

354TheStars Online Marketplace

SYSTEM REQUIREMENT SPECIFICATION

Version: 1.0



Michael Rowe 26101267

Steven Smith 40057065

Nagib Al-Bouery 27153031

Gheith Abi-Nader 25703387

Claire Shaw 40022429

Gia Khanh Nguyen 24880803

Olivier Fradette-Roy 40074024

Mark Jarjour 40057853

Guillaume Rochefort-Mathieu 40058103

Amrou Abdelhadi 40045799

Ghislain Clermont 40057664

Mahdi Chaari 27219946

Gina Cody School of Engineering and Computer Science

Concordia University

Comp 354 Fall 2019

Monday, October 7, 2019

Contents

1	Introduction	1
1.1	Purpose	1
1.2	Document Conventions	1
1.3	Intended Audience and Reading Suggestions	1
1.4	Project Scope	1
2	Overall Description	2
2.1	Product Perspective	2
2.2	Product Functions	2
2.3	User Stories and Personas	3
2.4	Design and Implementation Constraints	5
2.5	User Documentation	5
2.6	Assumptions and Dependencies	5
3	External Interface Requirements	6
3.1	User Interfaces	6
3.2	Hardware Interfaces	7
3.3	Software Interfaces	7
3.4	Communications Interfaces	8
4	System Features	9
4.1	Account Creation	9
4.1.1	Use Case Diagram	10
4.1.2	Functional Requirements	10
4.2	Profile Management	10
4.2.1	Functional Requirements	11
4.3	Account Recovery	11
4.3.1	Functional Requirements	12
4.4	Login	12
4.4.1	Use Case Diagram	12
4.4.2	Functional Requirements	12
4.5	Shopping Cart	13
4.5.1	Functional Requirements	13
4.6	Purchase System	13
4.6.1	Use Case Diagram	14
4.6.2	Functional Requirements	14
4.7	Product Listing	14
4.7.1	Use Case Diagram	15
4.7.2	Functional Requirements	15
4.8	Product Return	16
4.8.1	Functional Requirements	16
4.9	User Review	17
4.9.1	Use Case Diagram	17

4.9.2	Functional Requirements	17
4.10	Search, Sort and Trending Features	18
4.10.1	Use Case Diagram	18
4.10.2	Functional Requirements	18
4.11	Administrator Profile	19
4.11.1	Use Case Diagram	19
4.11.2	Functional Requirements	20
5	Other Non-Functional Requirements	21
5.1	Performance Requirements	21
5.2	Safety Requirements	21
5.3	Security Requirements	21
5.4	Software Quality Attributes	22
5.5	Business Rules	22
6	Unit Testing	23
6.1	Traceability Matrix	23
7	Conclusion	26
	Alphabetical Index	27
	Glossary	28
	Acronyms	29
	Bibliography	30

List of Figures

4.1	Main Use Case Diagram	9
4.2	Account Registration Use Case Diagram	10
4.3	Login Use Case Diagram	12
4.4	Purchase System Use Case Diagram	14
4.5	Product Listing Use Case Diagram	15
4.6	User Review Use Case Diagram	17
4.7	Search Use Case Diagram	18
4.8	Administrative Use Case Diagram	19

List of Tables

2.1	User Stories	3
2.2	Personas	4
3.1	Enumeration of Software, Libraries and API used.	8
4.1	Account Creation Functional Requirement.	10
4.2	Profile Management Functional Requirements	11
4.3	Account Recovery Functional Requirements	12
4.4	Login System Functional Requirements	13
4.5	Shopping Cart Functional Requirements	13
4.6	Purchase System Functional Requirements	14
4.7	Product Listing Functional Requirements	16
4.8	Product Return Functional Requirements	16
4.9	User Review Functional Requirements	17
4.10	Search, Sort and Trending Functional Requirements	19
4.11	Administrative Functional Requirements	20
5.1	Performance Non-Functional Requirements	21
5.2	Safety Non-Functional Requirements	21
5.3	Security Non-Functional Requirements	21
5.4	Software Quality Attributes Non-Functional Requirements	22
5.5	Business Rules Non-Functional Requirements	22
6.1	Traceability Matrix	24

1 Introduction

This section briefly covers the documents purpose, conventions, target audience and scope.

1.1 Purpose

The purpose of this document is to convey a detailed elucidation of the e-commerce website 354TheStars. It will explain the purpose and the features of the software, the interface of the website, and the functions and constraints under which the system must behave. This document is catering to users of the software; however, developers may also find it useful.

1.2 Document Conventions

This document follows the conventions established by the IEEE template for System Requirement Specification Documents [11].

1.3 Intended Audience and Reading Suggestions

This document is intended to be used by:

- Typical Users, such as parents, who want to use 354TheStars to purchase goods online and have them delivered to their home (Films, Books, Clothing, etc.).
- Commercial Users, such as Walmart, to sell used or new goods on the marketplace (Music Discs, Video Games, Video Game Consoles, Used Computer Hardware, etc.).
- Administrative Users who are interested in understanding the fundamental functions of the software and how to interact with and/or alter the other components in the system (Database, Telemetry, Access Privileges, etc.).

1.4 Project Scope

354TheStars is an online marketplace that provides users with a reliable and easy interface with which they can interact to buy and sell goods worldwide. The platform's objective is to increase our market share and, by extension, expand platform profitability. The system currently takes an eight percent fee per transaction completed in our market. In return, we aim to provide users with a secure, dependable and accessible platform that will make users' product listings stand out in contrast to the means employed by our competitors.

2 Overall Description

This section will discuss the perspective, functions, user classes and the assumptions made in regards to system.

2.1 Product Perspective

354TheStars is an online shopping website. On this website, anyone can set articles to sell worldwide for other customers to buy. The articles can be anything that is not porn or illegal in nature. Buyers can add multiple articles from different sellers into a cart, and then proceed to the payment all at once.

354TheStars is a stand-alone product that we are developing at the request of Bob TheRich. The website is designed to run on all main PC and mobile browsers.

2.2 Product Functions

The following are the functions of this product:

- Register: The user must register an account to purchase or sell, but can browse without one.
- Sign in: Users can login with an existing account.
- Log off: Users can log out of their account.
- Sell: Any registered user can create an article for selling one of their products.
- Search: Users will be able to search for items that are being sold on the website using keywords.
- Add to cart: The user can add any product for sale to their cart.
- Buy and Pay: Users can buy items in their cart in bulk, and proceed with a single payment.
- Review: Buyers can add reviews to item they have purchased. This review can include pictures.
- Refund: Sellers can enable or disable product refund. If enabled, buyers can return a defective item and request a refund, as long as the time limit set by the seller has not expired yet.

2.3 User Stories and Personas

This section covers the user stories and the personas interested in our website.

Table 2.1: User Stories.

As a/an	I want to...	so that...
user	have a basic account	I am distinct from other users.
user	be able to reset my password by email	I can retrieve my account if I forget my password.
user	be able to edit my personal info	I can change it if I need to.
buyer	have account creation requirements	I can be sure my account retains my important info.
buyer	write a review for a product	I can share my opinion with others.
buyer	have a shopping cart	I can save items for purchase later.
buyer	add an item to my shopping cart	I can purchase it later.
buyer	remove an item from my shopping cart	I can change my mind on whether or not to purchase an item.
buyer	see the total cost of items in my cart	I can know how much this purchase will cost.
buyer	select shipping from standard and express for my order	I can choose the optimal shipping method for my desired arrival date.
seller	be able to create an item for my store	I can sell the newly-added product.
seller	allow the return of goods	I can keep my customers happy.
seller	set the number of days for returns	I ensure buyers have plenty of time to return goods while not abusing returns.
admin	be able to add/remove/modify user accounts	I can ensure that no users abuse the system, and to help users with problems.
admin	check the website's earnings	I can keep the owner informed.
owner	regularly check on my website's earnings	I ensure that it is a profitable endeavour.

