



DALHOUSIE
UNIVERSITY

WISESHOPPING

GROUP 8

Team Members:

Nilesh Shivshankar Gupta (B00876771)
Adarsh Kannan Iyengar (B00900913)
Harsh Hariramani (B00899363)
Pavithra Gunasekaran (B00905274)
Elizabeth James (B00867159)

Table of Contents

1.	Application Overview.....	4
2.	Tech Stack	4
3.	Run and Deploy steps	4
4.	Dependencies:.....	5
5.	Database Schema	6
5.1.	ER Diagram	6
5.2.	Table description.....	6
6.	Architecture	7
7.	User scenarios.....	8
7.1.	User Functionality.....	8
7.1.1.	User Registration and Login	8
7.1.2.	Password Recovery	9
7.1.3.	View All Products and Stores	10
7.1.4.	View products by applying filter over seller	11
7.1.5.	View products by applying filter over zip code	12
7.1.6.	View products by applying filter over seller and zip code	13
7.1.7.	Search products by tags.....	14
7.1.8.	Compare product prices	15
7.1.9.	Add products to cart.....	16
7.1.10.	View cart.....	17
7.1.11.	Share cart	18
7.1.12.	Checkout cart to view the cheapest deal	19
7.1.13.	View Price Analytics	19
7.1.14.	Add subscription for price alerts	20
7.1.15.	View subscriptions.....	22
7.1.16.	Delete subscriptions	22
3.2.17	Update subscriptions	23
7.2.	Merchant Functionality	24
7.2.1.	Merchant registration and login	24
7.2.2.	Password Recovery	26
7.2.3.	Add store.....	26
7.2.4.	View store	29

7.2.5.	Update store	29
7.2.6.	Delete store	31
7.2.7.	Add products.....	32
7.2.8.	Update products.....	33
7.2.8.1.	Update price.....	33
7.2.8.2.	Update stock.....	35
7.2.8.3.	Update category name.....	37
7.2.8.4.	Update category description.....	38
7.2.9.	Update tags	40
7.2.10.	Analytics	41
7.3.	Common features	43
8.	Analytics	44
9.	Design Pattern.....	44
10.	Smell Analysis Summary.....	44
11.	CI CD status.....	47
12.	Limitations and future scope.....	48
13.	Member Contribution.....	48
13.1.	Features.....	48
13.2.	Other tasks	50

1. Application Overview

Wiseshopping is an application where consumers can buy a product by comparing its price among the different sellers and get the best deal. It also allows users to subscribe for price alert for price drop in products. It also helps user to make smart buying decisions by viewing the price of the desired product for the past one year.

The merchants need to update their product inventory to make the products available to the customers. The merchant can view the demand trends of a product for the previous year and plan the inventory accordingly or change the prices of a product.

2. Tech Stack

- Java
- SpringBoot
- MySQL

3. Run and Deploy steps

The application is a maven project using Spring boot framework and MySQL.

Minimum requirements: maven, IDE (IntelliJ, Eclipse), MySQL client, java version 11+

Steps to run the application:

In timberlea:

- Go to /users/webhome/elizabeth/csci5308/prod_builds/target. (All necessary folders are already created in this directory)
- Run command:
java -jar wiseshopping-0.0.1-SNAPSHOT --spring.profiles.active=prod
One can pass “prod”, “test”, “dev” to --spring.profiles.active parameter

This folder has world read, write, execute permissions. However, if you face access issue, please reach out to any of the team members.

In local machine or other environment:

- If you are using a database other than the CSCI5308_8_PRODUCTION, CSCI_5308_8_TEST, CSCI_5308_8_DEV_INT, create the DB schema using the .sql file in the repository. You will also have to add the database credentials in application-prod.properties and set “prod” --spring.profiles.active parameter
- Clone the project from <https://git.cs.dal.ca/courses/2022-winter/csci-5308/group8> in your workspace
- Run the build command **mvn clean install**
- After a successful build, create two directories in the target directory – **productDemandCharts** and **productPriceAnalyticsCharts**.

If you are moving the jar to a different directory, make sure to create these two directories in the folder containing the jar. The analytics graphs will be created in these two folders.

- v. Once the jar has been created in the target directory, run **java -jar wiseshopping-0.0.1-SNAPSHOT --spring.profiles.active=prod**
One can pass “prod”, “test”, “dev” to --spring.profiles.active parameter.
- vi. If you are running this project in IDE, edit the run configuration and set environmental variables as **--spring.profiles.active=prod**
- vii. You can also enable the Lombok plugin in your IDE. This is optional.

Deploy steps:

- i. The .gitlab-ci.yml is configure to deploy the application to the timberlea server
- ii. You can change the host and directory variable if you want to change the deploy host
- iii. Run the deploy pipeline in gitlab
- iv. Log in to timberlea and go to /users/webhome/elizabeth/csci5308/prod_builds/target
- v. Unzip the created jar and run **java -jar wiseshopping-0.0.1-SNAPSHOT --spring.profiles.active=prod**

4. Dependencies:

Table 1 shows the list of dependencies used in the project and its purpose

Table 1: POM dependencies

Artifact	Purpose
spring-boot-devtools	Improves spring development experience
spring-boot-starter-data-jpa	JPA integration
mysql-connector-java	MySQL driver
spring-boot-starter-test	Spring Testing
commons-codec	Password Hashing
commons-validator	Email Validation
slf4j-api	Logging
lombok	Clean code
spring-boot-starter-mail	Mail notification to send alerts to customers
guava	To use multimap datastructure
jfreechart	To draw graphs for analytical module

