

---

# **Software Requirements Specification**

**for**

# **GameVault**

**Version 1.1 approved**

**Prepared by Jalon Glenn, Jimmy Whitley, Felix Airhart, Caid Franks**

**SE Lab Group 3**

**09/30/25**

## Revision History

Version	Date	Author (s)	Comments
1.0	8/27/25	Glenn, Whitley, Airhart, Franks	Initial Version
1.1	9/30/25	Glenn, Whitley, Airhart, Franks	Revisions, rewording, and refining
1.2	10/2/25	Glenn, Whitley, Airhart, Franks	Revisions, rewording, and refining
1.3	10/3/25	Glenn, Whitley, Airhart, Franks	Requirement numbering adjustments

<b>1. Introduction.....</b>	<b>1</b>
1.1 Purpose.....	1
1.2 Document Conventions.....	1
1.3 Intended Audience and Reading.....	1
1.4 Product Scope.....	1
1.5 References.....	2
<b>2. Overall Description.....</b>	<b>2</b>
2.1 Product Perspective.....	2
2.2 Product Functions.....	3
2.3 User Classes and Characteristics.....	3
2.4 Operating Environment.....	4
2.5 Design and Implementation Constraints.....	4
<b>3. System Features.....</b>	<b>4</b>
3.1 Universal Features.....	4
3.1.1 Login (Milestone 1).....	4
3.1.2 Log Out.....	6
3.2 Customer Features.....	7
3.2.1 Customer Sign Up.....	7
3.2.2 Browse Products.....	9
3.2.2.1 Filter Products.....	10
3.2.2.2 View Product Listing (Milestone 2).....	10
3.2.2.3 Go to Next Page.....	11
3.2.3 Search Products.....	12
3.2.4 Add to Cart.....	13
3.2.5 View/Edit Cart.....	14
3.2.6 Checkout.....	15
3.2.7 View Order History.....	16
3.2.8 Make Returns.....	17
3.2.9 Edit Account Information.....	18
3.3 Vendor Features.....	19
3.3.1 Vendor Sign Up.....	20

3.3.2 Create, Modify, and Remove Listings.....	21
3.3.3 View and Act on Invoices.....	23
3.3.4 Manage Returns.....	25
3.3.5 Add/Edit Banking Info.....	26
3.4 Administrator Features.....	27
3.4.1 Create Additional Admin Accounts.....	27
3.4.2 Approve/Deny Listings.....	28
3.4.3 Approve/Deny Vendors.....	30
3.4.4 View Action Logs.....	32
3.4.5 Take Down Listings.....	32
3.4.6 Ban Users.....	33
<b>4. Other Nonfunctional Requirements.....</b>	<b>34</b>
4.1 Performance Requirements.....	34
4.2 Safety Requirements.....	35
4.3 Security Requirements.....	35
4.4 Software Quality Attributes.....	35
<b>5. Other Requirements.....</b>	<b>36</b>

# **1. Introduction**

## **1.1 Purpose**

This software requirements specification (SRS) document for GameVault is intended to outline in detail the scope and functionality of the web app. This includes all functional and non-functional requirements, constraints, limitations, and other features. It is meant to communicate all relevant information pertaining to the e-commerce website, so that anyone reading this document has a clear view of the design plans and implementation.

## **1.2 Document Conventions**

- Priorities:
  - Critical - must be implemented before any other feature
  - High - major functionality and many dependencies
  - Normal - required for the minimum viable product
  - Low - ideal features; not required
- Requirement ID:
  - Each requirement has an ID in the form of REQ-#
- User Stories:
  - Split into universal (3.1), customer (3.2), vendor (3.3), and admin (3.4) feature sections
- Feature References:
  - Written as “#.#.#Feature Name, User Story #, Step(s) #-#”
  - Provides feature ID and steps to follow to get user to certain point in other feature
- Formatting:
  - Main sections: Heading 1
  - Sub-sections: Heading 2 to Heading 4 as subsections get deeper
  - Font: Times New Roman
  - Body font size: 12
  - Body alignment: Justified

## **1.3 Intended Audience and Reading**

The intended audience for this document includes the customer, project stakeholders, and the development team. It serves as a guide for developers, a reference for testers and maintainers, and a clearly communicated outline of the website’s functions and requirements.

## **1.4 Product Scope**

Our goal is to build a fully functional E-Commerce website, where merchants can sell video game keys to customers. This market will let merchants sell physical video games for profit and let customers browse/buy video games from the merchants. The seller will be able to add, sell, and receive payments for their products. The buyer will be able to search, compare, buy, and return video games. The admin will have the ability to moderate users and products as they see fit,

including a log that gives an overview of all user activity. The website is compartmentalized based on user type, as the permissions and security levels increase from buyer to seller to admin. This website is a single subsystem and the rest of the SRS document will detail its capabilities.

## **1.5 References**

For our frontend development, we will use Blazor: a mix of HTML, CSS, C#, & JavaScript. Here is the documentation for it:

<https://learn.microsoft.com/en-us/aspnet/core/blazor/?view=aspnetcore-9.0>

For our backend development, we will use Firebase. Here is our documentation for it:

<https://firebase.google.com/docs/guides>

We will be using the Square sandbox system to safely test Square APIs/transactions without affecting real data. Here is the documentation for it:

<https://developer.squareup.com/docs/devtools/sandbox/overview>

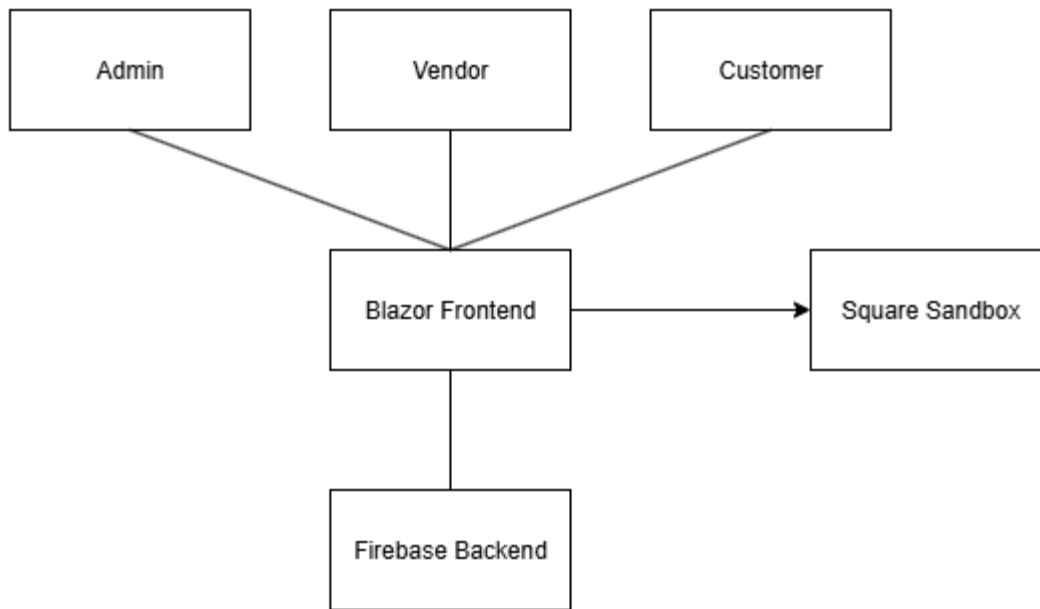
## **2. Overall Description**

### **2.1 Product Perspective**

This web application is a brand new product, which is self-contained and came to fruition as an agreed upon project idea by the members of our team. We will be building this from scratch - everything from frontend to backend.

