

Table of Contents

Table of Contents	2
Revision History	3
Introduction	4
1.1 Purpose	4
1.2 Document Conventions	4
1.3 Product Scope	4
1.4 Intended Audience	5
Overall Description	6
2.1 Product Perspective	6
2.2 Product Functions	6
2.3 User Classes and Characteristics	7
2.4 Operating Environment	8
2.5 Design and Implementation Constraints	8
2.6 User Documentation	8
2.7 Assumptions and Dependencies	8
External Interface Requirements	9
3.1 User Interfaces	9
Log In / Sign Up	9
Forget Password	9
Connect / Export	10
Products	12
Platforms	13
3.2 Hardware Interfaces	14
3.3 Software Interfaces	14
3.4 Communications Interfaces	14
System Features	14
4.1 Login	14
4.2 Register	16
4.3 Reset Password	17
4.4 Connect Shop	18
4.5 Export Data	19
4.6 Advanced Search	20
4.7 Update Quantity	20
4.8 Delete Product	21
5. Other Nonfunctional Requirements	22

5.1 Performance Requirements	22
5.2 Safety Requirements	22
5.3 Security Requirements	22
5.4 Software Quality Attributes	22
5.4.1 Reliability	22
5.4.2 Maintainability	23
5.4.3 Reusability	23
5.4.4 Portability	23
5.4.5 Supportability Requirements	23
5.4.6 Usability Requirements	23
5.5 Business Rules	24
Data Dictionary	24
Appendix B: Analysis Models	26
Conceptual Model	26
Conceptual Diagram	27
Dialog Map	28
Sequence Diagram	29
System Architecture	30
Appendix C: To Be Determined List	30
Appendix D: Test Cases	30
Appendix E: User Manual	31

Revision History

Name	Date	Reason For Changes	Version
Software Requirement Specifications	8/11/21	Draft	1.0
Software Requirement Specifications	13/11/21	Final Draft	1.1

1. Introduction

1.1 Purpose

Our team will develop a web application that enables the quantity of stock to be updated in real time across all the different ecommerce platforms. This project will be considered complete when the quantity of stock can be updated in real time. This project supports E-Commerce vendors to ensure a smooth online shopping for consumers.

1.2 Document Conventions

Conventions	Font Size	Font Type	Font Style
Headings	18	Times	Bold
Sub Headings	14	Times	Bold
Sub-sub Headings	11	Times	Italic
Normal Text	11	Times	None
Important Names	11	Times	Bold

1.3 Product Scope

The improvement in technology and convenience gave rise to the ever-growing market of E-Commerce. In Singapore, with the limited land space, E-Commerce is proving to be the popular alternative for the transaction of goods over the traditional brick and mortar stores.

The burgeoning E-Commerce space has resulted in large amounts of transactions occurring daily in real time. Consequently, Vendors who are operating on E-Commerce sites may have difficulty in managing their Inventory when their stocks differ from the store product quantity on their E-Commerce site. In the event where Vendors operate on multiple E-Commerce sites, the data discrepancy increases which may result in invalid transactions.

LazeAPI aims to grant vendors the ability to manage their store product details across multiple E-Commerce sites on a single centralised platform. A one-stop service portal that updates inventory across different E-Commerce sites concurrently. Synchronising product details to provide a centralised inventory management system for Vendors or Sellers. It will also allow them to export the latest inventory quantity for the respective ecommerce platforms.

1.4 Intended Audience

Target user: E-Commerce Vendors seeking a centralised inventory management solution across E-commerce platforms form our Client base

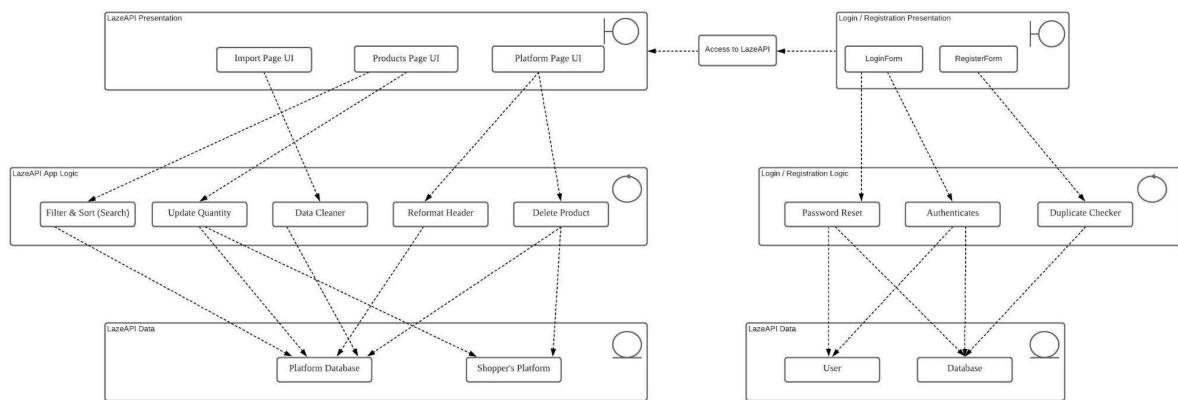
Group	Purpose of SRS document
Clients	Formal documentation identifying the problems to be solved and the software application for the team's solution. Clients may refer to the <i>Project Abstract, System features, Functional Requirements and Non-Functional Requirements</i> .
Developers	Formal documentation relaying design specification based on agreement of requirements with the client. Developers may refer to the <i>Project Abstract, System Features, UI Mock-up Prototype, Functional Requirements, Non-functional Requirements, Use Case Descriptions and Use Case Diagram</i>
Testers	Formal documentation forming the guideline to develop test cases, which will be applied to address requirements for verification for the testing phase. Testers may refer to the <i>Project Abstract, System Features, UI Mock-up Prototype, Functional Requirements, Non-functional Requirements and Use Case Descriptions</i>

Table 1: Purpose and LazeAPI references for stakeholders

2. Overall Description

2.1 Product Perspective

LazeAPI is a one stop real time inventory management web application, a utility tool created for vendors who sell on E-Commerce platforms. LazeAPI houses a centralised database built through the connection of shops across multiple platforms by vendors, consolidating stock details into a clean and formatted data output. LazeAPI grants real time inventory updates through order details on respective platforms, resolving the conflict of inventory inaccuracy and redundancy. Ultimately, LazeAPI seeks to empower vendors through the improvement of convenience and reduced manpower constraints for inventory management.



This diagram shows a 3-layered system architecture. The top layer represents the presentation , which is a user interface that the user interacts with. The middle layer represents the app logic which is like the function calls. The bottom layer is the persistent data which consists of where it collects and stores the information which can be accessed and updated easily.

The layered architecture on the right represents the architecture for handling the login , registration and resetting of password for our web application. Whereas the left represents how our main LazeAPI interacts. They are separated because both interact with different databases.

2.2 Product Functions

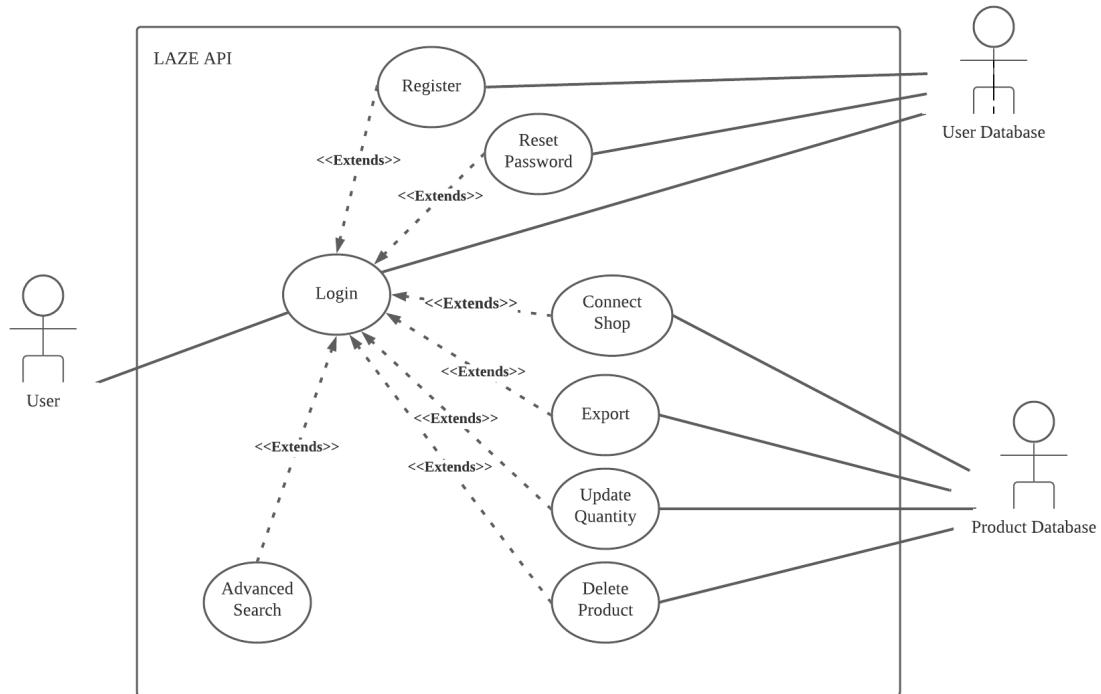


Fig. 1: Use Case Diagram for LazeAPI

The system must provide, at a minimum, the following functions in accordance with the other requirements described within this SRS document.

- Register and login as unique user of the application
- Resetting of password if the users forget his/her password
- Connect new ecommerce platform to the application for import of data
- Export platform data from the application to a downloaded excel file
- Update quantity of product across multiple platform through the application
- Search for the desired results through the search bar
- Deletion of product which will be reflected across all platforms

2.3 User Classes and Characteristics

The expected user for the web application, LazeAPI will be ecommerce store owners who want to manage their inventory across multiple ecommerce platforms.

There are 2 distinct users for the application:

1. Super User/ Administrative User:
 - Able to access user database to modify password

2. Registered User:
 - Able to reset password
 - Able to access the LazeAPI main page
 - Able to update the quantity for a specific product
 - Able to search for the desired product
 - Able to connect/ export platform data
 - Able to delete a specific product

2.4 Operating Environment

The web application LazeAPI is built using Visual Studio Code environment with Python Virtual Environment. Front-end development is made entirely using VueJS version 2, major features are divided into individual vue components. Back-end development is done with Django and Flask frameworks with SQLite3 for local databases. Flask manages the API calls and data manipulation from the ecommerce platform while Django handles the authentication and verification of the user login and registration process. The web application is supported by all major browsers that support JavaScript and is up to date.

2.5 Design and Implementation Constraints

A stable internet connection is required as users on our LazeAPI web app need to connect to their specific shopping platforms. If the user has not connected to any shopping platform, on the overview and product summary page, they will be shown an unpopulated table.

Our SQLite3 database will not hold the user's password due to security and privacy reasons.

2.6 User Documentation

Documentation of the user's view is further emphasised in Appendix D as a user manual. The following documents each step process upon using the LAZE web application.

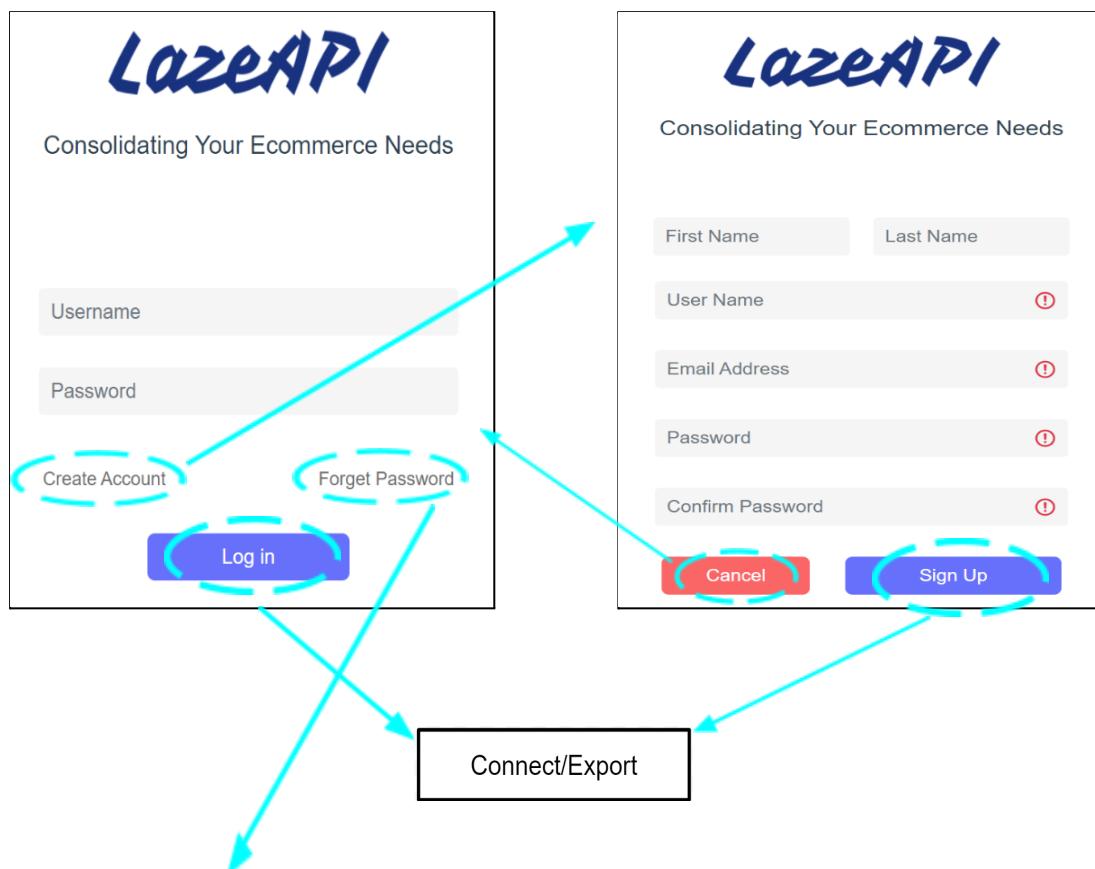
2.7 Assumptions and Dependencies

- 2.7.1 Connections to SQLite3 for database and third-party APIs is successful
- 2.7.2 Users have stable internet connection while using the webapp
- 2.7.3 Browser used by the user is up to date with the supported version of VueJS