Personas represent the different user types that might use the service, product, site, or brand in a similar way. A list of personas is shown in **Table 2.2**.

Table 2.2: List of Personas.

Persona Group	Name	Demographics	Goals and Tasks	Technical Experience
Buyer	Nancy Drew	22-year old Canadian woman, student	Nancy is conscious of her budget, but she still wants to keep up with the latest trends. She would like to find some new clothes at 354TheStars. She wants to be able to see the cheapest items first, and she would like to filter by type of item (shoes, pants, tops, etc.) to shop more efficiently.	She is a frequent user of social media and mobile applications.
Small Seller	George Nagano	65-year old Japanese Canadian man, married for 35 years, no higher-education as he took over the bonsai family business	With the increase of retail e-commerce, Nagano wants to sell online to broaden his potential customers. Nagano wants to be able to list his bonsais for sale with a description and photos. He also hopes that starting to sell on 354TheStars will increase his exposure.	Very little, he is more hands-on. His wife, on the other hand, takes care of accounting, bills, and helping clients at the front desk.
Top Seller	Mark Nguyen	45-year old Vietnamese-Canadian man.	Mark is a small business owner who owns an electronics shop. He has found recent success by selling computer parts online. Mark has been very successful in his new business, but he wants to retain his competitive edge. Mark wants to be able to move large volumes of products and needs the system to be well organized to keep track of his many different products. If possible, Mark would like to get some analytics on his sales, so he knows which products to purchase. Mark would also be interested in advertising his products on the site to gain more market share.	He has fifteen years of retail experience, but only two years of online sales.
Admin	Han Solo	32-year old Canadian man	Han is an admin at 354TheStars. He is responsible for ensuring that spammers do not take control of the website. He wants to be able to suspend and delete offending accounts. He also wants to be able to help users reset their lost passwords. Han also needs to be able to generate analytics on the site so he can track trends in traffic and sales.	He has ten years of experience in web design and management.
Owner	Bob TheRich	Male	Mr. TheRich is investing money in the marketplace from which he hopes that he can make money. Additionally, he wants to be able to track how much money he makes.	He has no background or expertise in software engineering.

2.4 Design and Implementation Constraints

The database will only be limited to a couple hundred items that we are using for testing purposes. Payments will be limited to PayPal only due to the security it offers, and we are simulating payment validation since the website will not be available to the public.

Checking out will be possible after the customer creates an account; therefore, no guest checkout will be allowed. A user will be limited to having only three addresses, and they will be able to choose one of them after he selects to check out. Accounts will be sufficiently secure with a minimum requirement of 8 characters. We will not be taking security against Brute Force attacks into consideration.

2.5 User Documentation

The JavaScript framework we will be using:

<https://vuejs.org>

<https://github.com/vuejs>

<https://vuetifyjs.com/en/>

<https://github.com/vuetifyjs/vuetify>

Flask:

<https://pypi.org/project/Flask/>

NodeJS:

<https://nodejs.org/en/>

<https://github.com/nodejs>

SQLAlchemy:

<https://www.sqlalchemy.org/>

2.6 Assumptions and Dependencies

Even though we are creating a website, we are assuming that eventually, we will have to make it mobile-friendly, depending on our progress. Advertising will be possible in the future, both for sellers and to promote other websites. We are also leaving the door open for chat support, analytics, search filtering, and browsing with different languages.

The front-end and back-end development are dependent on each other and the team's job execution. To maximize efficiency using Python, we will utilize SQLAlchemy, which is the preferred SQL toolkit in Python.

3 External Interface Requirements

This section outlines all the external requirements of the interfaces needed for the software to execute properly.

3.1 User Interfaces

One of the main design principles that we will be following when designing the software's user interface is "Material Design". This design is a UI standard that was introduced by Google in 2014 with the intent of unifying UI design across its own devices and web services.[1] Nowadays, it has been adopted by a multitude of sites in the industry as a way of standardizing user interfaces across both the web and different display form factors.

This design divides each UI element into "cards" that contain several features to enhance the user experience. One such feature is that these cards should retain their overall appearance regardless of which device you use the software. So, for instance, using a piece of software on a desktop computer with a 16:9 aspect ratio monitor should be roughly the same experience as using the software on a mobile device that has much less screen real estate. The standard ensures that responsible cross-platform design is a key component of your site and is as easy to implement as possible, which is important when you factor in the continuously growing number of users who use smaller form factor devices to browse and consume online media.

Another advantage of Material Design is the feedback of using its feature set. One of the key components to having a good user interface design is the satisfaction or "feedback" that a user receives when interacting with it. We could implement this as a nice tactile bump on a physical switch, the responsiveness of a scroll bar, or even trivial things such as having drop-shadows on the text. All these elements, while they may not be apparent, do affect the user experience and make it better, or more accurately worsen the experience when lack thereof is apparent.

Because of this, Material Design intelligently makes use of a lot of features that although do not affect the functionality of the software, do enhance the user feedback. One of these features is its use of light and shadows to simulate depth on interface elements. In the real world, lighting makes up a big part of how humans can figure out the depth of physical material. Thus, you can add drop-shadows to "forefront" elements of the UI to make them appear closer, even though that is an illusion created in our head. Minor design considerations such as this improve both the look and feel of a user interface.

On the functional side of things, Material Design encourages the use of minimalist icons to interact with the user. While this can lead to an overall cleaner design, it can also hinder a user's ability to understand how to operate your software, especially if they are unfamiliar with its genre. To combat this, any interact-able pieces of the software will require tool-tips. These are short pieces of text that will appear over hovered elements in the UI, which can explain the functionality and consequences of interacting with said element. The goal is for these tool-tips is to be as informative and concise as possible. The result is an interface with a clean aesthetic

that can also provide information to less savvy users.

These benefits, along with its set of standardized open-source UI templates and its standardization in the industry, make Material Design the gold standard that this software will follow to ensure the responsiveness and appeal of our interface.

3.2 Hardware Interfaces

Since the interface of the software is primarily web-based, the client will need to have internet-enabled hardware. The specific internet functionality needed will be outlined in Section 3.4 (Communications Interfaces). The input mechanism from the client will be a combination of mouse and keyboard usage, although touchscreen functionality is also available for mobile users. Feedback for these inputs and the web-page itself will output to a screen connected to the computer. The connection of all the I/O mentioned to the client's computer itself is beyond the scope of this software and is up to the client to configure properly.

On the server's end, our database will be connected to an always-on computer so that clients can access information such as their account details at any time. Information that needs saving between client sessions (product info, client account details, among other things) will be readily available and remembered in the database's storage. Like our clients, this server also needs to have internet access for the web server to query user requests and send out emails.

3.3 Software Interfaces

We will write the front end of our software in HTML. We will use JavaScript as its client-side scripting language alongside CSS for markup styling. All three of these tools are ubiquitous in web development, so there is little reason not to apply them to our software.

The hosting of the server-side of our software will be on a Linux-based machine, which will be running an SQL database. SQL enables the server to query the database quickly and is a veteran standard of the server industry. We will interact with this database using the Python programming language.

Below is a list of all the additional libraries and tools that we will be using for the development of this software. Each row contains the name and version of that tool, as well as a brief description of its functionality and its benefit to the development of our software.

Table 3.1: Software, Libraries and API used.

