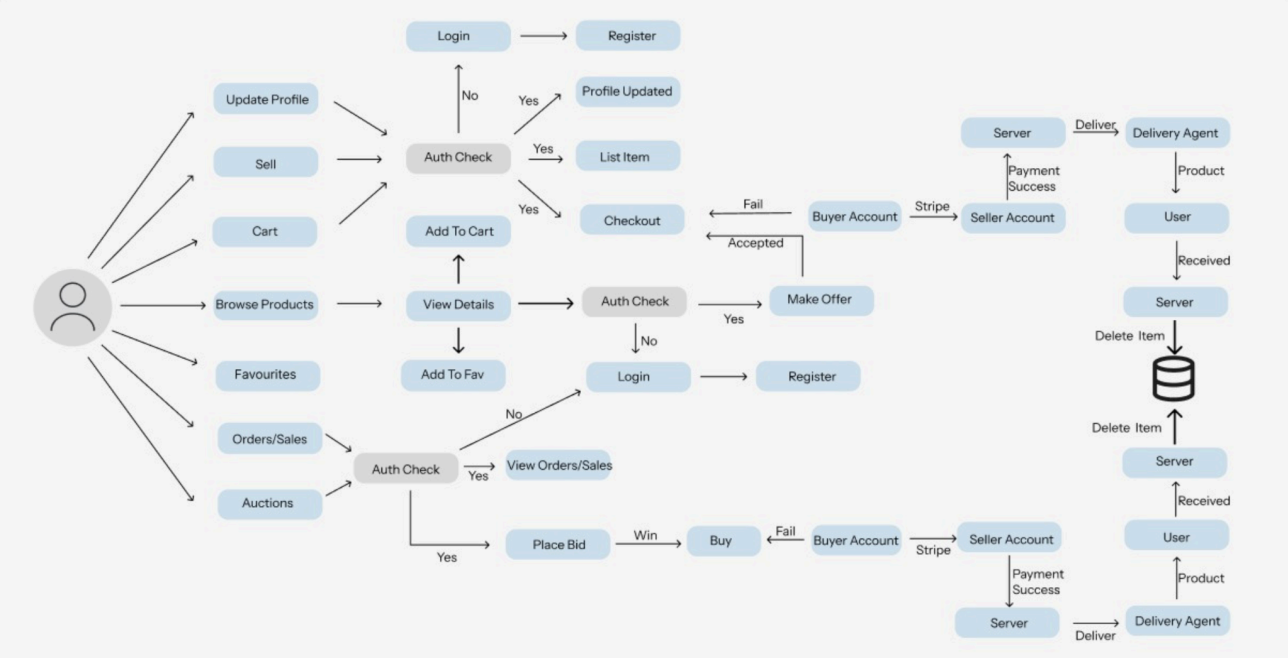
1.5 References and Acknowledgments

1. We designed the UML and use case diagrams with *Figma*.
2. For UI diagrams, we used *Canva*.
3. Used Database icon from figma for database representation.
4. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.
5. We'd like to thank Prof. Indranil Saha for providing the SRS template, teaching us about software development, and guiding us through. We'd also like to thank our TA, Mr Jeswaanth, for his valuable assistance in developing this document.

2 Overall Description

2.1 Product Overview

Re_Store is a user-friendly web platform designed for our campus community to facilitate buying and selling used items easily and securely. It also includes an auction-based interface for sellers to sell their products to interested buyers. Recognizing buyer's unique needs and budget constraints, our platform provides a dedicated and interactive space for buying and selling items essential for academic life. Whether they are reference books, lab equipment, or room essentials, students can easily navigate through categories, connect with peers, and make affordable transactions. This platform allows sellers to sell their items at a reasonable price and buyers to use reused items rather than going for new ones. This creates a supportive community by encouraging sharing and helps students manage their budgets more effectively.



2.2 Product Functionality

The platform includes the following features to enhance the user experience and create a dynamic marketplace within the campus community:

● Registration using the campus email address with OTP verification.

● The product details include a photo, price, and usage duration.

● Product filtering using keywords.

● Allow users to add products to their favorites/cart for easy access and future purchase.

● Messaging platform for negotiations between buyers and sellers.

● Allow users to rate their experience with buyers or sellers, helping build trust and transparency.

● User profile showing their listed products for sale and purchased items.

● Sellers can host auctions for their products, allowing buyers to bid and compete for items.

● Buyers can post requests for specific products, and sellers can make efforts to make those products available.

2.3 Design and Implementation Constraints

2.3.1 Memory Requirements

● The application's memory usage should be optimized, especially when dealing with large datasets in MongoDB.

● Efficient memory management is crucial to prevent performance bottlenecks and ensure smooth operation, particularly during high user activity periods.

2.3.2 Timing Requirements

● Users expect quick response times. Any delays in loading pages or processing user requests can lead to a poor user experience.

2.3.3 Security Considerations

● Usage of IITK email-based authentication for user registration to ensure that only legitimate users can register into the system.

● We will store the **salted hash** of the passwords of the users to make sure their passwords are more secure.

● All payment transactions will be processed through a **secure payment gateway**, ensuring encryption and protection of sensitive financial information.

● Messaging between buyers and sellers will be conducted through a **secure, encrypted communication channel** to prevent unauthorized access or tampering.

2.3.4 Tools, Languages and Databases

● We plan to use JavaScript for both the backend (Node.js with Express.js) and frontend (React) which impose language constraints.

● MongoDB is used to handle the large amount of data associated with the users.

● The Fetch API is to be used for communication with the backend

● We plan to ensure that the website uses all the required software needed to make it user-friendly, responsive, and portable GUI

2.3.5 Communication protocols

● Implementing WebSockets or other efficient communication protocols for real-time features like the buyer and seller chat.

● To ensure secure communication between the servers, we plan to use HTTPS protocol.

2.3.6 Conventions and Programming Requirements

● Since the software would be maintained by people other than the developers, the Object Oriented Programming Paradigm will enhance the software’s maintainability.

2.4 Assumptions and Dependencies

2.4.1 Assumptions

● We assume that the information and images in the profiles accurately represent the actual condition and appearance of the products, i.e., the products listed are authentic and in working condition.

● We assume that sellers and service providers will respond to booking and product requests, as well as customer services as soon as possible.

● We assume that users have reliable and high-speed internet access.

● We assume that users trust the platform's security measures for sharing their personal information.

● We assume that the user uploads each product only once.

● We assume that users will abide by the platform's terms and conditions, including refraining from uploading prohibited or illegal content.

● Users will handle payments securely, following the recommended practices outlined by the platform.

● Sellers will ensure auctions are conducted fairly, and buyers will bid responsibly.

2.4.2 Dependencies

● JWT Library/Framework: The project depends on the chosen JWT library or framework. Changes or issues with this library could affect the authentication and authorization mechanisms in the application.

● Database systems: The project relies on a specific database system, MongoDB. Database systems provide a structured way to organize and store data. It is also very crucial for the management of data.

● ReactLibrary/Framework: The project’s frontend part will be developed using the React ecosystem, including React DOM, React Router, React Hooks, etc. The project relies on React for building user interfaces and managing component-based architecture.

