

---

# **Software Design Specifications**

**for**

## **Campus Barter: A Point-Based Exchange System for Students Version 1.0**

Prepared by:

Deepesh Bhardwaj

1<sup>st</sup> Prophet

Document Information

Title: Campus Barter: A Point-Based Exchange System for Students Project Manager: Raghu Kishore Neelisetti		Document Version No: 1.0
Prepared By: Deepesh Bhardwaj- SE21UCSE049 Gnan Reddy- SE21UCSE064 Rachana Vinjamur- SE21UCSE159 Sanskar Nagda- SE21UCSE191 Ramya Swarna- SE21UARI113 Jahnvi Gali-SE21UARI053 Aditya Bhardwaj- SE21UARI007 Rishith N V- SE21UCSE163		Document Version Date: 14/04/2024
		Preparation Date: 11/04/2024

Version History

Ver. No.	Ver. Date	Revised By	Description	Filename
1.0	11/04/2024	Deepesh Bhardwaj	First Draft of SDS	Project Software Design Specification V1.0

## Table of Contents

<b>1 INTRODUCTION .....</b>	<b>4</b>
1.1 PURPOSE .....	4
1.2 SCOPE .....	4
1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS .....	4
1.4 REFERENCES .....	4
<b>2 USE CASE VIEW .....</b>	<b>4</b>
2.1 USE CASE .....	4
<b>3 DESIGN OVERVIEW .....</b>	<b>4</b>
3.1 DESIGN GOALS AND CONSTRAINTS .....	5
3.2 DESIGN ASSUMPTIONS .....	5
3.3 SIGNIFICANT DESIGN PACKAGES .....	5
3.4 DEPENDENT EXTERNAL INTERFACES .....	5
3.5 IMPLEMENTED APPLICATION EXTERNAL INTERFACES .....	5
<b>4 LOGICAL VIEW .....</b>	<b>5</b>
4.1 DESIGN MODEL .....	6
4.2 USE CASE REALIZATION .....	6
<b>5 DATA VIEW .....</b>	<b>6</b>
5.1 DOMAIN MODEL .....	6
5.2 DATA MODEL (PERSISTENT DATA VIEW).....	6
5.2.1 Data Dictionary.....	6
<b>6 EXCEPTION HANDLING .....</b>	<b>6</b>
<b>7 CONFIGURABLE PARAMETERS .....</b>	<b>6</b>
<b>8 QUALITY OF SERVICE .....</b>	<b>7</b>
8.1 AVAILABILITY.....	7
8.2 SECURITY AND AUTHORIZATION .....	7
8.3 LOAD AND PERFORMANCE IMPLICATIONS .....	7
8.4 MONITORING AND CONTROL .....	7

# 1 Introduction

*[The introduction of the Software Design Specifications should provide an overview of the entire Software Design. It should include the purpose, scope, definitions, acronyms, abbreviations, references, and overview of the Software Design Specifications.]*

## 1.1 Purpose

*This Software Design Specification (SDS) document is crafted to outline the architectural and design standards for the Campus Barter platform, a point-based exchange system designed for students. The SDS details the framework necessary to develop and implement the system according to the requirements specified in the Software Requirements Specification (SRS).*

## 1.2 Scope

*The scope of this SDS encompasses the design and structural layout of the Campus Barter system, including detailed descriptions of the user interface, data management, security measures, and interaction protocols. The design is aimed at ensuring scalability, reliability, and user engagement through an intuitive interface and robust backend architecture.*

## 1.3 Definitions, Acronyms, and Abbreviations

- *API - Application Programming Interface*
- *UI - User Interface*
- *UX - User Experience*
- *Database - Organized collection of data, generally stored and accessed electronically from a computer system.*