Name and Version	Description
Vuetify 2.0.19	UI components library based on the “Material Design” standard mentioned in Section 3.1. It provides a suite of components that enable fast and efficient production of the software’s front-end.[2]
ECMAScript 2015	The sixth edition of a standard for the JavaScript scripting languages used in universally all modern browsers. Due to its universal support, it allows the software to function on as many web browsers as possible.[6]
Flask 1.1.1	Python-based framework for developing a web application’s back-end. Enables the software to display dynamic web pages that change in response to user input.[3] For example, displaying the account information of a logged-in user whose content will differ from that of a different user.
Apache 2.4.41	One of the most commonly used web server tools on the world wide web. Used to serve web pages to clients whenever they connect to your site.[4]
Gunicorn 19.9.0	Acts as an interface between our web server and the web framework. Forwards user requests from the web server to our Python framework.[5]
SQLAlchemy	An SQL toolkit for the Python language which allows the web framework to communicate with the SQL database.[10]
JSON	File format for storing data in plain-text. Its ubiquity in web services enables us to store client information in our database under this format easily and to parse it into dynamic web pages quickly.[8]

3.4 Communications Interfaces

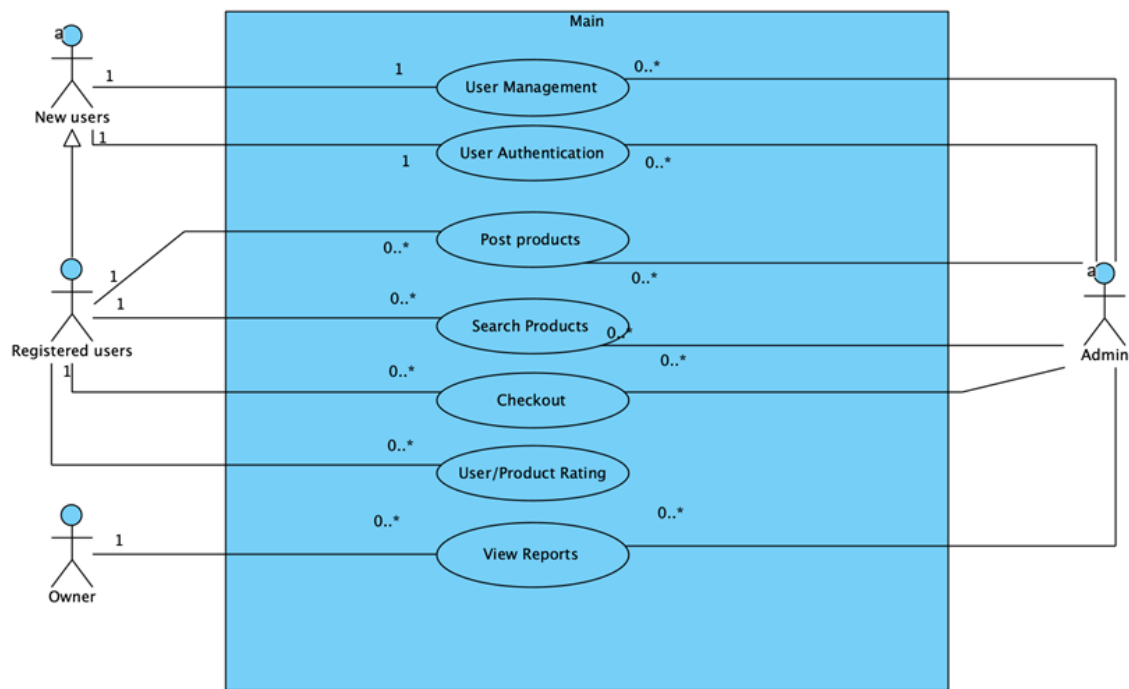
The primary communication interface of the software will be HTTPS over the world wide web as a means of transferring data between the client and server (sending web pages to the client, a seller adding new products to their catalogue, etc.). Currently, the only data that we will transfer between these two services are JSON objects, which, as mentioned in the preceding section, contain all the dynamic content to be displayed on the web site (account details, updated product info, etc.).

We will also utilize e-mails. Each user is required to have a unique email address attached to their account, so our software will be required to send the user emails for purposes such as verifying their registration, the sending of receipts when they purchase a product, among other things.

4 System Features

This section demonstrates the platform's core user features and explains how the users will interact with them and the results they return. Below are the use diagrams for the system features and a more in-depth description of each major feature.

Figure 4.1: Main Use Case Diagram



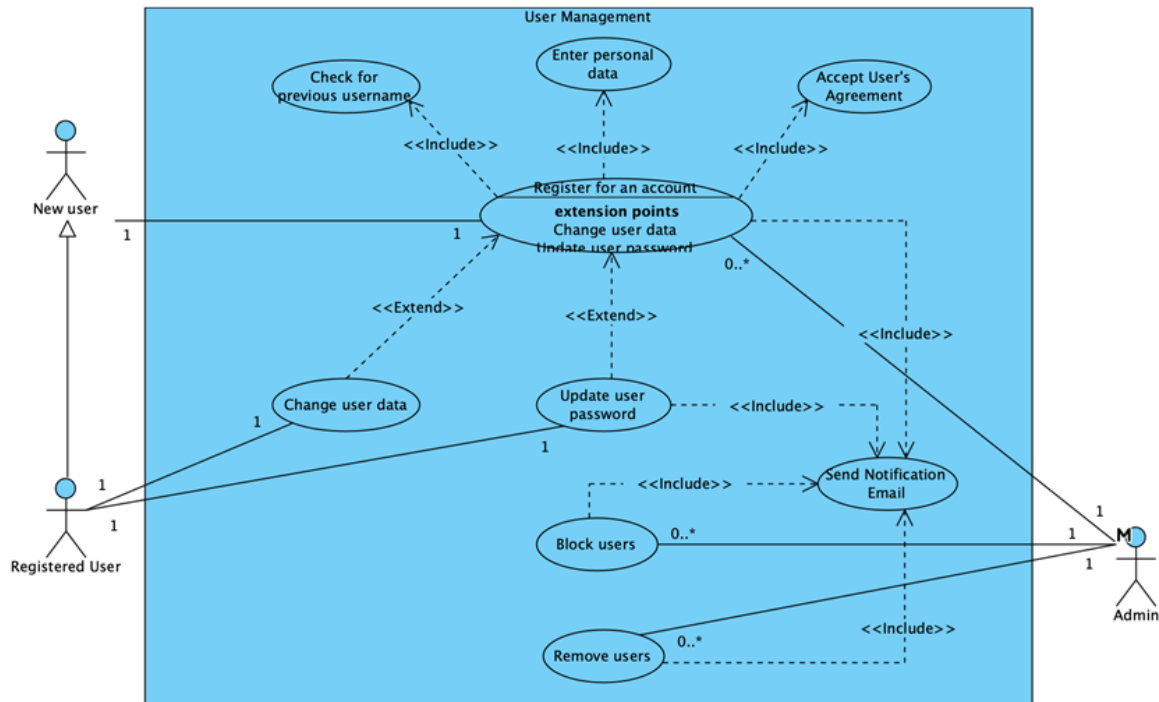
4.1 Account Creation

Account creation is a feature that encompasses the process by which users will interact with the system to register themselves onto the platform, which will create a user profile that users will use for the purchase and sale of items.

Feature Implementation Priority: High

4.1.1 Use Case Diagram

Figure 4.2: Account Registration Use Case Diagram



4.1.2 Functional Requirements

Functional requirements are shown in Table 4.1.

Table 4.1: Account Creation Functional Requirements (FR-ACS)[7].

Feature ID	Feature Name	Description
FR-ACS1	Validate Email Address	Verify that the email address follows the [WORDS]@[WORDS].[WORDS] sequence of characters.
FR-ACS2	Validate Password	Verify that the entered passwords contain at least one upper case, one special character and a minimum length of eight characters.
FR-ACS3	Emit Confirmation Email	The application will emit a confirmation email upon the creation of the account.
FR-ACS4	Register Account	The application shall allow a user to create an account from the login or register pages on the website.