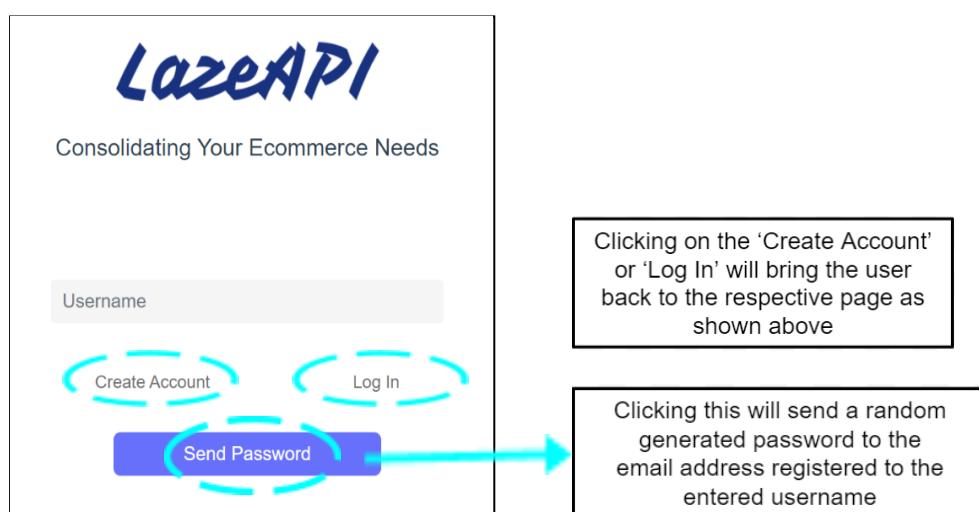
3. External Interface Requirements

3.1 User Interfaces

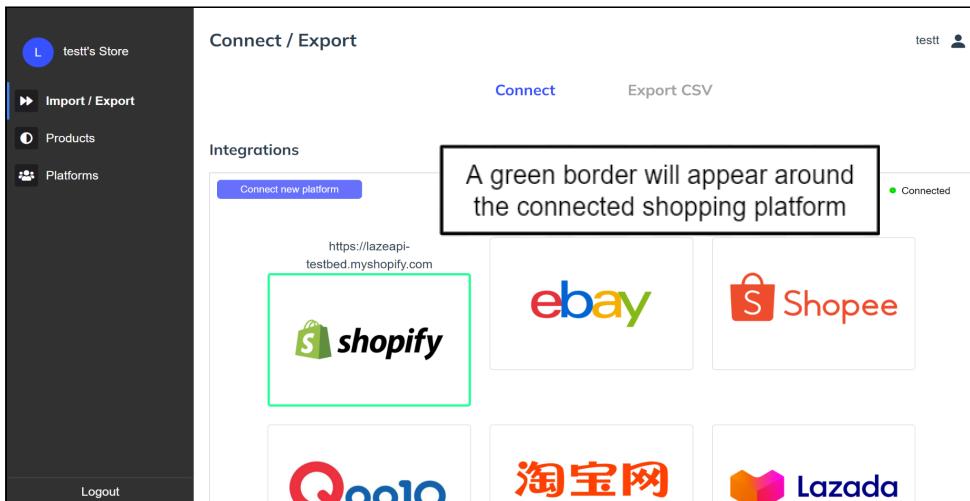
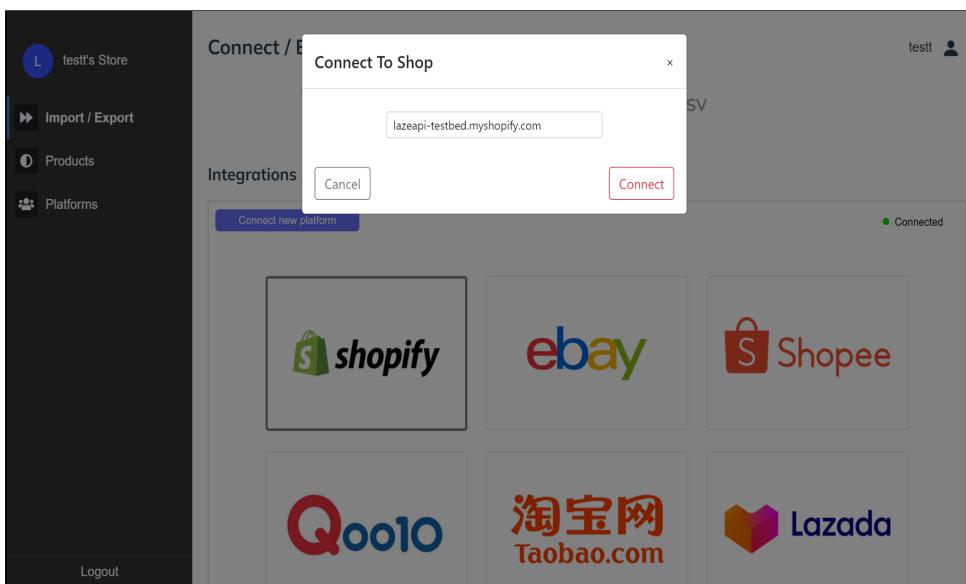
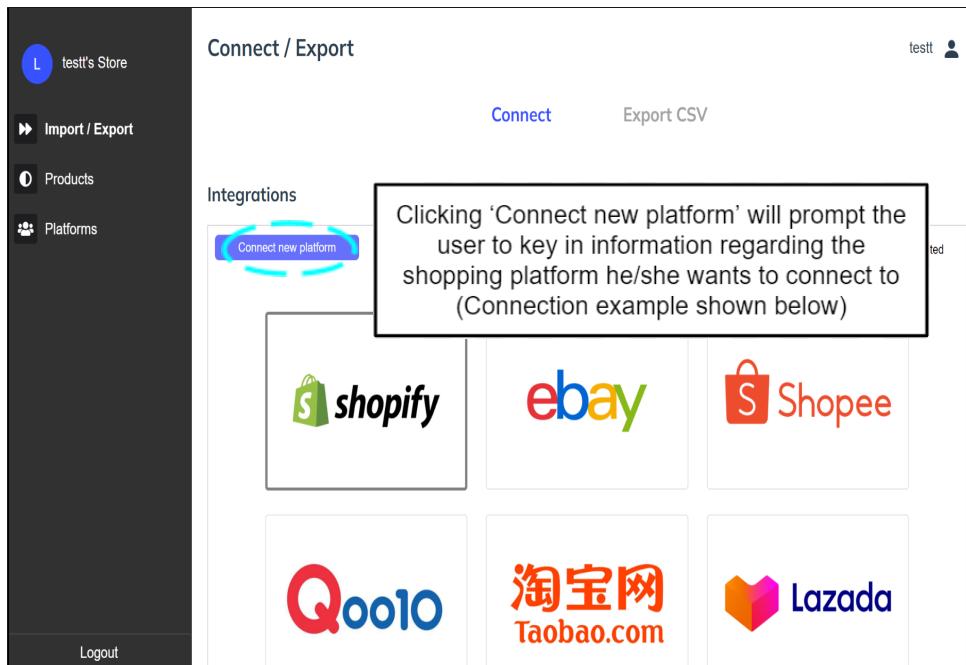
Log In / Sign Up



Forgot Password



Connect / Export



The screenshots show a user interface for managing store integrations, specifically focusing on the 'Connect / Export' section.

Screenshot 1: The 'Integrations' section displays several platform logos: Shopify (selected), ebay, Shopee, Qoo10, Taobao, and Lazada. A callout box highlights the 'Export' button, which is circled in red. Another callout box provides instructions: "To export, select the desired shopping platform and click 'Export'. After, a CSV file will automatically be downloaded".

Screenshot 2: The 'Integrations' section shows the same platforms. A callout box points to the 'Logout' button at the bottom left of the screen. Another callout box points to the 'export.csv' file in the download history, with the text: "User can find the CSV file in their 'Downloads' directory".

Screenshot 3: A modal window titled "Logout screen" asks "Are you sure you want to log out?". It has "Cancel" and "Logout" buttons. The background shows the same 'Connect / Export' interface with the 'Connected' status indicator.

Products

testt's Store

Import / Export

Products

Platforms

Logout

Products

Product

Image	Title	Tags
	Juliet (7 Roses & 2 Cotton flowers) Default Title	
	Preserved Rose Bag Charm with FurBall Blush Pink	
	Preserved Rose Bag Charm with FurBall Coral Peach	
	Preserved Rose Bag Charm with FurBall Lavender Blue	
	Preserved Rose Bag Charm with FurBall Indigo	

Clicking on an item allows user to see more details regarding the item

testt's Store

Import / Export

Products

Platforms

Logout

Products

Product

Image	Title	Tags
	Juliet (7 Roses & 2 Cotton flowers) Default Title	
	Preserved Rose Bag Charm with FurBall Blush Pink	
	Preserved Rose Bag Charm with FurBall Coral Peach	
	Preserved Rose Bag Charm with FurBall Lavender Blue	
	Preserved Rose Bag Charm with FurBall Indigo	

Item
Juliet (7 Roses & 2 Cotton flowers)
Default Title

SKU
JULIET

Quantity

Platforms
shopify

Platforms

Summary view of all the products user has across all shopping platforms

Platforms

Summary

Shopify

Logout

Image	Title	Platform	Collection	Delete
	Juliet (7 Roses & 2 Cotton flowers) Default Title	shopify		<button>Delete</button>
	Preserved Rose Bag Charm with FurBall Blush Pink	shopify		<button>Delete</button>
	Preserved Rose Bag Charm with FurBall Coral Peach	shopify		<button>Delete</button>
				<button>Delete</button>

User can choose an item to delete and it will be removed from the shopping platform

In the case where the user has too many variety of products, a search bar is available and reflects the searches in real time

Platforms

Summary

Shopify

Logout

Image	Title	Platform	Collection	Delete
	Juliet (7 Roses & 2 Cotton flowers) Default Title	shopify		<button>Delete</button>
	Sarah (2 Roses and 2 Cottons) Default Title	shopify		<button>Delete</button>
	Stella (3 Roses & 2 Cotton flowers) Default Title	shopify		<button>Delete</button>
	Trisha (2 Roses & 3 Cotton flowers) Default Title	shopify		<button>Delete</button>

cotton

User can select a platform to view products listed ONLY on that shopping platform

Platforms

Summary

Shopify

Logout

Image	Title	Platform	Collection	Delete
	Juliet (7			
	Preserved Rose Bag Charm with FurBall Blush Pink			
	Preserved Rose Bag Charm with FurBall Coral Peach			
	Preserved Rose Bag Charm with FurBall Lavender Blue			
	Preserved Rose Bag Charm with FurBall Indigo			

Item
is Selected

No Rows Selected

SKU

Quantity

No Rows Selected

Platforms

No Rows Selected

3.2 Hardware Interfaces

All server-side components must execute on server-class computers. All client-side components must execute on workstation-class and personal-class computers.

3.3 Software Interfaces

Databases:

- Ecommerce Platform: Platform API
- User Login Database: Django Database
- Product Data: Flask, SQLite3

App:

- Version: 1.00

Libraries/Dependencies:

- BootstrapVue
- Vue SideBar Menu
- Vue Router
- Vuetify
- FontAwesome Icon
- PortalVue
- RestAuth
- Django smtplib

3.4 Communications Interfaces

Email Service: Python SMTP Server

Data Transfer: REST API

4. System Features

4.1 Login

4.1.1 Description and Priority

Authenticate the user for access to the inventory database | High

4.1.2 Stimulus/Response Sequences

1. User inputs ‘USERNAME’ and ‘PASSWORD’.

2. User clicks on the ‘Log In’ button.
3. System interacts with the user database and authenticates the user.
4. User is logged in.

4.1.3 Alternative flows

AF1-S2: User clicks on the ‘Create Account’ button.

1. User is led to the user creation page.

AF2-S2: User clicks on the ‘Forget Password’ button.

1. User is led to the reset password page.

AF3-S3: If the PASSWORD or USERNAME is wrong

1. User is shown ‘Unable to log in with provided credentials’.

AF4-S3: If the PASSWORD is empty

1. User is shown ‘This field may not be blank’.

AF5-S3: If the USERNAME is empty

1. User is shown ‘Must include “username” and “password”’.

4.1.4 Functional Requirements

REQ-1: Users must be able to choose to login.

1.1 Registered users must be able to login.

1.1.1 Users must be able to sign out at any point in time.

1.1.2 System must be able to identify and prompt the user to re-enter their password if they entered it wrongly.

REQ-2: Information Databases

2.1 Database management systems must be able to retrieve the information needed when the app demands it.

2.1.1 Database must be able to store the login credentials where password is encrypted when storing.

2.2 Database management systems must be able to modify databases when the app has a new element to add to the database.

2.2.1 Database must be updated when a new user registers.

4.2 Register

4.2.1 Description and Priority

Register user into user database | Medium

4.2.2 Stimulus/Response Sequences

1. User selects the ‘Create Account’ button.
2. User keys in first name, last name, username, email address and password.
3. User clicks on the ‘Sign Up’ button.
4. System checks whether the user is already registered.
5. System creates a new user.
6. User is logged in.

4.2.3 Alternative flows

AF1-S3: User clicks on the ‘Cancel’ button.