● Node.js: The project depends on Node.js as the runtime environment for the backend. It enables JavaScript to run on the server, providing scalability and flexibility for handling requests and processing data.

● Express Framework: Express is used to build the server-side application. It is essential for handling routing, middleware, and API endpoints that connect the frontend with the backend logic and the database.

● Mongoose: A MongoDB ODM (Object Data Modeling) library used to interact with MongoDB. It simplifies database operations with schema-based modeling and validation.

● Fetch API: For making HTTP requests from the frontend to the backend.

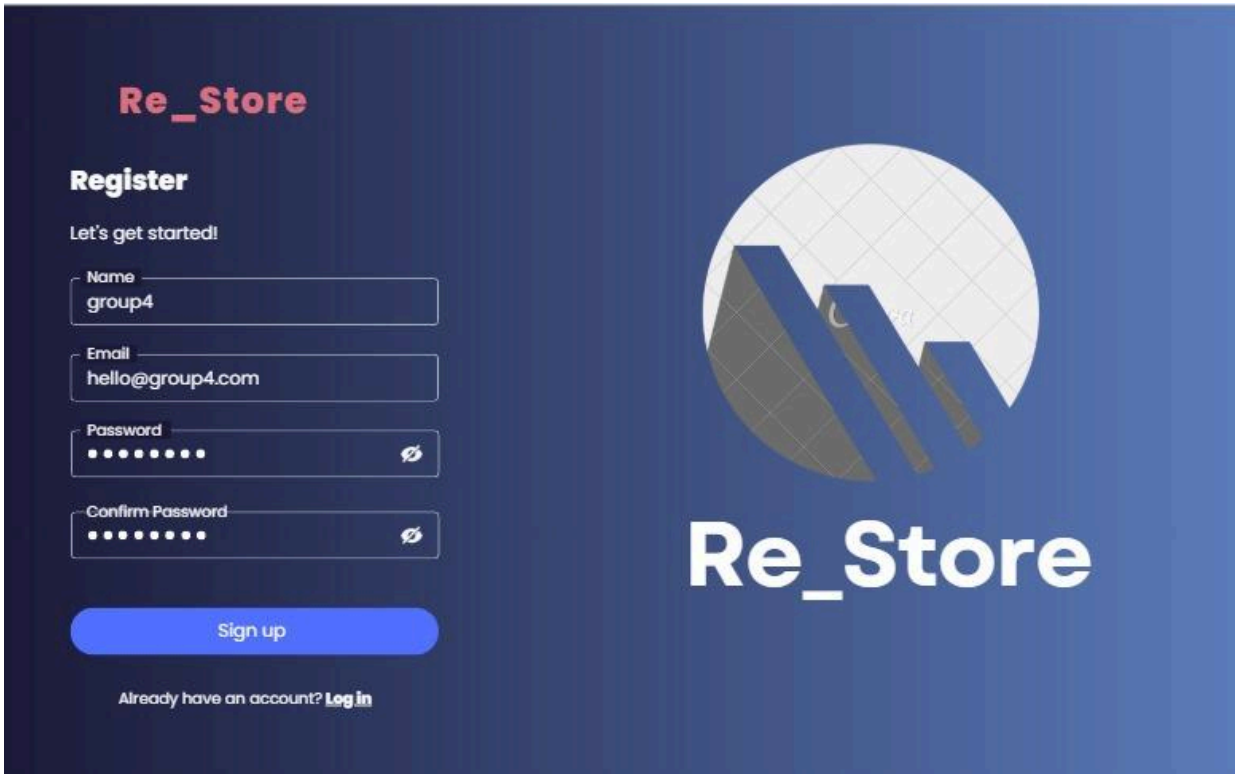
3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

Users can access the system through the website. To use the platform, they must create an account using their IITK email ID. After creating an account and logging in, users can access various web pages where they can engage in activities such as selling or buying used goods, chatting with sellers or buyers, and more. Additionally, users can manage their profiles, view their transaction history, and receive notifications about updates on their listings and communications.