5. Database Schema

5.1. ER Diagram

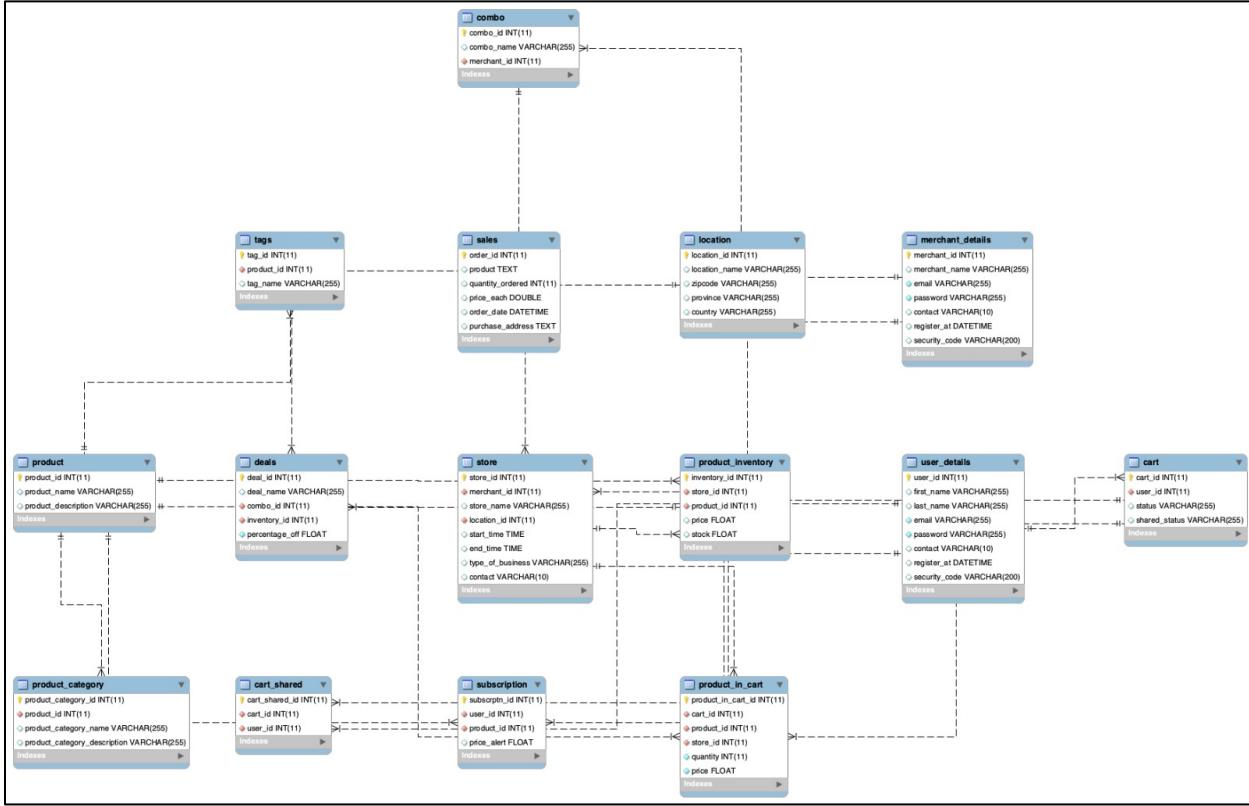


Figure 2: ER diagram

5.2. Table description

The database of the project contains the following tables along with their columns:

- **User details** – This table contains information like name, user id, email and password of the user.
- **Sales** – It stores the fields of quantity, price and order date of the product.
- **Location** – This table has the columns of zip code, province and country to know about the location of buyers and sellers.
- **Merchant details** – It contains the information about the merchant including name, email, password, and contact details.
- **Store** – The table includes data about store name, location and the type of business.
- **Product** – Product name, id and description are the fields stored in the table.
- **Product category** – It describes the ID and names of the available product categories.
- **Inventory** – The inventory has details about the stock and price of the products for each store
- **Subscription** – It has information about price alerts and user subscriptions.
- **Tags** – This table contains the tag name and id as its columns.

- **Cart** – It indicates the cart id and status of the cart if it is shared with another user.
- **Product in cart** – The table depicts the quantity and price of the product inside a cart along with the id of the store from where it was purchased.
- **Cart sharing** – This table is helpful to display the cart and user ids of the users who are sharing a cart.
- **Combo** – It presents the combo offers available when the user adds a product to his cart.
- **Deals** – The details about ongoing deals are shown in this table.

