Software Requirements Specification

for

Online Auction System for students to buy and sell their personal items

Prepared by BBS

Group Name: BBS

Abhilash P	**SE22UCSE004**	se22ucse004@mahindrauniversity.edu.in
Adithi Poddutoori	**SE22UCSE010**	se22ucse010@mahindrauniversity.edu.in
Surya Tej	**SE22UCSE065**	se22ucse065@mahindrauniversity.edu.in
Sreeja Bommagani	**SE22UCSE256**	se22ucse256@mahindrauniversity.edu.in
Sreeni Kura	**SE22UCSE258**	se22ucse258@mahindrauniversity.edu.in
Vijay Reddy T	**SE22UCSE305**	se22ucse305@mahindrauniversity.edu.in
Koushik Reddy	**SE22UCSE274**	se22ucse274@mahindrauniversity.edu.in

Instructor:	**Dr. Vijay Rao Duddu**
Course:	**Software Engineering**
Lab Section:	**CSE**
Teaching and Assistant:	**Dr. Vijay Rao Duddu**
Date:	**10-03-2025**

Contents

CONTENTS ..... **II**

REVISIONS ..... **III**

1 INTRODUCTION **1**

1.1 DOCUMENT PURPOSE 1
1.2 PRODUCT SCOPE 1
1.3 INTENDED AUDIENCE AND DOCUMENT OVERVIEW 1
1.4 DEFINITIONS, ACRONYMS AND ABBREVIATIONS 2
1.5 DOCUMENT CONVENTIONS 2
1.6 REFERENCES AND ACKNOWLEDGMENTS 2

2 OVERALL DESCRIPTION **3**

2.1 PRODUCT OVERVIEW 3
2.2 PRODUCT FUNCTIONALITY 3
2.3 DESIGN AND IMPLEMENTATION CONSTRAINTS 3
2.4 ASSUMPTIONS AND DEPENDENCIES 4

3 SPECIFIC REQUIREMENTS **4**

3.1 EXTERNAL INTERFACE REQUIREMENTS 4
3.2 FUNCTIONAL REQUIREMENTS ..... **ERROR! BOOKMARK NOT DEFINED.**
3.3 USE CASE MODEL 6

4 OTHER NON-FUNCTIONAL REQUIREMENTS **7**

4.1 PERFORMANCE REQUIREMENTS 7
4.2 SAFETY AND SECURITY REQUIREMENTS 7
4.3 SOFTWARE QUALITY ATTRIBUTES 7

5 OTHER REQUIREMENTS **8**

APPENDIX A – DATA DICTIONARY **8**

APPENDIX B - GROUP LOG **11**

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Draft Type and Number	*BBS*	*Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded.*	00/00/00

1 Introduction

The Bid, Buy, and Save (BBS) platform is a student-exclusive online auction system designed to simplify secure buying, selling, bidding, and renting of used items within a university environment. This platform provides a cost-effective and trusted marketplace for students, ensuring affordability, convenience, and accessibility. With features such as AI-driven recommendations and secure student authentication, BBS enhances campus trade while encouraging a peer-to-peer economy.

1.1 Document Purpose

This Software Requirements Specification document defines the functional and non -functional requirements for the Bid, Buy, and Save (BBS) platform, an exclusive online auction system for students. This document outlines the technical and operational aspects of the system to ensure a clear understanding among all stakeholders.

BBS is designed to provide university students with a secure and efficient marketplace to buy, sell, bid, and rent used items. The scope of this document covers the core auction functionalities, user authentication, AI-driven recommendations, and platform security. This Software Requirements Specification serves as a guide for developers, testers, and project managers to ensure a smooth development process.

1.2 Product Scope

The BBS platform is an online student-exclusive auction system that enables users to trade second-hand items within a secure campus environment. The platform is built to enhance affordability and resource-sharing by providing a trusted peer-to-peer economy.