1. System automatically returns to login page

AF2-S4: If username already exist

1. System informs the user ‘A user with that username already exists’.
2. User is prompted to enter another username.

AF3-S4: If email already exist

1. System informs the user ‘A user is already registered with this e-mail address’.
1. User is prompted to enter another email.

AF4-S4: If password is fails password complexity checker

1. System informs the user ‘This password is too common’
2. User is prompted to enter another password.

AF4-S4: If password is less than 8 characters

1. System informs the user ‘This password is too short. It must contain at least 8 characters’.
2. User is prompted to enter another password.

AF5-S4: If password is only consists of numbers

3. System informs the user ‘This password is entirely numeric’.
4. User is prompted to enter another password.

4.2.4 Functional Requirements

REQ-1: Users must register for an account before logging in

1.1 Username constraints

 1.1.1 Username must be unique

1.2 Email constraints

 1.2.1 Email must be unique

1.3 Password constraints

 1.3.1 Password must be at least 8 characters long

 1.3.2 Password must be not found in a list of common passwords

1.4 System must be able to store newly created accounts.

REQ-2: Information Databases

2.1 Database management systems must be able to retrieve the information needed when the app demands it.

 2.1.1 Database must be able to store the login credentials.

 2.1.1.1 Password must be encrypted before storing in the database

2.2 Database management systems must be able to modify databases when the app has a new element to add to the database.

 2.2.1 Database must be updated when a new user registers.

4.3 Reset Password

4.2.1 Description and Priority

Reset password and sends user new password | Low

4.2.2 Stimulus/Response Sequences

1. User selects the ‘Forget Password’ button.
2. User keys in the username of the registered account.
3. User clicks on the ‘Send Password’ button.
4. System checks whether the username is already registered.
5. System finds the corresponding email linked to the username.
6. System generates a new complex password for the account.
7. System sends the generated password to the email.

8. System informs the user ‘Please check your email for password.’.

4.2.3 *Alternative flows*

AF1-S3: If the user clicks on the ‘Create Account’ button

1. User is led to the user creation page.

AF2-S3: If the user clicks on the ‘Log In’ button

1. User is led to the log in page.

AF3-S4: If the username does not exist in the user database

1. System informs the user ‘username not found’.

4.2.4 *Functional Requirements*

REQ-1: System must be able to perform the relevant functionalities when the app demands it

1.1 System must be able to generate a new unique and random password

1.2 System must be able to handle email management system

1.2.1 System must be able to send out email containing generated password

REQ-2: Information Databases

2.1 Database management systems must be able to retrieve the information needed when the app demands it.

2.1.1 Database must be able to extract the user account credentials.

2.1.1.1 User’s email must be extracted

4.4 Connect Shop

4.4.1 *Description and Priority*

Connect a new ecommerce platform directly to the database | High

4.4.2 *Stimulus/Response Sequences*

1. User clicks on the respective platform icon which user wants to connect to
2. System will highlight icon selection with grey outline
3. User clicks the ‘CONNECT NEW PLATFORM’ button.
4. User will input the authentication path for the specified shop
5. System will establish connection to shop
6. Upon connection, platform icon will have a green highlight
7. System will retrieve shop products data
8. System will reformat the data to the match the display table

4.4.3 Functional Requirements

REQ-1: Registered users have access to connect features on web applications.

- 1.1 Registered users must be able to import the listed ecommerce platforms data to our web application.

REQ-2: Information Display

- 2.1 Data management systems must be able to retrieve the information needed when the app demands it.

2.1.1 Systems must be able to use the API provided by the ecommerce platform

2.1.2 Systems must be able to reformat data to match application format

- 2.2 System must be able to update visual indication of the shop icon based on the connection

4.5 Export Data

4.5.1 Description and Priority

Export data from platform into a CSV file | Medium

4.5.2 Stimulus/Response Sequences

1. User clicks on the respective platform icon which user wants to export
2. User clicks on the 'Export' button.
3. System would generate a export.csv file with all the products information from the selected shop
4. System downloads generated file directly to downloads folder

4.5.3 Alternative Flow

AF1-S3: If platform is not connected

1. System will not generate a csv file for download

4.5.4 Functional Requirements

REQ-1: Registered users have access to export features on web applications.

- 1.1 Registered users must be able to export their data from web application and convert them into csv files according to the different shopping platforms information

REQ-2: Information Display

- 2.1 Systems must be able to convert the product data into their corresponding header in the csv file

2.2 Systems must be download the file to the user's computer

4.6 Advanced Search

4.6.1 Description and Priority

Search for specific items in the database | High

4.6.2 *Stimulus/Response Sequences*

1. User types input into the search box
2. System will filter the existing data based on the user input
3. System will display the search result on the same page

4.6.3 Alternative Flow

- I. AF-S3: If no results are found

System informs the user 'There are no records to show.'

4.6.4 Functional Requirements

REQ-1: Information on the result must be based on the existing data extracted using an API.

1.1 App must display the search results with the description matching the search input

1.2 App must display the search results by original order

1.2.1 App must allow users to explore through search results manually.

1.3 App must display the fixed number of search results per page

1.3.1 App must allow users to change to other pages

REQ-2: Information Display

2.1 Data management systems must be able to retrieve the information needed when the app demands it.

4.7 Update Quantity

4.7.1 Description and Priority

User update product quantity | High

4.7.2 *Stimulus/Response Sequences*

1. User selects the product from the list of products to edit quantity
2. System displays product information and quantity on the right panel
3. User can change the quantity number of the product

4. User clicks on the ‘Update’ button to finalize the change in quantity.
5. System push data changes to platform database.

4.7.3 Alternative Flow

AF1-S1: If quantity is changed from the platform itself

1. System will identify the changes in quantity from the shop
2. System will update the product information on the right panel to match the platform quantity

4.7.4 Functional Requirements

REQ-1: Information Database

1.1 Database management systems must be able to retrieve the information needed when the app demands it.

1.1.1 System displays previous values on web applications.

1.1.2 Upon clicking the ‘Update’ button, database values should be updated immediately.

1.1.3 System displays updated values on web applications.

4.8 Delete Product

4.7.1 Description and Priority

User deletes a product for a platform | High

4.7.2 Stimulus/Response Sequences

1. User clicks on the ‘Delete’ button on the product user wants to delete
2. System prompts user to select the platform to delete product from
3. User clicks on the ‘Delete’ button in the prompt
4. System removes the selected product information from the application
5. System updates the platform database to remove the selected product

4.7.3 Alternative Flow

AF1-S3: If user clicks on the ‘Cancel’ button in the prompt

1. System will cancel the request for deletion

4.7.4 Functional Requirements

REQ-1: Information Database

1.1 Database management systems must be able to retrieve the delete the product when the

app demands it

1.1.1 System must allow user to choose which platform to delete the product from

1.2 Deletion of product in the platform database will be reflected in the application.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

1. All linked platforms and inventory systems must be updated within 3 seconds of any changes.
2. All other performance related to storage, memory, and processing should follow industry recommended practices to ensure resource requirements are minimized.

5.2 Safety Requirements

1. The password keyed in by the user is hashed to ensure security and privacy.
2. The user's login credentials are encrypted in the user's database to ensure security and privacy.

5.3 Security Requirements

1. User has to login successfully to access inventory system
2. User password is encrypted and stored in a separate database so in the event of hackers who managed to hack the main database, the password of registered users is not there.
3. If a user enters the wrong password multiple times, his/her account will be locked.
4. The software system must follow industry recommended practices for a secure software development.

5.4 Software Quality Attributes

5.4.1 Reliability

1. The web app should run smoothly and not crash frequently.
2. The web app must be separated between front-end and back-end to allow the admin to update the back end without interrupting the user.
3. Daily backup of data into cloud storage
4. The system should be available for 24 hours a day and 365 days a year
5. The web application should undergo feature testing, load testing, and regression testing prior to release , deployment and with any feature added.
6. The web app must be available at all times, even when there is maintenance on the database fetched through APIs from public sources (e.g. Shopee Open Platform and Shopify.dev)

5.4.2 Maintainability

1. The architecture, design, implementation, and documentation of the application must minimize the maintenance costs of the software system.

5.4.3 Reusability

3. Usage of vue.js allows component reusability enables reusing components as templates for similar system elements
4. An email notification containing a new password is sent to the user when the user clicks on “forget password”.

5.4.4 Portability

1. The web app must be portable and be able to work on various screen sizes.

5.4.5 Supportability Requirements

1. The system must be able to handle a large amount of product data in the event of the user having a huge product listing in his store
2. The system must handle up to 10000 edit and deletion of products a day
3. The system must be able to link up to 20 platforms at any time

5.4.6 Usability Requirements

1. The user should be able to connect, update or delete data in the system in less than 3 steps
2. Action buttons should be obvious (e.g., red for delete, green for update)
3. Save button to be included to “finalise changes” for updating database
4. The web app must have hints wherever user input is required.
5. The web app must guide and prompt error output for the user whenever an invalid to ease the user navigation
6. The registered users must be able to report or provide feedback on system errors through email.
7. The web application should be easy for new or infrequent users to learn how to use it

5.5 Business Rules

1. The app should comply with regulation from the various e-commerce platforms for example Shopee Open Platform API and Shopify.dev API website.
2. Only registered users are able to connect, export and update the quantity using our web application.
3. Superusers or Admin are able to update LazaAPI back end and database only when necessary.

Appendix A: Glossary

Data Dictionary

Terms	Description	Attributes
User	Clients or people that operate on our product. Including our main target audience, E-Commerce Sellers or vendors with products in store	user_id, username, password, contact_number, email, first_name, last_name, access token
Platform	E-Commerce software application where both parties Seller and Consumer can come together and perform transactions. For example, Shopee & Shopify	store_id, platform name
Store Products	Table that shows which platform sells which products and its pricing	SKU, price, price override
Order	Entity created when customer makes a purchase request to E-Commerce stores	order_id, customer name, customer address
Order Items	Table that shows quantity of each product ordered	SKU, quantity
Database	Main database table which consolidates all different csv headers from different shopping platforms into a common header	database_id, SKU, product name, CSV file, product quantity

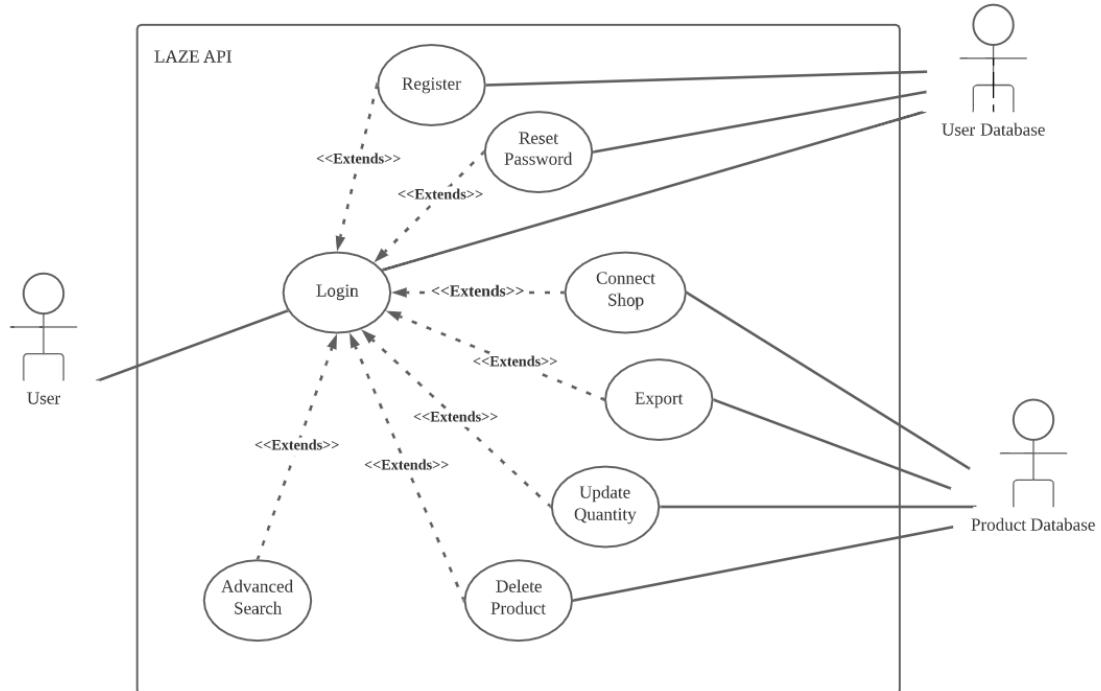
Table 4: Important terms definition and attributes

Object Entity	Object name	Data type	Description	Acceptable values	Accepts null value? (Y/N)
User	<u>user id</u>	Int	Unique identifier generated by database for users		N
	username	String	Login username for user		N
	password	String	Confidential password for personal access		N
	contact_number	Int	Landline or phone number of user		N
	email	String	Email of user for registration and password reset		N
	first_name	String	First name of user		N
	last_name	String	Last name of client		Y
	access token	String	Authentication key to perform API calls on E-Commerce platforms		N
Platform	<u>store id</u>	Int	Unique identifier of store generated by database		N
	name	String	Platform name		N
Store Products	<u>SKU</u>	String	Unique Identifier of product in store		N
	price	Double	Retail price of product		N
	price overwrite	Int	Price of product in store		N
Order	<u>order id</u>	Int	Unique identifier of store generated by database		N
	customer	String	Customer name		N
	address	String	Customer address where order is to be shipped to		N
	order status	String	Either shows accepted or refunded		N
Order Items	<u>SKU</u>	String	Identifier of store product in order		Y
	quantity	Int	Quantity of the products placed in order		N
Database	<u>database id</u>	Int	Unique identifier of store generated by database		N
	SKU	String	Identifier of product to store		N
	product name	String	Name of product		N
	CSV file	-	Imported data from platform's store analytics		N
	Product quantity	Int	Quantity of products in stock		N