6. Architecture

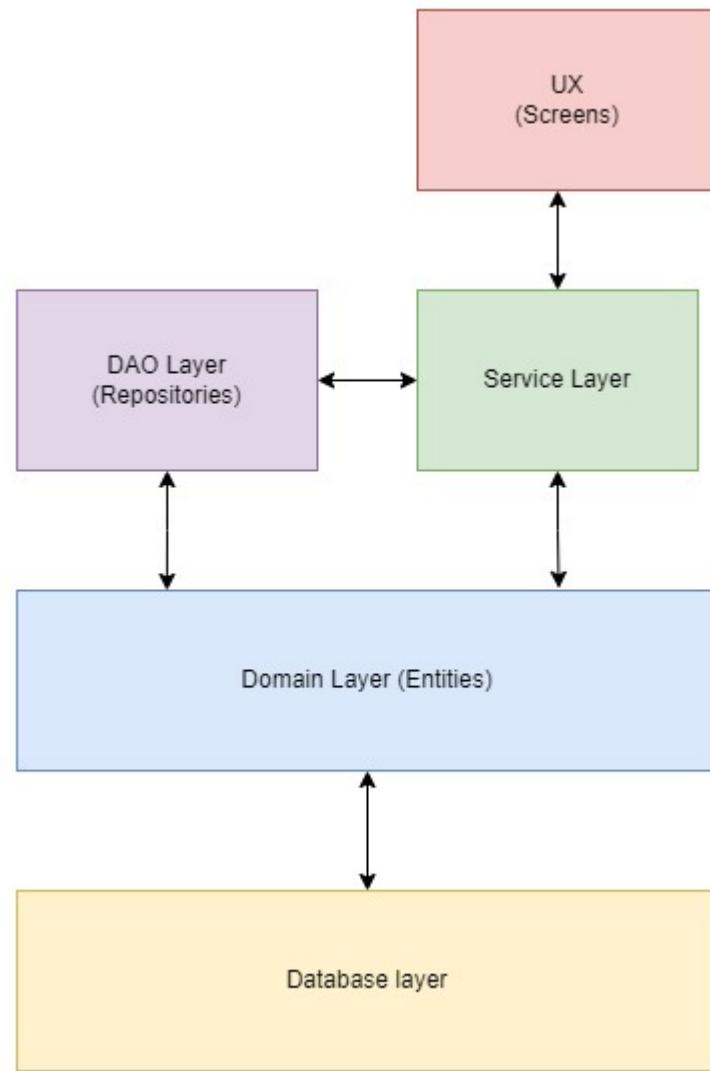


Figure 3: Architecture Diagram

The entities interact with the database. The entities reflect the tables of the database. The DAO layer contains the repositories that are responsible for the persistence of the objects to in the database. The service layer uses the entities defined in the domain layer along with the repository

layer to make services available to the screens. The end- user interacts with the screens to use the application.

7. User scenarios

7.1. User Functionality

7.1.1. User Registration and Login

Registration

The user can register to the application and then login. On application start up, the first page is the login page. The user needs to navigate to the registration page (Step 1 and 2). Then the user enters the details (Step 3 and 4). The user is then successfully registered post input validation. The security code here is used for user authentication for the forgot password feature.

Table 2: User Registration

Application Prompt	User Input
Application Starts	
Additional Navigation	:
Register	register
Are you a user or merchant	user
Enter registration details	<firstname> <lastname> <email> <password> <contact> <security code>

```
Are you a merchant or a user?  
:  
For additional navigation enter ":"  
Enter the page name  
Possible pages: register, resetPassword, login  
register  
***REGISTRATION***  
use : for additional navigation  
Are you a merchant or a user?  
user  
Enter <first name> <last name> <email> <password> <contact> <security code>  
kate parker kate@gmail.com kate 9897653344 123  
User has been successfully registered  
****USER MENU****  
  
Choose one of the following pages: products, subscriptions, profile, price_analytics, cart
```

Figure 4: User registration

Login

When the user selects the “user” option (Step 1), they will be redirected to the login details. Once the user enters the correct login credentials (Step 2), which is the “email and password”, they will be shown a Login successful message.

Table 3: User Login

Steps	Application Prompt	User Input
Step 1	Are you a merchant or user?	user
Step 2	Login Details	<email> <password>

```
Log...  
***LOGIN SCREEN***  
use : for additional navigation  
Are you a merchant or a user?  
[user  
Enter <email> <password>  
[kate@gmail.com kate  
****USER MENU****  
  
Choose one of the following pages: products, subscriptions, profile, price_analytics, cart
```

Figure 5: User login

7.1.2. Password Recovery

When the user enters invalid credentials, they are prompted with a message as “Invalid inputs received”. If by any chance the user forgets their password, they will be asked the question: “Forgot Password” or “Sign up for registering into the application”. When the user types “resetPassword”, the user must give the email, security code and the new password which the user must set for the account.

Table 4: User Password Recovery

Application Prompt	User Input
Forgot Password? Enter the email and security code along with the new password to recover your account.	resetPassword
Enter the details	<email> <security code> <new password>
Password Successfully Reset	

```

****LOGIN SCREEN****
use : for additional navigation
Are you a merchant or a user?
user
Enter <email> <password>
:
For additional navigation enter ":" 
Enter the page name
Possible pages: register, resetPassword, login
resetPassword
****RESET PASSWORD SCREEN****
Are you a merchant or a user?
user
*****
Forgot Password? Reset password by entering your <email id> <security code> <new password>
kate@gmail.com 123 katenew
****LOGIN SCREEN****
use : for additional navigation
Are you a merchant or a user?
user
Enter <email> <password>
kate@gmail.com katenew
****USER MENU****

Choose one of the following pages: products, subscriptions, profile, price_analytics, cart

```

Figure 6: User password recovery

7.1.3. View All Products and Stores

On successful login, the user is redirected to the user menu screen. The user should choose “products” to go to the products screen. The user should then choose “v_product” to view the products. The console will show the list of products available and their corresponding stores.