Our users will be able to access the e-commerce shop via web browsers, where they interact with the frontend to utilize the various features and specific privileged actions we build into the system. The frontend interface is where the buying, selling, and administrative managing will take place. The frontend will be created using Blazor, which ties in nicely with our backend.

As for the backend, we will be using Google's Firebase system to store and authenticate user information, including contact info, payment info, and transaction history. As this project is not meant to accept real transactions at this time, we will also be using Square's sandbox feature to simulate transactions and then store them using Firebase.



## **2.2 Product Functions**

- Users of three types, Customers, Vendors, and Administrators, can log in to their accounts to perform different actions and log out.
- Customers can apply for accounts and browse, filter, search for, buy, and return different categories of products.
- Vendors can apply for accounts, create, modify, and remove their listings, and manage and fulfill invoices and returns.
- Administrators can approve/deny all new users and product listings, as well as remove existing users and listings, and monitor all activity on the website.

## **2.3 User Classes and Characteristics**

There are three classes of users that will use this product: customers, vendors, and administrators. All user classes have the ability to login and log out, as well as change their personal information.

Customers have the least amount of control of the product, but are the most important type of user on the site. They will be browsing and buying items, which is what makes the product profitable. They have the lowest privilege levels, as they are essentially only allowed to browse, buy, change account information, and return items.

Vendors bring the items to the marketplace and submit their product listings for approval. They have a medium amount of privilege, as they are allowed to add/remove listings for buyers. They can also communicate with buyers about various things, including returns. They are also responsible for receiving and processing the payments for their listings.

Administrators have the highest privilege level, allowing them to control almost every aspect of the market. They can approve/deny new sellers, as well as block sellers who they deem problematic for various reasons. This extends to individual products that sellers list, which may not align with the rules, regulations, or values of the market. They have a log of every user action, whether it be a buyer or a seller. From this log, they can make decisions as previously mentioned to manipulate the site's users and products as they see fit.

## **2.4 Operating Environment**

- Google Firebase backend
- Blazor WASM (Web Assembly)
- Platforms: Safari, Chromium-based browsers (Version numbers TBD)

Our website will have a Blazor Web Assembly frontend and have Google Firebase as the backend. The platforms this website will work on are Safari and any Chromium-based browsers from the last few years.

## **2.5 Design and Implementation Constraints**

Limitations:

- This application is not guaranteed to work on Firefox or other unspecified web platforms.
- There are no reviews or user communication systems.

# **3. System Features**

## **3.1 Universal Features**

### **3.1.1 Login (Milestone 1)**

A universal login page, shared by all three user types, which every page redirects to if not logged in, forcing every user to authenticate before using the website.

**Priority: Critical**

#### **User Story 1**

As a user, I want to log in.

1. The user navigates to the website.
2. The user enters their email and password.
3. The user clicks "Log In".
4. The system validates credentials.
5. The system directs the user to the home page corresponding to the user type.



**Exceptions:**

- Incorrect email/password combination - See User Story 2
- User is offline - system requests will fail, and the user will be notified that the server cannot be reached

**User Story 2**

As a user, I want feedback from the system if my credentials are incorrect and try again.

1. Invoke 3.1.1 Login, User Story 1, Steps 1-2
2. The user enters an incorrect email/password combination.
3. The system rejects credentials.
4. The system notifies the user that the combination is invalid via on-screen message and is prompted to try again.

**User Story 3**

As a user, I want to reset my password if I forget it.

1. The user navigates to the website.
2. The user clicks “Forgot password?” option.
3. The system directs the user to the forgot password page.
4. The system prompts the user to enter their email.
5. The user enters their email then presses “Send reset link.”
6. The system sends an email to the user with a reset password link and redirects to the login page.

**User Story 4**

As a banned user, I want to be notified that my account is banned when I try to log in.

1. Invoke 3.1.1 Login, User Story 1, Steps 1-4
2. The system generates an on-screen error message stating that the user is banned and login is not allowed.

**Requirements:**

REQ-1: The user shall be able to access the website over the internet.

REQ-2: The user shall be served the login page first by the website.

REQ-3: The user shall be required to input an email address on the login page.

REQ-4: The user shall be required to input a password on the login page.

REQ-5: The user shall be allowed to submit a login request on the login page.

REQ-6: The user shall be authenticated by the system based on a submitted email address and password.

REQ-7: The user shall be redirected to their corresponding homepage upon login.

REQ-8: The user shall receive an error message on the login page if the system is inaccessible.

REQ-9: The user shall receive an error message if the system rejects the credentials.

REQ-10: The user shall be able to reset their password on the login page.

REQ-11: The user shall be required to input their email address in the password reset request.

REQ-12: The user shall receive a temporary password generated by the system on a successful password reset request.

REQ-13: The user shall receive the temporary password on a successful password reset request

REQ-179: The user shall receive an error message on the login page if they have been banned.

### **3.1.2 Log Out**

Accessible from any page, a user may log out of their account and be automatically redirected back to the login page.

**Priority: High**

#### **User Story 1**

As a user, I want to log out of my account.

1. The user scrolls down to the bottom of the page.
2. The user clicks the “Log Out” button.
3. The system logs the user out and redirects the user to the login page.

#### **Requirements:**

REQ-15: The user shall be able to log out from every page.

REQ-16: The user’s session shall end when they submit a successful log out request.

REQ-17: The user shall be redirected to the login page when they log out.

### **3.1.3 Change Password**

Allows the user to change their password while logged in.

**Priority: Normal**

#### **User Story 1**

As a customer, I want to update the password on my account.

1. Invoke 3.1.1 Log In, User Story 1
2. The user clicks the account icon in the top right of the page.
3. The user clicks “Update Password”.
4. The user is prompted to enter their current password, then click “Submit”.
5. The system authenticates the password. On success, the user is prompted to enter their new password twice. On failure, the user is prompted to try again.
6. The user enters their new password twice.
7. The user clicks “Update”.
8. The system validates credentials, then updates email in the database.
9. The system notifies the user of success via on-screen message.