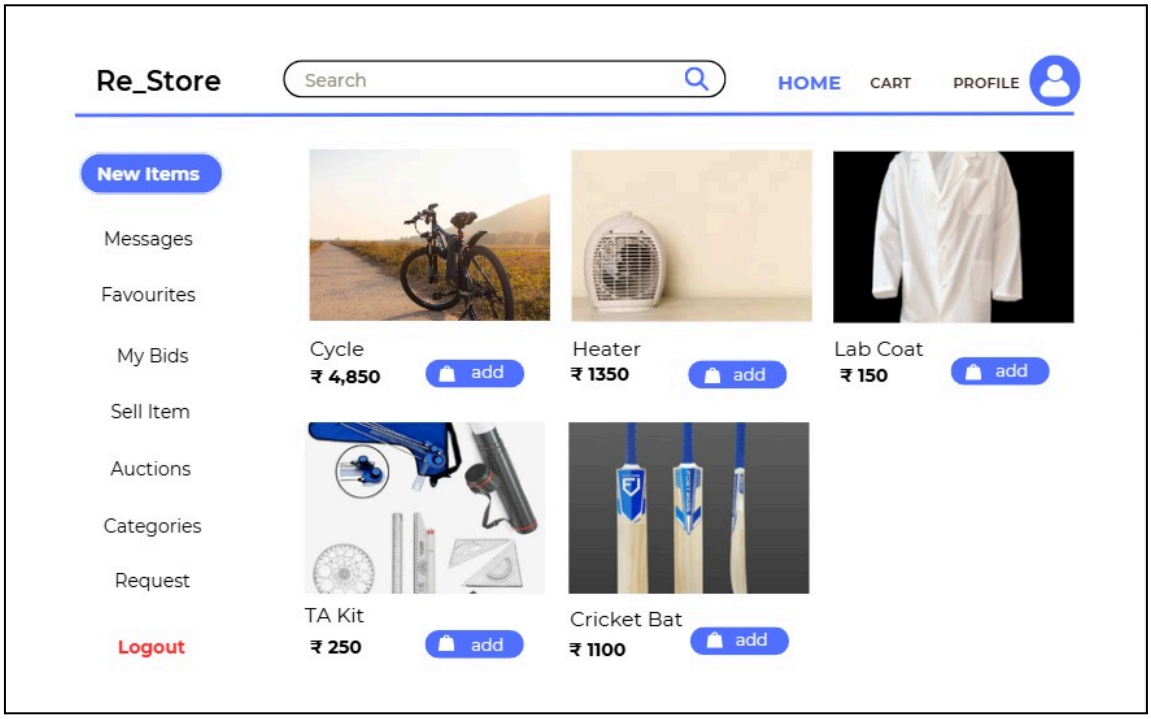
- **Create a new account page**



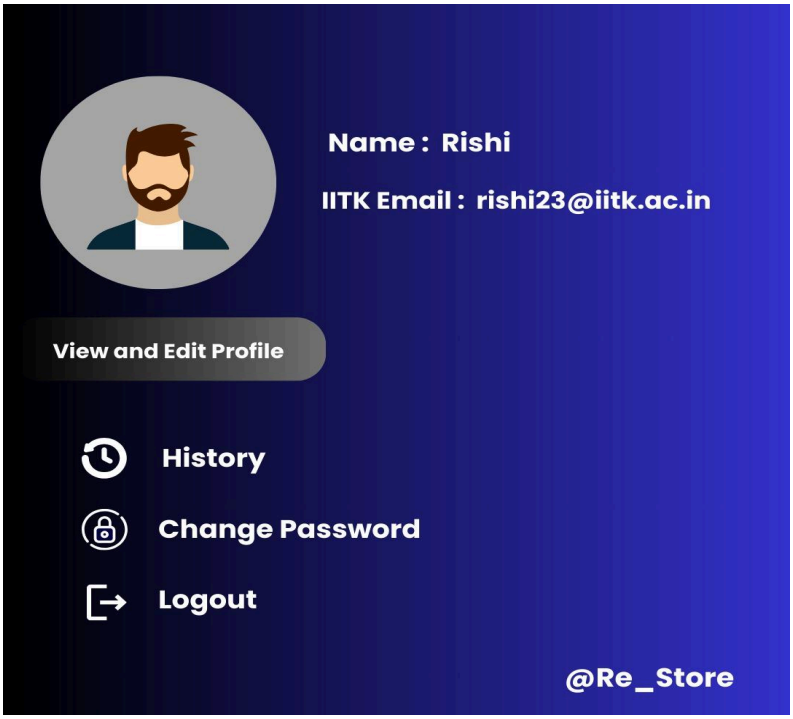
- **Login Page**

- **Forgot Password Page**

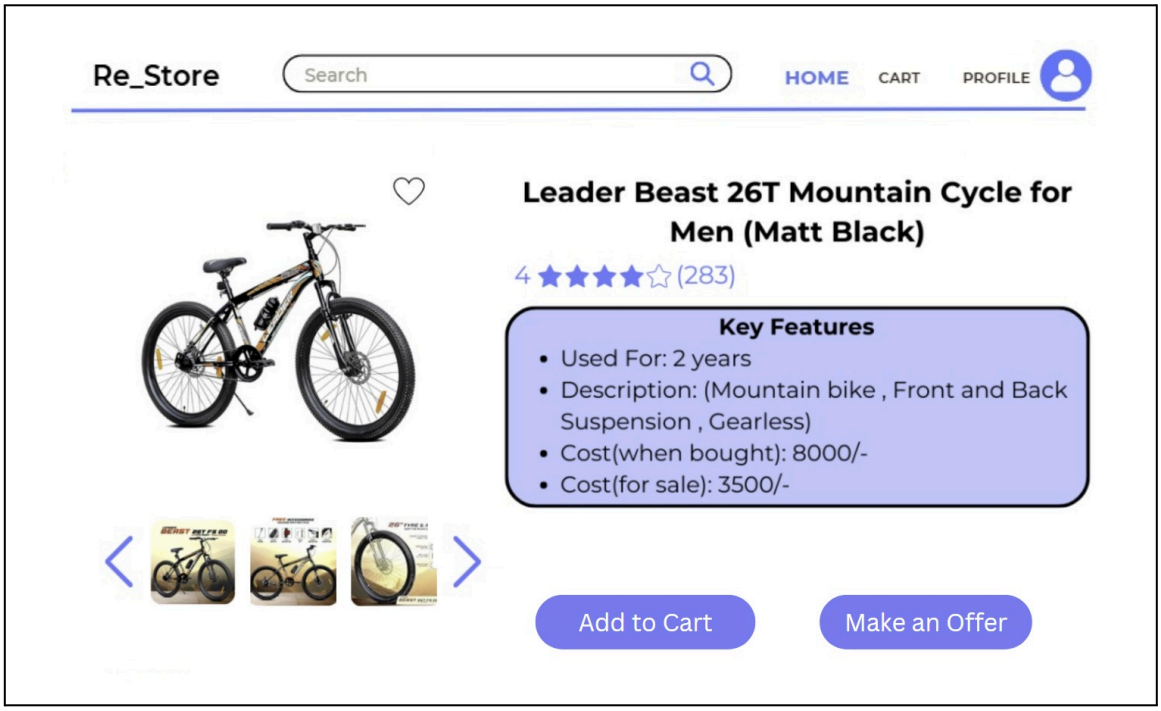
- **Home Page**



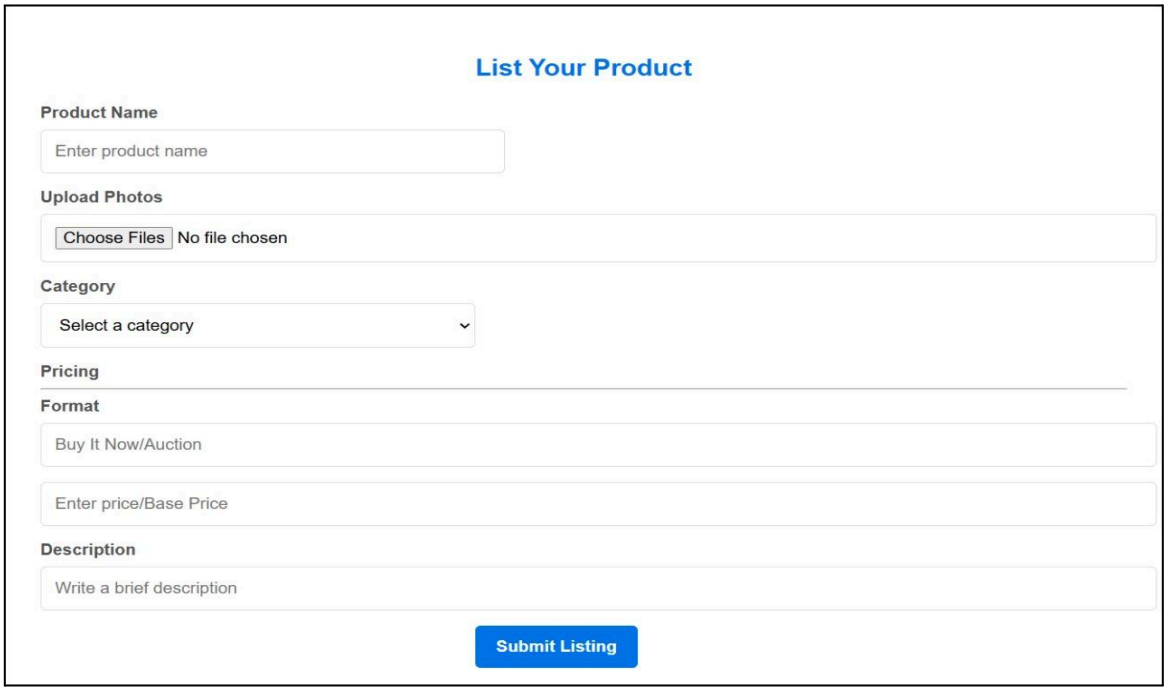
- **Profile Page**



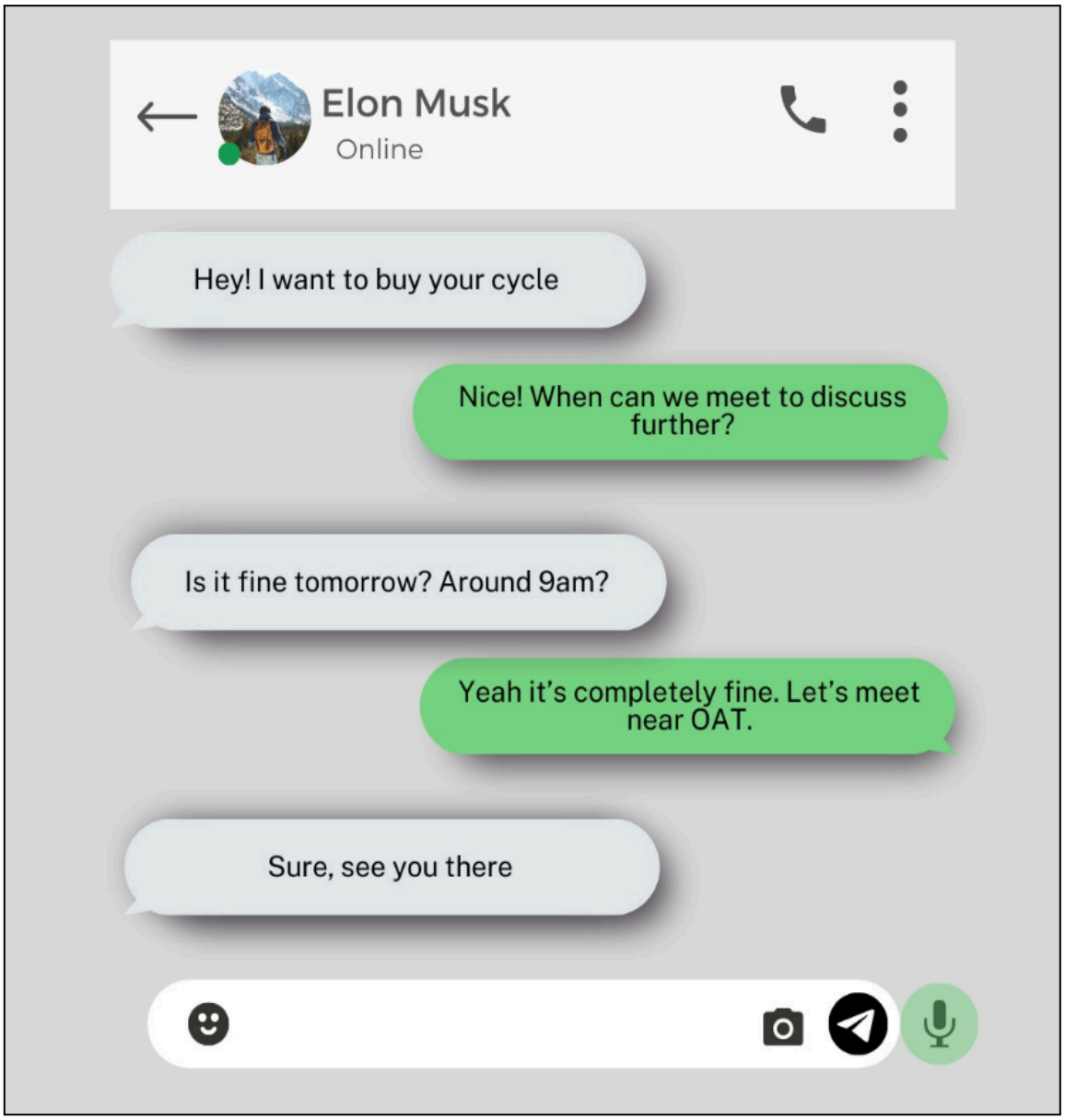
- **Product view Page**



- **Sell Page**



- **Chat Page**


Elon Musk
Online
Hey! I want to buy your cycle
Nice! When can we meet to discuss further?
Is it fine tomorrow? Around 9am?
Yeah it's completely fine. Let's meet near OAT.
Sure, see you there

3.1.2 Hardware Interfaces

- The software interacts with web servers to host the website, handling HTTP requests and responses efficiently. This ensures smooth user interactions, data retrieval, and seamless browsing experiences across different pages.
- Users can access the platform using a smartphone or computer with a modern web browser. The website's responsive design ensures a smooth and accessible user interface on both desktop and mobile devices.

3.1.3 Software Interfaces

The software is built with three key components: front-end, back-end, and database.

- **Front-end**: This is the user interface that users directly interact with. It includes everything users see, such as buttons, forms, and the overall layout, providing a user-friendly experience.
- **Back-end**: The back-end handles the server-side logic and processes user requests from the front end. It also manages application functionality and data processing and ensures that everything runs smoothly behind the scenes.
- **Database**: The database is responsible for storing and retrieving all the application's data. It keeps track of user profiles, product listings, orders, chats, request list and transaction history ensuring that data is securely stored and easily accessible.

3.2 Functional Requirements

3.2.1 User Registration

All users are required to sign up on the application by providing a username, and their IITK email address, and setting a secure password.

3.2.2 Email Verification

The application will verify the user’s email by sending a one-time password (OTP) to their registered email address.

3.2.3 User Verification

The application restricts registration to email addresses with the IITK domain, ensuring access is limited to members of the campus community.