Table 5: User views products and stores

Application Prompt	User Input
Choose one of the following pages: <u>products</u> , subscriptions, profile, price_analytics, cart	products
To add product, enter a_product To filter products, enter f_zipcode, f_seller, f_zipcode_seller To search by tags, enter s_tags <u>To view all products, enter v_product</u> To compare the product prices, enter c_product	v_product

```

****PRODUCTS SCREEN****

To add product, enter a_product.
To filter products, enter f_zipcode, f_seller, f_zipcode_seller
To search by tags , enter s_tags
To view all products, enter v_product.
To toggle product availability based on location, enter t_product.
To compare the product prices, enter c_product
Enter "exit" to exit the product menu
[v_product
*****LIST OF PRODUCTS*****
Product Name: Milk
Id: 1
Available in: Walmart - halifax,123456
Price: 10.0
*****
Product Name: Wired Headphones
Id: 2
Available in: Atlantic - dartmouth,B3J2K9
Price: 15.0
*****
Product Name: FarmersMilk
Id: 7
Available in: Walmart - halifax,123456
Price: 5.0
*****
Product Name: Milk
Id: 1
Available in: Atlantic - dartmouth,B3J2K9
Price: 5.0
*****
Product Name: Ketchup
Id: 8
Available in: AsianMart - InglesiSt,BFG852
Price: 8.99
*****
Product Name: Chilli
Id: 9
Available in: AsianMart - InglesiSt,BFG852
Price: 6.99
*****

```

Figure 7: User view products

7.1.4. View products by applying filter over seller

On successful login, the user is redirected to the user menu screen. The user should choose “products” to go to the products screen. The user should then choose “f_seller” to view the products filtered by seller. The application will show the list of seller's names available. The user will input the seller of their choice. The list of products sold by the inputted seller will be displayed

Table 6: User views product availability filtered by sellers

Application Prompt	User Input
Choose one of the following pages: <i>products</i> , <i>subscriptions</i> , <i>profile</i> , <i>price_analytics</i> , <i>cart</i>	products
To add product, enter a_product To filter products, enter f_zipcode, <i>f_seller</i> , f_zipcode_seller To search by tags, enter s_tags To view all products, enter v_product	f_seller

To compare the product prices, enter c_product	
** list of sellers will be shown here**	<seller_name>

```

To add product, enter a_product.
To filter products, enter f_zipcode, f_seller, f_zipcode_seller
To search by tags , enter s_tags
To view all products, enter v_product.
To toggle product availability based on location, enter t_product.
To compare the product prices, enter c_product
Enter "exit" to exit the product menu
f_seller
List of sellers:
AsianStore
walmart
zig
Enter the seller
AsianStore
Product Name: Ketchup
Id: 8
Available in: AsianMart - InglsiSt,BFG852
*****
Product Name: Chilli
Id: 9
Available in: AsianMart - InglsiSt,BFG852
*****

```

Figure 8: User filter products by seller

7.1.5. View products by applying filter over zip code

On successful login, the user is redirected to the user menu screen. The user should choose “products” to go to the products screen. The user should then choose “f_zipcode” to view the products filtered by zipcode. The application will show the list of serviceable sellers. The user should input the zipcode of their choice. The list of products sold by the inputted seller will be displayed

Table 7: User views product availability filtered by zipcode

Application Prompt	User Input
Choose one of the following pages: <i>products</i> , <i>subscriptions</i> , <i>profile</i> , <i>price_analytics</i> , <i>cart</i>	products
To add product, enter a_product To filter products, enter <i>f_zipcode</i> , <i>f_seller</i> , <i>f_zipcode_seller</i> To search by tags, enter <i>s_tags</i> To view all products, enter <i>v_product</i> To compare the product prices, enter <i>c_product</i>	<i>f_zipcode</i>
** list of zipcode will be shown here**	<zipcode>

```

To add product, enter a_product.
To filter products, enter f_zipcode, f_seller, f_zipcode_seller
To search by tags , enter s_tags
To view all products, enter v_product.
To toggle product availability based on location, enter t_product.
To compare the product prices, enter c_product
Enter "exit" to exit the product menu
f_zipcode
List of serviceable zipcodes
123456
B3J2K9
BFG852
Enter the zipcode
BFG852
Product Name: Ketchup
Id: 8
Available in: AsianMart - InglsiSt,BFG852
*****
Product Name: Chilli
Id: 9
Available in: AsianMart - InglsiSt,BFG852
*****

```

Figure 9: Filter products by zipcode

7.1.6. View products by applying filter over seller and zip code

On successful login, the user is redirected to the user menu screen. The user should choose “products” to go to the products screen. The user should then choose “f_zipcode_seller” to view the products filtered by zipcode and seller. The application will show the list of serviceable sellers. The user should input the zipcode of their choice. The application will then show the list of seller's names available. The user should input the seller of their choice. The list of products sold by the inputted seller and the zipcode will be displayed

Table 8: User views product availability filtered by zipcode and seller

Application Prompt	User Input
Choose one of the following pages: <i>products</i> , subscriptions, profile, price_analytics, cart	products
To add product, enter a_product To filter products, enter f_zipcode, f_seller, <i>f_zipcode_seller</i> To search by tags , enter s_tags To view all products, enter v_product To compare the product prices, enter c_product	f_zipcode
** list of zipcode will be shown here**	<zipcode>
** list of sellers will be shown here**	<seller_name>

```

To add product, enter a_product.
To filter products, enter f_zipcode, f_seller, f_zipcode_seller
To search by tags , enter s_tags
To view all products, enter v_product.
To toggle product availability based on location, enter t_product.
To compare the product prices, enter c_product
Enter "exit" to exit the product menu
f_zipcode_seller
List of serviceable zipcodes
123456
B3J2K9
BFG852
Enter the zipcode
B3J2K9
List of sellers:
AsianStore
walmart
zig
Enter the seller
AsianStore
Product Name: Wired Headphones
Id: 2
Available in: Atlantic - dartmouth,B3J2K9
*****

```

Figure 10: Filter products by seller and zipcode

7.1.7. Search products by tags

On successful login, the user is redirected to the user menu screen. The user should choose “products” to go to the products screen. The user should then choose “s_tags” to search by tags. The user should input the tag of their choice. The list of products tagged with the input will be displayed.