#### **Requirements:**

REQ-14: The user shall be able to change their password on the account information page.

REQ-15: The user’s changes to password shall be logged.

REQ-16: The user shall be required to enter their password to change their password.

REQ-17: The user shall be prevented from changing their password unless the new password is different from their old password, meets the security requirements, and matches their second inputted password.

REQ-18: The user shall be required to input their new password a second time to change their password.

REQ-19: The user shall be shown a success or error message based on the results of changing account information.

## **3.2 Customer Features**

### **3.2.1 Customer Sign Up**

A page that allows a new customer to request an account if they do not already have one.

**Priority: High**

#### **User Story 1**

As a customer, I want to sign up.

1. The user navigates to the website
2. The user selects “Don’t have an account? Sign up!” option at the bottom of the login screen.
3. The user is directed to the sign up page.
4. The user selects “Customer” from “User Type” drop down.
5. The user enters an email address.
6. The user enters a password meeting requirements, then re-enters the password for verification purposes.
7. The user clicks the “Sign Up” button.
8. The user is redirected to the login page.

#### **User Story 2**

As a customer, I want to be notified if the email I try to sign up with already exists so that I can try again or go back to the login screen.

1. Invokes 3.2.1, User Story 1, Steps 1-7
2. The system checks the database to see if the email already exists, finds an instance, and returns an error.
3. The user is notified that an account already exists with this email.
4. The system redirects to the login page.

#### **User Story 3**

As a customer, I want to be notified if my password does not meet the requirements, and I want to be shown the requirements when I attempt again.

1. Invoke 3.2.1, User Story 1, Steps 1-7
2. The user is notified that the password doesn’t meet the requirements.
3. The system displays the password requirements and redirects the user to the login page.

**Requirements:**

REQ-20: The user shall be able to access the sign up page from the login page.

REQ-21: The user shall be required to select between “Customer” and “Vendor” account types on the sign up page.

REQ-22: The user shall be required to input an email address on the sign up page.

REQ-23: The user shall be required to input a password on the sign up page.

REQ-24: The user shall be required to input their chosen password a second time on the sign up page.

REQ-25: The user shall be prevented from signing up if both passwords do not match.

REQ-26: The user shall be prevented from signing up if the password does not meet the security requirements.

REQ-27a: The user shall be able to create a new customer account (conditional on REQ-20 and REQ-21).

REQ-27b: The user shall be able to submit a new vendor request (conditional on REQ-20 and REQ-21).

REQ-28: The user shall be redirected to the login page when they submit a new account request.

REQ-29: The user shall be informed on the login page that their new vendor request has been received and is pending.

REQ-30: The user shall have their new account request logged by the system.

REQ-31: The administrator(s) shall be notified when a new vendor request is submitted.

REQ-32: The user shall have their new vendor request stored for action by administrators.

REQ-33: The user’s new account request shall be verified not to conflict with an existing account of the same email address.

REQ-34: The user shall receive an error message on the sign-up page if their new account conflicts with an existing account. (see REQ-31).

REQ-35: The user shall receive an error message if both passwords do not match.

REQ-36: The user shall receive an error message and instructions if their chosen password does not meet the security requirements.

### **3.2.2 Browse Products**

A page that allows a customer to view and browse through available products, displaying pertinent information on each listing.

**Priority: High**

#### **User Story 1**

As a customer, I want to browse the products.

1. Invoke 3.1.1 Login, User Story 1.
2. The system checks user type and loads its corresponding home page.
3. The user is directed to the home page.
4. The user sees a grid of suggested products.
5. The user can select products from this page (3.2.2.2), filter listings (3.2.2.1), search (3.2.3), or go to the next page.

#### **User Story 2**

As a logged in customer, I want to browse products from any page.

1. The user clicks the home button on the navigation bar.
2. The system directs the user to the home page and populates an unfiltered listing grid from the listings database.

#### **Requirements:**

REQ-37: The customer's home page must be the browse products page.

REQ-38: The customer shall be shown a grid of product listings on the browse products page.

REQ-39: The customer shall be shown a thumbnail image, product name, price, and the listing owner's display name for each product listing on the browse products page.

REQ-40: The user shall be able to return home with a button press from every page.

REQ-41: The user shall be redirected to their corresponding homepage when they press the home button while logged in.

#### **3.2.2.1 Filter Products**

While browsing products, allows customers to filter shown listings by category.

**Priority: Normal**

## **User Story 1**

As a customer, I want to filter the product listings.

1. Invoke 3.2.2 Browse Products, User Story 1
2. The user checks or unchecks boxes from the filter list on the left.
3. The user clicks “Apply Filter”.
4. The system repopulates the listing grid with listings from the database meeting the filter criteria.

### **Requirements:**

REQ-42: The user shall be shown a list of filters on the browse products page.

REQ-43: The user shall be able to select none, one, or multiple options from the list of filters on the browse products page.

REQ-44: The user shall be able to apply a filter by submitting a filter search request on the browse products page.

REQ-45: The user shall cause the website to retrieve a group of listings that meet the filter when they submit a filter search request.

REQ-46: The user shall be shown the results from the system on the browse products page when a filter search request is submitted.

### **3.2.2.2 View Product Listing (Milestone 2)**

While browsing products, or based on a search result or link, allows users to view more information about a product or begin the purchasing process for that product.

### **Priority: Critical**

## **User Story 1**

As a customer, I want to view a product listing.

1. Invoke 3.2.2 Browse Products, User Story 1
2. The user clicks a listing.
3. The system directs the user to the listing page, which shows the product photos, price, description, and related items; also shows an “Add to Cart” button. (Includes a back option that reloads previously viewed page of listings)

**Requirements:**

REQ-47: The user shall be able to access each listing page via a unique URL, corresponding to each unique listing.

REQ-48: The user shall be sent to the corresponding listing page when they click a listing on the browse products page.

REQ-49: The user shall be shown a thumbnail image, product name, price, listing owner's display name, product categories, and description on each listing page.

REQ-50: The customer shall be able to add the product to their cart from each listing page.

REQ-51: The user shall be shown a list of other "Suggested" products in the same categories on each listing page

**3.2.2.3 Go to Next Page**

Only a reasonable number of listings will be queried and displayed while browsing, but more can be accessed via pages.

**Priority: Normal**

**User Story 1**

As a customer, I want to click to the next page of products.

1. Invoke 3.2.2, User Story 1, Steps 1-4
2. The user scrolls to the bottom.
3. The user clicks the "Next Page" button.
4. The system loads the next set of listings onto a new page.
5. The customer is directed to the next page.

**Requirements:**

REQ-52: The user shall only cause the system to retrieve up to TBD listings from a query.