3.2.4 User Profile

The application enables users to manage their profiles by updating their contact details and uploading profile pictures and passwords. Users can also view their sold and purchased items and their currently listed products for sale.

3.2.5 Listing Products for Sale

Users can upload products they want to sell, providing details such as the product's name, images, description, price, original purchase price, duration of use, mode of selling(Buy Now or Auction). Sellers can also assign their products to different categories.

3.2.6 Searching and Browsing Products

Buyers can either search for specific products they’re interested in or browse through different categories to discover items they might like. To make the process easier, the system offers filters that help narrow down results, allowing users to sort by factors like price range and how long the item has been used.

3.2.7 Contacting Sellers

Buyers can contact sellers to ask about products and negotiate prices. Communication takes place through the application's messaging feature, allowing users to chat directly with each other.

3.2.8 Notifications

The application sends email notifications to users whenever they receive new messages, keeping them updated on their listings. Users also get email notifications and updates during the payment process to keep them informed about the status of their payment and delivery.

3.2.9 Removing Products

The application enables users to remove products they have uploaded for sale, either when the products are purchased and delivered successfully or when the user decides not to sell the product anymore.

3.2.10 Favourites

The application allows users to create a favorites list of products they are interested in but may not be ready to buy immediately.

3.2.11 Auction Feature

The auction feature allows users to participate in dynamic, time-bound bidding for used products. Sellers can list products for auction, setting a starting base price and a bidding deadline. Buyers can place competitive bids, with the highest bidder at the end of the auction winning the product. This feature helps set fair prices through competitive bidding, making the buying and selling process more exciting and engaging for users.