Table 9: User views products by search tags

Application Prompt	User Input
Choose one of the following pages: <i>products</i> , subscriptions, profile, price_analytics, cart	products
To add product, enter a_product To filter products, enter f_zipcode, f_seller, f_zipcode_seller <i>To search by tags , enter s_tags</i>	s_tags
To view all products, enter v_product To compare the product prices, enter c_product	
enter a tag name. Eg. Milk	<tag> (Eg. “Dairy”, “milk”)
** List of products with the similar tag names will be displayed **	

```

To add product, enter a_product.
To filter products, enter f_zipcode, f_seller, f_zipcode_seller
To search by tags , enter s_tags
To view all products, enter v_product.
To toggle product availability based on location, enter t_product.
To compare the product prices, enter c_product
Enter "exit" to exit the product menu
s_tags
enter a tag name. Eg: milk
milk
Product Name: Milk
Id: 1
Available in: Walmart - halifax,123456
Price: 10.0
*****
Product Name: FarmersMilk
Id: 7
Available in: Walmart - halifax,123456
Price: 5.0
*****

```

Figure 11: search by tags

7.1.8. Compare product prices

On successful login, the user is redirected to the user menu screen. The user should choose “products” to go to the products screen. The user should then choose “c_product” to compare product prices. The application will show the list of products. The user should input the product of their choice. The console will then show the prices in each store.

Table 10: User compare product price between sellers

Application Prompt	User Input
Choose one of the following pages: <i>products</i> , subscriptions, profile, price_analytics, cart	products
To add product, enter a_product To filter products, enter f_zipcode, f_seller, f_zipcode_seller To search by tags , enter s_tags To view all products, enter v_product <i>To compare the product prices, enter c_product</i>	c_product
** list of products will be shown**	<product> (Eg. Eggs)

```

To add product, enter a_product.
To filter products, enter f_zipcode, f_seller, f_zipcode_seller
To search by tags , enter s_tags
To view all products, enter v_product.
To toggle product availability based on location, enter t_product.
To compare the product prices, enter c_product
Enter "exit" to exit the product menu
[c_product
Enter a product name to compare price
Possible names that you can enter : Milk, Wired Headphones, Eggs, Pasta, Burgers, sauce, FarmersMilk
|milk
Atlantic => 5.0
Walmart => 10.0

```

Figure 12: Compare product prices

7.1.9. Add products to cart

On successful login, the user is redirected to the user menu screen. The user should choose “products” to go to the products screen. The user should then choose “v_product” or “f_<condition>” to view the products. They can also choose to search the product by tags using “s_tags”. The application will show the list of products. The user should input the id of the product followed by the quantity as <id> <quantity>

Table 11: User adds products to cart

Application Prompt	User Input
Choose one of the following pages: <i>products</i> , subscriptions, profile, price_analytics, cart	products
<u>To add product, enter a_product</u> To filter products, enter f_zipcode, f_seller, f_zipcode_seller To search by tags, enter s_tags To view all products, enter v_product To compare the product prices, enter c_product	a_product
To add product to cart, enter <id> <quantity>	<id> <quantity>
After adding the inputs, enter “done”	done

```

To add product, enter a_product.
To filter products, enter f_zipcode, f_seller, f_zipcode_seller
To search by tags , enter s_tags
To view all products, enter v_product.
To toggle product availability based on location, enter t_product.
To compare the product prices, enter c_product
Enter "exit" to exit the product menu
a_product
To add product to cart, enter <id> <quantity>
Enter "done" when finished adding products to cart
1 10
Product 1 from store Walmart is added to cart
2 1
Product 2 from store Atlantic is added to cart
done
To add product, enter a_product.
To filter products, enter f_zipcode, f_seller, f_zipcode_seller
To search by tags , enter s_tags
To view all products, enter v_product.
To toggle product availability based on location, enter t_product.
To compare the product prices, enter c_product
Enter "exit" to exit the product menu

```

Figure 13: View Cart

7.1.10. View cart

The user can choose to view the cart details to get all the products present in the cart. The user should input “cart” from the user menu to go to the cart screen. The user should then choose “v_cart”. This option will display the items in cart and to which user it belongs to.

Table 12: User views the product in cart

Application Prompt	User Input
Choose one of the following pages: products, subscriptions, profile, analytics, <u>cart</u>	cart
Cart belongs to user@email.com Choose one of the following Use v_, s_, c_ to view, share, checkout followed by cart	v_cart
** list of products in cart and who it belongs to**	

```

Choose one of the following pages: products, subscriptions, profile, price_analytics, cart
cart
****CART SCREEN****

Choose one of the following
Use v_, s_, c_ to view, share, checkout followed by cart
eg: v_cart, s_cart, c_cart
v_cart
product size 2
Cart belongs to pv304053@dal.ca
Product Wired Headphones -> Quantity: 1
Product Milk -> Quantity: 10

```

Figure 14: Share cart

7.1.11. Share cart

The user can choose to share the cart with other users to get all the products from a single store for the cheapest. The user should input “cart” from the user menu to go to the cart screen. The user should then choose “s_cart”. Next, the user has to input the email ids of the other users they want to share with, followed by “done”.