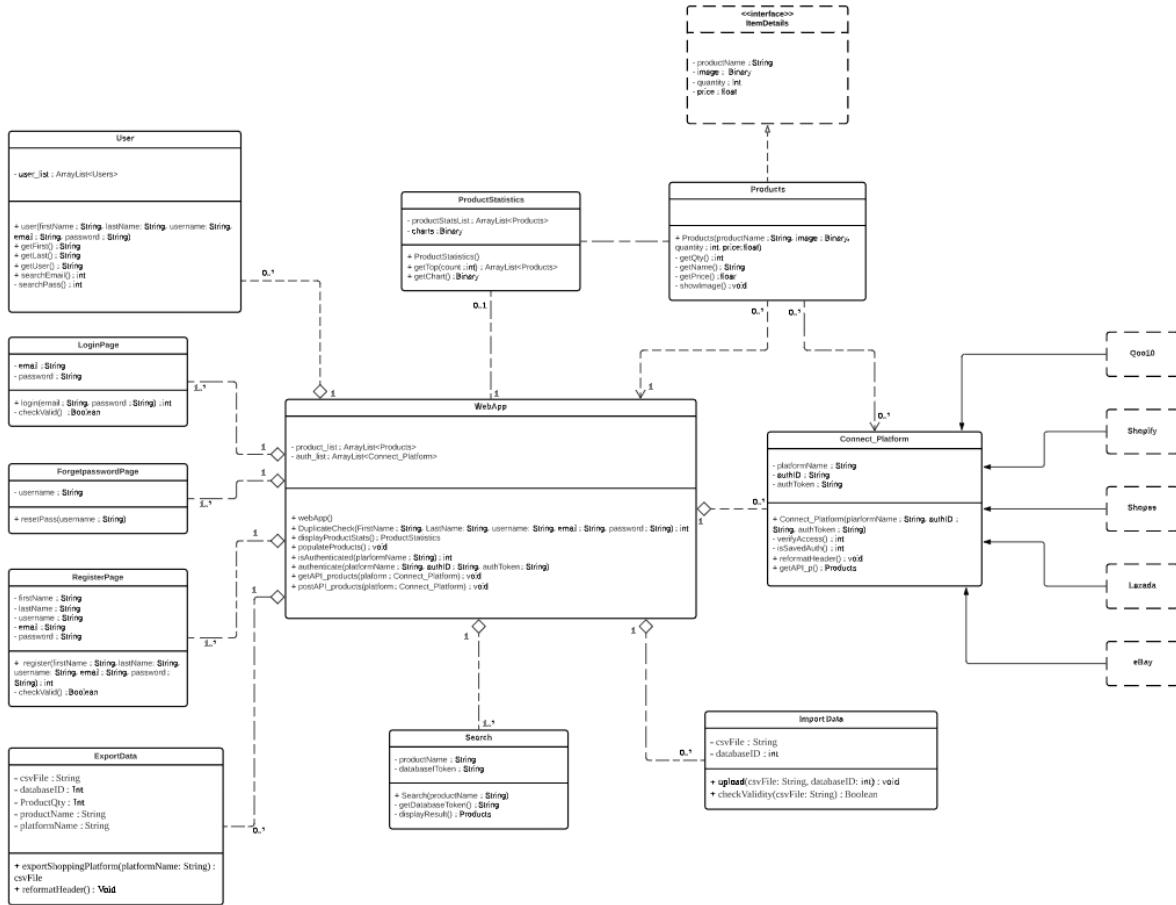
Table 5: Additional definition of terms

Appendix B: Analysis Models

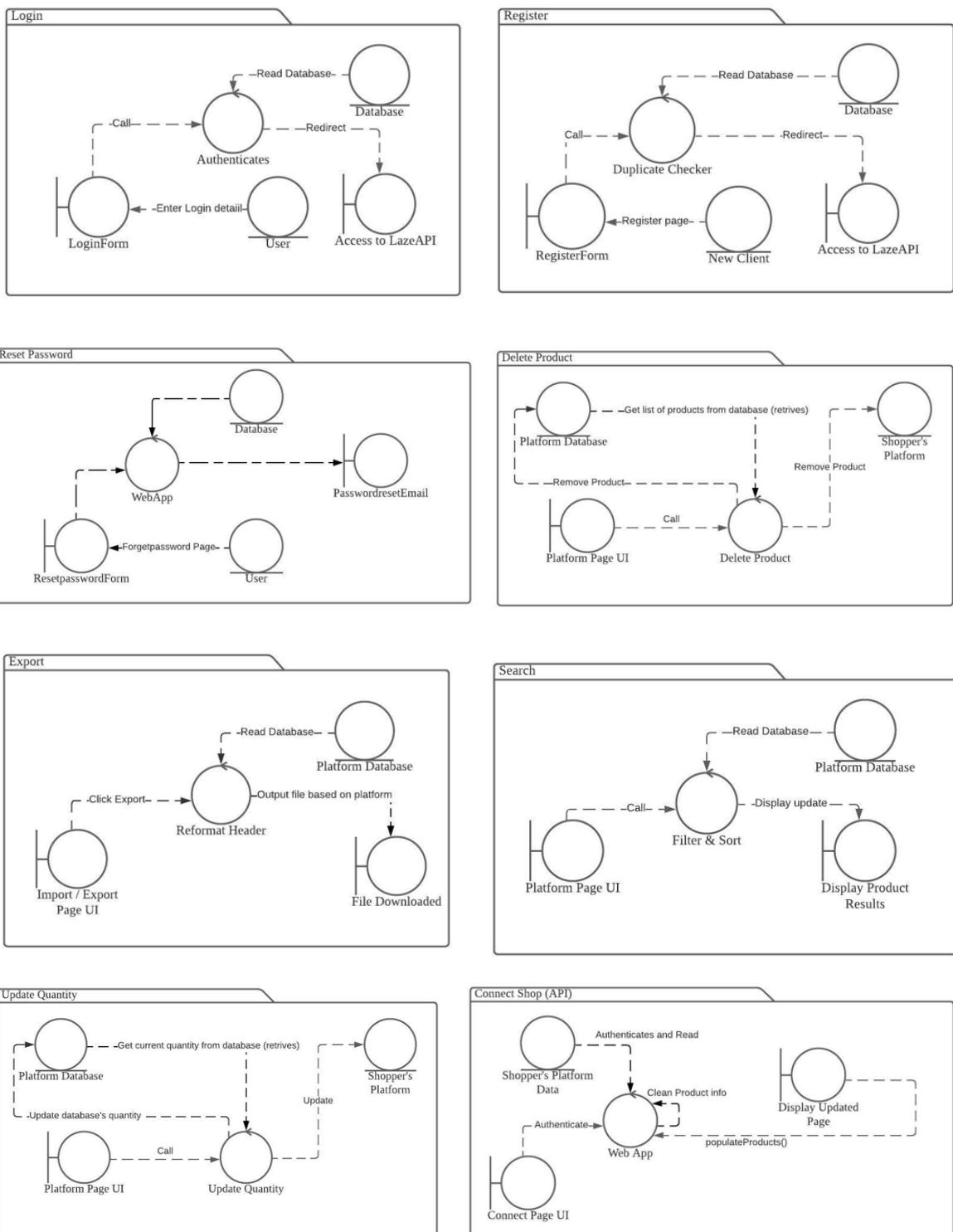
Use Case Diagram



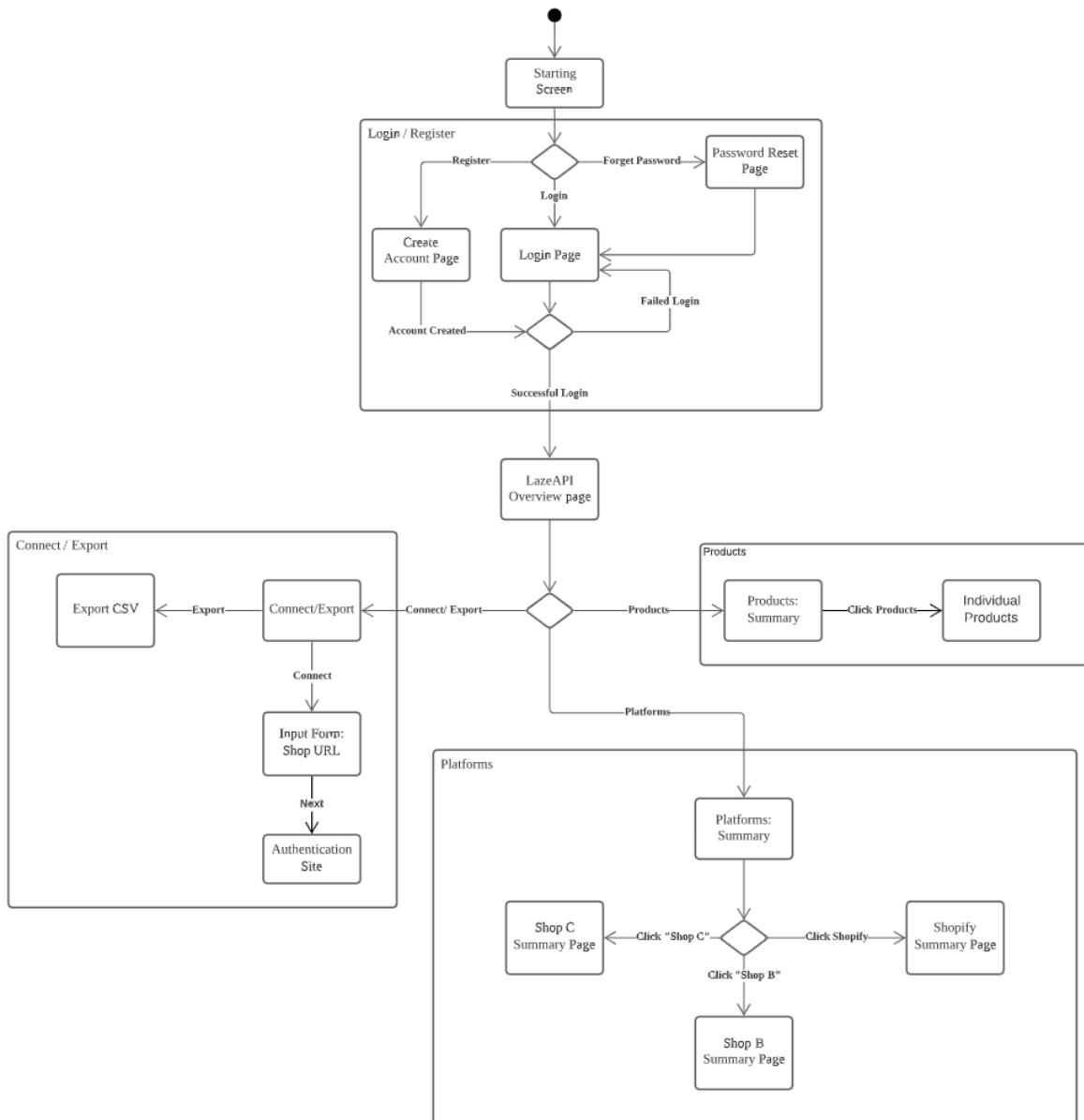
Conceptual Model



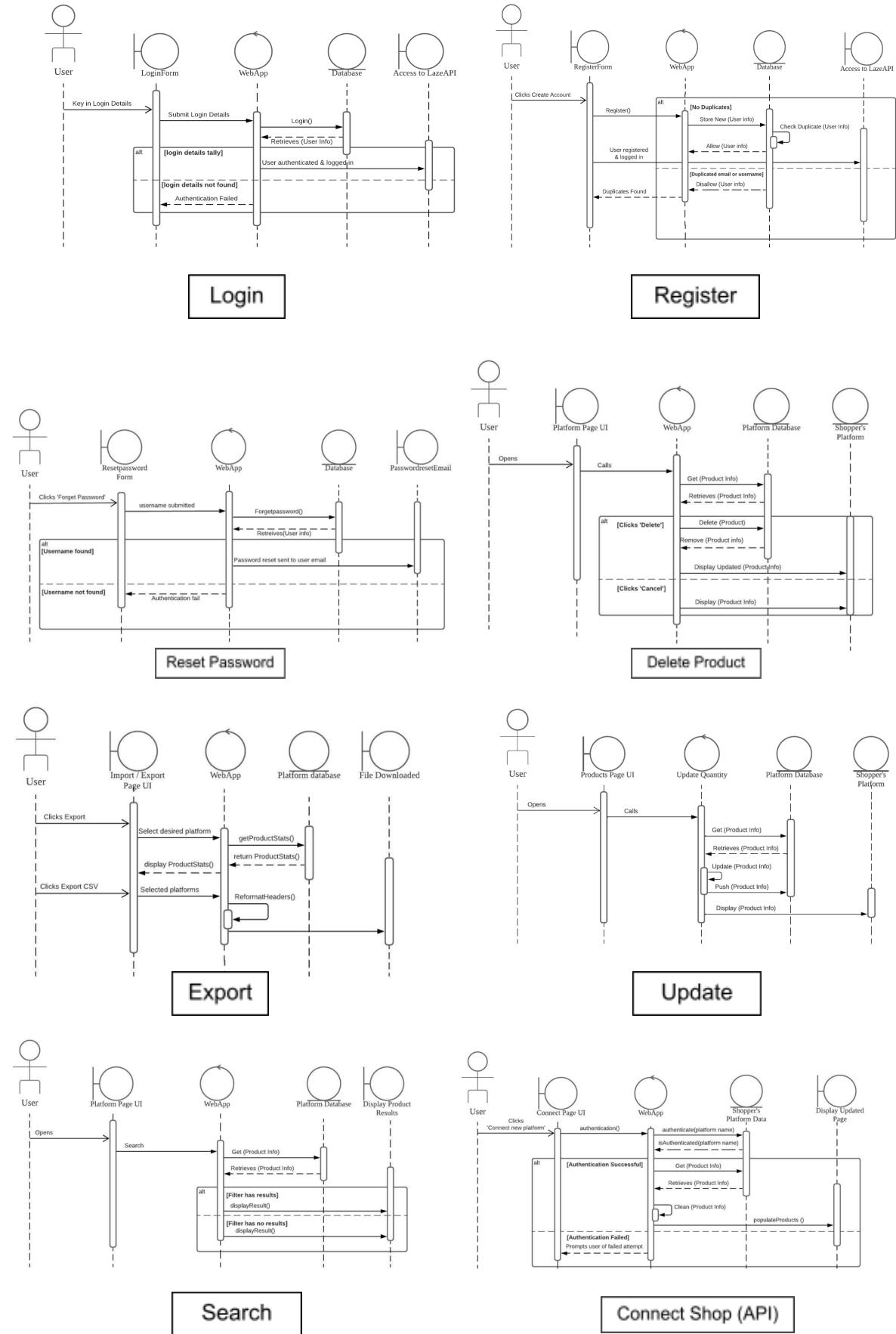
Conceptual Diagram



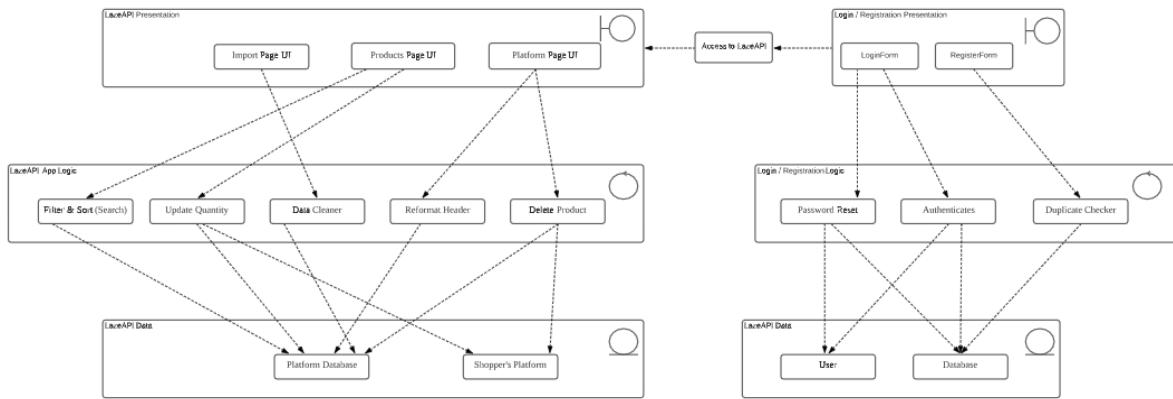
Dialog Map



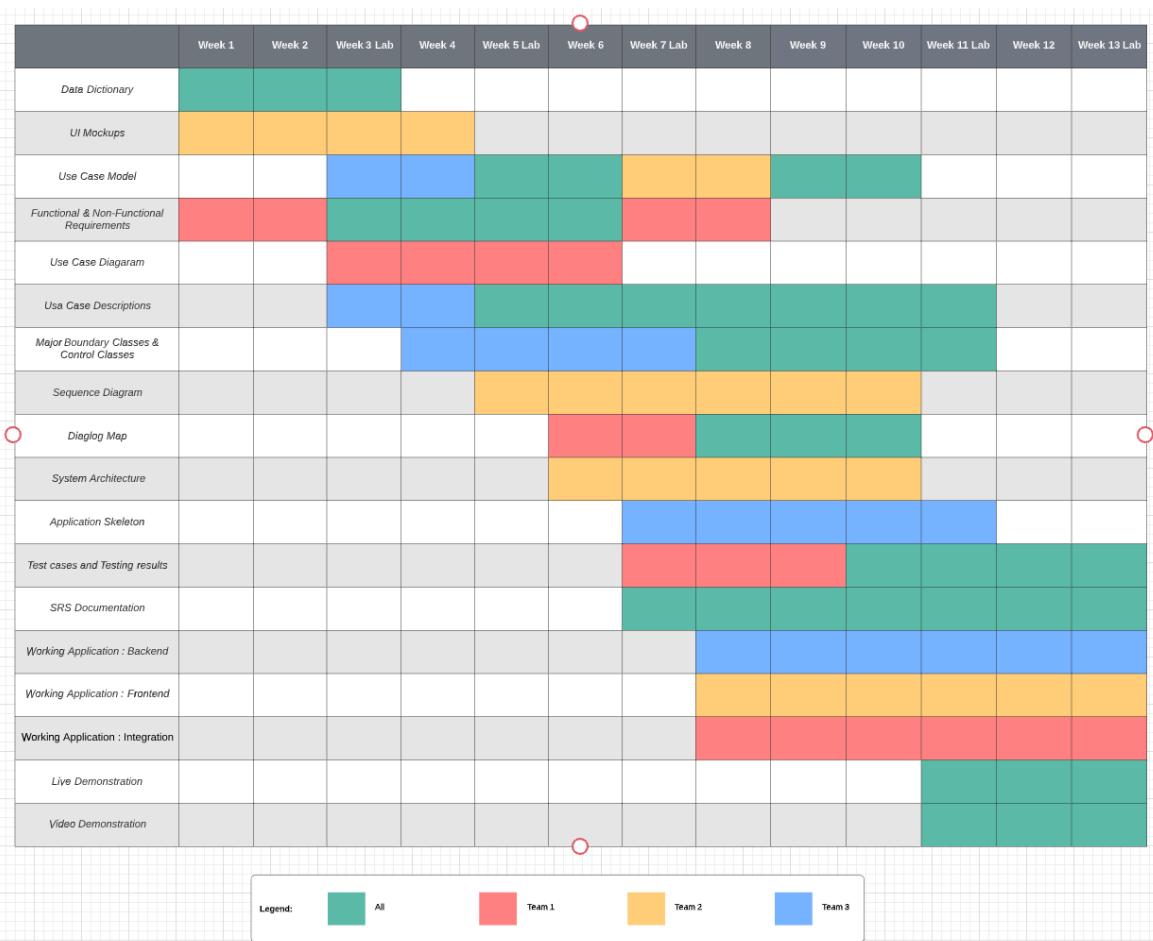
Sequence Diagram



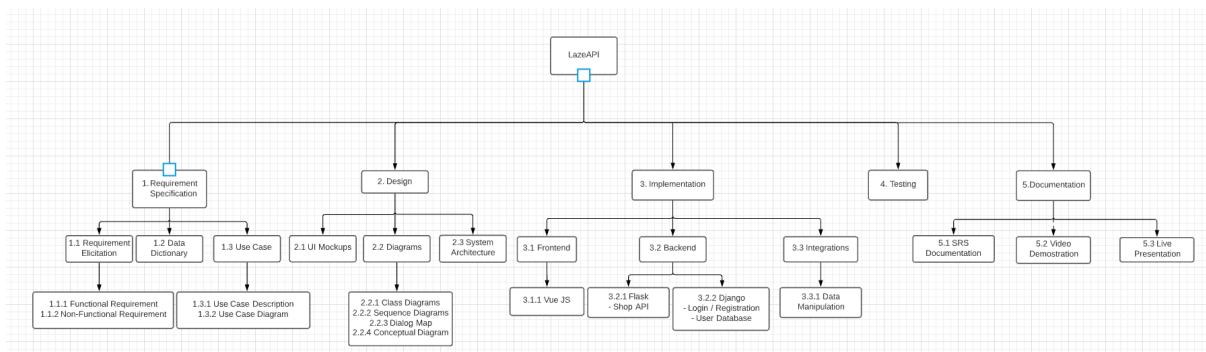
System Architecture



Gantt Chart for Timeline of Tasks



Workbench Chart (Tasks Breakdown)



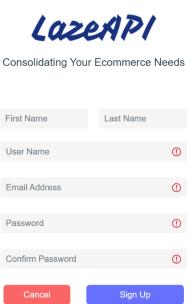
Appendix C: To Be Determined List

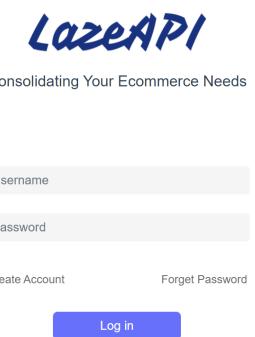
Scheduled for updates	Objectives	Description	To Be Determined
1	<u>Supportability</u>	Add support for all major ecommerce sites	Y

Appendix D: Test Cases

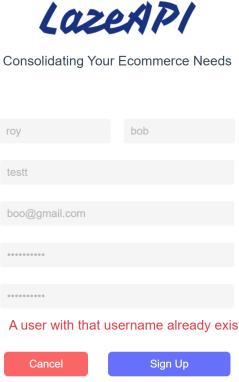
WhiteBox Testing

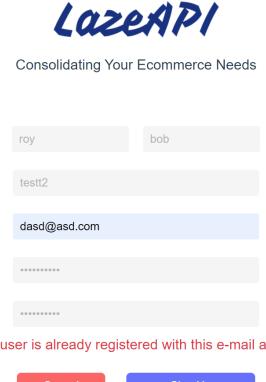
Test Case 1	
Test Case Type	User Interface
Description	'Create account' button on login page leads to registration page
Expected Results	Registration page opens

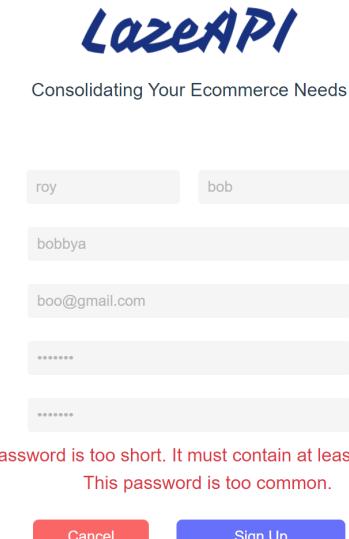
Process Diagram	
Pass or Fail	Pass

Test Case 2	
Test Case Type	User Interface
Description	'Cancel' button on registration page leads back to login page
Expected Results	Login page opens
Process Diagram	
Pass or Fail	Pass

Test Case 3	
Test Case Type	User Interface

Description	‘Sign Up’ button on registration page will prompt user with error msg when user tries to sign up with a registered username
Expected Results	System will display ‘A user with that username already exists’.
Process Diagram	 <p>The screenshot shows a registration form for LazeAPI. The form fields are: Username (roy), Password (bob), First Name (testt), Last Name (boo@gmail.com), and Email (dasd@asd.com). Below the form, an error message is displayed: "A user with that username already exists." At the bottom are two buttons: "Cancel" (red) and "Sign Up" (blue).</p>
Pass or Fail	Pass

