Software Requirements Specification

for

SneakerHeadz

Version 1.0 approved

Prepared by

Group 10:

Ehren Achtermann	**eta55**	**eta55@msstate.edu**
Trey Fullwood	**rf802**	**rf802@msstate.edu**
Kennedy Keyes	**kfk38**	**kfk38@msstate.edu**
Austin Wheeler	**aaw345**	**aaw345@msstate.edu**
Ander Talley	**ajt432**	**ajt432@msstate.edu**

SneakerHeadz Co.

February 10th, 2023

0. Contents

1. Introduction

2. Overall Description

3. System Features

4. Other Nonfunctional Requirements

5. Other Requirements

1. Introduction

SneakerHeadz is an e-commerce website that allows the user to readily find shoes the user wants to buy. The website will also allow the user the option to sell shoes the user may not want anymore. The storing of the data required to portray different shoe options and compare shoe options with each other will be accomplished with an SQL database. The SQL database will also help with the transaction of purchases.

1.1 Purpose

This SRS document is designed to give an explanation in full about the software requirements for SneakerHeadz. The software requirements will thoroughly communicate the design and features of SneakerHeadz to the stakeholders and to anyone who reads the SRS document. This document is designed to describe the entire website.

1.2 Document Conventions

This document uses 12 pt size font for body paragraphs, 24 pt size font for title headers, 18 pt size font for subheaders, 12 pt size font for sub subheaders, and 10 pt size font for page numbers/headers. All headers of sections are bolded. All of the text font type is Arial, and all of the text font style is normal text. The margins are all the same at one inch. The contents and cover pages are exceptions.

1.3 Intended Audience and Reading

The intended audience for reading the SRS document is the professor and teaching assistant of the class. As well as, anyone interested in developing or the development of an e-commerce website. The document is designed to be read linearly. First, the reader should start with the introduction, and move to the overall description of the product. After getting familiar with the product and what the overall design is, the reader should next read the system features to get a more in-depth description on features used to create the website/product.

1.4 Product Scope

The product is to be an e-commerce shoe website called SneakerHeadz which is used to allow a user to log on to the website, and enter a username and password to get on either a customer or an admin account. A customer can either search for shoes to buy or sell shoes they already own. A customer will also be able to compare shoes from different sellers. An admin account will be allowed to oversee the website's transactions and functions to make sure purchases go through properly, and moderate new user accounts. These functions will be designed using various softwares and databases. A few examples are VS code and MySQL.

1.5 References

Amazon.com. Spend less. Smile more.

Electronics, Cars, Fashion, Collectibles & More | eBay

Buy Shoes & New Sneakers - Real Yeezys, Retro Jordans, Nike (stockx.com)

Example Software Requirements Specification (SRS) | ReqView Documentation

2. Overall Description

2.1 Product Perspective

The products are various brands, sizes, and classifications of shoes. The website will offer the most popular types of sneakers based on comfort, performance, and popularity. Each unit is able to be altered in quantity, colorway, and size based on availability and popular demand. SneakerHeadz will be a shoe resell aftermarket company. SneakerHeadz will also operate via a user database where users can create, manage, or delete accounts.

2.2 Product Functions

- The seller will be able to describe their shoes by selecting radio buttons, drop down menus, and entering in data fields with the web interface.
- The seller can add as many shoes to the marketplace.
- The seller can receive payments by different payment options our system can handle.
- The seller can analyze a dashboard or hub that visualizes overall credit from every purchase.
- The buyer will be able to search/filter, compare, watch, buy, and ship shoes throughout the web interface by using the marketplace’s system, buttons, display filters, and text fields portrayed on the device.
- The buyer can watch shoes by clicking a button that will store the shoe data in the buyer’s account’s watchlist. The data will stay, regardless if it sells or not, and will allow the buyer to delete the shoe from the list or return to the shoe’s selling page.
- The buyer can select different payment types, shipping rates and addresses during checkout, which can be stored in a database if wanted, by selecting radio/check mark buttons and entering in data fields. May get shipping and invoice notifications by email.
- The buyer will be able to analyze their purchase history and incoming/pending orders.
- The buyer can issue a return by selecting a shoe order out of the purchase history in their account. This will trigger a “reason” and refund statement that will be addressed by the seller. They can decline or accept. If issues occur, the admin will have to interfere.
- The product will allow buyers and sellers to delete their accounts if wanted. The admin will permanently delete everything relevant. However, buyers will still see their purchases from deleted sellers.

- The product will demand all users to authenticate by signing up/logging in with an email, username, and password before taking any monetarily action and logging out when finished.
- The product will allow buyers and sellers to customize their profiles to prove more authenticity to each other ([private] verifying identity with personal information, [public - can be optional] uploading their bio, location, picture, and displaying mandatory seller ratings).
- The product will allow buyers and sellers to communicate before, during, and after each purchase by sending inquiries through email.
- The product will allow buyers and sellers to rate each other after each successful shoe purchase and delivery by entering in an optional text field and selecting an amount of stars (out of 5). If nobody or only one complies, the admin will force a rating through the system.
- The product will send data to and from different payment and delivery companies.
- The admin will be able to perform permissible abilities through the database for authenticity of products, buyers, and sellers.
- The admin will be able to look in the database to see different user actions and interfere.