3.2.12 Secure Payment Gateway

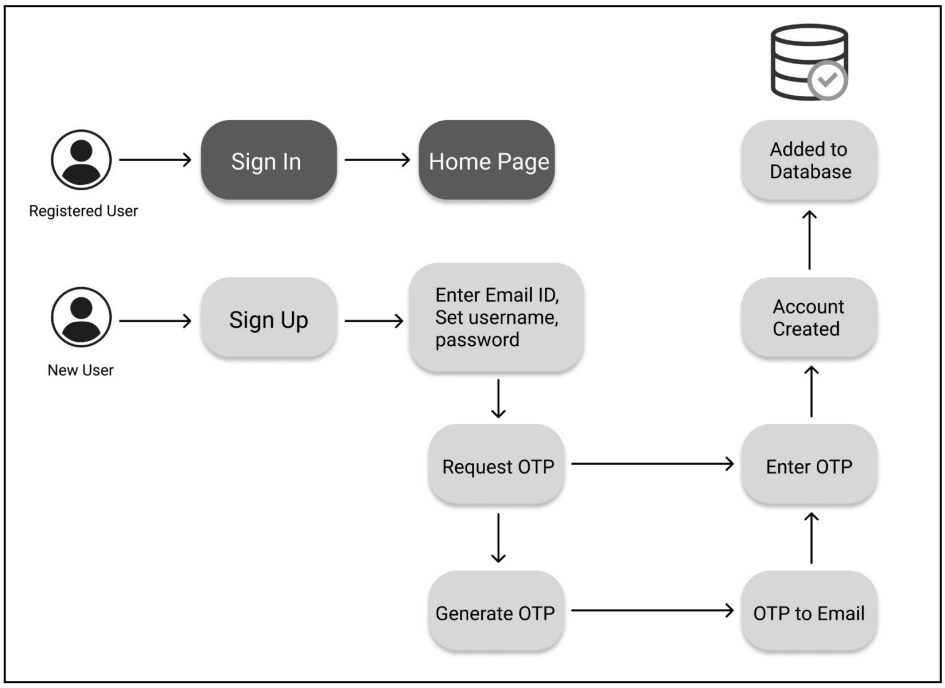
The user is guided through a secure payment process, ensuring a smooth transaction. The user receives immediate feedback that the payment has been processed successfully. This confirmation often comes in the form of a transaction ID or order summary to their registered email address.

3.2.13 Product Request

The product request feature allows buyers to request products that are not currently available on the platform. If a buyer is looking for a specific product, they can submit a request, and the system will notify all users about the demand for the product. This creates an opportunity for sellers to list the requested product, ensuring that buyers can find what they're looking for and also allowing sellers to respond to the market’s demand. This feature promotes a more interactive and responsive marketplace for all users.

3.3 Use Case Model

3.3.1 Use Case #1 (SignUp and Login)



Author – Anirudh

Purpose – This use case is for the login and signup of the user.

Requirements Traceability – Profile page interface, signup, and login interface,

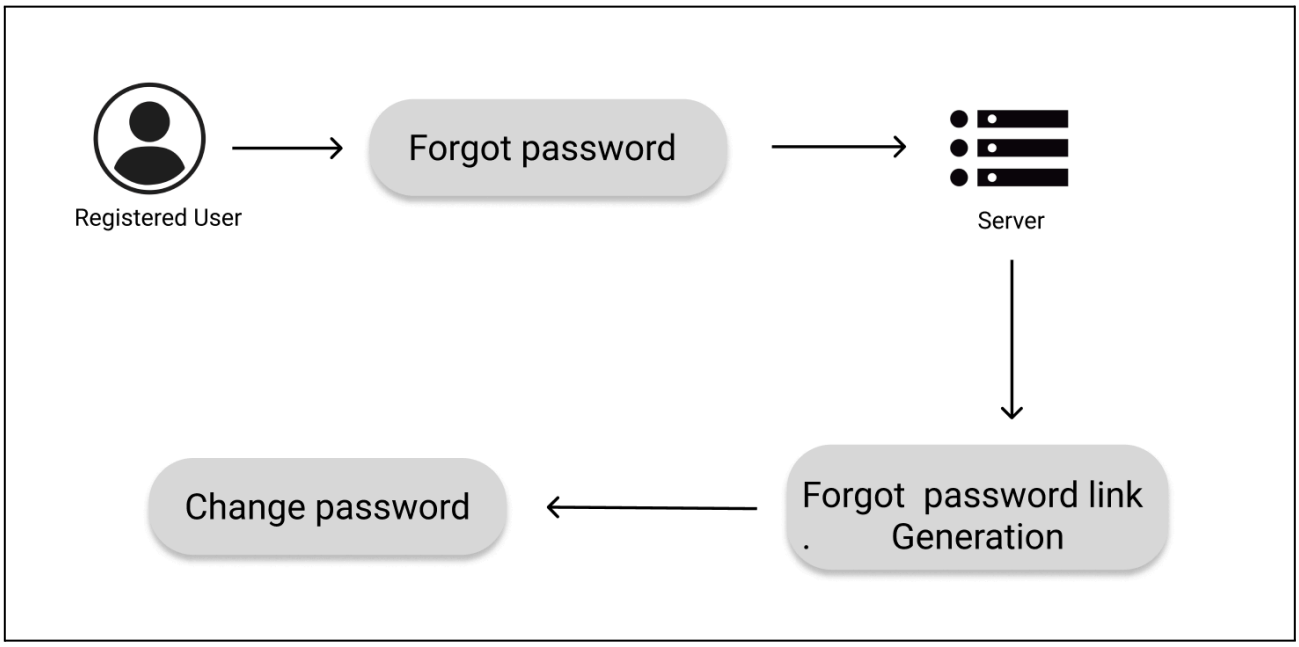
Priority – High

Preconditions – Must be a student of IITK

Post Conditions – The user is logged in and able to interact with the server.

Actors – User of the website, and server.
Exceptions – The user may forget the password and go for the forgot password option.
Includes – Only one account is created for a mail ID
Notes – Here the user will have to enter some necessary details(name, email, password, etc.) during signup/login

3.3.2 Use Case #2(Forgot password)



Author – Prem Santhosh

Purpose – This use case is to reset passwords

Requirements Traceability – Profile page, User Database

Priority – High

Preconditions – The user should have an account

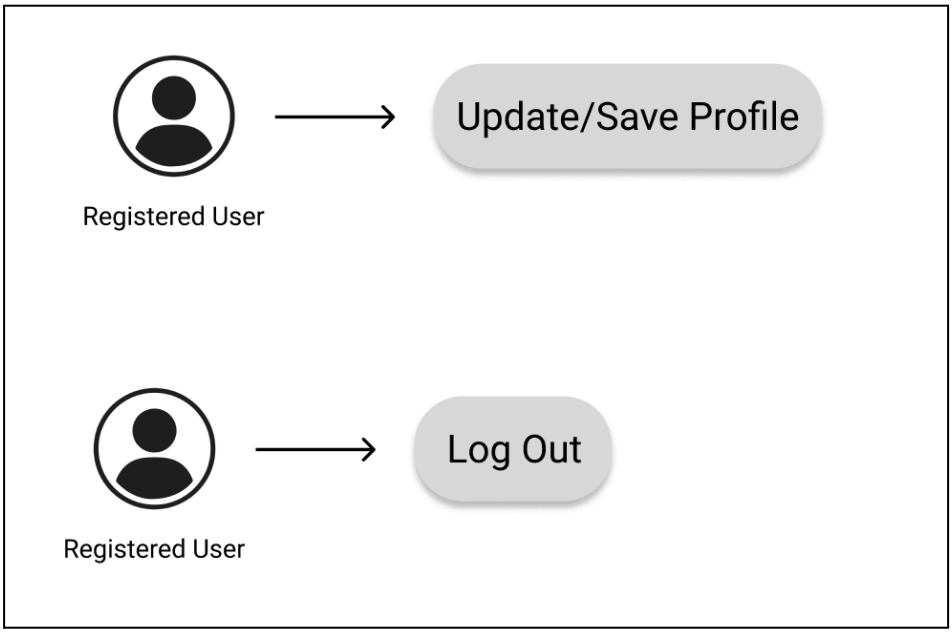
Post Conditions – The user gets a computer-generated OTP to his mail and the User's new password is updated.