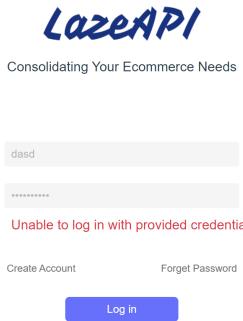
Test Case 4	
Test Case Type	User Interface
Description	‘Sign Up’ button on registration page will prompt user with error msg when user tries to sign up with a registered email
Expected Results	System will display ‘A user is already registered with this e-mail address’.
Process Diagram	 <p>The screenshot shows a registration form for LazeAPI. The form fields are: Username (roy), Password (bob), First Name (testt2), Last Name (dasd@asd.com), and Email (dasd@asd.com). The email field is highlighted in blue. Below the form, an error message is displayed: "A user is already registered with this e-mail address." At the bottom are two buttons: "Cancel" (red) and "Sign Up" (blue).</p>
Pass or Fail	Pass

Test Case 5	
Test Case Type	User Interface
Description	'Sign Up' button on registration page will prompt user with error msg when user tries to sign up with a simple password
Expected Results	System will display 'This password is too common'.
Process Diagram	 <p>The screenshot shows a registration form for 'LazeAPI' with the tagline 'Consolidating Your Ecommerce Needs'. The form fields are as follows:</p> <ul style="list-style-type: none"> First Name: roy Last Name: bob Email: bobbya Email: boo@gmail.com Password: (redacted) Confirm Password: (redacted) <p>Below the form, two error messages are displayed in red:</p> <ul style="list-style-type: none"> This password is too short. It must contain at least 8 characters. This password is too common. <p>At the bottom are two buttons: 'Cancel' (red) and 'Sign Up' (blue).</p>
Pass or Fail	Pass

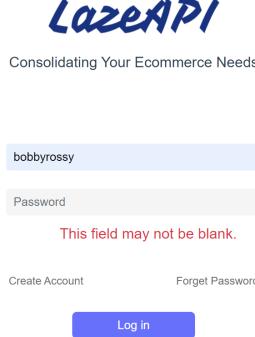
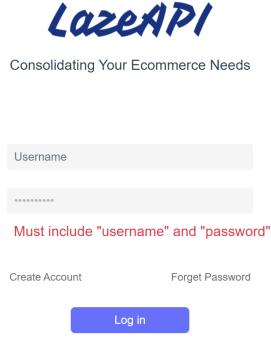
Test Case 6	
Test Case Type	User Interface
Description	'Sign Up' button on registration page will prompt user with error msg when user tries to sign up with a password shorter than 8 characters
Expected Results	System will display 'This password is too short. It must contain at least 8 characters'.

