Software Engineering
Software Requirements Specification (SRS) Document

[Shop Management Web App]
[Shoppeasy]

[9/24/23]

[Version]

By: [Dragon, Samuel, Jeong]

[Honor Code]

Table of Contents

1. Introduction

1.1. Purpose

[The goal of your project and the objectives it wishes to accomplish]

The Shop Management Web App is a shopping tool web app and listing aggregator designed to be an easy and simple all-around shop-and-sell tool for newcomers to online shopping and selling. The goal of this web app is to be used both by buyers and sellers to be introduced to the ability to be able to easily shop, browse, and configure items online.

1.2. Document Conventions

[Full description of the main objectives of this document in the context of your project.
Here's how you should begin this section:
"The purpose of this Software Requirements Document (SRD) is to..."
"In it, we will . . ., . . ., and . . .."]

The purpose of this Shop Management Web App (SMWA) is to provide an easy web application for buyers and sellers to shop and sell. In it, we will include a way to search for products, create listings of products for its users (for both buyers and sellers), configure product specifications such as pricing, description, etc. and communication between users. Providing a straightforward interface that gives a clear view of the web application's features.

1.3. Definitions, Acronyms, and Abbreviations

[Include any specialized terminology dictated by the application area or the product area.
For example:]

Java	A programming language originally developed by James Gosling at Sun Microsystems. We will be using this language to build the shopping web app.
MySQL	Open-source relational database management system.
HTML	Hypertext Markup Language. This is the code that will be used to structure and design the web application and its content.
SpringBoot	An open-source Java-based framework used to create a micro Service. This will be used to create and run our application.
MVC	Model-View-Controller. This is the architectural pattern that will be used to implement our system.
Spring Web	Will be used to build our web application by using Spring MVC. This is one of the dependencies of our system.
Thymeleaf	A modern server-side Java template engine for our web environment. This is one of the dependencies of our system.
NetBeans	An integrated development environment (IDE) for Java. This is where our system will be created.
API	Application Programming Interface. This will be used to implement a function within the software where the current date and time is displayed on the homepage.

1.4. Intended Audience

[Describe which part of the SRS document is intended for which reader. Include a list of all stakeholders of the project, developers, project managers, and users for better clarity.]

The intended audiences of this SRS document are buyers, individual sellers, and businesses (the users), as well as ourselves (the developers). For the former, the first and fifth sections of the document are most suitable for use.

1.5. Project Scope

[Specify how the software goals align with the overall business goals and outline the benefits of the project to business.]

The goal of the web application is to provide an easy-to-use interface for all consumers, sellers, and other users of businesses to buy, sell, or configure products to fit their needs. This aligns with the business goals of online shopping, as a shopping web app requires an organized, fast and clear interface that can lead users to easily find whatever they want to fulfill their goals.

The benefits of the project to business include:

- Relieves stress of finding products and price limits of products and ensures they have the tools to navigate through the web application.
- Increasing pleasure to customers as they are able to communicate with sellers and admins to guarantee the best possible service.
- Increases pleasure among consumers and sellers to be able to create listings to keep products in a user's shopping cart organized and to be able to configure their listings.

1.6. Technology Challenges

[Any technological constraints that the project will be under. Any new technologies you may need to use]

1.7. References

[Mention books, articles, web sites, worksheets, people who are sources of information about the application domain, etc. Use proper and complete reference notation. Give links to documents as appropriate. You should use the APA Documentation model (Alred, 2003, p. 144).]

2. General Description

2.1. Product Perspective

[Describe the context and origin of the product.]

Shopping Management Web Application (SMWA) found its origin through a person that simply doesn't have the time to go out shopping and has the desire to shop from the comforts of their home, delivered right at their doorstep. The idea was originated by a person with not much free time.

2.2. Product Features

[A high-level summary of the functions the software would perform and the features to be included.]