Table 13: User shares cart with other users

Application Prompt	User Input
Choose one of the following pages: products, subscriptions, profile, analytics, cart	cart
Cart belongs to user@email.com Choose one of the following Use v_, s_, c_ to view, share, checkout followed by cart eg: v_cart, s_cart, etc.	s_cart
Enter the email id of user, one per line and "done" when completed	Multiple <email_id> followed by done

```
To add product, enter a_product.
To filter products, enter f_zipcode, f_seller, f_zipcode_seller
To search by tags , enter s_tags
To view all products, enter v_product.
To toggle product availability based on location, enter t_product.
To compare the product prices, enter c_product
Enter "exit" to exit the product menu
a_product
To add product to cart, enter <id> <quantity>
Enter "done" when finished adding products to cart
1 1
Product Milk from store Walmart is added to cart for a price 10.0 each
Product Milk from store Atlantic is added to cart for a price 5.0 each
done
To add product, enter a_product.
To filter products, enter f_zipcode, f_seller, f_zipcode_seller
To search by tags , enter s_tags
To view all products, enter v_product.
To toggle product availability based on location, enter t_product.
To compare the product prices, enter c_product
Enter "exit" to exit the product menu
exit
Exiting user menu...
Choose one of the following pages: products, subscriptions, profile, price_analytics, cart
cart
****CART SCREEN****

Choose one of the following
Use v_, s_, c_ to view, share, checkout followed by cart
eg: v_cart, s_cart, c_cart
s.cart
Enter the email id of user, one per line and "done" when completed
pv304053@dal.ca
done
We have clubbed your carts now!
```

Figure 15: Share cart

7.1.12. Checkout cart to view the cheapest deal

The user can check-out a cart from going to cart menu by hitting “cart” after login and then “c_cart”. All the carts belonging to the user and other users who share the same cart will be checked out. The cheapest store selling those products will also be shown.

Table 14: User check out cart

Application Prompt	User Input
Choose one of the following pages: products, subscriptions, profile, <u>price_analytics</u> , cart	cart
Cart belongs to user@email.com Choose one of the following Use v_, s_, c_ to view, share, checkout followed by cart eg: v_cart, s_cart, etc.	c_cart

```
Choose one of the following
Use v_, s_, c_ to view, share, checkout followed by cart
eg: v_cart, s_cart, c_cart
[c_cart
Your total price is:
$100.0 from Walmart
$15.0 from Atlantic
Cheapest store: Atlantic for $15.0
cart 1 checked out
```

Figure 16: checkout cart

7.1.13. View Price Analytics

The user can check the price analytics of a particular product by every month by hitting the “price_analytics” option. Next, the user will be prompted to enter the name of the product. This will generate a graph that shows the price analytics of the product(e.g., iPhone).

Table 15: User views the best time to buy a product

Application Prompt	User Input
Choose one of the following pages: products, subscriptions, profile, <u>price_analytics</u> , cart	price_analytics
Enter a product name to get the price analytics by every month	<product_name> (Eg. iPhone)

```

Choose one of the following pages: products, subscriptions, profile, price_analytics, cart
price_analytics      228 KB  PNG
*****PRICE ANALYTICS*****
12:27 PM           155 KB  PNG
Enter a product name to get the price analytics by every month
Enter exit to return to user analytics screen
iPhone
*****
The graph can be viewed at ./productPriceAnalyticsCharts
exit
Exiting Price analytics...

```

Figure 17: View price analytics

```

elizabeth@timberlea:~/csci5308/deploy_builds/target$ cd productPriceAnalyticsCharts/
elizabeth@timberlea:~/csci5308/deploy_builds/target/productPriceAnalyticsCharts$]
ls
iPhone.png

```

Figure 18: price analytics location

Figure 19 shows the analytics chart generated for iphone.

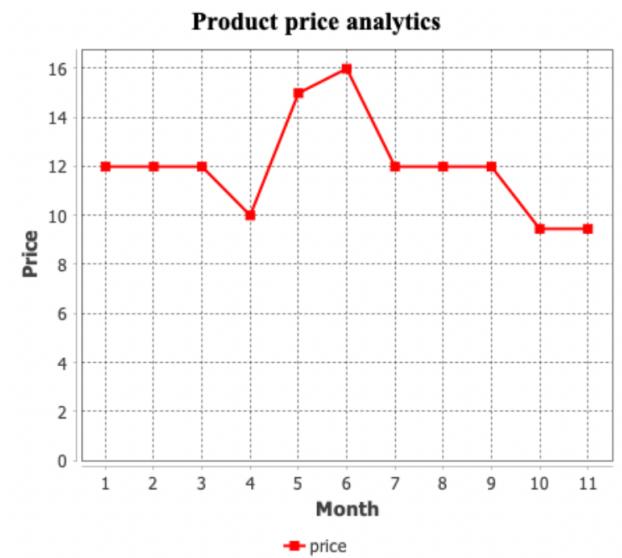


Figure 19: Price analytics for iphone

7.1.14. Add subscription for price alerts

The user can choose to subscribe for price alerts. If there is any drop in price for the product that the user has selected, they will get a mail like Figure 34.