REQ-53: The user shall be able to access the next and previous pages of results via buttons at the bottom of the browse products page.

REQ-54: The user shall cause the system to cache more listings than the current page size whenever they access the browse products page or submit a search or filter query to decrease loading times during browsing.



### **3.2.3 Search Products**

From any page, allows a user to search for a product by name, description, category, or vendor.

**Priority: Normal**

#### **User Story 1**

As a customer, I want to type a product's name inside the search bar to find a specific product.

1. Invoke 3.1.1 Login, User Story 1.
2. The user types a game's name into the search bar.
3. The user presses the "Enter" key.
4. The system searches the database for listings matching the entry.
5. The system displays a grid of listings containing the phrase searched.

#### **User Story 2**

As a customer, I want feedback when no games match with my search so I know the game is unavailable.

1. Invoke User Story 1, Steps 1-4
2. If no matches are found, the system displays a message of "No products matching '(search entry)' found."

#### **Requirements:**

REQ-55: The user shall have access to a search bar on every page.

REQ-56: The user shall be able to enter text into the search bar and submit a search query

REQ-57: The user shall cause the system to retrieve listings matching, in order, name, vendor, description, and/or category based on the search term.

REQ-58: The user shall receive a message on the browse products page if no listings match their query.

REQ-59: The user shall be shown the results of the most recent search query on the browse products page, until cleared.

REQ-60: The user shall be shown products with 0 stock with a "Sold Out" message.

REQ-61: The customer or vendor shall not be shown unpublished listings (except their own) in their query results.

### **3.2.3.1 Search Suggestions**

While searching, customers get search suggestions based on partial or complete strings they type into the search bar, updating every character added or removed.

**Priority: Low**

#### **User Story 1**

As a customer, I want search suggestions to appear when I type or delete in the search bar.

1. Invoke User Story 1, Steps 1-2.
2. The user begins typing a search query.
3. The system begins searching and returning listings that best match the current entry. Activates after every character is typed or deleted.
4. A dropdown appears beneath the search box showing suggested queries matching the user entry.
5. The user clicks a suggested query, presses enter, or clicks the search button.
6. The system displays a grid of listings containing the query phrase.

#### **Requirements:**

REQ-62: The user shall cause the system to retrieve listings based on similar queries to their exact search query.

REQ-63: The user shall cause the system to generate alternate query suggestions based on their search query.

REQ-64: The user shall be shown suggested alternate queries in a list below the search bar.

### **3.2.4 Add to Cart**

Allows the user to add a product to their cart from a listing, beginning the checkout process.

**Priority: High**

#### **User Story 1**

As a customer, I want to add an item to my cart.

1. Invoke 3.2.2.2 View Product Listing, User Story 1
2. The user selects the quantity (default qty. = 1) they want to add to their cart.

3. The user clicks the “Add to Cart” button.
4. The system adds the specified number of the product to the user’s cart in the database.
5. The user is prompted with two options: “Continue Shopping” or “Checkout”.
6. If the user clicks “Continue Shopping”, the system redirects the user to the listing page.
7. If the user clicks “Checkout”, they are directed to the cart page where 3.2.5 View/Edit Cart is invoked.

### **Requirements:**

REQ-65: The customer shall be shown a quantity selector on each listing page.

REQ-66: The customer shall be prevented from selecting more than the quantity in stock on each listing page.

REQ-50: The customer shall be able to add the product to their cart from each listing page.

REQ-67: The customer shall cause the system to keep track of what items and what quantities of those items are in their cart.

REQ-68: The customer shall be prompted to choose between “Continue Shopping” or “Checkout” when they add an item to their cart.

REQ-69: The customer shall be taken to the cart page when they choose checkout from the prompt.

### **3.2.5 View/Edit Cart**

Keeps track of items a user wants to buy, as well as their quantities. Allows a user to quickly review and purchase multiple items at once, as well as quickly removing items they no longer want to purchase.

### **Priority: High**

### **User Story 1**

As a customer, I want to view my cart.

1. From any page, the user clicks the shopping cart icon in the top right of the page.
2. The system loads the shopping cart page.
3. The user is directed to the shopping cart page, which shows all items (including quantity) in their cart and the total price.

## **User Story 2**

As a customer, I want to edit item quantities in my cart.

1. Invoke 3.2.5 View/Edit Cart, User Story 1
2. The user clicks “-“, “+”, “Remove from Cart”, or types in a custom quantity. They can also click any combination of these.
3. The system uses number change to update their cart in the database and reloads the updated cart page.

### **Requirements:**

REQ-70: The customer shall have access to a shopping cart button on every page when logged in.

REQ-71: The customer shall be shown the name, quantity, and price for each item in their cart on the shopping cart page.

REQ-72: The customer shall be shown the total price of all items in their cart on the shopping cart page.

REQ-73: The customer shall be able to edit the quantities of items in their cart on the shopping cart page.

REQ-74: The customer shall be able to remove items from their cart on the shopping cart page.

## **3.2.6 Checkout**

Allows the user to place an order for all the items currently in their cart, including paying for the items and providing shipping information.

### **Priority: High**

#### **User Story 1**

As a customer, I want to check out my cart.

1. Invoke 3.2.5 View/Edit Cart, User Story 1
2. The user clicks the “Checkout” button at the bottom of the page.
3. The system directs the user to the checkout page, pulling database information from the cart.
4. The user enters shipping, billing, and payment information.
5. The user has the option to “Go back” or “Checkout”.
6. If the user clicks “Go Back”, the system redirects the user to the cart page.
7. After clicking “Checkout”, the system creates the invoice(s) and moves the order to pending orders for vendor verification(s).

### **Requirements:**

REQ-75: The customer shall be able to continue to the checkout page from the shopping cart page.

REQ-76: The customer shall be able to retrieve saved shipping, billing, and payment information on the checkout page.

REQ-77: The customer's shipping, billing, and payment information must be stored securely by the system.

REQ-78: The customer shall be able to input unsaved shipping, billing, and payment information, including addresses and credit card information.

REQ-79: The customer shall be able to return to their cart or browsing instead of checking out.

REQ-80: The customer shall be able to submit an order on the checkout page (conditional on REQ-80).

REQ-81: The customer shall not be allowed to submit an order unless all required fields are filled in on the checkout page.

REQ-82: The customer shall have their submitted order logged by the system.

REQ-83: The customer shall have their payment transferred from their payment method to the vendor's account when the order is confirmed by the vendor.

### **3.2.7 View Order History**

Allows the customer to view all of their placed orders, as well as their totals and statuses.

#### **Priority: Normal**

#### **User Story 1**

As a customer, I want to view order history.

1. From any page, the user clicks the account icon in the top right of the page.
2. The system redirects the user to the account page.
3. The user clicks the "Order History" tab.
4. The system loads user's five most recent orders.
5. Five most recent orders are listed, including a "next page" option if the amount of orders exceeds page limit.