Key Benefits:

Cost Savings*: Students can buy affordable second-hand goods.*
Flexibility: Users can bid, buy, or rent items.
Security*: Only verified students can access the platform.*
AI-driven Recommendations*: Smart suggestions based on user preferences.*

1.3 Intended Audience and Document Overview

This document is aimed at:

Developers: To grasp system requirements and implementation details.
Project Managers: To track milestones and ensure requirements are fulfilled.
Testers: To verify system functionality and security.
Clients & Professors: To evaluate the system's objectives and overall scope.

Document Structure:
- Introduction & Scope: Overview of the product.
- Technical Details: Technologies used and system design.
- Deliverables & Milestones: Development timeline.
- Roles & Responsibilities: Team members duties.

1.4 Definitions, Acronyms and Abbreviations

AI: Artificial Intelligence
BBS: Bid, Buy, and Save
CSS: Cascading Style Sheets
HTML: Hypertext Markup Language
JS: JavaScript
MySQL: Structured Query Language-based relational database
SRS: Software Requirements Specification
UI/UX: User Interface / User Experience

1.5 Document Conventions

This document adheres to IEEE formatting standards:
Font: Arial, size 11 or 12
Text Style: Italics for comments, bold for section titles
Spacing: Single-spaced with 1-inch margins
Headings: Follow the prescribed template structure for sections and subsections

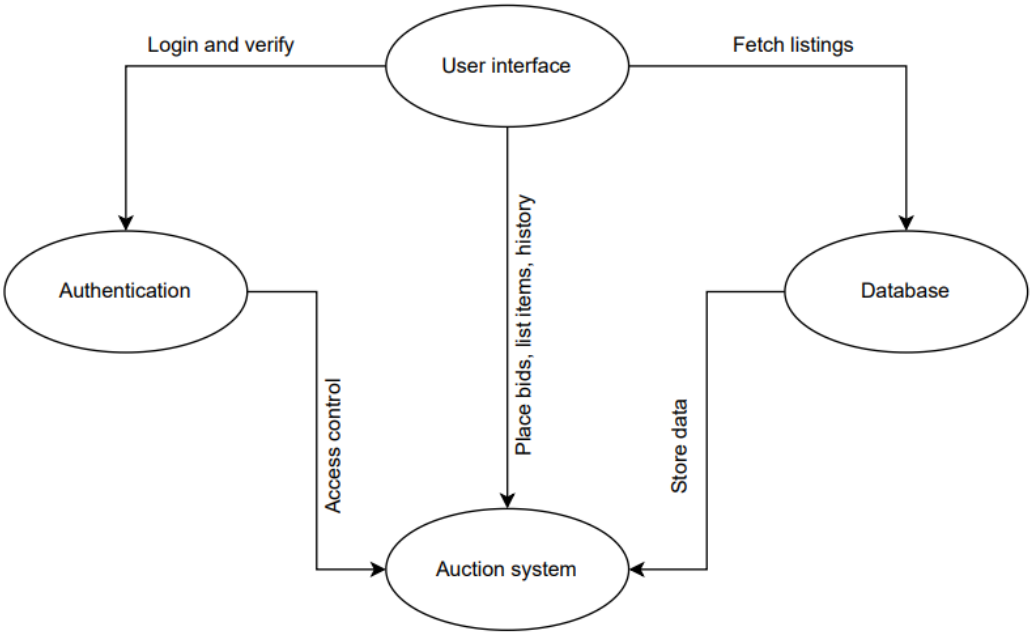
1.6 References and Acknowledgments

This document is based on the Team-9 Online Auction for Students (BBS) Project and references internal project documents, UI/UX designs, and system architecture guidelines. Additional references include university policies on student marketplaces and online auction systems.

2 Overall Description

2.1 Product Overview

The Online Auction System for Students is a web application that provides a dedicated auction platform for university students. It replaces traditional campus bulletin boards for trading items, ensuring security through strict student verification and offering a structured interface for listing items and placing bids.