The product features include the ability for users to buy and sell listings, with identification being handled through individual account creation, and also the ability for administrators to manage user accounts and listings for moderation purposes. Users can also search and sort listings, as well as see listings posted by specific users.

2.3. User Class and Characteristics

[A categorization and profiling of the users the software is intended for and their classification into different user classes]

Our website application does not require consumers and sellers to have any prior knowledge of a computer, as we are making our features as clear and visible as possible. Although, basic knowledge of using a web browser and one's own knowledge of shopping is advised. The website application disables the need to have prior knowledge of shopping online, as the visible features will guide them easily.

2.4. Operating Environment

[Specification of the environment the software is being designed to operate in.]

The web application is designed to operate on all web browsers across many computers (desktop, laptops, etc.).

2.5. Constraints

[Any limiting factors that would pose challenge to the development of the software. These include both design as well as implementation constraints.]

2.6. Assumptions and Dependencies

[A list of all assumptions that you have made regarding the software product and the environment along with any external dependencies which may affect the project]

The web application will be dependent on Spring Web, SpringBoot and API in order to create and run the application that will be developed within NetBeans. The application will use source code in Java and HTML to build our web application.

3. Functional Requirements

[Statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations.]

3.1. Primary

[All the requirements within the system or sub-system in order to determine the output that the software is expected to give in relation to the given input. These consist of the design requirements, graphics requirements, operating system requirements and constraints if any.]

- FR0: The system will allow the user to generate a listing from entered data, including name, price point, method of exchange, display image, and tags.
- FR1: The system will allow the user to search through listings generated by other users, categorizing them either chronologically, by tag, or by user.
- FR2: The system will allow the user to send a purchase request to another user who has generated a listing.
- FR3: The system will allow users who have interacted through a purchase request to contact each other through the use of a messaging interface.
- FR4: The system will allow administrators to manage the information stored on user accounts.

3.2. Secondary

[Some functions that are used to support the primary requirements.]

- Account authorization setup so that users cannot alter each others’ listings and orders
- Logged pricing history to ensure that a user is charged only the price of an item at the time of purchase

4. Technical Requirements

4.1. Operating System and Compatibility

[The environments that will be needed to operate the system]

The application will be compatible with any operating system that is able to view and to interact with traditional web pages.

4.2. Interface Requirements

4.2.1. User Interfaces

[The logic behind the interactions between the users and the software. This includes the sample screen layout, buttons and functions that would appear on every screen, messages to be displayed on each screen and the style guides to be used.]

The application will have buttons to interact with between the users and our software include:

- Home Page (Title at top of screen with a name):
  - “add to cart” button
  - “Manage listing”/”Manage shopping cart” button
  - “login” function that displays the message is username/password is incorrect.
  - “Add product” button under products.
- Manage Listing Page: Seller
  - “Add a product” button that allows sellers to enter their product information.
  - A “message user” button with a message indicating that the message has been sent and a message indicating another message from another user.
- Listing Page: Buyer
  - A “message user” and “give feedback” button with a message confirming the message/feedback being sent, and a message indicating a message from another user.
  - “Remove item” button that will remove the product from the cart.
- Admin Page (Title displayed at top of screen):
  - View user button, will display user information.
  - Under the “view user” button, there will be a remove user button, and there will be a message after clicking the button.
  - A “message user” button with a message indicating that the message has been sent and a message indicating another message from another user.
  - “Add promotion” button to set discounts to a set category with a message indicating that the promotion is established.

4.2.2. Hardware Interfaces

[All the hardware-software interactions with the list of supported devices on which the software is intended to run on, the network requirements along with the list of communication protocols to be used.]

The application will be able to run on any device that has internet access such as smartphones, desktop computers, laptops and tablets.

4.2.3. Communications Interfaces

[Determination of all the communication standards to be utilized by the software as a part of the project]

The application will utilize the HTTP protocol to connect to a web API and fetch info.