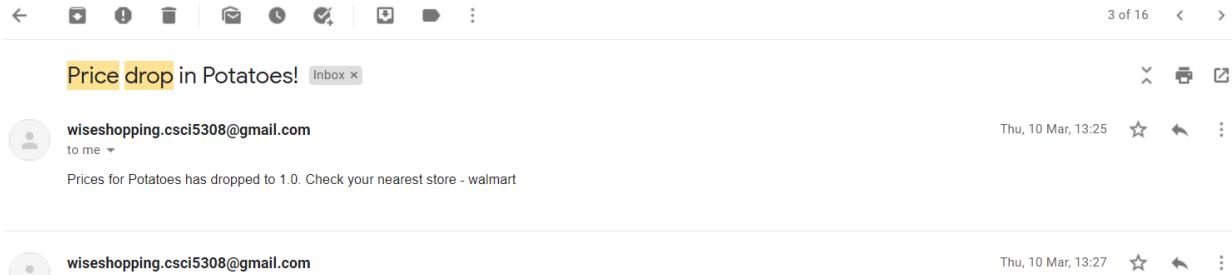


Figure 20: Price alert mail screenshot

The user can check-out the subscriptions option from going to subscription menu by hitting “subscription” after login. In order to add a subscription for price alerts, the user should enter “a_subscription”. Then the user will be prompted to enter the product ID and the price alert. This way, the subscription for price alert will be set.

Table 16: User adds subscription

Application Prompt	User Input
Choose one of the following pages: products, subscriptions, profile, <u>price_analytics</u> , cart	subscriptions
Choose one of the following Use v_, u_, d_ to add, update or delete followed subscription eg: v_subscriptions, a_subscriptions, d_subscriptions, etc. Use exit to exit the subscriptions screen	a_subscriptions
Enter <product_id> <price_alert>	<product_id> <price_alert> Eg: 2 3
Displays the subscription has been added: Eg: Subscription with user: ad123@gmail.com, productId: milk, priceAlert: 22.0 saved	

```

Choose one of the following pages: products, subscriptions, profile, price_analytics, cart
subscriptions
****SUBSCRIPTION SCREEN****

[Choose one of the following
Use v_, a_, d_ to add, update or delete followed subscription
eg: v_subscriptions, a_subscriptions, d_subscriptions, etc.
[Use exit to exit the subscriptions screen
a_subscriptions
Enter <product_id> <price_alert>
1 2.99
[Subscription with user: pv304053@dal.ca, productId: Milk, priceAlert: 2.99 saved

```

Figure 21: add subscription of price alerts

7.1.15. View subscriptions

The user can choose to view the subscriptions. For viewing a subscription, the user can enter v_subscription. This option displays the current subscription alerts that are active for that user.

Table 17: User views subscription

Application Prompt	User Input
Choose one of the following pages: products, subscriptions, profile, <u>price_analytics</u> , cart	subscriptions
Choose one of the following Use v_, u_, d_ to add, update or delete followed subscription eg: v_subscriptions, a_subscriptions, d_subscriptions, etc. Use exit to exit the subscriptions screen	v_subscriptions
Displays the subscriptions the user has: Eg: Product name: milk, price alert: 22.0	

```
*****SUBSCRIPTION SCREEN****

Choose one of the following
Use v_, a_, d_ to add, update or delete followed subscription
eg: v_subscriptions, a_subscriptions, d_subscriptions, etc.
Use exit to exit the subscriptions screen
[v_subscriptions
Product name: Milk, price alert: 2.99
```

Figure 22: view subscriptions

7.1.16. Delete subscriptions

The user can choose to delete a particular subscription alert for a product. For deleting a subscription, the user can enter d_subscriptions. After this, the user will be prompted to enter the ID of the product for which the user wants to delete the subscription. On successfully entering the product ID, the corresponding subscription will be deleted.

Table 18: User deletes subscription

Application Prompt	User Input
Choose one of the following pages: products, subscriptions, profile, <u>price_analytics</u> , cart	subscriptions
Choose one of the following	d_subscriptions

Use v_, u_, d_ to add, update or delete followed subscription eg: v_subscriptions, a_subscriptions, d_subscriptions, etc. Use exit to exit the subscriptions screen	
Enter the <product_id> to be deleted	2
Displays the subscription has been deleted.	

```

Choose one of the following pages: products, subscriptions, profile, price_analytics, cart
subscriptions
****SUBSCRIPTION SCREEN****

Choose one of the following
Use v_, a_, d_ to add, update or delete followed subscription
eg: v_subscriptions, a_subscriptions, d_subscriptions, etc.
Use exit to exit the subscriptions screen
d_subscriptions
SubscriptionId: 1, Product name: Milk, price alert: 2.99
Enter the <id> to be deleted
1
Subscription deleted

```

Figure 23: delete subscriptions

3.2.17 Update subscriptions

The user can choose to update the current alert for a product to a new alert. For updating the subscription and adding a new price alert for a product, the user has to enter u_subscriptions. This option first displays the current active subscription and the price alert associated with it. Then, the user is prompted to enter the product ID and the new price alert for the product. On successfully entering the details, the subscription alert for the product is updated.