2.2 Product Functionality

User registration and authentication.
Item listing with images and detailed descriptions.
Auction-based bidding system with real-time updates.
Secure transaction handling.
Admin panel for managing users and listings.
Reviews and notifications: Users can review sellers/items and receive real-time notifications regarding bid status and auction outcomes

2.3 Design and Implementation Constraints

Must comply with university policies.
Limited budget for hosting and maintenance.
Web-based only.
Backend: Java, Spring Boot, Hibernate, MySQL (and optionally MongoDB if needed).
Frontend: React.js

2.4 Assumptions and Dependencies

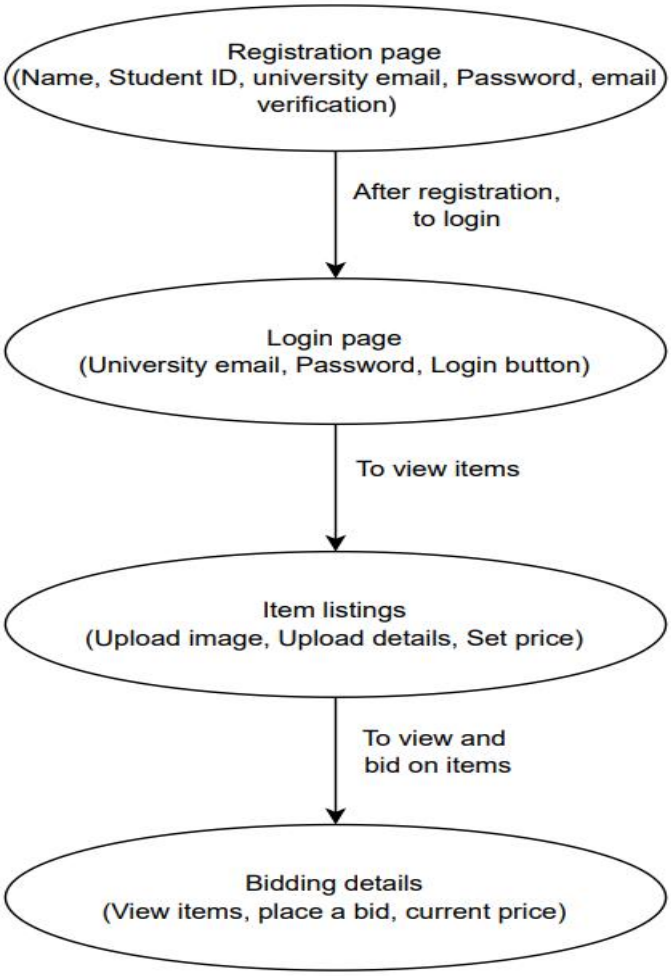
Only university students can access the platform.
Users will have reliable internet access.
Transactions are managed externally, there is no integrated payment gateway.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

Web-based interface with forms for registration, login, item listing, and bidding.



3.1.2 Hardware Interfaces

Standard web browsers and personal laptops.

3.1.3 Software Interfaces

Database interaction through MySQL.

3.2 Functional Requirements

The functional requirements define the expected behaviour of the system, including the features and services it provides to users.

3.2.1 F1: User Registration & Authentication

The system shall allow users to register using their student email.
The system shall verify student identities before granting access.
The system shall allow users to log in securely using a password.

3.2.2 F2: Item Listing & Management

The system shall provide an interface to list items for sale.
The system shall allow users to upload images and descriptions for listed items.
The system shall allow users to edit or remove their listings.

3.2.3 F3: Bidding & Auction Process

The system shall allow users to place bids on listed items.
The system shall track the highest bid for each auctioned item.
The system shall allow sellers to accept or reject bids.

3.2.4 F4: Notifications & Alerts

The system shall notify users about bid updates via email.
The system shall alert sellers when their item receives a new bid.
The system shall notify the highest bidder when an auction ends.