4.2.4. Software Interfaces

[The interaction of the software to be developed with other software components such as frontend and the backend framework to be used, the database management system and libraries describing the need and the purpose behind each of them.]

HTML, CSS to build the front end of our web application and using Java to support our back end code. Also using list libraries/APIs to hold and retrieve data for products/items and users, as well as maps for login credentials.

5. Non-Functional Requirements

[Constraints on the services or functions offered by the system (e.g., timing constraints, constraints on the development process, standards, etc.). Often apply to the system as a whole rather than individual features or services.]

5.1. Performance Requirements

[The performance requirements need to be specified for all the functional requirements.]

- NFR0(R): Once the user has inputted all necessary data, creating and managing a listing will take no longer than thirty seconds to complete.
- NFR1(R): The listing search algorithm will return the current results within ten seconds, regardless of how much content has been sorted
- NFR2(R): A given listing's history will never be greater than twenty entries. Any entries older than this will be culled.

5.2. Safety Requirements

[List out any safeguards that need to be incorporated as a measure against any possible harm the use of the software application may cause.]

- NFR3(R): Create a listing history so that any price changes to a listing after an order is made don't change the order

5.3. Security Requirements

[Privacy and data protection regulations that need to be adhered to while designing of the product.]

- NFR4(R): No user will be able to access the purchasing history or details of another user
- NFR5(R): No user will be able to access the pickup and shipping details of another user
- NFR6(R): Messages associated with transactions should be kept entirely private, unless a user explicitly reports a message to an administrator

5.4. Software Quality Attributes

[Detailing on the additional qualities that need to be incorporated within the software like maintainability, adaptability, flexibility, usability, reliability, portability etc.]

5.4.1. Availability

The software should be available to anyone with a personal computer and an internet connection, and the ability to purchase listings should be available to anyone with an account.

5.4.2. Correctness

When given two identical search queries, unless a new listing has been added, the software should provide precisely the same results to the search.

5.4.3. Maintainability

The software should be able to run without intervention, even when errors occur, and should be built in such a way that new features can be added to modular elements (such as available listing fields, the searching algorithm, etc.) without altering the existing features.

5.4.4. Reusability

Features should be built in a modular fashion, so that if a new view is added that requires them it may merely incorporate the functionality to access them that has already been constructed.

5.4.5. Portability

The software should be shipped as Java jars, so it can be run on any system with Java installed.

5.5. Process Requirements

5.5.1. Development Process Used

[Software Process Model]