## 1.4 References

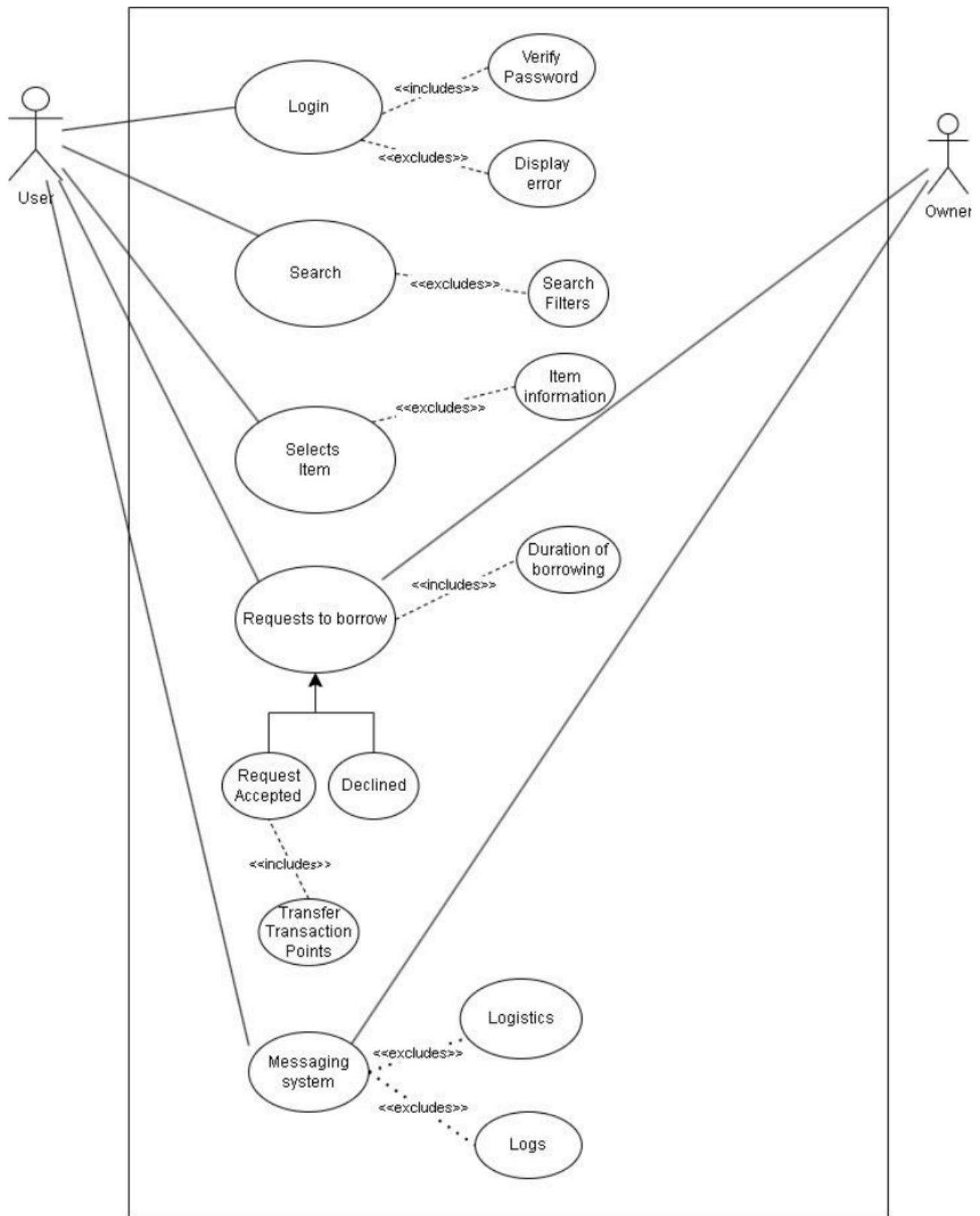
- *Software Requirements Specification for Campus Barter*
- *Campus Barter Project Glossary*