3.2.5 F5: User & Transaction Management

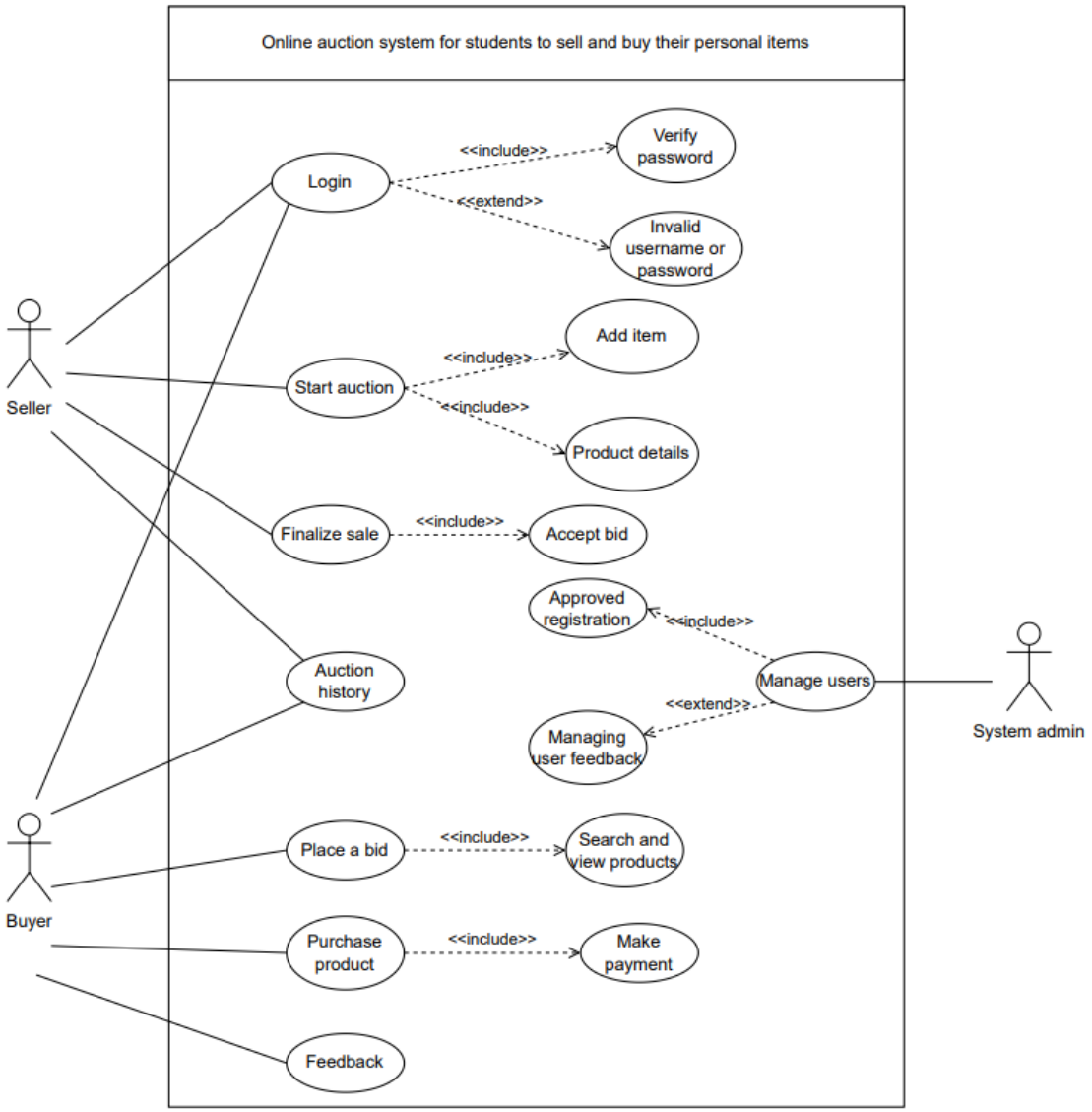
The system shall enable administrators to manage and verify users.
The system shall allow administrators to remove fraudulent or inactive users.
The system shall generate reports on auction activities.