4.2 Profile Management

Profile Management is an agglomeration of user management requirements. It consists of everything from changing account email address or shipping address to browsing the user transaction history or viewing current listings.

Feature Implementation Priority: Medium

4.2.1 Functional Requirements

Functional requirements are shown in **Table 4.2**.

Table 4.2: Profile Management Functional Requirements (FR-PM1)[7].

Feature ID	Feature Name	Description
FR-PM1	Change Email Address	The application shall allow the user to change the email address of their profile.
FR-PM2	Change Password	The application shall allow the user to change the password of their account.
FR-PM3	Add Shipping Address(es)	The application shall allow a user to add up to three different shipping addresses.
FR-PM4	Remove Shipping Address(es)	The application shall allow a user to remove a shipping address from their profile.
FR-PM5	Edit Shipping Address(es)	The application shall allow a user to edit a shipping address in their profile.
FR-PM6	Add Phone Number	The application shall allow a user to add a phone number to their profile.
FR-PM7	Remove Phone Number	The application shall allow a user to remove a phone number from their profile.
FR-PM8	Edit Phone Number	The application shall allow a user to modify their profile phone number.
FR-PM9	View Transaction History	The application shall allow a user to see a list of all completed and pending transactions.
FR-PM10	View Receipts	The application shall allow a user to access the receipt for any transaction in their transaction history.
FR-PM11	View Product Listings	The application shall allow a user to access a list of all active and completed listings.
FR-PM12	View Ratings	The application shall allow a user to view their current seller ratings.
FR-PM13	View Reviews	The application shall allow a user to view a list of all the reviews they have registered.
FR-PM14	View Order Status	The application shall allow a user to display the status of all their pending orders.
FR-PM15	Change Order Status	The application shall allow a seller to change the status of their pending orders. For example, the seller will be able to notify the buyer when the item has shipped.

4.3 Account Recovery

When a user tries to log in, they will have the option: “I Forgot My Password” that will send an email containing a temporary password to the user. The password can be used by the user to recover the account.

Feature Implementation Priority: Low

4.3.1 Functional Requirements

Functional requirements are shown in **Table 4.3**.

Table 4.3: Account Recovery Functional Requirements (FR-AR)[7].

Feature ID	Feature Name	Description
FR-AR1	Recover Password	The application shall allow a user to recover their profile password.
FR-AR2	Send Temporary Password	The system shall change the user profile password to a temporary password that it will send to the user via email.

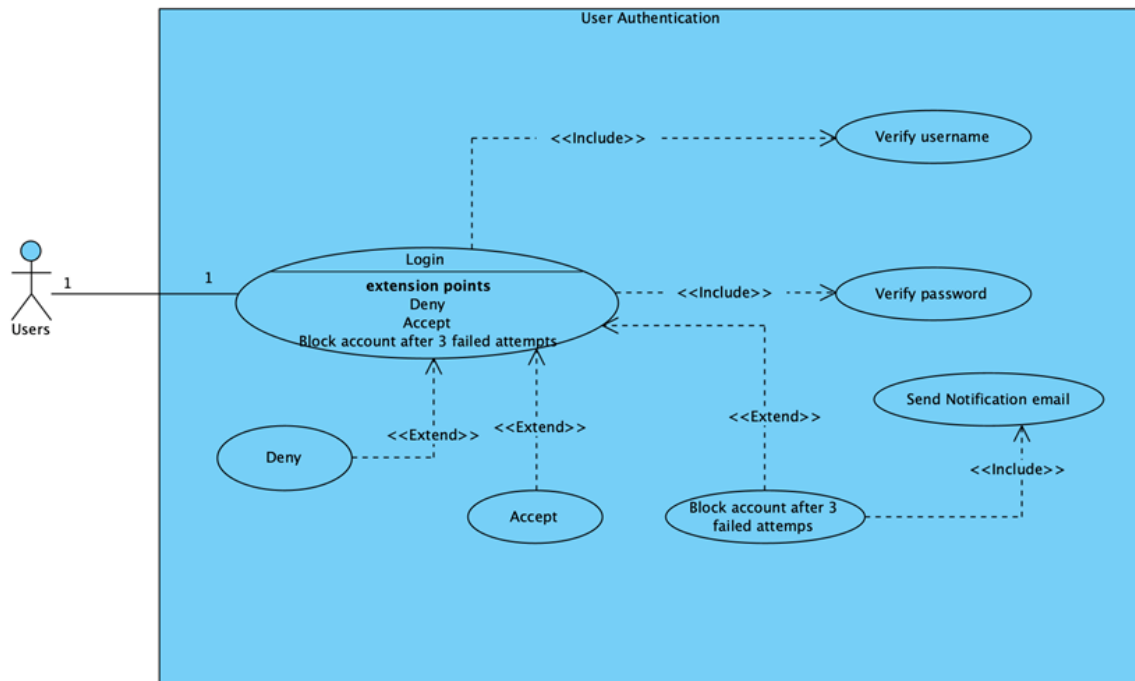
4.4 Login

Users may browse the store as a guest or log in to purchase products on the marketplace. To login, the user must first have created an account profile. The login feature grants the client user-level privileges. User-level privileges come with various benefits, such as the ability to check out or save products to their cart.

Feature Implementation Priority: High

4.4.1 Use Case Diagram

Figure 4.3: Login Use Case Diagram



4.4.2 Functional Requirements

Functional requirements are shown in **Table 4.4**.

Table 4.4: Login System Functional Requirements (FR-LOG).[7].

Feature ID	Feature Name	Description
FR-LOG1	Login User	The application shall allow a user to login using their profile credentials either at the main page or while browsing the shop. At the end of the session a user can logout.
FR-LOG2	Login Administrator	The application shall allow an administrator to login using their special credentials on the main page.

4.5 Shopping Cart

The shopping cart is an essential feature that keeps track of all the items a user wants to purchase while they are browsing the marketplace. Users can also save their shopping cart to their profile as a watch-list.

Feature Implementation Priority: Low

4.5.1 Functional Requirements

Functional requirements are shown in **Table 4.5**.

Table 4.5: Shopping Cart Functional Requirements (FR-SC).[7].

Feature ID	Feature Name	Description
FR-SC1	Add Product Listing(s)	The application shall allow a user to add product listings to their shopping cart.
FR-SC2	Remove Product Listing(s)	The application shall allow a user to remove a product listing from their shopping cart.
FR-SC3	Update Quantity	The application shall allow a user to update the number of items they are purchasing from product listings in the shopping cart.
FR-SC4	Save Cart	The application shall allow a user to save their shopping cart to their profile.