# 2 Use Case View

## 2.1 Use Case

*Use Case Diagram and Description: Borrowing an Item*

- *Scenario: A user wants to borrow an item using barter points.*
- *Steps:*
  1. *User logs in to the platform.*
  2. *User navigates to the item listings and selects an item.*
  3. *User sends a borrow request to the item owner.*
  4. *The item owner reviews the request and either approves or denies it.*
  5. *If approved, barter points are transferred from the borrower to the lender.*
  6. *The borrower and lender coordinate for item pickup.*



### 3 Design Overview

### 3.1 Design Goals and Constraints

- **Goals:**
  - To create a scalable and secure system that facilitates the barter of items using a point-based system.
  - To ensure that the system is user-friendly and accessible to all students.
- **Constraints:**
  - Must operate within the regulatory frameworks of data security and user privacy.
  - The system should be fully operational with minimal downtime.

### 3.2 Design Assumptions

- The system assumes that all users will have a reliable internet connection.
- Users are assumed to have a basic understanding of web navigation.

### 3.3 Significant Design Packages

- **User Management:** Handles user registration, authentication, and profile management.
- **Item Management:** Manages item listings, searches, and transaction histories.
- **Transaction Processor:** Processes and logs all transactions made within the system.

### 3.4 Dependent External Interfaces

This section describes the external interfaces that are crucial for the operation of the Campus Barter system. These interfaces allow the application to interact with external services and systems to enhance functionality and user experience.

**Table: Public Interfaces Required by the Design**

External Interface Name	Module Using the Interface	Description of Functionality	How the Interface is Used
Email Service API	User Account Management	To send verification emails and notifications to users.	The Email Service API is crucial for sending out email confirmations during user registration and for other notifications related to account management.
Payment Gateway API	Financial Transactions	Handles secure processing of transactions for purchasing points.	Integrated within the Financial Transactions module to enable users to buy barter points securely.
Mapping Service API	Item Location Services	Provides mapping data to help users locate items and arrange meet-ups.	Used to display item locations on maps and suggest convenient exchange locations.

#### Example Interface Details

- **Email Service API**
  - **Used by:** User Account Management
  - **Purpose:** Automates sending of operational emails.
  - **Application:** Sends welcome emails, password resets, and notification updates.
- **Payment Gateway API**
  - **Used by:** Financial Transactions
  - **Purpose:** Ensures secure handling of financial transactions online.
  - **Application:** Processes payments when users purchase additional barter points.
- **Mapping Service API**
  - **Used by:** Item Location Services
  - **Purpose:** Provides geographical data services.
  - **Application:** Helps users find items available for barter in nearby locations.

### 3.5 Implemented Application External Interfaces (and SOA web services)

*This section details the interfaces provided by the Campus Barter system that are available for other applications. These interfaces allow external systems to leverage specific functionalities of the Campus Barter platform.*

**Table: Implementation of Public Interfaces**

Interface Name	Module Implementing the Interface	Description of Functionality	How the Interface is Implemented
User Profile API	User Management	Allows external systems to retrieve user profile information.	This API provides a secure way for external applications to access basic user profile data, improving interoperability with other systems.
Item Search API	Search System	Enables external systems to search for items available for barter.	Implemented to allow third-party applications to query available items based on various filters like category, condition, and location.
Transaction API	Transaction Management	Facilitates external applications in initiating and tracking barter transactions.	Provides endpoints for starting, monitoring, and completing exchanges, thereby enhancing the platform's integration capabilities.

#### **Example Interface Implementation**

- **User Profile API**
  - **Implemented by:** User Management Module
  - **Purpose:** Provides access to user profiles.
  - **Details:** Securely outputs user data such as name and transaction history upon request.
- **Item Search API**
  - **Implemented by:** Search System
  - **Purpose:** Allows item queries from external parties.
  - **Details:** Supports search operations by external apps, returning items that match specified criteria.
- **Transaction API**
  - **Implemented by:** Transaction Management
  - **Purpose:** Manages barter transactions between users.
  - **Details:** Handles requests to initiate, process, and finalize transactions through external applications.