Process Diagram	<p>The screenshot shows a sign-up form for LazAPI. The user has entered 'roy' and 'bob' into the first two input fields. In the third field, they have entered 'bobbya'. In the fourth field, they have entered 'boo@gmail.com'. Below these fields are two error messages: 'This password is too short. It must contain at least 8 characters.' and 'This password is too common.'. At the bottom right are 'Cancel' and 'Sign Up' buttons.</p>
Pass or Fail	Pass

Test Case 7	
Test Case Type	User Interface
Description	'Log In' button on login page will open import/export page with the connect tab opened when user correctly inputs login credentials
Expected Results	User transits to import/export page as default first page, it will show the option to connect new shops and show the currently connected shops
Process Diagram	<p>The screenshot shows the 'Import / Export' section of the LazAPI interface. On the left, there's a sidebar with 'testt's Store' and navigation links for 'Import / Export', 'Products', and 'Platforms'. On the right, there's a 'Connect / Export' section with a 'Connect' button and an 'Export CSV' button. Below that is an 'Integrations' section with a 'Connect new platform' button. Several integration cards are shown, including one for 'shopify' which is highlighted with a green border and a URL 'https://lazeapi-testbed.myshopify.com'. Other cards include 'ebay', 'Shopee', 'Qoo10', '淘宝网', and 'Lazada'. A status indicator 'Connected' with a green dot is visible next to the Shopee card.</p>
Pass or Fail	Pass

Test Case 8	
Test Case Type	User Interface
Description	'Log In' button on prompt user with error msg when user credentials does not match the database
Expected Results	System will display 'Unable to log in with provided credentials'.
Process Diagram	 <p>The screenshot shows a login form for 'LazeAPI'. The page title is 'LazeAPI' with the subtitle 'Consolidating Your Ecommerce Needs'. Below the title is a login input field containing 'dasd' and a password input field containing '*****'. A red error message 'Unable to log in with provided credentials.' is displayed below the fields. At the bottom of the form are links for 'Create Account' and 'Forgot Password', and a blue 'Log in' button.</p>
Pass or Fail	Pass

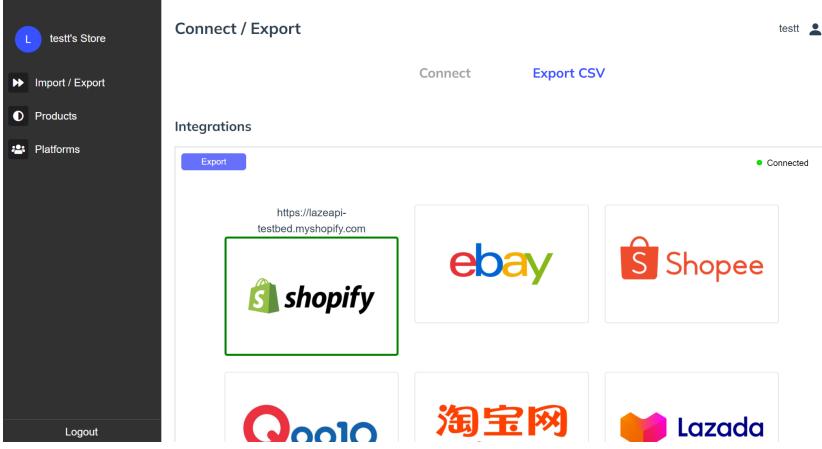
Test Case 9	
Test Case Type	User Interface
Description	'Log In' button on prompt user with error msg when either username or password is left blank
Expected Results	System will display 'This field may not be blank' when password is empty and display 'Must include "username" and "password" when username is empty'

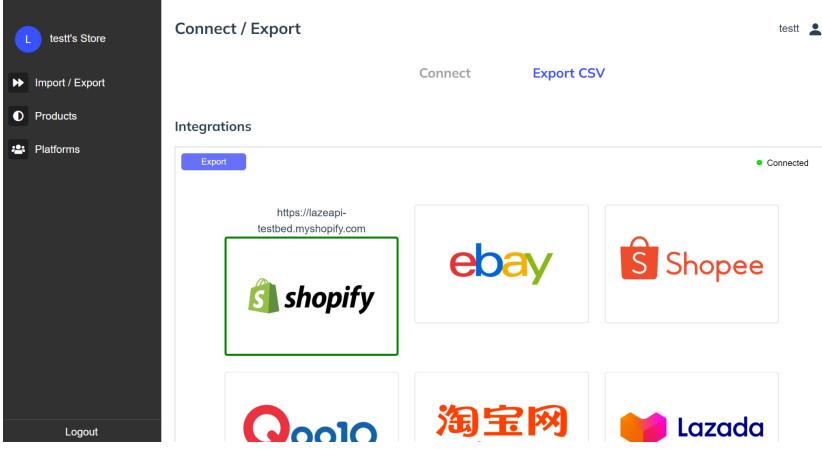
Process Diagram	 <p>The screenshot shows the LazeAPI login page. At the top, the LazeAPI logo and tagline "Consolidating Your Ecommerce Needs" are displayed. Below the logo is a light blue input field containing the text "bobbyrossy". Underneath it is a standard grey password input field. A red error message "This field may not be blank." is shown above the password field. At the bottom of the form are two links: "Create Account" and "Forgot Password", followed by a prominent blue "Log in" button.</p>  <p>This screenshot shows the same LazeAPI login page, but both the username and password fields are empty. A red error message "Must include 'username' and 'password'." is centered below the fields. The rest of the page, including the logo, links, and "Log in" button, remains identical to the first screenshot.</p>
Pass or Fail	Pass

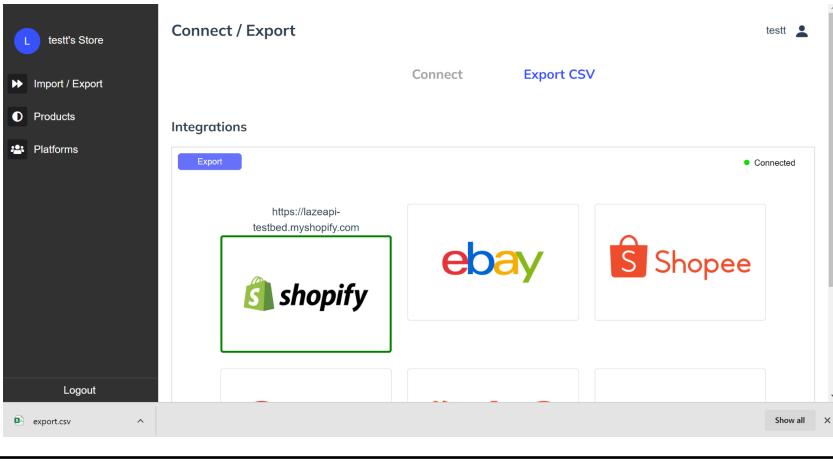
Test Case 10	
Test Case Type	User Interface
Description	'Logout' button on the bottom of the sidebar menu opens allows user to return to login page upon confirmation
Expected Results	User will go back to login page

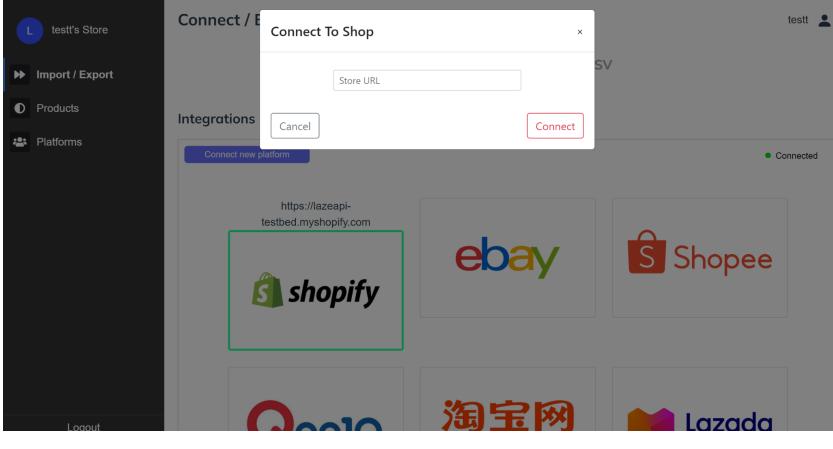
Process Diagram	
Pass or Fail	Pass

Test Case 11	
Test Case Type	User Interface
Description	User clicking on 'Export CSV' tab in the import/export page will open a page which shows the connected shops and option to export shop product data
Expected Results	'Export CSV' page opens

Process Diagram	
Pass or Fail	Pass

Test Case 12	
Test Case Type	File Download
Description	User clicking on 'Export' button after selecting one of the connected shop icon in the 'Export CSV' page will have a file downloaded to the computer
Expected Results	File will be downloaded
Process Diagram	

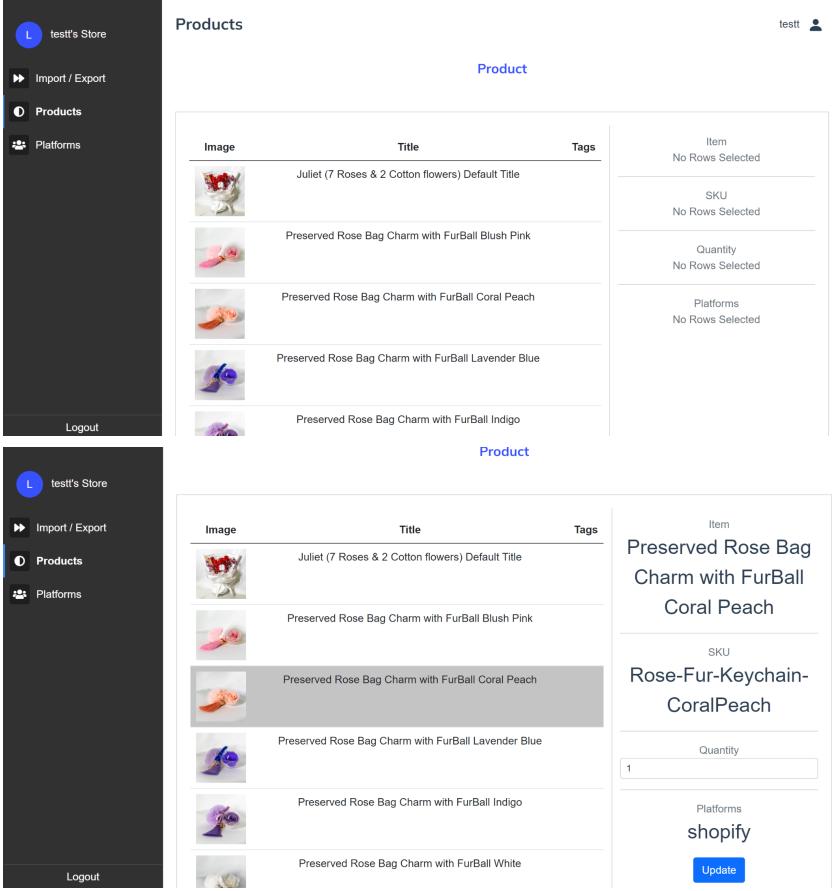
	
Pass or Fail	Pass

Test Case 13	
Test Case Type	API
Description	User clicking on the ‘Connect new platform’ button after selecting one of the shop icons will open a panel of options for connecting to the chosen shop. Once successfully connected, the shop icon will have a green outline
Expected Results	Eg. Shopify is connected, a green outline will form around the Shopify icon
Process Diagram	

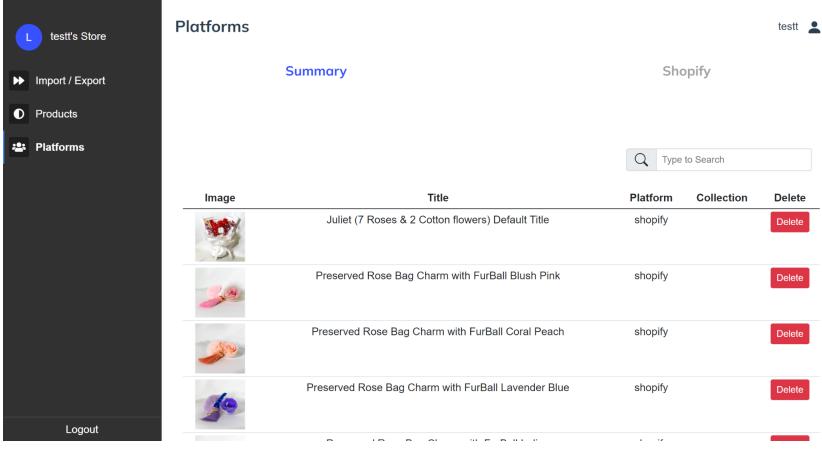
	<p>Connect / Export</p> <p>testt </p> <p>Integrations</p> <p>Connect new platform </p> <ul style="list-style-type: none"> https://lazeapi-testbed.myshopify.com
Pass or Fail	Pass