**Requirements:**

REQ-84: The customer shall have their submitted order logged by the system.

REQ-85: The customer shall be able to access order history from their account info page

REQ-86: The customer shall be shown five of their most recently placed orders on the order history page.

REQ-87: The customer shall be shown date, time, order ID, status, and total price for each order on the order history page.

REQ-88: The customer shall be able to move between pages of orders on the order history page

### **3.2.8 Make Returns**

Allows the customer to start the returns process for a received order.

**Priority: Normal**

**User Story 1**

As a customer, I want to return an item.

1. Invoke 3.2.7 View Order History
2. The user finds the order they want to return.
3. The user clicks the “Start a Return” option.
4. The system prompts the user to select the items they want to return.
5. The user selects the item(s) they want to return.
6. The system prompts the user to enter a reason for the return.
7. The user fills out prompts required for vendor/admin.
8. The user clicks “Submit”.
9. The system changes the status from “Completed” to “Pending Return” and notifies the vendor.

**Requirements:**

REQ-89: The customer shall be able to start a return for any orders with the completed status on the order history page.

REQ-90: The customer shall be able to choose which items they wish to return in the return prompt.

REQ-91: The customer shall be able to input information about why they are returning products in the return prompt.

REQ-92: The customer shall be able to submit a return request in the return prompt.

REQ-93: The customer shall have their return request logged by the system.

## **3.3 Vendor Features**

### **3.3.1 Vendor Sign Up**

Allows a new user to apply for a vendor account

**Priority: High**

#### **User Story 1**

As a vendor, I want to create an account.

1. The user navigates to the website
2. The user selects “Don’t have an account? Sign up!” option at the bottom of the login screen.
3. The system directs the user to the sign up page.
4. The user selects “Vendor” from “User Type” drop down.
5. The system loads additional fields for vendor information to be entered.
6. The user enters an email address.
7. The user enters a password meeting requirements, then re-enters the password for verification purposes.
8. The user fills out a vendor questionnaire.
9. The user clicks the “Sign Up” button.
10. The system sends the account application to the admin for approval and directs the user to the pending account page detailing the vendor approval process.

#### **User Story 2**

As a vendor, I want to be notified if the email I enter during sign-up is already attached to an account so that I can retry or return to the log in screen.

1. Invoke 3.3.1 Vendor Sign Up, User Story 1, Steps 1-10
2. The system finds an account in the database with an email matching the entered one and notifies the user via on-screen message.
3. The user has the option to try again or go back to the login page.

#### **User Story 3**

As a vendor, I want to be notified if my password does not meet the requirements, and I want to be shown the requirements when I attempt again.

1. Invoke 3.3.1 Vendor Sign Up, User Story 1, Steps 1-10. (Note: Step 7 is done incorrectly)
2. The system finds requirements are not met, then notifies the user via an on-screen message that they are not met.
3. The system shows the full requirements on the page and prompts the user to try again.

### **Requirements:**

REQ-1: The user shall be able to access the website over the internet.

REQ-2: The user shall be served the login page first by the website.

REQ-20: The user shall be able to access the sign up page from the login page.

REQ-21: The user shall be required to select between “Customer” and “Vendor” account types on the sign up page.

REQ-22: The user shall be required to input an email address on the sign up page.

REQ-23: The user shall be required to input a password on the sign up page.

REQ-24: The user shall be required to input their chosen password a second time on the sign up page.

REQ-25: The user shall be prevented from signing up if both passwords do not match.

REQ-26: The user shall be prevented from signing up if the password does not meet the security requirements.

REQ-94: The user shall receive vendor-specific questions on the sign-up page if they select “Vendor.”

REQ-95: The user shall be required to answer the vendor-specific questions on the sign-up page if they select “Vendor.”

### **3.3.2 Create, Modify, and Remove Listings**

Allows a vendor to view all of their listings, create more listings, edit listings, submit listings for approval, and unpublish listings.

**Priority: High (Milestone 3)**

**User Story 1**



As a vendor, I want to list a new item.

1. Invoke 3.1.1 Log In, User Story 1
2. The user clicks the “Listings” button.
3. The system directs the user to a listing page, pulling all listing data related to the user from the database.
4. The user clicks “Create Listing”.
5. The system opens a listing template for user entry.
6. The user uploads a thumbnail, enters item information (name, price, description), and enters quantity held.
7. The user clicks “Submit Listing”.
8. The system moves the listing to pending listings for the admin to approve and redirects to the “Listings” page.
9. The system notifies the user when the listing is approved or denied.

## **User Story 2**

As a vendor, I want to view my listings.

1. Invoke 3.3.2 Create, Modify, and Remove Listings, User Story 1, Steps 1-3
2. The user clicks “View Listings”.
3. The system opens a listing page that shows tabs of listing data (published listings, unpublished listings, pending listings, and listings removed by admin) pulled from the database.

## **User Story 3**

As a vendor, I want to unpublish one of my published listings.

1. Invoke 3.3.2 Create, Modify, and Remove Listings, User Story 2, Steps 1-2
2. The user locates the listing they would like to remove and press the “Unpublish” button.
3. The system prompts the user with a message asking if they are sure they would like to remove the listing.
4. If the user proceeds with deletion, the system reloads the listing page with a success message..
5. If the user backs out, the prompt disappears and nothing changes on the page.

## **User Story 4**

As a vendor, I want to modify one of my published listings.

6. Invoke 3.3.2 Create, Modify, and Remove Listings, User Story 2, Steps 1-2

7. The user locates the listing they would like to remove and press the “Edit” button and makes changes, then clicks “Update”
8. The system prompts the user with a message asking to confirm edits.
9. If the user proceeds with edits, the system changes to listing status to pending and submits to admin for approval.
10. If the user backs out, the prompt disappears and nothing changes on the page.

**Requirements:**

REQ-96: The vendor shall be able to view listings or invoices on their homepage.

REQ-97: The vendor shall be able to create a new listing from the vendor listings page.

REQ-98: The vendor shall be required to provide a thumbnail, name, price, description, and current stock when creating/editing a listing on the edit/create listings page.

REQ-99: The vendor shall be able to submit a listing for approval on the edit/create listings page.

REQ-100: The vendor shall have their listing approval requests logged by the system.

REQ-101: The administrator(s) shall be notified when a listing approval request is submitted.

REQ-102: The vendor shall be shown their listings separated by Pending, Published/Active, Inactive, and Removed on the vendor listings page.

REQ-103: The vendor shall be able to edit only their published or removed listings on the vendor listings page.

REQ-104: The vendor shall be able to view and edit pending and published/active listings on the vendor listings page.

REQ-105: The vendor shall be able to unpublish their listing from the listing page.