Incremental Development Model

5.5.2. Time Constraints

Workload of multiple classes in folders, availability of group members

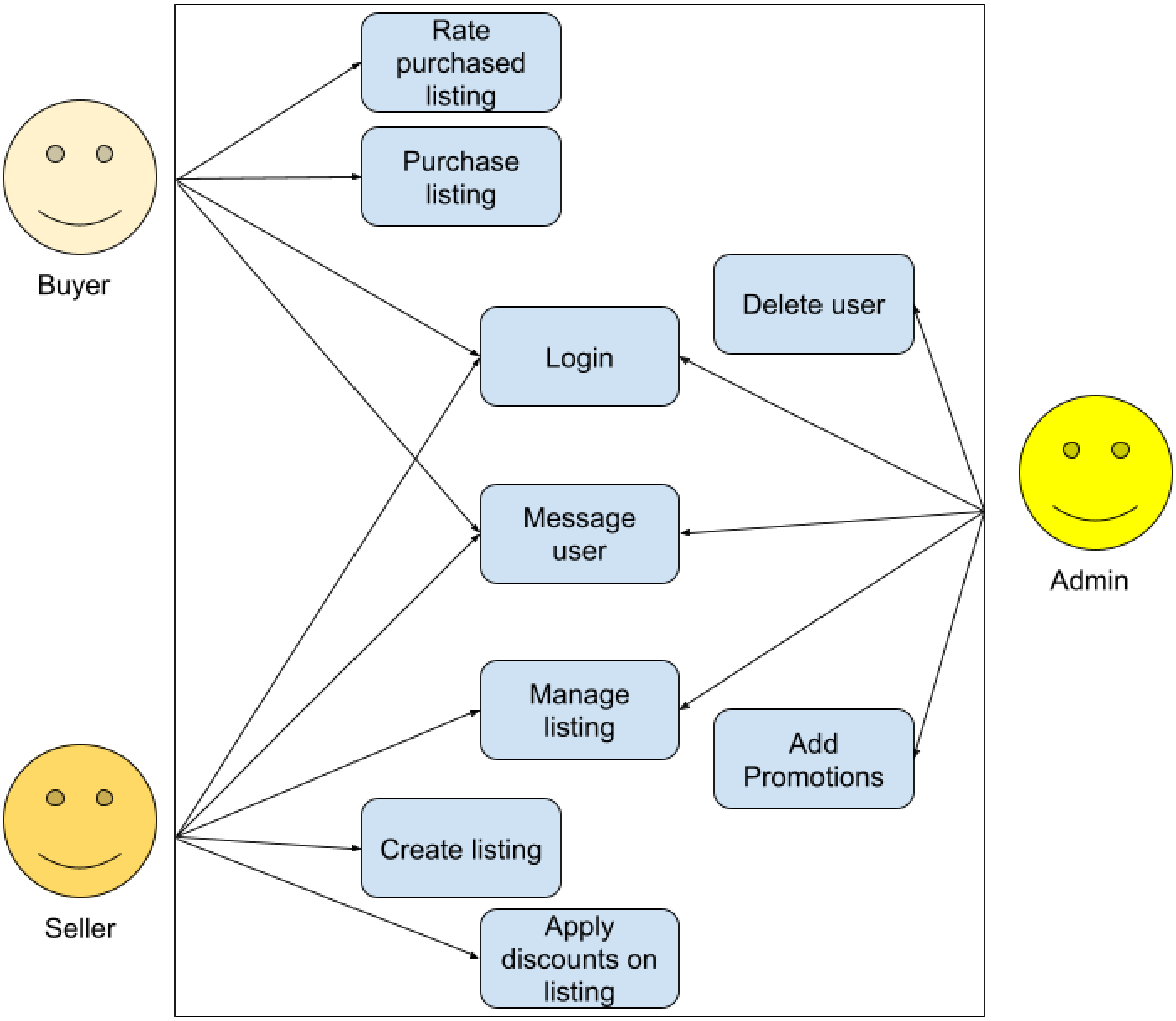
5.5.3. Cost and Delivery Date

Cost of 0$ and Delivery date is on December 5th, 2023

5.6. Other Requirements

TBD

5.7. Use-Case Model Diagram



5.8. Use-Case Model Descriptions

5.8.1. Actor: Admin (Dragon Phiansin)

- **Login:** Username and password are inputted to identify a specific admin.
- **Message User(s)**: The admin may respond to users to assist them, answer questions, and deal with problems.
- **Delete User(s)**: The admin may delete/ban users accordingly, based on user permissions or based on user behavior.
- **Add Promotions**: The admin can add certain promotions depending on certain aspects such as seasons, quantity of items, or events (back to school, Christmas, etc.).

5.8.2. Actor: Buyer (Jeong Kim)

- **Login:** Username and password are inputted to identify the user as a buyer
- **Message User:** Buyer's way to contact a seller about questions they have prior to a purchase or issues with a purchased item.

- **Make a purchase:** The buyer may purchase an item according to their needs with a chosen payment method.
- **Rate the purchase:** The buyer can give a rating of an item that they purchased on a scale of 1 to 10.