Actors – User of the website, and server.

Exceptions – If the user doesn't receive OTP, he may go for the resend OTP option.

Includes – None

3.3.3 Use Case #3 (Update Profile)



Author – Koushik

Purpose – This use case is for the profile updation

Requirements Traceability – Profile page, User Database

Priority – Medium

Preconditions –The user should have an account

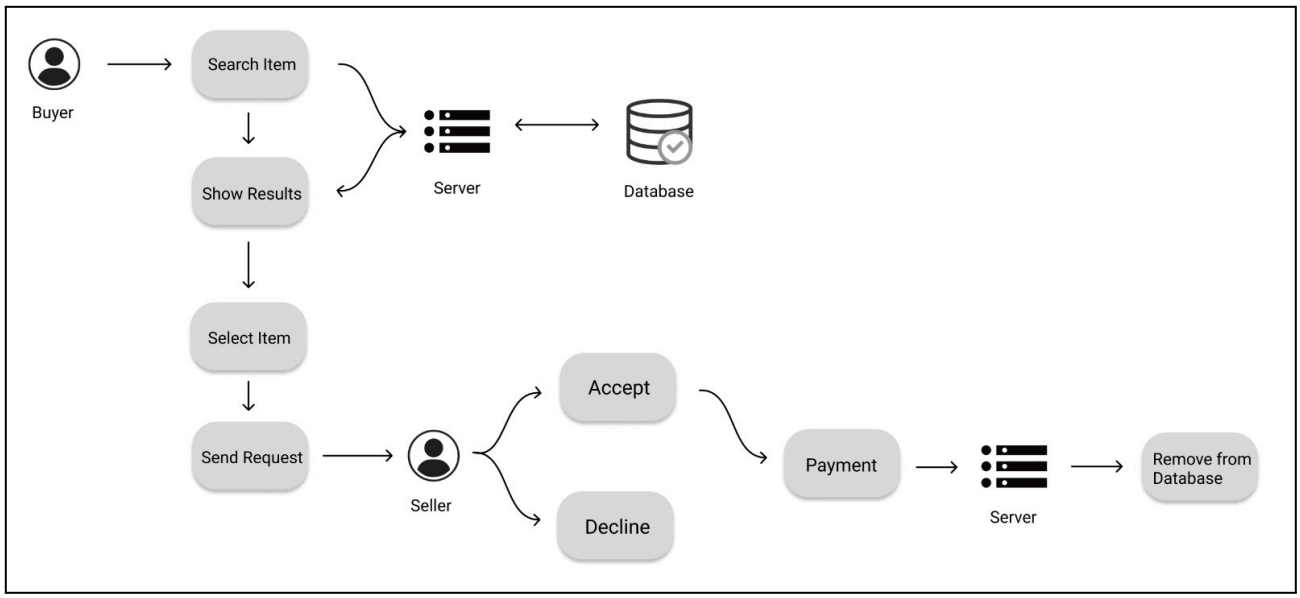
Post Conditions – Profile data updated

Actors – User of the website, and server.

Exceptions – None

Includes – None

3.3.4 Use Case #4 (Purchasing items)



Author – Anirudh, Saatvik

Purpose – This use case displays the interface for an item purchase.

Requirements Traceability – Database, Search Interface, Buyer-Seller messaging interface

Priority – High

Preconditions – The user should have an account, Database should contain item data.

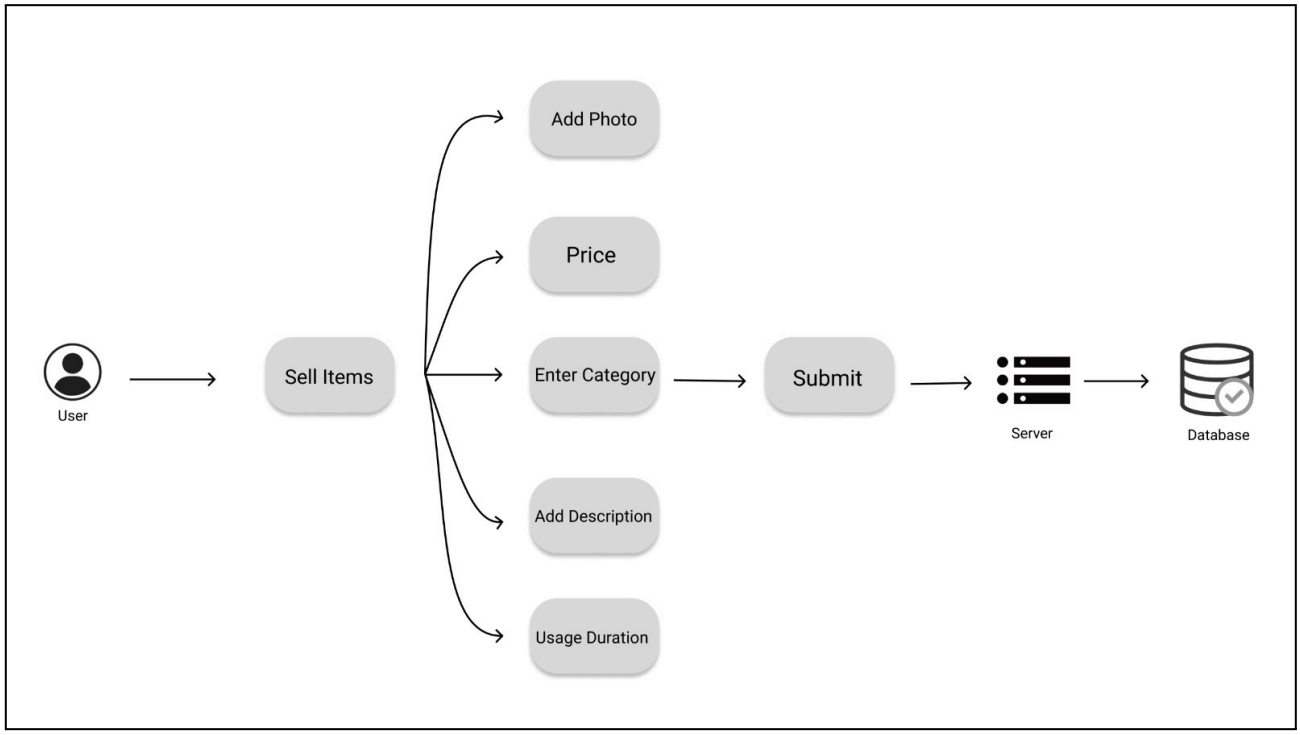
Post Conditions – Items are removed from product listings after successful payment and permanently deleted from the database after delivery.

Actors – Buyer, seller, delivery agent, server.

Exceptions – Items can be ordered by only a single user.