REQ-106: The vendor shall be able to edit the stock of their listing from the listing page.

REQ-107: The vendor shall have their listing stock changes logged by the system.

REQ-108: The vendor shall be required to confirm that they want to unpublish a listing.

**3.3.3 View and Act on Invoices**

Allows a vendor to view all invoices of every status, access invoice details, and move invoices between statuses

**Priority: High**

**User Story 1**

As a vendor, I want to view invoices placed by customers.

1. Invoke 3.1.1 Log In, User Story 1
2. The user clicks the “Invoices” button.
3. The system directs the user to an invoices page that shows tabs of invoice data (pending, awaiting shipment, completed, and returns) pulled from the database.

## **User Story 2**

As a vendor, I want to confirm an invoice placed by a customer.

1. Invoke 3.3.3 View and Act on Invoices, User Story 1
2. The user clicks the “Pending” tab.
3. The system loads invoices from the database which have a “Pending” status.
4. The user clicks a pending invoice.
5. The system shows the options “Confirm Order” or “Decline Order”.
6. The user clicks “Confirm Order”.
7. The system moves the invoice status from pending to confirmed and notifies the customer of the order confirmation.

## **User Story 3**

As a vendor, I want to decline an invoice placed by a customer.

1. Invoke 3.3.3 View and Act on Invoices, User Story 2 Steps 1-5
2. The user clicks “Decline Order”.
3. The system prompts the user for a declination reason.
4. The user enters the reason for declining the invoice and clicks “Send Feedback”.
5. The system moves the invoice status from pending to declined and notifies the customer of the order declination.

## **User Story 4**

As a vendor, I want to update an invoice status from confirmed to shipped.

1. Invoke 3.3.3 View and Act on Invoices, User Story 1
2. The vendor clicks the “Awaiting Shipment” tab.
3. The system shows the user a list of confirmed invoices that now have an “Awaiting Shipment” status, sorted from oldest to newest.
4. The vendor clicks an invoice.

5. The system displays the invoice information, including customer address and shipping information.
6. The vendor clicks the “Order Shipped” button after shipping the invoice.
7. The system moves the invoice status from “Confirmed” to “Shipped” and notifies the customer of the status change.

### **Requirements:**

REQ-109: The vendor shall be shown their invoices separated by pending, awaiting shipment (confirmed), shipped, completed, declined, canceled, pending return, and awaiting return on the vendor invoices page.

REQ-110: The vendor shall be able to confirm or decline invoices from the pending invoices section of the vendor invoices page.

REQ-111: The vendor must be shown all invoice information on the vendor invoices page.

REQ-112: The vendor must be able to mark an invoice as shipped from the confirmed section of the vendor invoices page.

REQ-113: The user shall have their invoice status changes logged.

REQ-114: The customer shall be notified of all their invoice’s status changes.

REQ-115: The vendor must be able to view details about an invoice including shipping address on the vendor invoices page.

### **3.3.4 Manage Returns**

Allows a vendor to view all pending and completed returns and fulfill or dispute them.

**Priority: Normal**

#### **User Story 1**

As a vendor, I want to accept a return from a customer.

1. Invoke 3.3.3 View and Act on Invoices, User Story 1
2. The user clicks the “Returns” tab.
3. The system loads invoices with a “Pending Return” status from the database.
4. The user clicks a customer request for return.
5. The system loads invoice details and user return feedback.
6. The user clicks “Accept Return”.
7. The system prompts the user for return information.

8. The user enters information on how the customer should return the product.
9. The user clicks “Accept Return”.
10. The system notifies the user and redirects to the returns page.

## **User Story 2**

As a vendor, I want to deny/dispute a return from a customer.

1. Invoke 3.3.4 Manage Returns, User Story 1, Steps 1-5
2. The user clicks the “Dispute” button.
3. The system prompts the user for an explanation.
4. The user enters reasoning and clicks “Dispute Return”
5. The system notifies both the admin and the customer of the return dispute.

## **Requirements:**

REQ-116: The vendor must be able to confirm or dispute a return from the pending return section of the vendor invoices page.

REQ-117: The vendor shall be able to view the customer’s contact information, reason for requesting return, and all associated invoice information from the pending return section of the vendor invoices page.

REQ-118: The vendor shall be required to provide return instructions when confirming a return.

REQ-119: The vendor shall be required to provide a reasoning when disputing a return.

REQ-120: The users’ return-related communications shall be logged by the system.

REQ-121: The customer shall be informed when their return changes status.

REQ-122: The administrator(s) shall be notified when a return is disputed.

## **3.3.5 Add/Edit Banking Info**

Allows a vendor to add banking information to their account to receive payments for their sales.

**Priority: Normal**

## **User Story 1**

As a vendor, I want to add my banking information.

1. Invoke 3.2.9 Edit Account Information, User Story 1
2. The vendor is shown a subsection titled Banking Information with an Add button since no information has previously been entered.
3. The vendor clicks the Add button.
4. The page prompts them for their routing number and account number.
5. The vendor inputs their information and presses confirm.
6. The system stores the vendor's routing number and account number.
7. The vendor is shown a success message.

## **User Story 2**

As a vendor, I want to edit my banking information

1. Invoke 3.3.5 Add/Edit Banking Info, User Story 1
2. The vendor is shown a subsection titled Banking Information with the last 4 digits of their routing number and account number and an Update button.
3. The vendor clicks the Update button.
4. The page prompts them for their password.
5. The vendor enters their password and clicks confirm.
6. The page prompts them for their new routing number and account number.
7. The vendor inputs their information and presses confirm.
8. The system stores the vendor's routing number and account number.
9. The vendor is shown a success message.

## **Requirements:**

REQ-123: The vendor shall be shown a Banking Information section on the account info page.

REQ-124: The vendor shall be shown an Add/Update button based on whether they have saved banking information on the account info page.

REQ-125: The vendor shall be able to add their banking information on the account info page.

REQ-126: The vendor shall be required to enter their password before updating their banking information on the account info page.

REQ-127: The vendor shall be able to see the last 4 digits of their banking information on the account info page.

## **3.3.6 View/Edit Account Information**

Allows the vendor to view and change their name/email on account.

**Priority: Low**

### **User Story 1**

As a vendor, I want to view my account information.

1. Invoke 3.1.1 Log In, User Story 1
2. The user clicks the account icon in the top right of the page.
3. The user clicks the “Account Information” tab.
4. The user is shown account information including email & name.

### **User Story 2**

As a vendor, I want to update the email on my account.

1. Invoke 3.2.9, User Story 1
2. The user clicks “Update E-mail/Name”
3. The user enters the new email/name.
4. The user enters their password to allow the change.
5. The user clicks “Update”.
6. The system validates credentials, then updates email/name in the database.
7. The system notifies the user of success via on-screen message and an email to the new address.