## 4 Logical View

*This section outlines the architectural and detailed design of the Campus Barter system. It describes how the various components and modules of the system interact and work together to fulfil the application's key functionalities.*

#### **Design Hierarchy and Interactions:**

- **Top Layer (Interaction Layer):** This layer includes the user interface components and external API gateways. It manages interactions with the users and external systems.
- **Middle Layer (Business Logic Layer):** Consists of various service managers that handle the business logic of the system. For instance, the Transaction Manager handles all logic related to barter transactions.
- **Bottom Layer (Data Access Layer):** This layer interacts with the database and data storage systems to retrieve, update, and store data.

### 4.1 Design Model

The Campus Barter system is modularized into several key components, each responsible for a distinct aspect of the application's functionality. Here are the significant modules and classes:

**Modules and Classes:**

- **User Module**
  - **Classes:**
    - **User:** Manages user profile data and credentials.
    - **UserManager:** Handles operations such as user registration, login, and profile updates.
  - **Responsibilities:** Ensures user data integrity and security. Facilitates user interactions with the platform.
  - **Relationships:** Interacts with the Transaction Module to update user point balances.
- **Item Module**
  - **Classes:**
    - **Item:** Represents items available for barter.
    - **ItemController:** Manages item listings, searches, and item data retrieval.
  - **Responsibilities:** Maintains item listings and coordinates item exchanges.
  - **Relationships:** Uses User data to link items with owners.
- **Transaction Module**
  - **Classes:**
    - **Transaction:** Manages data regarding barter transactions.
    - **TransactionManager:** Processes and logs transactions.
  - **Responsibilities:** Handles the logistics of barter transactions, including point transfers and transaction confirmations.
  - **Relationships:** Collaborates with the User and Item modules to facilitate transactions.

## 4.2 Use Case Realization

Each key use case defined in Section 2 is implemented through interactions between these modules. Below are descriptions and diagrams for a sample use case.

**Use Case: Item Borrowing**

- **Overview:** Describes how users can borrow items using barter points.
- **Interaction Between Modules:**
  1. **User Module:** Authenticates the user and checks the user's point balance.
  2. **Item Module:** Searches for the requested item and verifies its availability.
  3. **Transaction Module:** Initiates the transaction if the user has enough points, then updates the point balances of both parties involved.

**Diagrams:**

- **Sequence Diagram:** Shows the sequence of interactions between the User, Item, and Transaction modules for an item borrowing scenario.
- **Activity Diagram:** Details each step involved in the borrowing process, from item search to transaction completion.