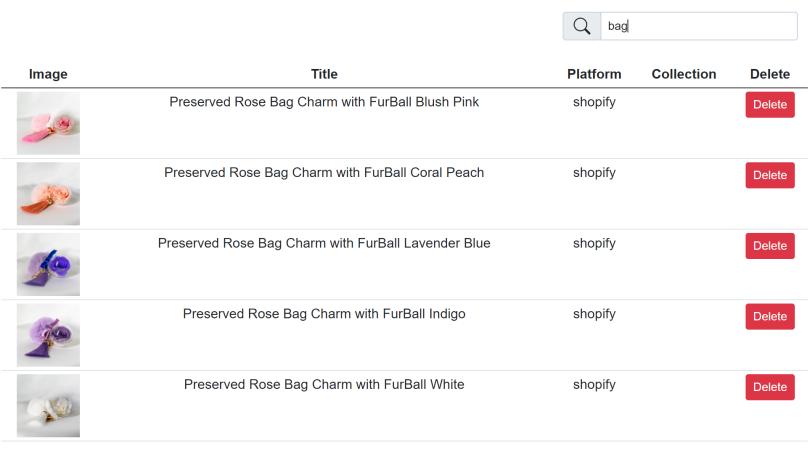
Test Case 14																			
Test Case Type	User Interface																		
Description	User clicking on 'Products' option on the sidebar menu will lead the user to the products page																		
Expected Results	System will lead user to a page with a list of all products from all existing connected platforms, the scroll is set to unlimited and user will be able to see all products in one page																		
Process Diagram	<p>Products </p> <table border="1"> <thead> <tr> <th>Image</th> <th>Title</th> <th>Tags</th> </tr> </thead> <tbody> <tr> <td></td> <td>Juliet (7 Roses & 2 Cotton flowers) Default Title</td> <td></td> </tr> <tr> <td></td> <td>Preserved Rose Bag Charm with FurBall Blush Pink</td> <td></td> </tr> <tr> <td></td> <td>Preserved Rose Bag Charm with FurBall Coral Peach</td> <td></td> </tr> <tr> <td></td> <td>Preserved Rose Bag Charm with FurBall Lavender Blue</td> <td></td> </tr> <tr> <td></td> <td>Preserved Rose Bag Charm with FurBall Indigo</td> <td></td> </tr> </tbody> </table> <p>Item No Rows Selected SKU No Rows Selected Quantity No Rows Selected Platforms No Rows Selected</p>	Image	Title	Tags		Juliet (7 Roses & 2 Cotton flowers) Default Title			Preserved Rose Bag Charm with FurBall Blush Pink			Preserved Rose Bag Charm with FurBall Coral Peach			Preserved Rose Bag Charm with FurBall Lavender Blue			Preserved Rose Bag Charm with FurBall Indigo	
Image	Title	Tags																	
	Juliet (7 Roses & 2 Cotton flowers) Default Title																		
	Preserved Rose Bag Charm with FurBall Blush Pink																		
	Preserved Rose Bag Charm with FurBall Coral Peach																		
	Preserved Rose Bag Charm with FurBall Lavender Blue																		
	Preserved Rose Bag Charm with FurBall Indigo																		
Pass or Fail	Pass																		

Test Case 15

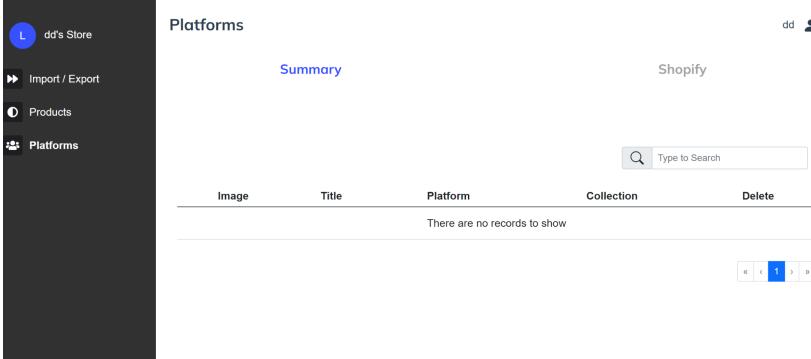
Test Case Type	User Interface
Description	User clicking on any product in the ‘Products’ page will enable the system to show the details of the selected product on the right side panel.
Expected Results	The details of the selected product such as the item name, the SKU number, the quantity in the connected shop platforms and the platforms that holds the product for sale will appear on the right panel
Process Diagram	 <p>The screenshot displays a user interface for managing products. The left sidebar includes options for 'Import / Export', 'Products' (which is currently selected), and 'Platforms'. The main content area shows a table of products with columns for 'Image', 'Title', and 'Tags'. One specific product, 'Preserved Rose Bag Charm with FurBall Coral Peach', is highlighted. To the right, a detailed view provides more information about this product, including its item name, SKU, quantity, and the platform it is connected to (Shopify). An 'Update' button is also visible.</p>
Pass or Fail	Pass

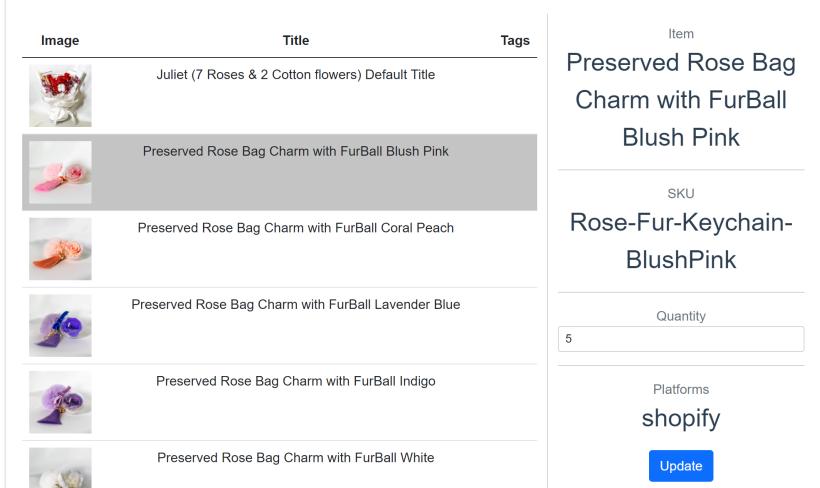
Test Case 16	
Test Case Type	User Interface
Description	User clicking on ‘Platforms’ option on the sidebar menu will lead the user to the platforms page with the ‘Summary’ tab within the page open as default
Expected Results	Summary tab in the platforms page will be opened with an empty search bar and a table showing the records of products connected to

	the web application divided up into different pages.
Process Diagram	
Pass or Fail	Pass

Test Case 17	
Test Case Type	Data Manipulation
Description	User entering a search input into the search bar
Expected Results	System will filter the existing data connected to the platform and return the results of the search. The search results will be divided into different pages if it exceeds the limit of each page. If there is no results, the system will return 'There are no records to show'
Process Diagram	

	<p>Platforms</p> <p>Summary</p> <p>Shopify</p> <p>testt </p> <p><input type="text"/> bagd</p> <table border="1"> <thead> <tr> <th>Image</th><th>Title</th><th>Platform</th><th>Collection</th><th>Delete</th></tr> </thead> <tbody> <tr> <td colspan="5">There are no records matching your request</td></tr> </tbody> </table> <p> 1 2 3 4 ... »</p>	Image	Title	Platform	Collection	Delete	There are no records matching your request				
Image	Title	Platform	Collection	Delete							
There are no records matching your request											
Pass or Fail	Pass										

Test Case 18											
Test Case Type	User Interface										
Description	User arrives at the ‘Platforms’ page when there are no platforms connected yet										
Expected Results	The ‘Summary’ tab in the ‘Platforms’ page will show a table with 0 records, displaying ‘There are no records to show’										
Process Diagram	 <p>Platforms</p> <p>Summary</p> <p>Shopify</p> <p>dd </p> <p><input type="text"/> Type to Search</p> <table border="1"> <thead> <tr> <th>Image</th> <th>Title</th> <th>Platform</th> <th>Collection</th> <th>Delete</th> </tr> </thead> <tbody> <tr> <td colspan="5">There are no records to show</td></tr> </tbody> </table> <p> 1 2 3 4 ... »</p>	Image	Title	Platform	Collection	Delete	There are no records to show				
Image	Title	Platform	Collection	Delete							
There are no records to show											
Pass or Fail	Pass										

Test Case 19																							
Test Case Type	API																						
Description	In the ‘Products’ page, user change the quantity of a selected product and clicks on ‘Update’ button																						
Expected Results	On the web application, the quantity will be changed to the new quantity. The system will also update the platform database through API calls and the platform quantity will be updated to the new changes as well																						
Process Diagram	 <p>The screenshot shows a list of products on the left and a sidebar for updating a selected item on the right.</p> <table border="1"> <thead> <tr> <th>Image</th> <th>Title</th> <th>Tags</th> </tr> </thead> <tbody> <tr> <td></td> <td>Juliet (7 Roses & 2 Cotton flowers) Default Title</td> <td></td> </tr> <tr> <td></td> <td>Preserved Rose Bag Charm with FurBall Blush Pink</td> <td></td> </tr> <tr> <td></td> <td>Preserved Rose Bag Charm with FurBall Coral Peach</td> <td></td> </tr> <tr> <td></td> <td>Preserved Rose Bag Charm with FurBall Lavender Blue</td> <td></td> </tr> <tr> <td></td> <td>Preserved Rose Bag Charm with FurBall Indigo</td> <td></td> </tr> <tr> <td></td> <td>Preserved Rose Bag Charm with FurBall White</td> <td></td> </tr> </tbody> </table> <p>Item: Preserved Rose Bag Charm with FurBall Blush Pink</p> <p>SKU: Rose-Fur-Keychain- BlushPink</p> <p>Quantity: <input type="text" value="5"/></p> <p>Platforms: shopify</p> <p>Update</p>		Image	Title	Tags		Juliet (7 Roses & 2 Cotton flowers) Default Title			Preserved Rose Bag Charm with FurBall Blush Pink			Preserved Rose Bag Charm with FurBall Coral Peach			Preserved Rose Bag Charm with FurBall Lavender Blue			Preserved Rose Bag Charm with FurBall Indigo			Preserved Rose Bag Charm with FurBall White	
Image	Title	Tags																					
	Juliet (7 Roses & 2 Cotton flowers) Default Title																						
	Preserved Rose Bag Charm with FurBall Blush Pink																						
	Preserved Rose Bag Charm with FurBall Coral Peach																						
	Preserved Rose Bag Charm with FurBall Lavender Blue																						
	Preserved Rose Bag Charm with FurBall Indigo																						
	Preserved Rose Bag Charm with FurBall White																						
Pass or Fail	Pass																						

Test Case 20		
Test Case Type	API	
Description	In the ‘Platforms’ page, the user clicks on the ‘Delete’ button of a product. User selects the platform to delete the product listing from.	
Expected Results	On the web application, the product listing will be removed from its local database. It will no longer appear in the ‘Platforms’ ‘Summary’ and the ‘Products’ page. On the ecommerce platform, the product listing will be removed from the store as well.	

Process Diagram	<p>The screenshot shows a user interface with a dark sidebar on the left containing links: 'testt's Store', 'Import / Export', 'Products', and 'Platforms'. The main area has a heading 'Delete: Juliet (7 Roses & 2 Cotton flowers) Default Title'. It asks 'Do you want to delete this product?' with a checked 'Shopify' checkbox. Below is a table with columns 'Platform', 'Collection', and 'Delete'. It lists five products under the 'shopify' platform. The first product is selected. At the bottom are 'Cancel' and 'Delete' buttons.</p>
Pass or Fail	Pass

Test Case 21	
Test Case Type	User Interface
Description	In the 'Platforms' page, the user clicks on one of the available ecommerce platform shop tabs instead of the 'Summary' tab.
Expected Results	System will display products that are only listed in the specific shops.
Process Diagram	<p>The screenshot shows a user interface with a dark sidebar on the left containing links: 'testt's Store', 'Import / Export', 'Products', and 'Platforms'. The main area has a heading 'Platforms' and a 'Summary' link. A 'Shopify' tab is highlighted in blue. On the right, there is a table with columns 'Image', 'Title', and 'Tags'. The table lists five products. To the right of the table, there are four sections: 'Item' (No Rows Selected), 'SKU' (No Rows Selected), 'Quantity' (No Rows Selected), and 'Platforms' (No Rows Selected).</p>
Pass or Fail	Pass

Test Case 22	
---------------------	--

Test Case Type	Password Encryption
Description	When a new user registers for an account, the password is encrypted and not available for viewing the database of Django
Expected Results	Password information not stored visibly within the Django database
Process Diagram	
Pass or Fail	Pass

Test Case 23	
Test Case Type	Data Limit
Description	In the 'Products' page, user change the quantity of a selected product to 0 and clicks on 'Update' button
Expected Results	On the web application, the quantity will be changed to 0 without removing the product. The system will also update the platform database through API calls and the platform quantity will be updated to the new changes as well
Process Diagram	

Pass or Fail	Pass

Test Case 24	
Test Case Type	Data Limit
Description	In the ‘Products’ page, user change the quantity of a selected product to 1,000,000 and clicks on ‘Update’ button
Expected Results	On the web application, the quantity will be changed to 1,000,000 without removing the product. The system will also update the platform database through API calls and the platform quantity will be updated to the new changes as well
Process Diagram	

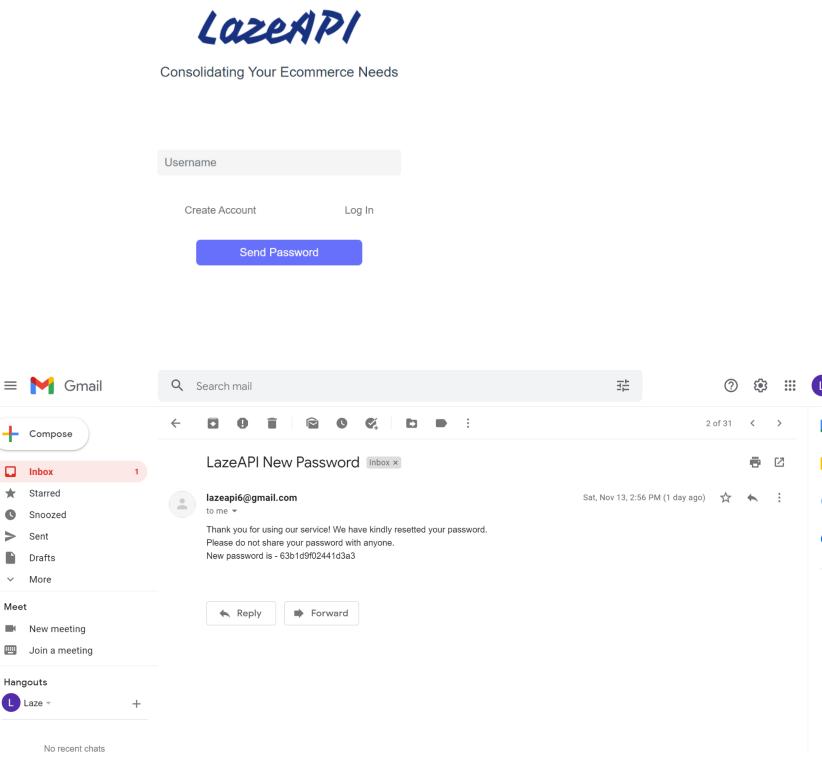
The screenshot shows a product management interface with the following details:

- Update Success**: A modal message at the top right.
- Product Details (Left Panel):**
 - Image**: Shows five product variants: Juliet (7 Roses & 2 Cotton flowers) Default Title, Preserved Rose Bag Charm with FurBall Blush Pink, Preserved Rose Bag Charm with FurBall Coral Peach, Preserved Rose Bag Charm with FurBall Lavender Blue, and Preserved Rose Bag Charm with FurBall Indigo.
 - Title**: Juliet (7 Roses & 2 Cotton flowers) Default Title
 - Tags**: Not explicitly listed but implied by the variant descriptions.
- Right Panel (Product View):**
 - Item**: Juliet (7 Roses Cotton flower Default Title)
 - SKU**: JULIET
 - Quantity**: 1000000
 - Platforms**: shopify
 - Update**: A button to update the product information.
- Bottom Left (Sidebar):**
 - LAZEAPI_testbed
 - Home
 - Orders
 - Products** (selected)
 - All products
 - Inventory
 - Transfers
 - Collections
 - Gift cards
 - Customers
 - Analytics
 - Marketing
 - Discounts
 - Apps
 - SALES CHANNELS
 - Online Store
- Bottom Center (Main Content):**
 - Products**
 - Filter products: All, Active, Draft, Archived
 - Search bar: Search
 - Table headers: Product, Status, Inventory, Type, Vendor
 - Data rows:
 - Juliet (7 Roses & 2 Cott on flowers) - Active, 1,000,000 in stock, Preserved Roses, Daisy Dreams
 - Preserved Rose Bag Cha rm with FurBall - Active, 10 in stock for 5 variants, Preserved Roses, Daisy Dreams
 - Preserved Rose Bag Cha rm with Pearls - Active, 38 in stock for 9 variants, Preserved Roses, Daisy Dreams
 - Preserved Rose Bag Cha rm - (partially visible)

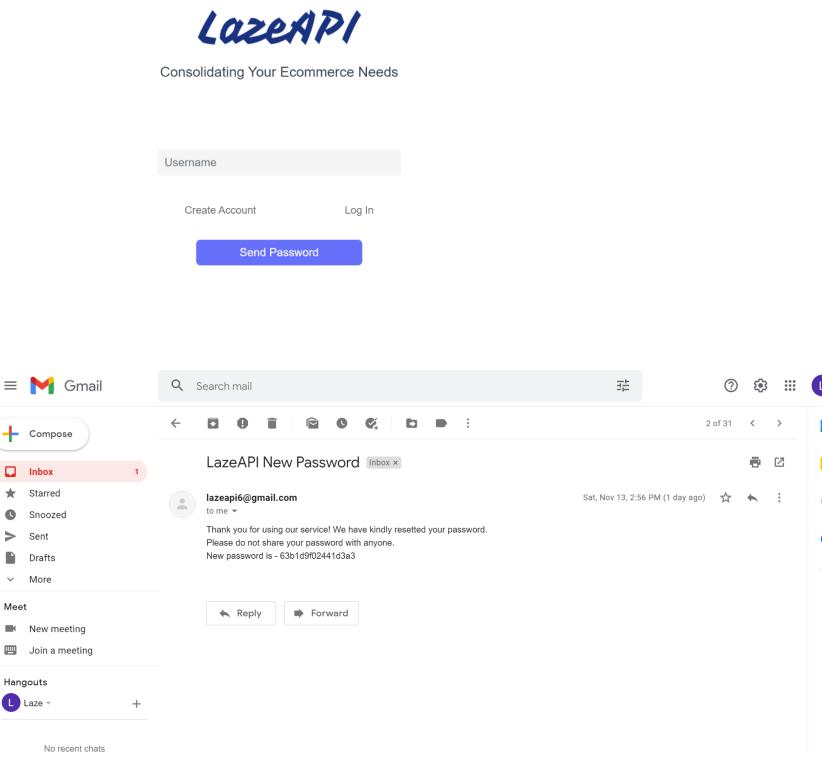
Pass or Fail

Pass

Test Case 25	
Test Case Type	User Interface
Description	User can select the 'Forgot Password' option in the 'Log In' page and will be redirected to forget password page
Expected Results	User will be led to forget password page

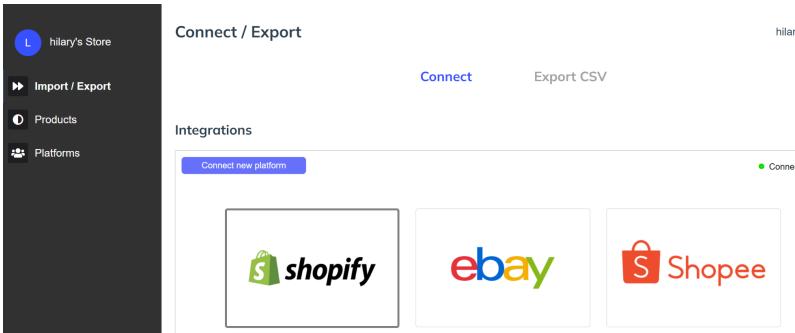
Process Diagram	 <p>The screenshot shows two parts of a user interface. The top part is the LazeAPI login page with fields for 'Username' and 'Password', and buttons for 'Create Account' and 'Log In'. Below it is a Gmail inbox showing an email from 'lazeapi6@gmail.com' with the subject 'LazeAPI New Password'. The email body contains a message: 'Thank you for using our service! We have kindly resetted your password. Please do not share your password with anyone. New password is - 63b1d9f02441d3a3'. There are 'Reply' and 'Forward' buttons below the email.</p>
Pass or Fail	Pass

Test Case 26	
Test Case Type	User Interface
Description	User enters a registered username and clicks on the 'Send Password' button.
Expected Results	System will find the email linked to the registered account and send an auto generated password for user to login with the username

Process Diagram	 <p>The screenshot shows two parts of a user interface. The top part is the LazeAPI login page with fields for 'Username' and 'Password', and buttons for 'Create Account', 'Log In', and 'Send Password'. The bottom part is a Gmail inbox showing an email from 'lazeapi6@gmail.com' titled 'LazeAPI New Password'. The email body contains a message: 'Thank you for using our service! We have kindly resetted your password. Please do not share your password with anyone. New password is - 63b1d9f02441d3a3'. Below the email are standard Gmail controls for reply, forward, and settings.</p>
Pass or Fail	Pass

BlackBox Testing

Test Case 1	
Test Case Type	User Interface
Description	User is asked to create a new account and login with newly registered account
Expected Results	User log into import/export page

Process Diagram	
Pass or Fail	Pass

Test Case 2	
Test Case Type	User Interface
Description	User is asked to navigate through all the pages in the application without direction
Expected Results	User are able to navigate through all the pages with ease
Process Diagram	-
Pass or Fail	Pass

Test Case 3	
Test Case Type	User Interface
Description	User is asked to connect the shopify shop with given shopify url and account
Expected Results	User able to connect successfully
Process Diagram	-
Pass or Fail	Pass

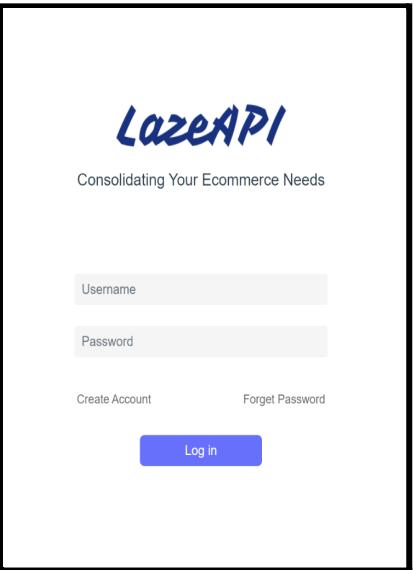
Test Case 4	
Test Case Type	Update Quantity
Description	User is asked to update the quantity of any product they wish to change
Expected Results	User able to update quantity on both web application and platform successfully
Process Diagram	-
Pass or Fail	Pass

Test Case 5	
Test Case Type	Search
Description	User is asked to search for any product
Expected Results	User able to find the search bar and use it easily to find the desired product
Process Diagram	-
Pass or Fail	Pass

Test Case 5	
Test Case Type	Search
Description	User is asked to search for any product
Expected Results	User able to find the search bar and use it easily to find the desired product
Process Diagram	-
Pass or Fail	Pass

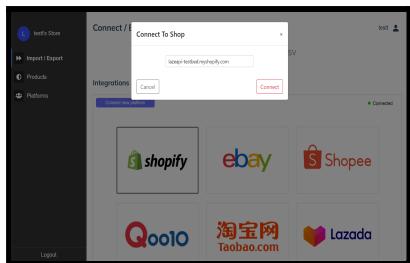
Test Case 6	
Test Case Type	Search
Description	User is asked to delete a product from the shopify platform through the web application
Expected Results	User able to find the delete button for the selected product and remove it from listing
Process Diagram	-
Pass or Fail	Pass

Appendix E: User Manual

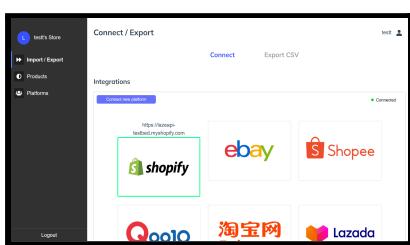
User Manual	
User view	User's Action
<p>1)</p> 	<p>Users open the 'Login' page on the web page application.</p> <p>Users can select the 'Login' button or 'Create Account' button.</p> <p>Click the 'Login' button and enter Username and Password if the user has an account. User is then redirected to the 'LazeAPI overview page'.</p> <p>Click the 'Create Account' button on the 'Login' page to be redirected to the 'Register' page if the user does not have an account and wishes to create an account.</p> <p>Click the 'Forgot Password' button on the 'Login' page to be redirected to the 'Forgot Password' page if the user has an account but forgot the password and wishes to reset it.</p>

2)	<p>User selects the ‘Create Account’ button and the ‘Register’ page will be displayed.</p> <p>User to enter the first name into the ‘First Name’ field, last name into the ‘Last Name’ field and the desired user name into the ‘User Name’ field</p> <p>User to enter a valid email address to link with the account to be created into the ‘Email Address’ field</p> <p>User to enter a desired password into the ‘Password’ and ‘Confirm Password’ field. Passwords that are too common will be rejected.</p> <p>User is then logged in and redirected to ‘LazeAPI Platform’ page</p>
----	--

User view	User's Action
3)	<p>Users can select ‘Connect / Export’, ‘Products’ or ‘Platforms’ on the side toolbar.</p> <p>User selects ‘Connect / Export’ on the side toolbar to connect their ecommerce platforms and export platform’s product data as a CSV file. The connect page will be shown,</p> <p>User can click on ‘Connect new platform’ button, a pop out will appear which would require the user to key in the shop URL they wish to connect to.</p>



Once successfully connected, the shopping platform will display a green border around it.

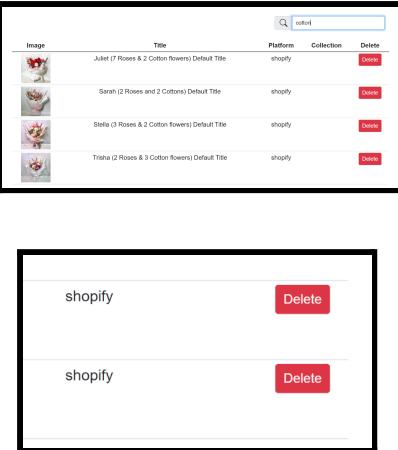
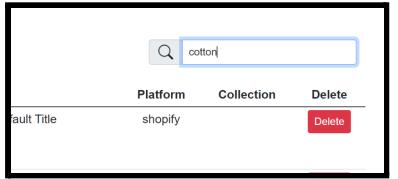
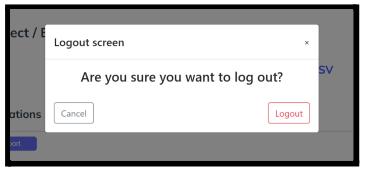


Users select '**Products**' on the **side toolbar** to view and edit the product details.

Users select '**Platform**' on the **side toolbar** to view products grouped by individual ecommerce platforms and edit product details directly on ecommerce platforms.

User Manual

User view	User's Action
<p>4)</p>	<p>User selects 'Products' on the side toolbar and the 'Products' page will be shown.</p> <p>User can view all products by name, sold on connected ecommerce platforms.</p> <p>User to input product details for the selected product to change the quantity and prices of the products.</p> <p>LazeAPI's system pushes data changes to platform database</p>

<p>5)</p> 	<p>User selects 'Platforms' on the side toolbar and the 'Platform' page will be shown.</p> <p>User can view all products by name, grouped based on the individual connected ecommerce platform.</p> <p>User to click on the 'Delete' button to remove selected products.</p> <p>User to input product details for the selected product to change the quantity and prices of the products.</p> <p>LazeAPI's system pushes data changes to the platform database.</p>
<p>6)</p> 	<p>User looking for a specific product, can enter input into the Search box.</p> <p>System will filter the database based on the user input and display the search result on the same page.</p>
<p>7)</p> 	<p>Users click on 'Logout' on the side toolbar to end their session, which displays a pop up for confirmation. User then selects the 'Logout' button to confirm logout.</p> <p>Users will be redirected to the 'Login' page once they have been signed out of their account.</p>