### **Requirements:**

REQ-128: The user shall be able to go to the account information page via a button on every page.

REQ-129: The user shall be able to see their email on the account information page.

REQ-130: The user shall be able to change their email on the account information page.

REQ-131: The vendor shall see their display name displayed on the account information page.

REQ-132: The vendor shall be able to change their display name on the account information page.

REQ-133: The user’s changes to their email, password, or display name shall be logged.

REQ-134: The user shall be required to enter their password to change their email address or password.

REQ-135: The user shall be prevented from changing their password unless the new password is different from their old password, meets the security requirements, and matches their second inputted password.

REQ-136: The user shall be required to input their new password a second time to change their password.

REQ-137: The user shall be shown a success or error message based on the results of changing account information.

## **3.4 Administrator Features**

### **3.4.1 Create Additional Admin Accounts**

Allows administrators to create additional administrator accounts.

**Priority: Low**

#### **User Story 1**

As an admin, I want to add an additional admin account.

1. 3.1.1 Log In, User Story 1
2. The user clicks the “Manage Accounts” button.
3. The system directs the user to an account management page.
4. The user clicks “Add Admin Account”.
5. The system directs the user to an invite page with an email field.
6. The user enters the email for a person whom they would like to invite as an admin.
7. The user re-enters the email for verification.
8. The user clicks “Invite Admin”.
9. The system sends an email with an admin invite to the address given and reloads the invite admin screen with a success message. On failure, the system redirects the user to the invite admin page with a failure message.

#### **Requirements:**

REQ-138: The administrator shall be able to access pending users, pending requests, action logs, and manage users from their home page.

REQ-139: The administrator shall be able to create a new administrator account from the admin user management page.



REQ-140: The administrator shall be required to input an email address in the new administrator account dialog.

REQ-141: The administrator shall be required to verify the email address by inputting it a second time in the new administrator account dialog.

REQ-142: The new administrator shall receive an email with instructions for accessing their account when a new administrator request is processed.

### **3.4.2 Approve/Deny Listings**

Allows administrators to audit listings and approve/deny all listings before they are published.

**Priority: High**

#### **User Story 1**

As an admin, I want to view pending, published, removed, and unpublished listings.

1. 3.1.1 Log In, User Story 1
2. The user clicks the “Listings” button.
3. The system redirects the user to a listings page that has four tabs: Pending, Published, Removed, and Unpublished.
4. The user clicks the tab with the status they want to see.
5. The system directs the user to a browse page containing the first page of listings with the status.

#### **User Story 2**

As an admin, I want to approve a pending vendor listing.

1. Invokes 3.4.2 Approve/Deny Listings, User Story 1, Steps 1-3
2. The user clicks “Pending Listings”.
3. The system shows a list of listings that have the “Pending” status.
4. The user clicks a pending listing.
5. The system loads the listing page in a pop-up.
6. The user clicks “Approve Listing”.
7. The system changes the status of the listing to “Published” and notifies the vendor.
8. The pop-up window closes and the user gets a success message.

#### **User Story 3**

As an admin, I want to deny a pending vendor listing.

1. Invoke 3.4.2, User Story 2, Steps 1-5
2. The user clicks “Deny Listing”.
3. The system prompts the user for a denial reason.
4. The user enters the reason for denial and clicks “Send Feedback”.
5. The system notifies the vendor of the denial with the reasoning attached.
6. The pop-up window closes and the user gets a removal message.

**Requirements:**

REQ-143: The administrator shall be shown all pending listings arranged by oldest first on the admin pending listings page.

REQ-144: The administrator shall be shown the date and time of the request, vendor, name of product, and price for each pending listing on the admin pending listings page.

REQ-145: The administrator shall be able to access details about each listing on the admin pending listings page.

REQ-146: The user shall only be able to access the listing page for an unpublished or pending listing if they are the owner or an administrator.

REQ-147: The administrator shall be able to approve or deny a pending listing request on its listing page..

REQ-148: The administrator’s listing approvals and denials shall be logged.

REQ-149: The administrator shall be required to supply a reason when denying a pending listing request.

REQ-150: The administrator shall be able to remove (unpublish) a published listing on its listing page.

REQ-151: The administrator shall be required to supply a reason when removing a published listing.

REQ-152: The administrator’s listing removals shall be logged.

REQ-153: The vendor shall be notified when their listing is approved, denied, or removed.

### **3.4.3 Approve/Deny Vendors**

Allows the administrators to selectively allow certain vendors to publish products on the platform.

**Priority: High**

#### **User Story 1**

As an admin, I want to view active/banned vendor/user accounts.

1. 3.1.1 Log In, User Story 1
2. The user clicks the “Accounts” button.
3. The system loads a page with four tabs: Pending Vendor Accounts, Vendor Accounts, User Accounts, and Banned Accounts.
4. The user clicks the tab with the status they want to see.
5. The system loads a list of the corresponding account type.

#### **User Story 2**

As an admin, I want to view pending vendor accounts.

1. Invoke 3.4.3, User Story 1, Steps 1-3
2. The user clicks the “Pending Vendor Accounts” tab.
3. The system loads a list of pending vendor account applications.

#### **User Story 3**

As an admin, I want to approve a vendor account application.

1. Invoke 3.4.3, User Story 2, Steps 1-3
2. The user clicks a vendor application.
3. The system directs the user to a page showing the vendor application information.
4. The user clicks “Approve Vendor Application”.
5. The system moves the application from pending status to approved and notifies the vendor.
6. The system redirects the user to the pending applications tab and generates a success message.

#### **User Story 4**

As a customer, I want to deny a vendor account application.

1. Invoke 3.4.3, Approve/Deny Vendors, User Story 3, Steps 1-3
2. The user clicks “Deny Vendor Application”.

3. The system prompts the user for a reason.
4. The user enters the reason for denial and clicks “Send Feedback”.
5. The system moves the application from pending status to denied and notifies the vendor.
6. The page redirects to the pending applications and generates a removal message.

### **Requirements:**

REQ-154: The administrator shall be shown the date and time of the request, vendor display name, and justification for each pending vendor request on the admin pending users page.

REQ-155: The administrator shall be shown all pending vendor requests arranged by oldest first on the admin pending users page.

REQ-156: The administrator shall be able to view details about each pending user, including vendors’ answers to the vendor questionnaire on the admin pending users page.

REQ-157: The administrator must be shown all users on the admin user management page.

REQ-158: The administrator shall be able to approve a vendor account on the admin pending users page.

REQ-159: The administrator’s vendor account approval or denial shall be logged.

REQ-160: The vendor shall be notified when their account is approved.

REQ-161: The administrator shall be able to deny a vendor account on the admin pending users page.

REQ-162: The administrator shall be required to enter a reasoning for denying a vendor account.

REQ-163: The user shall be notified if their pending vendor account is denied.