5.8.3. Actor: Seller (Samuel)

- **Login:** Username and password are inputted to identify the current user as a specific seller.
- **Message User:** If a buyer has gotten in contact regarding a purchase listing, a seller can send messages to the buyer through that listing page.
- **Create Listing:** By inputting name, price point, exchange method, and optional tags and images, a seller can create a listing for a product that is visible to all buyers.
- **Manage Listing:** Previously generated listings can have fields edited, with earlier forms of the listing being stored for posterity.
- **Discount Listing:** The seller may apply time- or buyer-specific discounts to the price point of a listing, with automated creation, management, and removal after the time or purchase has elapsed.

5.9. Use-Case Model Scenarios

5.9.1. Actor: Admin (Dragon Phiansin)

- **User-case Name: Delete/Ban User**
  - **Initial Assumption**: A user asks for login credentials (user account) to be deleted or the user does not act according to code of conduct.
  - **Normal**: Admin will delete a user's login credentials (user account) corresponding to the user's behalf
  - **What Can Go Wrong:** The user's login credentials are not deleted or are temporarily deleted.
  - **Other Activities**:
  - **System State on Completion**: The user's account is deleted.
- **User-case Name: Add Promotions**
  - **Initial Assumption**: The admin has established a promotion sale for an event or category of products related to promotional sale.
  - **Normal**: A promotional sale is applied to a set category of products.
  - **What Can Go Wrong**: A promotional sale is applied to products unrelated to a certain product or the sale of products is too high.
  - **Other Activities**: The admin can change or add a discount to prices to certain products and not just a whole category.
  - **System State on Completion**: A promotional sale is applied to products of the same category related to promotion events.
- **Use-Case Name**: Login
  - **Initial Assumption**: The admin has a registered account to login. That account is stored in the application database. The admin has access to the username and password to access their account.
  - **Normal**: The admin will enter the username and password and be granted access to their account.
  - **What Can Go Wrong**: The username or password may fail to match the stored username and password hash, preventing the admin from accessing their account.

    - **Other Activities**: The admin can reset their account password with a forgot-password page.
    - **System State on Completion**: The admin is logged in and can view their and other user's information page, as well as their listings aggregators.
- **Use-Case Name**: Message User
    - **Initial Assumption**: The admin has either messaged the user (Buyer or seller).
    - **Normal**: A message is sent to the user, which they can then view and respond to.
    - **What Can Go Wrong**: A message doesn't get sent.
    - **Other Activities**: A report option feature can report suspicious activity during messaging between users.
    - **System State on Completion**: A message sent by an admin is sent and can be seen and responded to from other users.
- **Use-Case Name**: Manage Listing
    - **Initial Assumption**: The admin is logged into their account, and has the ability to view listings of other users.
    - **Normal**: The admin can view listings of other users by clicking on another user's account.
    - **What Can Go Wrong**: A user's listing(s) cannot be seen and cannot click on the user account.
    - **Other Activities**: Admin can also delete a user's listing according to the user.
    - **System State on Completion**: The listing now shows the changes on all pages, as well as a link to show previous versions of the same listing.

5.9.2. Actor: Actor: Buyer (Jeong Kim)

- **Use-Case Name**: Login
    - **Initial Assumption**: The system asks for the username and password of the buyer and the buyer logs in with the username and the password they signed up with.
    - **Normal**: The user will enter the username and password and have access to their account.
    - **What Can Go Wrong**: The username or password may fail to match the stored username and password.
    - **Other Activities**: The buyer can reset their account password with a forgot-password page.
    - **System State on Completion**: The buyer is logged in and can view their user profile, items that are currently listed to be sold and their purchase history.
- **Use-Case Name**: Message User
    - **Initial Assumption**: the buyer has either messaged the seller on one of the seller's listings pages or the user is an administrator.
    - **Normal**: A message is sent to the user, which they can then view and respond to.
    - **What Can Go Wrong**: A message doesn't get sent.
    - **Other Activities**: Report option for suspicious listings.
    - **System State on Completion**: A message gets sent to a user and can receive a response by them.
- **Use-Case Name**: Make a purchase
    - **Initial Assumption**: The buyer is logged in to their account and has the listing of their interest opened.