2.3 User Classes and Characteristics

The typical individuals (buyers and sellers) interacting with our web interface should have the basic knowledge of web environments, managing mouse and keyboard tools, payment and shipping types, and navigating usage on various devices (phone, desktop/laptop, and a tablet). The product's front end layout will be eye appealing and easy to understand for those users.

The individuals interacting with the backend, specifically the admin(s), will see more of a MySQL database system and will share data with contracted third parties. The users should have a structured understanding of MySQL database systems in order to operate.

2.4 Operating Environment

SneakerHeadz is a web service that will be available on all supported web platforms such as Google Chrome, Mozilla Firefox, and Microsoft Edge. The common operating systems used are Windows OS, Linux OS, and macOS. SneakerHeadz will be running on a web service that will contain MySQL, JavaScript, and Python.

2.5 Design and Implementation Constraints

Shoe retailers and resellers are heavily affected by a myriad of outside factors which play a pivotal role in sales and overall company performance. In order to track and manage inventory, a tracking mechanism would have to be created or purchased. There are multiple services available that specialize in inventory management (this may or may not be necessary for the project). Also, the business must have some level of customer service via a phone number or email. There are several liabilities/incidentals in which third party services could be held responsible such as the shipping company, inventory system, or even the customers themselves. Ultimately, SneakerHeadz is only responsible for receiving and selling the product to the customer.

3. System Features

3.1 TEMPLATE

3.1.1 Description and Priority

Login

- The ability for returning users to log into their account (9)

Sign Up

- Ability for a user to make an account (9)

Logout

- The ability for a user to safely log out of the system (9)

Delete Account

- Ability for a user to delete their account and their information (8)

Buy

- Ability for a user to buy whatever shoe they are shopping for (8)

Sell

- Ability for a user to upload a shoe to the website with the intention to sell (8)

Purchase History

- Show user their previous purchases (3)

Account Info

- Allow users w/ account to view their profile and edit it as necessary (6)

Search

- Ability for users to search for a specific shoe. If shoe is being sold, it will pop up, will not otherwise (6)

Filter

- Deploys a dropdown that will show various brands and filter database based on brand for user's choice (4)

Delete Item

- Ability for a seller to take down a shoe they previously listed to sell (7)

Checkout

- Allows user to buy the shoe they found listed (9)

3.1.2 Stimulus/Response Sequences

Login

- From the home page -> User clicks "Login/Sign up"
- Will log in user if they have an account w/ proper email & password

Logout

- Only exists if user is signed in
- dropdown window with several options limited to users w/ accounts, click logout, system logs the user out

Delete Account

- Only exists if user is signed in
- Dropdown window with several options -> click "Account Options" -> link to delete account

Buy

- click link to browse shoes available -> click item -> click buy option

Sell

- Link on homepage "Sell" -> upload details of the shoe -> click "Add to Market"

Purchase History

- Dropdown window with several options -> click "Account Options" -> click "Purchase History"

Account Info

- Dropdown window with several options -> click “Account options” -> click “Account information”

Search

- Search bar to look up specific shoes

3.1.3 Functional Requirements

REQ-1: Database (storage)

- ○ Server to store user account info and anything being sold on the site. This will also keep track of order histories (who bought what for how much from whom and when). Languages TBD

REQ-2: Web Interface

- ○ User must be able to interact with the software to buy/sell/browse shoes
- ○ User must be able to log in/out
- ○ User must be able to create an account

3.2 User Interface

3.2.1: Description and Priority

The User Interface should hold all of the components of what a user can access when navigating the website. For example, the home page should clearly state the majority of the other functions that the service provides, such as “Registration,” “Login,” “Cart,” “Search,” and others. Users, depending on their privilege in the website, should be able to view different menus and selections based on their privilege. The User Interface is a high priority, as it is the direct connector of consumer to product.

3.2.2: Stimulus / Response Sequences

Home Page: The User will be able to find most of the other functions that the service provides here. It should be the landing page for all new users, unless they navigated via a direct link to a product, as well as the landing page for returning users when they are navigating specifically to the website without a link to somewhere else. This should be considered the “root” page for users.

Interaction 1: The user selects “Registration.” The website takes the user to the registration page. The user enters in their information to register for an account. The website sends the details to the database, storing a new user ID for the user as well as creating a user page for the user.

Interaction 2: The user selects “Search.” The website takes the user to the search page, allowing it to enter in filters, keywords, price ranges, and other filter settings to find the shoes the user is looking for.