### **3.4.4 View Action Logs**

Allows an administrator to view all actions taken on the website, prompting further action if necessary.

**Priority: Normal**

#### **User Story 1**

As an admin, I want to view a log with transactions and vendor account activity.

1. Invoke 3.1.1 Log In, User Story 1.
2. The user clicks the “Action Logs” tab/link

3. The system displays a certain number of recent actions, sorted by most recent actions first.
4. The user scrolls through the list and notes which actions are of the “Transaction” type.

## **User Story 2**

As an admin, I want to find a failed transaction.

1. Invoke 3.3.4 View Action Logs, User Story 1, Steps 1-3
2. The user scrolls through the list until they find the faulty transaction - labeled “Failed”

## **Requirements:**

REQ-164: The administrator shall be shown TBD recent actions, sorted by most recent first, on the action logs page.

REQ-165: The administrator shall be able to navigate through older actions in pages on the action logs page.

REQ-166: The administrator shall be shown the date, time, action type, vendor (if applicable), customer (if applicable), and action details on the action logs page.

REQ-167: The administrator shall be able to quickly access accounts, listings, orders, and other objects associated with each log by clicking them on the action logs page.

## **3.4.5 Take Down Listings**

Allows administrators to take down listings that may be fraudulent or inaccurate or fail to meet content guidelines.

## **Priority: Normal**

## **User Story 1**

As an admin, I want to take down a listing page.

1. The user navigates to a listing via the link or the browse page.
2. The system shows an “Unpublish” button where the check out button would be for normal users.
3. The user clicks the “Unpublish” button.
4. The system asks the user to confirm unpublishing.
5. The user clicks “Confirm Unpublish”.

6. The system removes the listing from customer view, notifies the vendor, and moves the listing status into “Removed”.

### **Requirements:**

REQ-168: The administrator shall be able to remove (unpublish) a published listing on its listing page.

REQ-169: The administrator shall be required to supply a reason when removing a published listing.

REQ-170: The administrator’s listing removals shall be logged.

REQ-171: The vendor shall be notified when their listing is approved, denied, or removed.

### **3.4.6 Ban Users**

Allows administrators to ban users.

#### **Priority: Normal**

#### **User Story 1**

As an admin, I want to ban a user from one of their listing pages.

1. The admin navigates to an account via a listing, the browse page, or the action logs.
2. The admin clicks on the vendor’s name to access their information.
3. The system loads the admin view of the vendor page.
4. The admin clicks the “Ban” button beneath their account information.
5. The system prompts user to “Confirm Ban” or “Cancel”
6. The user clicks “Confirm Ban”.
7. The system sets the account status of the user to “Banned”, notifies the vendor, removes all of the vendor’s listings, and disables log in.

#### **User Story 2**

As an admin, I want to ban a user from the activity log.

1. Invoke 3.4.4 View Action Logs, User Story 1, Step 1-3
2. The admin clicks on any user’s name to access their information
3. The admin clicks the “Ban” button beneath their account information.
4. The system loads the admin view of the vendor page.

5. The admin clicks the “Ban” button beneath their account information.
6. The system prompts user to “Confirm Ban” or “Cancel”
7. The user clicks “Confirm Ban”.
8. The system sets the account status of the user to “Banned”, notifies the vendor, removes all of the vendor’s listings, and disables log in.

### **Requirements:**

REQ-172: The administrator shall be able to ban existing users from the admin user management page.

REQ-173: The administrator shall be able to navigate to a vendor’s account details on the admin user management page by clicking the vendor’s name on the listing page.

REQ-174: The administrator’s user bans shall be logged.

REQ-175: The user shall be immediately notified when they have been banned.

REQ-176: The administrator shall be required to confirm that they want to ban the specified account before it is banned.

REQ-177: The vendor’s listings shall be unpublished if they are banned.

REQ-178: The vendor’s pending invoices shall be denied if they are banned.

## **4. Other Nonfunctional Requirements**

### **4.1 Performance Requirements**

- Page load times shall display text content within 2 seconds, with images rendered within 4 seconds.
- The system shall support 5,000 concurrent users.
- Search shall return text results within 2 seconds, with associated product images rendered within 4 seconds.
- Payment processing shall be completed within 7 seconds.
- Inventory updates shall update in real time.
- Updating personal information shall take less than 2 seconds.

### **4.2 Safety Requirements**

- When the user types their password to login, it should be hidden.

- The customer's web browser shall not display a customer's password in account information. It shall be represented with special characters representing typed characters.
- The customer's web browser shall never display a customer's credit card number after retrieving from the database. It shall always be shown with just the last 4 digits of the credit card number.

### **4.3 Security Requirements**

- The user's data should be protected through Firebase security rules.
- The user's login information should be secure enough so that no one except for the admin will know that user's username/email and password.
- The system should be secure enough so that the user's personal information will not be disclosed to unauthorized users.
- The system should be secure enough to prevent the corruption of data from unauthorized users.
- The system should be secure enough to protect against denial-of-service attacks.
- The system shall not leave any cookies on the customer's computer containing the user's password.
- The system's back-end servers shall only be accessible to authenticated administrators.
- The system's back-end databases shall be encrypted.
- The system's back-end servers shall never display a customer's password. The customer's password may be reset but never shown.

### **4.4 Software Quality Attributes**

#### **4.4.1 Availability**

- The system will remain available at 99.9% outside of stated maintenance periods.

#### **4.4.2 Interoperability**

- The system shall work with third-party payment processors through Square.

#### **4.4.3 Portability and Flexibility**

- The system shall function across major browsers (Safari, Chrome, Edge, Firefox, and other Chromium-based browsers).

#### **4.4.4 Reliability**

- The system shall reliably perform its functionality.
- The system shall reliably perform the correct function 99.9% of the time it is executed.

#### **4.4.5 Usability**

- The system shall be intuitively understandable by users to allow for them to operate it on the first interaction.



## **5. Other Requirements**

- Accessibility Requirements: The website shall adhere to web accessibility standards as outlined by the WCAG (Web Content Accessibility Guidelines).
- Database Requirements: The system shall utilize a relational database to perform its functionality.

## **Appendix A: Glossary**

- Admin - highest user level; can manage accounts, listings, orders, and see all activity
- Blazor - frontend development framework
- Customer - user who can buy products from vendors
- Firebase - backend from google for authentication and databasing
- Square Sandbox - testing environment for simulating transactions
- WASM - WebAssembly, framework used by Blazor to run C# code
- WCAG - Web Content Accessibility Guidelines

## **Appendix B: To Be Determined List**

1. Multiple images/videos on listings
2. Built in dashboard with charts and data analysis statistics

### **TBD Requirements:**

REQ-162: The administrator shall be shown TBD recent actions, sorted by most recent first, on the action logs page.