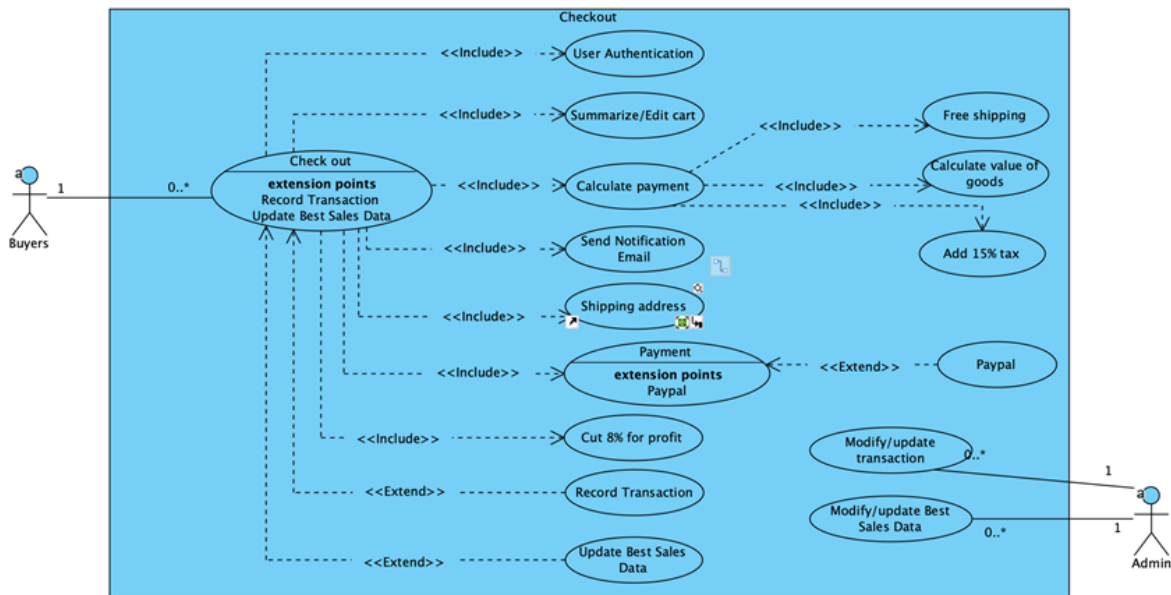
4.6 Purchase System

The purchase system is the agglomeration of tools and requirements from checkout to payment processing and issuing a notice to the seller and buyer if an object sells. Payment processing is handled using the third-party software, PayPal.

Feature Implementation Priority: Medium

4.6.1 Use Case Diagram

Figure 4.4: Purchase System Use Case Diagram



4.6.2 Functional Requirements

Functional requirements are shown in Table 4.6.

Table 4.6: Purchase System Functional Requirements (FR-PS)[7].

Feature ID	Feature Name	Description
FR-PS1	Select Shipping Address	The application shall allow a user to select a shipping address.
FR-PS2	Select Shipping Method	The application shall allow a user to select a shipping method.
FR-PS3	Proceed to Checkout	The application shall allow a user to proceed to pay for one or more items in their shopping cart.
FR-PS4	Payment Processing	The application shall allow a user to pay online using PayPal. PayPal will also handle the authorization.
FR-PS5	Emit Notification	The system will notify the seller by email when a product sells and when a transaction is complete.
FR-PS6	Transaction Confirmation	The system shall emit a transaction confirmation to both the seller and the buyer.

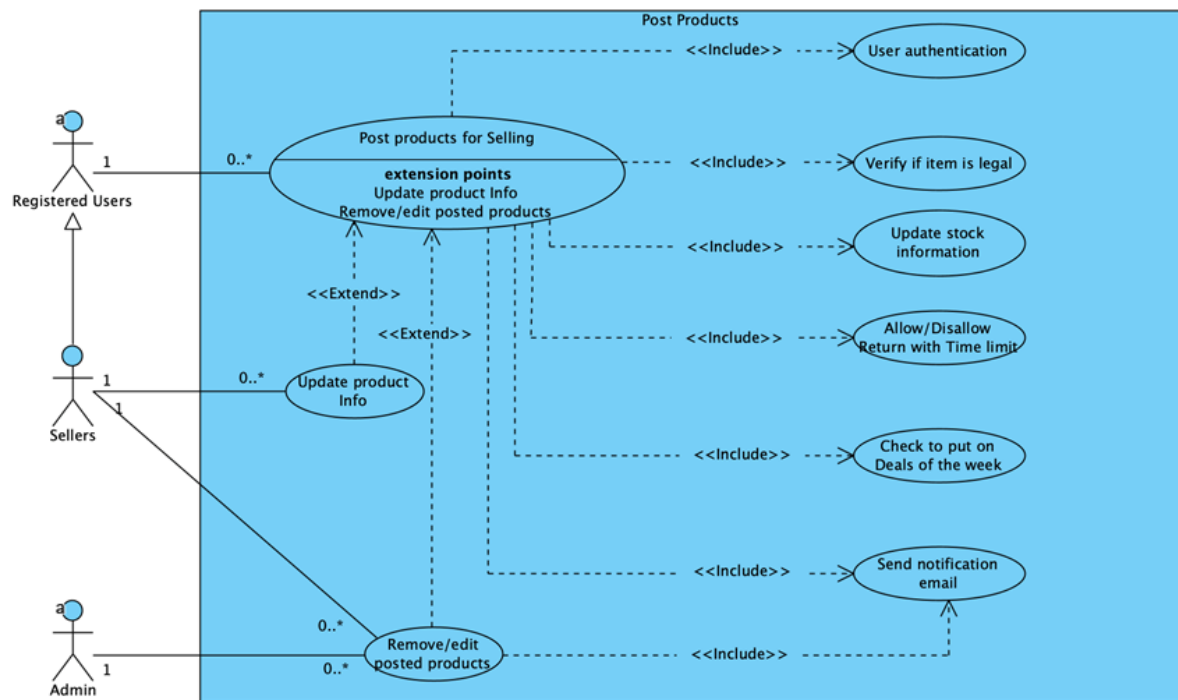
4.7 Product Listing

Product Listing is a feature that contains the information relevant to a product. It contains information such as the seller username, price, description, product images, etc. It contains all the tools and functions needed to create and view product listings.

Feature Implementation Priority: High

4.7.1 Use Case Diagram

Figure 4.5: Product Listing Use Case Diagram



4.7.2 Functional Requirements

Functional requirements are shown in Table 4.7.

Table 4.7: Product Listing Functional Requirements (FR-PLS)[7].

Feature ID	Feature Name	Description
FR-PLS1	Create Listing	The application shall allow users to post new product listings, free of charge.
FR-PLS2	Edit Listing	The application shall allow users to edit their current posts, which includes changing the title, description (maximum 1000 characters), price, category, and condition.
FR-PLS3	Set Refund	The application shall allow users to set whether or not they accept refunds and the period allocated to the buyer.
FR-PLS4	Update Quantity	The application shall allow a user to update the available quantity on a listing.
FR-PLS5	Set Condition	The application shall allow a user to set the condition of their product.
FR-PLS6	Set Category	The application shall allow a user to set the category of their listing.
FR-PLS7	Set Description	The application shall allow a user to set a description for their listing. Albeit a maximum of 1000 characters will be allowed.
FR-PLS8	Set Title	The application shall allow a user to set a title for their listing.
FR-PLS9	Upload Images	The application shall allow a user to upload images for their listing.
FR-PLS10	Hide Listing	The application shall allow a user to hide a listing, removing it from the marketplace.
FR-PLS11	Unhide Listing	The application shall allow a user to unhide a listing, adding it to the marketplace.
FR-PLS12	Delete Listing	The application shall allow a user to remove a listing.

4.8 Product Return

Product return is the feature that enables buyers to ask for refunds on defective products or items that arrived differently than advertised. All returns are refunds, but the seller will only emit a refund once the original item returns to them.

Feature Implementation Priority: Low

4.8.1 Functional Requirements

Functional requirements are shown in **Table 4.8**.

Table 4.8: Product Return System Functional Requirements (FR-RS)[7].

Feature ID	Feature Name	Description
FR-RS1	Demand Refund	The application shall allow a user to demand a refund on authorized transactions. Furthermore, the user can only demand a refund during a period allocated by the seller, which will vary from seller to seller.
FR-RS2	Emit Refund	The application shall allow a seller to accept and emit a refund. The system will refund the buyer and charge the seller's account.