3.2.3: Functional Requirements

REQ-1: “Database:” The software will require functioning databases to hold all of the relevant information and tags to ensure the user can use the service correctly. All items for sale will have tags that describe what the item is, as well as associated prices and other relevant information. All users will have their userID stored in a database, as well as general account information. There should be another database to store financial information properly, so that when a sale goes through, the payment is received and distributed correctly.

REQ-2: “Search:” The search feature of the service will need to be able to filter through various keywords and tags to find items that fit what the user is looking for. We will need to be able to implement an efficient sorting algorithm that can search for and find specific tags in the item database that correlate with whatever the user is looking for, and, assuming the tags the user is looking for do not coincide with anything in the database, a case for “no matching items found.”

REQ-3: “Login/Registration:” The registration feature of the service will need to be able to accept usernames/passwords and store them in the database. Depending on how we limit password creation / username creation, we may need to implement a method of warning the user that their inputs were invalid. For login, there should be a method of retrieving a forgotten password(?)

REQ-4: “User Account Page:” As a customer of the service, a normal user will have access to a cart, their order history, their account details, and any preferences they have set. Most of these details will need to be stored in the various databases that we will have storing data, and should be able to be found via the user’s internal ID. Assuming the user is an admin, then the user should have access to any sort of moderation log, such as removing items from the store.

3.2.3 User Stories

User 1: “Searching” User 1 wants to search through different shoes so they can find the perfect pair for them: User 1 will visit the website that will display the homepage. User 1 will click and enter in the search bar their certain criteria. The page will update and display shoes with similar or specific content related to the search. User 1 can also interact with the search filters, which includes price range/ascending/descending, color, relevance, shoe size, width, free shipping, and seller rating. User 1 will find, compare, and click on the shoe that will take them to the buying page. They can either “Add to Cart” or place item(s) on their “Watchlist” to save for later purchase.

User 2: “Buying” User 2 decides to buy one of the pairs of shoes that they have found. User 2 selects “Add to Cart,” and is then sent to “Checkout,” where they then enter their information and set any shipping alerts to successfully purchase the shoes and await their delivery. Their order will appear in the “Purchase History” to view and will instruct a rating form.

User 3: “Selling” User 3 has shoes they want to sell. User 3 navigates to the “Make a Listing” tab, and puts shoes up as a listing, providing relevant tags. Another user buys the shoes, and User 3 receives money for the shoes. User 3 can analyze any monetary data in their profile later.

User 4: “Deleting Account” User 4 has used the service for a few months now, but would like to no longer use the service. User 4 navigates to their account page and selects “delete account.” The user is prompted to enter their password, and the account is successfully deleted and removed from the corresponding databases.

User 5: “Login” User 5 navigates to the Login page. User 5 enters their username and password in the corresponding boxes, and then clicks “login.” If User 5 has successfully entered their correct information, they login. Otherwise, text appears informing the user that they have entered either a wrong username or password. Other guidance would include signing up with email, username, and password in text fields or submitting a forgotten information form.

User 6: “Logout” User 6 selects the “logout” button. User 6 is logged out of their account.

User 7: “Return” User 7 wants to return an item and receive a refund. User 7 will visit their profile, then “Purchase History” to select their order. User 7 will select “Return Order” to start a return/refund statement form. User 7 will have to respond to the form their reasoning of returning, shipping options, and wait for the seller to respond. If there is no response, the issue can be taken care of by the admin. The refund will be processed when the item is delivered back to the seller.

User 8: “Profile Customization” User 8 wants to customize their profile and have a way to communicate with others. User 8 will enter in the selected fields their information (like bio, location, payment methods, general saved account information, and pictures). Afterwards, User 8 can chat or send emails to sellers with correct information by clicking on their profile’s “Chat” button.

User 9: “Admin Backlog” User 9 is an admin for the website and wants to look through the transaction history on the website to dispute a claim made by another User. User 9 goes to their profile page and selects “Admin,” which is only there as their account is an Admin account. User 9 then selects “Backlog” and is greeted with the website’s transactions.

User 10: “Admin Delete” User 10 is an admin for the website and has found a user that is listing an item that doesn’t match the images provided. User 10 has a button on the listing that non admin users do not have, which is “Delete Listing.” User 10 selects it, and is then prompted if they are sure they want to delete the listing.

User 11: “Admin Ban” User 11 is an admin for the website and has found a user that is abusing the selling feature by trying to scam other users. User 11 navigates to User 11’s profile, and selects the “Admin” button. User 11 then selects “Ban a User,” and then types in the offending user’s UserID. User 11 then confirms the UserID after being prompted via the website.

4. Other Nonfunctional Requirements

4.1 Performance Requirements

- Logging in to the users/admins account should take less than 1 second.
- Displaying different shoes should take less than 5 seconds.
- Being able to sell shoes should take less than 1 second.
- Being able to buy shoes should take less than 1 second.
- Updating and saving a user's personal information should take less than 1 second.
- Displaying a Master Log to admins should take less than 5 seconds.

4.2 Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>

4.3 Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

4.4 Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

5. Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Privacy Policy:

DISCLAIMER:

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>