3.2.6 F6: Rental Functionality

The system shall enable rental functionality for listing items available for rent.
The system shall allow users to set rental durations and prices.
The system shall manage rental transactions and agreements.

3.3 Use Case Model

3.3.1 Use Case #1



Author – *Adithi Poddutoori*

Purpose *- To allow users to register and participate in the online auction system for buying and selling products.*

Requirements Traceability – *Maps to the functional requirement for user registration.*

Priority *- High. The ability for students to register successfully and participate in auctions is critical for system deployment.*

Preconditions *- The user must be an existing university student with a valid email.*

Post conditions - *The user is registered and can access the platform.*

Actors – *a. Student – The primary actor operates on the platform.*

b. System – Validates student credentials and creates an account
– Manages Users and maintains platform integrity

Extends – *Login, Manage users*

Flow of Events

1. **Basic Flow** – *The system verifies the email and ensures it's a valid student email.*
   The system creates the account and stores the user information.
   The student receives a confirmation email with activation instructions.
   The student can log in and access the system.
2. **Alternative Flow** - *If the email is invalid, the system displays an error message.*
3. **Exceptions** - *Registration fails if the system cannot verify the email.*

Includes - Verify *password, add item, Product details, accept bid, Search and view products, Make payment, Approved registration.*

4 Other Non-functional Requirements

4.1 Performance Requirements

The system shall support at least 100 concurrent users.
The response time for page loads shall not exceed 5 seconds under normal operating conditions.

4.2 Safety and Security Requirements

Only verified students can register and access the platform.
User passwords must be stored securely using 128-bit encryption.
All item listings must be approved by an administrator before publication.

4.3 Software Quality Attributes

Reliability*: The system should function 24/7 with minimal downtime.*
Usability: The interface must be simple and intuitive to ensure ease of use.
Scalability*: The architecture must accommodate an increasing number of student users.*

5 Other Requirements

The database shall store transaction history for at least one year.

The system shall generate and maintain logs for administrative monitoring.

Compliance Requirements*: The platform must comply with university marketplace policies regarding student-to-student transactions. It must meet data protection and privacy laws such as GDPR. All financial transactions must follow PCI DSS (Payment Card Industry Data Security Standard) guidelines.*

Environmental Requirements*: The platform must be optimized for low energy consumption to ensure sustainability when hosted on cloud servers.*

Legal and Ethical Requirements*: Users must agree to terms of service before listing items or bidding. There will be strict policies against counterfeit or illegal items.*

Appendix A – Data Dictionary

Data Dictionary for BBS Platform

Variable/Constant	Description	Possible Values/States	Related Operations	Requirements
User_ID	*Unique ID for each student user*	*Alphanumeric (eg, SE22UCSE256)*	Authentication, Profile management	*Unique and verified against student records*
Username	*User's display name*	*Any valid string*	Profile creation, Login	*Unique, letters only*
Email	*Registered email address*	*Valid email format (eg, se22ucse256@mahindrauniversity.edu.in)*	Registration, Notifications	*Unique and verified*

Password	*User's login credential*	*Encrypted string*	Login, Authentication	*Minimum 8 characters, special symbols required*
Account_Type	*Type of user account*	*Buyer, Seller, Admin*	Role-based access	*Default Buyer upon registration*
Item_ID	*Unique ID for each item*	*Alphanumeric (e.g., ITEM5678)*	Listing, Bidding, Buying	*Auto-generated upon listing*
Item_Name	*Name of the product/service*	*Any valid string*	Listing, Searching, Filtering	*Must be descriptive*
Item_Description	*Description of the item*	*Any valid text*	Listing, Viewing	*Minimum 10 characters*
Starting_Price	*Initial price set by seller*	*Numeric (eg, 100.00)*	Bidding, Listing	*Must be greater than zero*
Current_Bid	*Highest bid on an item*	*Numeric*	Bidding, Displaying	*Updates dynamically*
Bid_ID	*Unique ID for a bid*	*Alphanumeric (eg, BID6789)*	Bidding, Transaction Records	*Auto-generated upon bid placement*
Bid_Amount	*Bid amount placed by user*	*Numeric*	Bidding, Auction Closing	*Must be higher than current bid*
Bid_Status	*Status of a bid*	*Active, Outbid, Won*	Bidding Process	*Auto-updates based on auction progress*
Transaction_ID	*Unique ID for transactions*	*Alphanumeric (eg, TXN3456)*	Payment, Order History	*Auto-generated upon purchase*