- **Normal**: The buyer is able to make a purchase by pressing the purchase button and choosing a method of payment.
- **What Can Go Wrong**: A buyer can make an accidental purchase.
- **Other Activities**: Items can be added to the cart before making a purchase.
- **System State on Completion**: Item is purchased and the confirmation message gets sent to both buyer and seller .

- **Use-Case Name**: Rate a purchase
  - **Initial Assumption**: The buyer has purchased and received an item from the seller and has put it to use.
  - **Normal**: The buyer leaves a rating for the item that they purchased and it goes towards the seller's rating.
  - **What Can Go Wrong**:
  - **Other Activities**: Make a claim of an item if a wrong item is received or never arrived.
  - **System State on Completion**: The rating is given to the seller.

5.9.3. Actor: Seller (Samuel)

- **Use-Case Name**: Login
  - **Initial Assumption**: The seller has a registered account to login. That account is stored in the application database. The seller has access to the requisite username and password to access the account.
  - **Normal**: The seller will enter the username and password and be granted access to their account.
  - **What Can Go Wrong**: The username or password may fail to match the stored username and password hash, preventing the seller from accessing their account.
  - **Other Activities**: The seller can reset their account password with a forgot-password page.
  - **System State on Completion**: The seller is logged in and can view their user information page and listings aggregator.
- **Use-Case Name**: Message User
  - **Initial Assumption**: Both the seller and other user have registered accounts, and are logged in to those accounts. In addition, the user has either messaged the seller on one of the seller’s listings pages or the user is an administrator who has previously contacted the seller.
  - **Normal**: A message is sent to the user, which they can then view.
  - **What Can Go Wrong**: The message can fail to be sent, preventing the other user from viewing it. In this case, a popup should alert the sender that the message failed.
  - **Other Activities**: A report feature, to allow users to alert administration regarding suspicious messages.
  - **System State on Completion**: The message is available to the other user, and an interface is provided for a response to be written by them.
- **Use-Case Name**: Create Listing
  - **Initial Assumption**: The seller is logged into their account.
  - **Normal**: The seller inputs the necessary information to generate a listing, which is then added to the database.
  - **What Can Go Wrong**: The listing can fail to be created, in which case we should remain on the current page for the seller to try again.
  - **Other Activities**: A time for a listing to be made visible to all users can be set.

    - **System State on Completion**: The listing is now visible through both the search interface and the seller's listings page.

- **Use-Case Name**: Manage Listing
    - **Initial Assumption**: The seller is logged into their account, which has an active listing.
    - **Normal**: The seller can alter single elements of the listing without recreating it or altering the other elements.
    - **What Can Go Wrong**: The listing can fail to update.
    - **Other Activities**: The listing can be deleted altogether through a batch edit button.
    - **System State on Completion**: The listing now shows the changes on all pages, as well as a link to show previous versions of the same listing.
- **Use-Case Name**: Apply Discounts on Listing
    - **Initial Assumption**: The seller is logged into their account, which has an active listing. In the case that the discount is buyer-specific, the seller must also know the username of the buyer account.
    - **Normal**: The seller can set a percentage or exact discount to a listing.
    - **What Can Go Wrong**: The discount can fail to be set, or the time can be set incorrectly.
    - **Other Activities**: Batch listing discount; i.e., a discount can be applied to a group or all listings created by the seller.
    - **System State on Completion**: A discount is visible on the listing for either all users for the duration set, or for one specific user until the order is completed.

6. Design Documents

6.1. Software Architecture

6.2. High-Level Database Schema

6.3. Software Design

6.3.1. State Machine Diagram: Actor Name (Responsible Team Member)

6.3.2. State Machine Diagram: Actor Name (Responsible Team Member)

6.3.3. State Machine Diagram: Actor Name (Responsible Team Member)

6.4. UML Class Diagram

7. Scenario

7.1. Brief Written Scenario with Screenshots