Includes – Buyer-Seller messaging interface and payments interface are provided.

3.3.5 Use Case #5 (Uploading Items)



Author – Ayush, Saatvik

Purpose – This use case displays the procedure to add an item to the sales list.

Requirements Traceability – Database.

Priority – High

Preconditions – The user should have an account and the Item data.

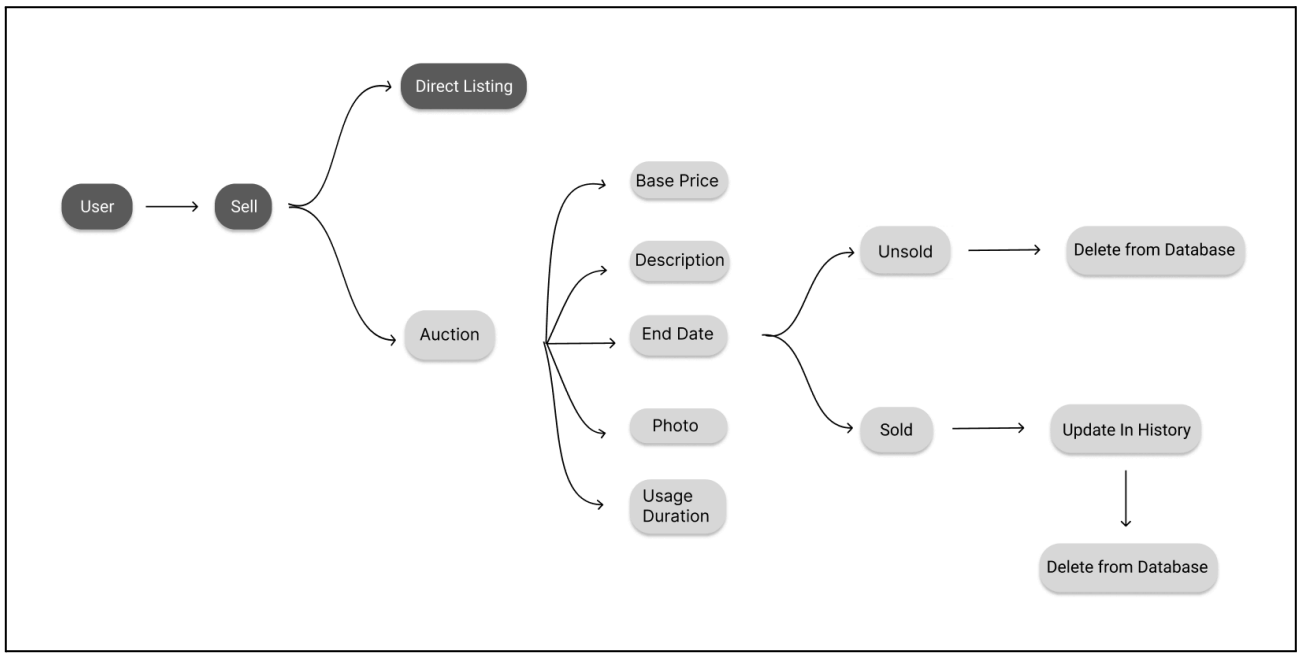
Post Conditions – Item list updation in the Database.

Actors – User of the website, and server.

Exceptions – None

Includes – None

3.3.6 Use Case #6 (Auction)



Author – Ayush, Prem, Saatvik

Purpose – This use case displays the auction procedure.

Requirements Traceability – Database

Priority – High

Preconditions – The user should have an account.

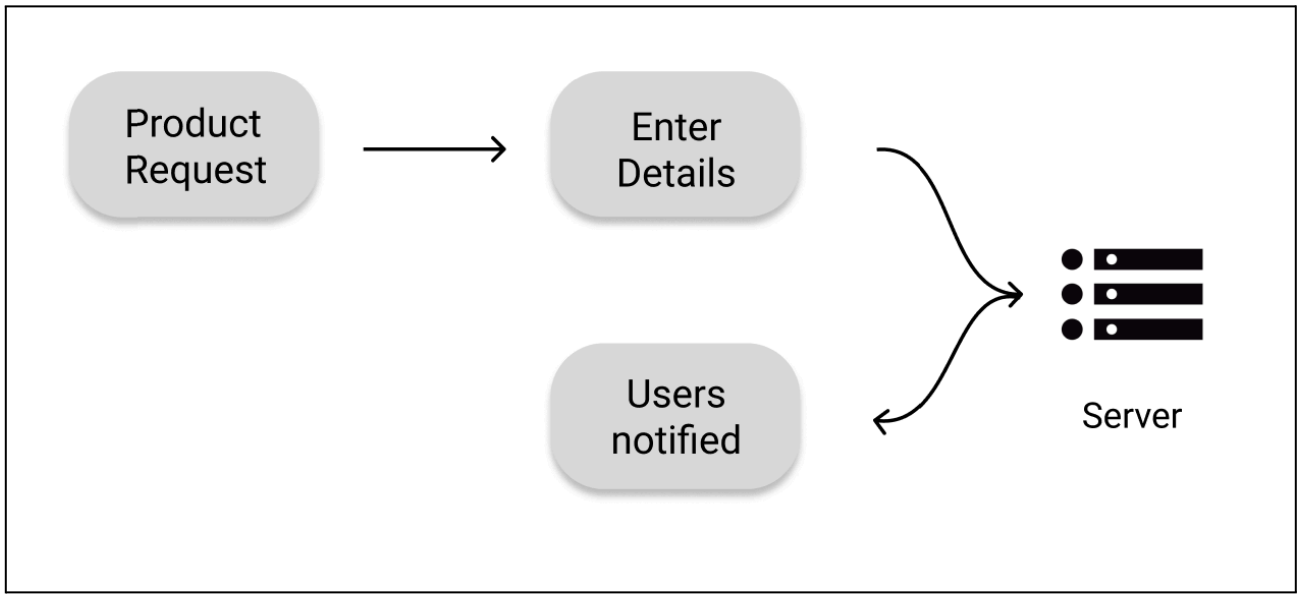
Post Conditions – Notification is sent to the users and the winner should pay the amount within a specified time to initiate the delivery.

Actors – Users of the website

Exceptions – None

Includes – None

3.3.7 Use Case #7 (Product Request)



Author – Anirudh, Koushik

Purpose – This use case displays the product request feature.

Requirements Traceability – Database

Priority – High

Preconditions – The user should have an account

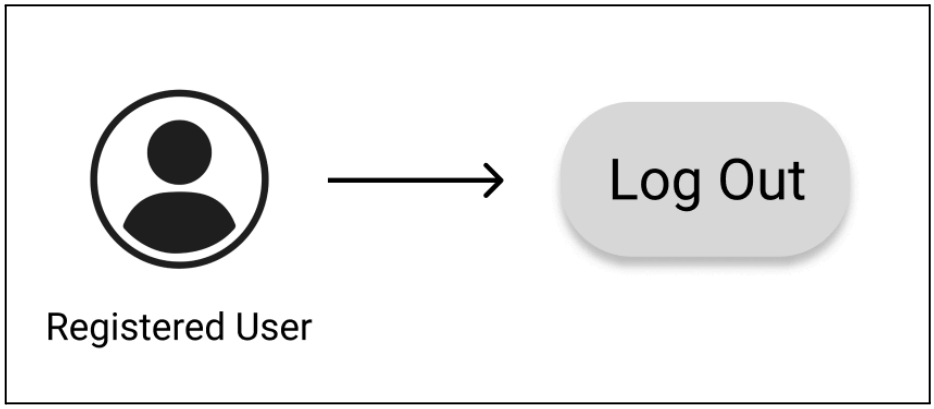
Post Conditions – The request list is updated in the database, and all users are notified.