Payment_Status	*State of payment confirmation*	*Pending, Completed, Failed*	Payment Processing, Order Completion	*Must be updated post-payment*
Order_ID	*Unique ID for a completed purchase*	*Alphanumeric (eg, ORD1234)*	Order Management, Shipment	*Auto-generated upon successful payment*
Order_Status	*Status of the order*	*Processing, Shipped, Delivered, Cancelled*	Order Tracking	*Updates dynamically*
Saved_Items	*List of items saved by user*	*List of Item_IDs*	Wishlist, Future Purchase	*Optional*
Review_ID	*Unique ID for a review*	*Alphanumeric (eg, REV4567)*	Reviews, Ratings	*Auto-generated*
Review_Rating	*User rating for an item*	*Numeric (1 to 5)*	Reviews, Seller Reputation	*Required field*
Review_Comment	*User feedback for an item*	*Any valid text*	Reviews, Seller Reputation	*Optional but encouraged*
Date_Created	*Timestamp of entry creation*	*Date-Time Format*	Logging, Order Processing	*Auto-generated*

Appendix B - Group Log

A) Minutes from Group Meetings

Meeting 1: Project Kickoff

Attendees: All team members
Meeting duration: *1 hour*
Agenda: *Overview of the project, roles and responsibilities, timeline*
Key Points:
Discussed project goals and objectives
Assigned roles: Abhilash and Surya (Back-End Developers), Sreeja and Adithi (Front-End Developers/UI Designers), Sreeni (Testing and Marketing), Vijay and Koushik (Database Management)
Set deadlines for key milestones (SOW, SRS, SDD, Test Plan, Project Demos)

Meeting 2: Requirements Gathering

Attendees: *All team members*
Meeting duration: *4 hours*
Agenda: *Gather and document requirements from student surveys and interviews*
Key Points:
Analyzed survey results and interview feedback
Compiled a list of required features for the platform
Decided on essential functionalities: user registration, item listing, bidding system, payment integration, auction history, Feedback analysis

Meeting 3: Design Phase Kickoff

Attendees: *All team members*
Meeting duration: 5 *hours*
Agenda: Discuss UI/UX design, create design sketches and blueprints
Key Points:
Created initial design sketches for the platform's user interface
Ensured responsive design for mobile and desktop devices
Reviewed and refined design based on team feedback

B) Group Activities

Surveys and Interviews:

a. *Conducted surveys and interviews with students to understand their needs and preferences for the platform.*
b. *Compiled and analyzed feedback to determine essential features and functionalities.*

Design and Development:

c. *Created UML diagrams and developed demo UI.*
d. *Developed core features, including user registration, item listing and bidding system.*
e. *Implemented end to end encryptions to protect user data and ensure safe transactions.*

C) Effort and Contributions

Abhilash and Surya: *Working on Backend to effectively incorporate into Frontend and Database.*
Sreeja and Adithi: *Working on frontend development and UI/UX design.*
Sreeni: *Conducted surveys and gathered User-Feedback to decide on the additional features which can be integrated to improve User experience.*
Vijay and Koushik: *Created a database prototype based on information which will be added to the web application*

This detailed documentation of our group meetings, activities, and individual contributions highlights the effort and collaboration involved in producing the document and developing the BBS platform.