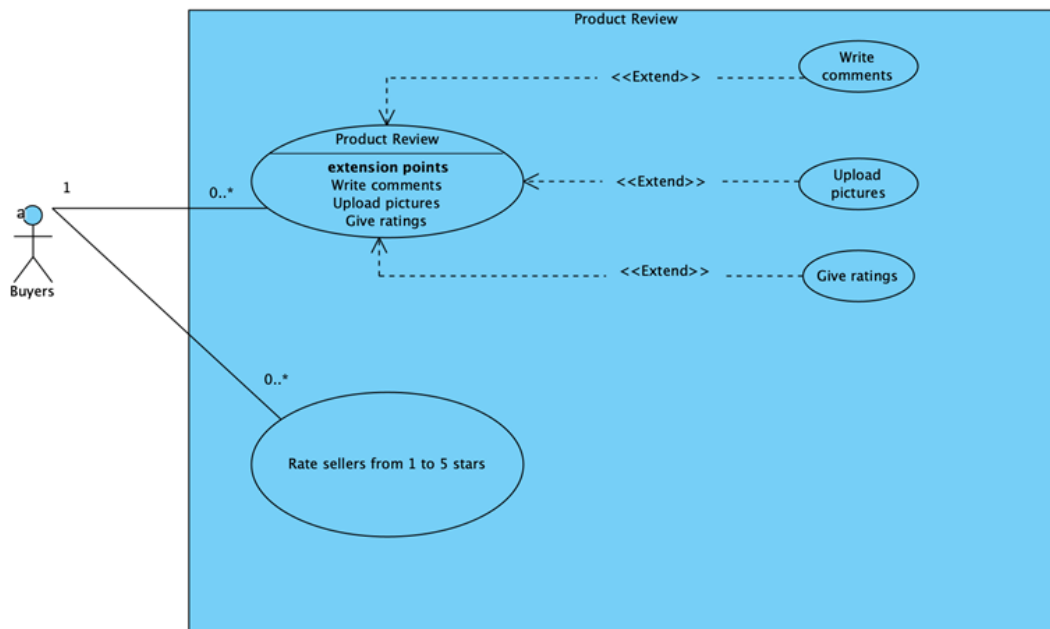
4.9 User Review

Similar to eBay, sellers will be evaluated based on their number of positive and negative reviews and given a quantitative rating. These ratings are incredibly important in determining the reliability of a seller. Nevertheless, a quantitative rating alone is not sufficient as it can be easily manipulated by “false” reviews. To this end, it is incredibly important that users can also leave comments and images to justify their positive or negative ratings. A user can only review a seller fifteen days following the purchase of a product(s).

Feature Implementation Priority: Low

4.9.1 Use Case Diagram

Figure 4.6: User Review Use Case Diagram



4.9.2 Functional Requirements

Functional requirements are shown in **Table 4.9**.

Table 4.9: User Review Functional Requirements (FR-UR)[7].

Feature ID	Feature Name	Description
FR-UR1	Post Review	The application shall allow a user to review a product that they have purchased. A review can only be posted fifteen days following the date of purchase.
FR-UR2	Display Reviews	The application shall allow a user to view all reviews made on a seller's profile.

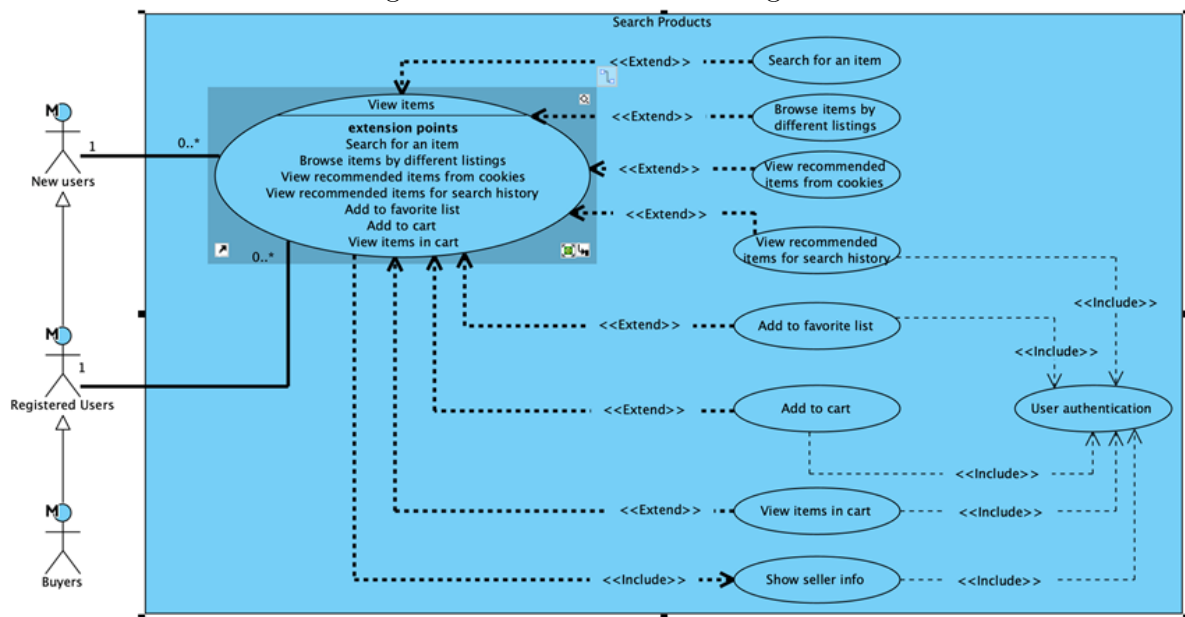
4.10 Search, Sort and Trending Features

The search-and-sort function allows the users to browse the marketplace by limiting their options to a set of inputted keywords, which is an essential feature for platforms that are expected to host several thousands of users. The omission of this feature will make navigating through the market a lot harder for those who are looking for a particular product. However, for simplicity, the search keywords will be limited to the title and category of the listings and will not include searching the description of a listing for matching keywords.

Feature Implementation Priority: Medium-High

4.10.1 Use Case Diagram

Figure 4.7: Search Use Case Diagram



4.10.2 Functional Requirements

Functional requirements are shown in Table 4.10.

Table 4.10: Search, Sort and Trending Functional Requirements (FR-SST)[7].

Feature ID	Feature Name	Description
FR-SST1	Search Store	The application shall allow a user to input keywords into a search bar and initiate a search across the entire marketplace.
FR-SST2	Sort Store by price	The application shall allow a user to sort the store page by ascending and descending price.
FR-SST3	Sort Store by Release Date	The application shall allow a user to sort the store page by the newest/oldest release.
FR-SST4	Sort Store by Alphabetical Order	The application shall allow a user to sort the store page by alphabetical or reverse alphabetical order.
FR-SST5	Display Trending Listings	The application shall display the trending listings in a banner on the storefront.
FR-SST6	Update Trending List	The system shall periodically perform an update on the list of trending products and sellers.