Actors – Users of the website

Exceptions – None

Includes – None

3.3.8 Use Case #8 (Logout)



Author – Prem Santhosh, Koushik

Purpose – This use case displays the logout

Requirements Traceability – None

Priority – High

Preconditions – The user should have an account

Post Conditions: The user is logged out of their account, and the login page is displayed. The user needs to log back in to come back.

Actors – Users of the website

Exceptions – None

Includes – None

4 Other Non-Functional Requirements

4.1 Performance Requirements

The system should be fast and responsive, even when multiple users are online at the same time. Data should stay updated in real-time so that users always have access to the latest information. The system needs to be stable and consistent, offering a smooth experience without frequent crashes or frustrating errors.

As the community grows, the platform should easily scale to handle more users without slowing down. It should work well across a variety of devices, adapting seamlessly to different hardware and software setups. Even on older or less powerful devices, it should remain easy to use and responsive.

When updates or changes are needed, they should happen smoothly with minimal disruptions. The platform should also be flexible and ready to grow and evolve alongside the needs of its users in the community.

4.2 Safety and Security Requirements

Account Verification: Users must register with an IITK email address to verify the user.
User Privacy: All user credentials should be securely encrypted to prevent unauthorized access.
Communication Privacy: Chats between users are fully private, and only the involved users can view them. No access is granted to anyone else, including the system administrators.
Protection Against Fraud: Users can flag suspicious activities, fraudulent listings of products, or any inappropriate behavior from other users, maintaining a safe environment for all users.

4.3 Software Quality Attributes

4.3.1 Reliability

The system should be easy and convenient to use, letting users create and manage listings anytime. It efficiently handles large amounts of data, ensuring smooth operations and quick processing of the data. Whether users are buying or selling they can expect a smooth and dependable experience, all offered at a reasonable price.

4.3.2 Usability

The website should be built with simplicity as a priority, making it user-friendly for everyone. Sellers save time with helpful features like automated listing management, transaction history tracking, and effortless inventory updates.

4.3.3 Portability

The application works flawlessly on any Windows system and is built using React JS, making it responsive and adaptable as a progressive web application. This ensures the user enjoys a consistent and reliable experience across all the devices.

4.3.4 Flexibility

The Re_Store platform is designed to meet the user's needs. It allows users to customize listings, adjust search preferences, and manage content easily. Advanced search and filtering options allow users to find what they need quickly. Updating or editing listings is simple, giving users full control over their experience.

4.3.5 Maintainability

The platform is designed to run smoothly and efficiently. Its well-organized structure and detailed documentation make maintenance straightforward. Minor security fixes are addressed within a few days, whereas updates or improvements, including testing and documentation, are typically implemented within a week.

4.3.6 Adaptability

The application automatically keeps up with the changes. For example, when a product is sold, it’s immediately removed from the listings, ensuring the information you see is always accurate and up-to-date.

4.3.7 Security

Passwords are securely stored using advanced hashing techniques. If the user forgets the password, resetting it is easy with a secure link sent directly to the user’s registered email ID.

5 Other Requirements

Appendix A – Data Dictionary

Table for user's information

Field Name	Data Type	Field size for display	Description	Example
Name	String	25	Name of the user	Siddhanth Nandan Saaho
Username	String	25	Username(unique)	abc123
Email	Email	30	IITK email id of the user(unique)	saaho@iitk.ac.in
Password	hex	20	Password for login	00112233-4455-6 6 77-c899-aabbccddeeff

Table for sell page

Field Name	Data Type	Field size for display	Description	Example
Upload image	Composite	N/A	Images of the item	-----
Product Description	String	300	Information about item	Mountain bike, front and back suspension
Base Price	Integer	15	Price of the item	4,000
Using Since	Date	15	Date from when he is using the product	18/11/2022

Appendix B - Group Log

Meet Date	Members Present	Topics Discussed	Duration
10-01-2025	Koushik, Saatvik, Anirudh, Meghana, Pranaya, Rakesh, Yashwanth, Prem Santhosh, Sanjay Raghav, Ayush.	We explored ideas for the project, focusing on practical solutions to benefit the campus community.	120 Minutes
11-01-2025	Koushik, Saatvik, Anirudh, Meghana, Pranaya, Rakesh, Yashwanth, Prem Santhosh, Sanjay Raghav, Ayush.	Shortlisted three projects, prioritized them, discussed the ideas with the professor, and finalized the project.	120 Minutes
12-01-2025	Koushik, Saatvik, Anirudh, Meghana, Pranaya, Rakesh, Yashwanth, Prem Santhosh, Sanjay Raghav, Ayush.	We discussed our final idea, filled out the Google form, and created two groups to work on the project, with each group working on alternate days.	90 Minutes
13-01-2025	Anirudh, Koushik, Saatvik, Prem Santhosh, Ayush.	Developed a working plan for the project, identified the software tools required for documentation, and completed a few parts of the introduction.	90 Minutes
14-01-2025	Rakesh, Sanjay, Yashwanth, Meghana, Pranaya.	Completed the remaining parts of the introduction and wrote the text for Section 2.	90 Minutes

15-01-2025	Anirudh, Koushik, Saatvik, Prem Santhosh, Ayush.	Discussed the webpage interface, focusing on how it should function, and thought of some ideas for flowcharts.	90 Minutes
16-01-2025	Rakesh, Sanjay, Yashwanth, Meghana, Pranaya.	Learned Canva from scratch and created basic UI images.	90 Minutes
17-01-2025	Anirudh, Koushik, Saatvik, Prem Santhosh, Ayush.	Finalized the plans for the flowcharts and prepared them using Figma.	120 Minutes
18-01-2025	Rakesh, Sanjay, Yashwanth, Meghana, Pranaya.	Developed the final UI images using Canva, ensuring they aligned with our project's design requirements.	120 Minutes
19-01-2025	Anirudh, Koushik, Saatvik, Prem Santhosh, Ayush.	Completed Section 3 using the UI images and flowcharts we prepared.	120 Minutes
20-01-2025	Rakesh, Sanjay, Yashwanth, Meghana, Pranaya.	Discussed the non-functional requirements and completed Section 4.	120 Minutes
21-01-2025	Koushik, Saatvik, Anirudh, Meghana, Pranaya, Rakesh, Yashwanth, Prem Santhosh, Sanjay Raghav, Ayush.	We had a meeting with the TA to clarify doubts in the document.	30 Minutes
23-01-2025	Koushik, Saatvik, Anirudh, Meghana, Pranaya, Rakesh, Yashwanth, Prem Santhosh, Sanjay Raghav, Ayush.	Finished the Data Dictionary and Group Log, made final changes, and completed the document.	150 Minutes