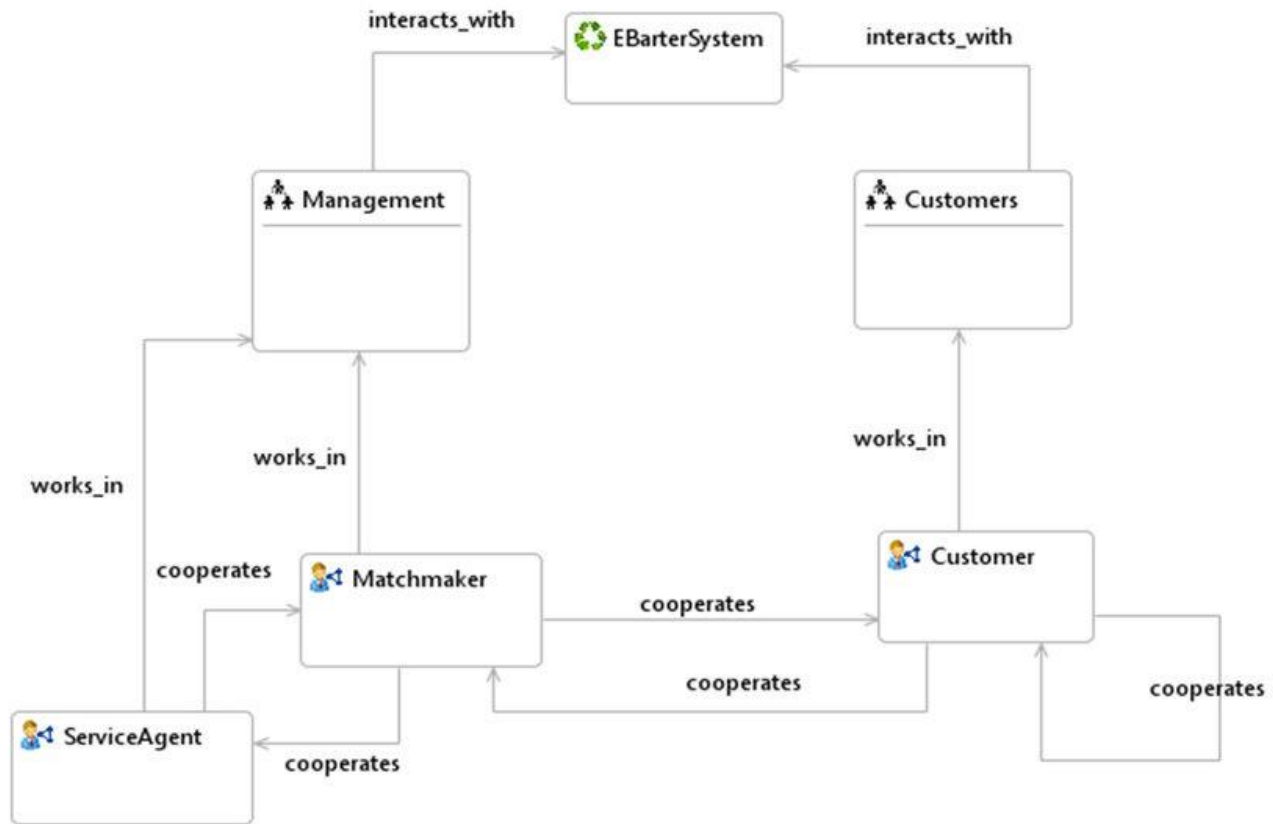
**Detailed Class Collaborations:**

- **User Class:** Verifies user identity and point balance.
- **Item Class:** Checks item availability.
- **Transaction Class:** Processes the exchange of points and updates transaction records.

**Expand for each Use Case:**

- Repeat the above format for each use case, providing relevant diagrams and descriptions to illustrate how the modules and classes interact to realize the functionality.





## 5 Data View

### 5.1 Domain Model

- *The domain model includes entities such as User, Item, and Transaction, with relationships defined to reflect ownership and transactions.*

### 5.2 Data Model (persistent data view)

- *User Table: Stores user data with fields for userID, username, password, etc.*
- *Item Table: Stores item data with fields for itemID, description, ownerID, etc.*
- *Transaction Table: Logs transactions with fields for transactionID, borrowerID, lenderID, itemID, etc.*

#### 5.2.1 Data Dictionary

- *userID: Unique identifier for a user.*
- *itemID: Unique identifier for an item.*
- *transactionID: Unique identifier for a transaction.*

## 6 Exception Handling

*Describes how the system handles potential errors such as database failures, authentication errors, and data validation issues.*

*Example: If a user fails to authenticate, the system logs the attempt and returns an error message.*

## 7 Configurable Parameters

*Parameters that can be adjusted include point values for items, user roles, and access permissions.*

This table describes the simple configurable parameters (name / value pairs).

Configuration Parameter Name	Definition and Usage	Dynamic?
<i>maxBorrowLimit</i>	<i>Defines the maximum number of items a user can borrow at one time.</i>	Yes

## **8 Quality of Service**

### **8.1 Availability**

*The system is designed for high availability with redundant servers and failover mechanisms in place.*

### **8.2 Security and Authorization**

*Implements OAuth for secure authentication and SSL/TLS for data encryption.*

### **8.3 Load and Performance Implications**

*Designed to handle up to 10,000 concurrent users without performance degradation.*

### **8.4 Monitoring and Control**

*Includes logging of all critical operations and performance metrics to monitor the system's health.*