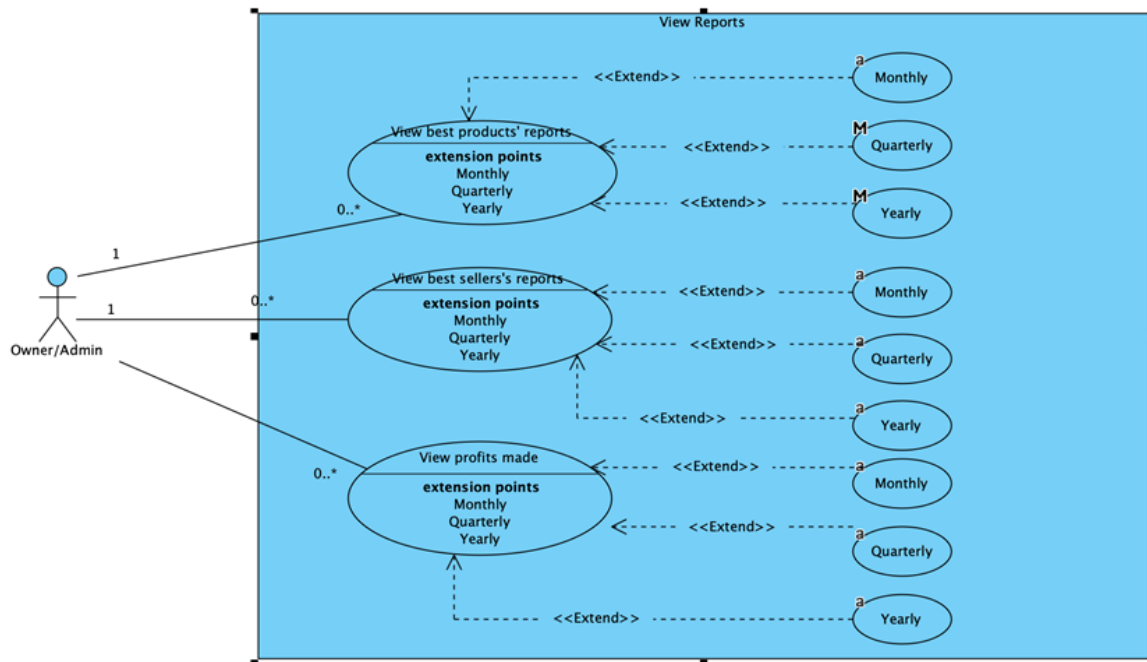
4.11 Administrator Profile

An administrative profile is a user feature that is necessary for system maintenance and also grants access to the “Admin” privileged tools such as sales reports, and revenue generated, list of user profiles, etc.

Feature Implementation Priority: Medium-High

4.11.1 Use Case Diagram

Figure 4.8: Administrative Use Case Diagram



4.11.2 Functional Requirements

Functional requirements are shown in **Table 4.11**.

Table 4.11: Administrative Functional Requirements (FR-AP)[7].

Feature ID	Feature Name	Description
FR-AP1	Display Sales Data	The application shall allow an administrator to view all sales data, which includes all transactions made through the website.
FR-AP2	Display Platform Revenue	The application shall allow an administrative user to view data related to the growth and profitability of the platform.
FR-AP3	Edit Transaction Fee	The application shall allow an administrative user to change the fee charged to the seller upon completing a transaction.
FR-AP4	Delete User	The application shall allow an administrator to remove a user account.
FR-AP5	Edit User	The application shall allow an administrator to edit the user profile of any registered user.
FR-AP6	Force Maintenance	The application shall allow an administrative user to force system maintenance.
FR-AP7	Deposit Earnings to PayPal	The application shall allow an administrative user to deposit the site's earnings in the owner's saved PayPal account.

5 Other Non-Functional Requirements

This section discusses the non-functional requirements that outline how the system is designed to behave.

5.1 Performance Requirements

Non-functional performance requirements are shown in **Table 5.1**.

Table 5.1: Performance Non-Functional Requirements (NFR-PREQ)[7].

Feature ID	Feature Name	Description
NFR-PREQ1	Multi Browser Support	A type of online platform will not limit the website design.
NFR-PREQ2	Scalability	The design will be scalable to maximize the volume of the transactions (support at least 100 clients and 100 sellers).
NFR-PREQ3	Ease Of User Creation	Account creation and authentication is a minimalist process.
NFR-PREQ4	Timely Transaction Processing	Transactions should be processed quickly, alerts to the users are immediate, to minimize delays.

5.2 Safety Requirements

Non-functional safety requirements are shown in **Table 5.2**.

Table 5.2: Safety Non-Functional Requirements (NFR-SFT)[7].

Feature ID	Feature Name	Description
NFR-SFT1	Receipt Generation	The system will provide the customer with a proof of purchase upon payment completion.
NFR-SFT2	Concealed Seller Information	The system hides the contact details of the seller from buyers.
NFR-SFT3	Continuous Support	Customer support contact information will be accessible to all buyers upon purchase.
NFR-SFT4	Inclusive Environment	The platform will not tolerate any illegal or sexual content.

5.3 Security Requirements

Non-functional security requirements are shown in **Table 5.3**.

Table 5.3: Security Non-Functional Requirements (NFR-SRT)[7].

Feature ID	Feature Name	Description
NFR-SRT1	User Monitoring	The users have to authenticate themselves.
NFR-SRT2	Payment Security	All payments are handled by a third-party.
NFR-SRT3	Complete Account Management	The users' passwords are re-settable.

5.4 Software Quality Attributes

Non-functional software quality attributes requirements are shown in **Table 5.4**.

Table 5.4: Software Quality Attributes Non-Functional Requirements (NFR-SQA)[7].

Feature ID	Feature Name	Description
NFR-SQA1	Back-ended Site Data	Payment processing and data management are separate and not handled by the client.
NFR-SQA2	Mobile Support	The design is mobile friendly.
NFR-SQA3	Quality of UI Experience	The user interface does not contain illegible fonts, and the selected colours complement each other.
NFR-SQA4	Content Simplicity	The web pages will contain a fair balance of text and visual material.
NFR-SQA5	Payment Reliability and Consistency	The system should not lose payments that pass-through PayPal.

5.5 Business Rules

Non-functional business requirements are shown in **Table 5.5**.

Table 5.5: Business Rules Non-Functional Requirements (NFR-BR)[7].

Feature ID	Feature Name	Description
NFR-BR1	Real-Time Accounting	The website posts all transactions to the ledger, and a report can be viewed by management in real-time.
NFR-BR2	Omnipresent User Support	The website users can reach out to customer support for any help with our site.
NFR-BR3	Shipping Time Objectives	The seller has a time limit to notify the shipment of the product.
NFR-BR4	Buyer Account Flexibility	We will not limit buyers to one address, but they will be limited to a maximum of three different addresses.

6 Unit Testing

This section tabulates the unit tests that are to be performed on the system features under functional and non-functional requirements.

6.1 Traceability Matrix

The association between the test cases, and the functional and non-functional requirements will be tracked using a traceability matrix. This describes the test cases in which a particular requirement will be tested and will also display whether the requirement has been met or has failed.

Table 6.1: Traceability Matrix

Feature ID	Feature Name	Test Cases	Status
FR-ACS1	Validate Email Address	T-0001 T-0002	No run No run
FR-ACS2	Validate Password	T-0003 T-0004	No run No run
FR-ACS3	Emit Confirmation Email	T-0005 T-0006	No run No run
FR-ACS4	Register Account	T-0007	No run
FR-PM1	Change Email Address	T-0008	No run
FR-PM2	Change Password	T-0009	No run
FR-PM3	Add Shipping Address(es)	T-0010 T-0011	No run No run
FR-PM4	Remove Shipping Address(es)	T-0012	No run
FR-PM5	Edit Shipping Address(es)	T-0013	No run
FR-PM6	Add Phone Number	T-0014	No run
FR-PM7	Remove Phone Number	T-0015	No run
FR-PM8	Edit Phone Number	T-0016	No run
FR-PM9	View Transaction History	T-0017 T-0018	No run No run
FR-PM10	View Receipts	T-0019 T-0020	No run No run
FR-PM11	View Product Listings	T-0021 T-0022	No run No run
FR-PM12	View Ratings	T-0023	No run
FR-PM13	View Reviews	T-0024	No run
FR-PM14	View Order Status	T-0025	No run
FR-PM15	Change Order Status	T-0026	No run
FR-AR1	Recover Password	T-0027	No run
FR-AR2	Emit Temporary Password	T-0027	No run
FR-LOG1	Login User	T-0028 T-0029	No run No run
FR-LOG2	Login Administrator	T-0030	No run
FR-SC1	Add Product Listing(s)	T-0031	No run
FR-SC2	Remove Product Listing(s)	T-0032	No run
FR-SC3	Update Quantity	T-0033	No run
FR-SC4	Save Cart	T-0034	No run
FR-PS1	Select Shipping Address	T-0035 T-0036	No run No run
FR-PS2	Select Shipping Method	T-0037	No run
FR-PS3	Proceed to Checkout	T-0038	No run
FR-PS4	Payment Processing	T-0039	No run
FR-PS5	Emit Notification	T-0040	No run
FR-PS6	Transaction Confirmation	T-0041	No run
FR-PLS1	Create Listing	T-0042	No run
FR-PLS2	Edit Listing	T-0043	No run
FR-PLS3	Set Refund	T-0044	No run
FR-PLS4	Update Quantity	T-0043	No run
FR-PLS5	Set Condition	T-0045	No run
FR-PLS6	Set Category	T-0046	No run

FR-PLS7	Set Description	T-0047 T-0048	No run No run
FR-PLS8	Set Title	T-0049	No run
FR-PLS9	Upload Images	T-0050	No run
FR-PLS10	Hide Listing	T-0051	No run
FR-PLS11	Unhide Listing	T-0052	No run
FR-PLS12	Delete Listing	T-0053	No run
FR-RS1	Demand Refund	T-0054 T-0055 T-0056	No run No run No run
FR-RS2	Emit Refund	T-0057	No run
FR-UR1	Post Reviews	T-0058 T-0059 T-0060	No run No run No run
FR-UR2	Display Reviews	T-0061	No run
FR-SST1	Search Store	T-0062	No run
FR-SST2	Sort Store by price	T-0063	No run
FR-SST3	Sort Store by Release Date	T-0064	No run
FR-SST4	Sort Store by Alphabetical Order	T-0065	No run
FR-SST5	Display Trending Listings	T-0066	No run
FR-SST6	Update Trending List	T-0067	No run
FR-AP1	Display Sales Data	T-0068	No run
FR-AP2	Display Platform Revenue	T-0069	No run
FR-AP3	Edit Transaction Fee	T-0070	No run
FR-AP4	Delete User	T-0071	No run
FR-AP5	Edit User	T-0072	No run
FR-AP6	Force Maintenance	T-0073 T-0074 T-0075	No run No run No run
FR-AP7	Deposit Earnings to PayPal	T-0076	No run
NFR-PREQ1	Multi Browser Support	TBA	No run
NFR-PREQ2	Scalability	TBA	No run
NFR-PREQ3	Ease of User Creation	TBA	No run
NFR-PREQ4	Timely Transaction Processing	TBA	No run
NFR-SFT1	Receipt Generation	TBA	No run
NFR-SFT2	Concealed Seller Information	TBA	No run
NFR-SFT3	Continuous Support	TBA	No run
NFR-SFT4	Inclusive Environment	TBA	No run
NFR-SRT1	User Monitoring	TBA	No run
NFR-SRT2	Payment Security	TBA	No run
NFR-SRT3	Complete Account Management	TBA	No run
NFR-SQA1	Back-ended Site Data	TBA	No run
NFR-SQA2	Mobile Support	TBA	No run
NFR-SQA3	Quality of UI Experience	TBA	No run
NFR-SQA4	Content Simplicity	TBA	No run
NFR-SQA5	Payment Reliability and Consistency	TBA	No run
NFR-BR1	Real-Time Accounting	TBA	No run
NFR-BR2	Omnipresent User Support	TBA	No run
NFR-BR3	Shipping Time Objectives	TBA	No run
NFR-BR4	Buyer Account Flexibility	TBA	No run

7 Conclusion

Through the requirements and technologies outlined in this document, this software will produce a website that hosts an e-commerce website. The result is to provide a platform that is responsive and intuitive to use for both sellers and consumers alike. A set of requirements and system features, as requested by clients, have been defined to meet their expectations better.

We will use this document throughout development as a manifesto for the software development process. Any or all of the requirements and scope specifications in this document are subject to change due to client requests. We will explicitly mark any changes to this document in a “Revision History” section listed below the Table of Contents.

Alphabetical Index

account, 2, 3, 5, 7–12, 16, 20–22

admin, 3, 4

buyer, 3, 11, 13, 14, 16

database, 5, 7, 8

e-commerce, 1, 4, 26

feature, 6, 9–14, 16–22

Flask, 5, 8

framework, 5, 8

hardware, 7

interface, 1, 6–8, 22

JavaScript, 5, 7, 8

JSON, 8

library, 8

NodeJS, 5

performance, 21

personas, 3

platform, 1, 6, 9, 20, 21, 26

product, 1–3, 7, 8, 13, 14, 16–18, 22

Python, 5, 7, 8

revenue, 19

section, 26

security, 5, 21

seller, 2, 3, 8, 11, 13, 14, 16, 17, 20–22

server, 7, 8

software, 1, 4, 6–8, 13, 22, 26

SQLAlchemy, 5, 8

use case, 9

user, 2–14, 16, 17, 19, 20, 22

vuejs, 5

vuetify, 5, 8

website, 1–5, 10, 20–22, 26

Glossary

application programming interface A set of functions and procedures allowing the creation of applications that access the features or data of an operating system, application, or other service[9].

back-end The part of a computer system or application that is not directly accessed by the user, typically responsible for storing and manipulating data[9].

Brute Force Attack A brute force attack is a trial-and-error method used to obtain information such as a user password or personal identification number (PIN).

database A structured set of data held in a computer, especially one that is accessible in various ways[9].

drop-shadows is a visual effect consisting of a drawing element which looks like the shadow of an object, giving the impression that the object is raised above the objects behind it.

form factor The physical size and shape of a piece of computer hardware[9].

framework A basic structure underlying a system[9].

front-end relating to or denoting the part of a computer system or application with which the user interacts directly[9].

interface Is a mark up language specially suited for scientific documents[9].

library A collection of programs and software packages made generally available, often loaded and stored on disk for immediate use[9].

server A computer or computer program which manages access to a centralized resource or service in a network[9].

stand-alone of computer hardware or software able to operate independently of other hardware or software[9].

standardized open-sources an open-source interface, API or toolkit that has become an industry standard.

toolkit A set of software tools[9].

Acronyms

API Application Programming Interface.

HTTPS Hypertext Transfer Protocol Secure.

IEEE The Institute of Electrical and Electronic Engineers.

JSON JavaScript Object Notation.

Bibliography

- [1] Matt Brian. *Google's new 'Material Design' UI coming to Android, Chrome OS and the web*. June 2014. URL: <https://www.engadget.com/2014/06/25/googles-new-design-language-is-called-material-design/>.
- [2] Vivian Cromwell. *Interview with Evan You*. Nov. 2016. URL: <https://web.archive.org/web/20170603052649/https://betweenthewires.org/2016/11/03/evan-you/>.
- [3] *Flask*. URL: <https://palletsprojects.com/p/flask/>.
- [4] Apache Software Foundation. *The Apache HTTP Server Project*. URL: <http://httpd.apache.org>.
- [5] *Gunicorn*. URL: <https://gunicorn.org>.
- [6] Ecma International. *ECMAScript® 2015 Language Specification*. June 2015. URL: <http://www.ecma-international.org/ecma-262/6.0/>.
- [7] *Interview with Dr. Hakim Mellah and Dr. Aiman Hanna*. personal communication. In-Person Interview. Sept. 2019.
- [8] *JSON*. URL: <http://www.json.org>.
- [9] Oxford University Press. *Lexico dictionary*. Oct. 2019. URL: <https://www.lexico.com>.
- [10] *SQLAlchemy*. URL: <https://www.sqlalchemy.org>.
- [11] Karl E. Wieggers. *IEEE Template for System Requirement Specification Documents*. 1999. URL: https://web.cs.dal.ca/~hawkey/3130/srs_template-ieee.doc.