Table 19: User updates subscription

Application Prompt	User Input
Choose one of the following pages: products, subscriptions, profile, <u>price_analytics</u> , cart	subscriptions
Choose one of the following Use a_, v_, u_, d_ to add, view, update or delete followed subscription eg:a_subscription, v_subscriptions, a_subscriptions, d_subscriptions, etc. Use exit to exit the subscriptions screen	u_subscriptions
Displays the current subscription and to Enter the <id> <new_price_alert> to be updated	<id> <new_price_alert> Eg: 12 20

Eg: SubscriptionId: 130, Product name: milk, price alert: 22.0	
Displays the Subscription has been updated	
<pre> Choose one of the following pages: products, subscriptions, profile, price_analytics, cart subscriptions ****SUBSCRIPTION SCREEN**** Choose one of the following Use v_, a_, d_ to add, update or delete followed subscription eg: v_subscriptions, a_subscriptions, d_subscriptions, etc. Use exit to exit the subscriptions screen a_subscriptions Enter <product_id> <price_alert> 1 2.99 Subscription with user: pv304053@dal.ca, productId: Milk, priceAlert: 2.99 saved exit Exiting subscription menu... Choose one of the following pages: products, subscriptions, profile, price_analytics, cart subscriptions ****SUBSCRIPTION SCREEN**** Choose one of the following Use v_, a_, d_ to add, update or delete followed subscription eg: v_subscriptions, a_subscriptions, d_subscriptions, etc. Use exit to exit the subscriptions screen u_subscriptions SubscriptionId: 2, Product name: Milk, price alert: 2.99 Enter the <id> <new_price_alert> to be updated 2 2.09 Subscription updated exit Exiting subscription menu... Choose one of the following pages: products, subscriptions, profile, price_analytics, cart subscriptions ****SUBSCRIPTION SCREEN**** Choose one of the following Use v_, a_, d_ to add, update or delete followed subscription eg: v_subscriptions, a_subscriptions, d_subscriptions, etc. Use exit to exit the subscriptions screen v_subscriptions Product name: Milk, price alert: 2.09 </pre>	

Figure 24: Update subscriptions

7.2. Merchant Functionality

7.2.1. Merchant registration and login

Registration

The merchant can register to the application and then login. On application start up, the first page is the login page. The merchant needs to navigate to the registration page. Then the merchant enters the details. The merchant is then successfully registered post input validation. The security code here is used for user authentication for the forgot password feature. After successful registration, the merchant is redirected to merchant menu screen.

Table 20: Merchant Registration

Application Prompt	Merchant Input
Application Starts	

Additional Navigation	:
Register	register
Enter registration details	<email> <password> <contact> <security code>

```
***LOGIN SCREEN****
use : for additional navigation
Are you a merchant or a user?
:
For additional navigation enter ":" 
Enter the page name
Possible pages: register, resetPassword, login
register
***REGISTRATION****
use : for additional navigation
Are you a merchant or a user?
merchant
Enter <name> <email> <password> <security code>
AsianStore asian@store.com password 123
merchant is not registered
You have been registered successfully
****MERCHANT MENU****

Choose one of the following
products, store, analytics
```

Figure 25: Merchant Registration

Login

When the selects the “merchant” option (Step 1), they will be redirected to the merchant login details. Once the merchant enters the correct login credentials (Step 2), which is the “email and password”, they will be shown a Login successful message.

Table 21: Merchant Login

Application Prompt	Merchant Input
Are you a merchant or user?	merchant
Login Details	<email> <password>

```

Started WiseshoppingApplication in 5.018 seconds (JVM running for 5.614)
***LOGIN SCREEN***
use : for additional navigation
Are you a merchant or a user?
merchant
Enter <username> <password>
asian@store.com password
You have logged in successfully
****MERCHANT MENU****

Choose one of the following
products, store, analytics

```

Figure 26: Merchant login

7.2.2. Password Recovery

The merchant is prompted with the notification "Invalid inputs received" when they enter invalid credentials. If a merchant forgets his password, he will be prompted to choose "Forgot Password" or "Sign up for enrolling into the program." When the merchant inputs "resetPassword," the merchant must provide the email address, security code, and a new password for the account.

Table 22: Merchant Password Recovery

Application Prompt	Merchant Input
Forgot Password and Account Recovery	resetPassword
Reset Password Details	<email> <security code> <new password>

```

For additional navigation enter ":" 
Enter the page name
Possible pages: register, resetPassword, login
resetPassword
***RESET PASSWORD SCREEN***
Are you a merchant or a user?
merchant
*****
Forgot Password? Reset password by entering your <email id> <security code> <new password>
asian@store.com 123 changedpassword
***LOGIN SCREEN***
use : for additional navigation
Are you a merchant or a user?
merchant
Enter <username> <password>
asian@store.com changedpassword
You have logged in successfully
****MERCHANT MENU****

Choose one of the following
products, store, analytics

```

Figure 27: Merchant Password Recovery

7.2.3. Add store

On successful login, the merchant is redirected to the merchant menu screen. Then, they should choose "store" to go to the stores screen. The merchant should then choose "a_store" to add the

stores. Then the merchant will be prompted to enter the store name, type of business, start and end time of the store, as well as the contact information. Then the merchant is supposed to enter the location name, the corresponding zip code, the province, and the country details. On successful addition of the details, the merchant will be displayed the information about the added store.

Table 23: Merchant Add Store

Application Prompt	Merchant Input
Choose one of the following: products, store, analytics	store
Enter one of the following - add or delete or update Enter a_store to add store Enter d_store to delete store Enter u_store to update stores Enter v_store to view stores Enter exit to exit store menu	a_store
Enter <store_name> <business_type> <start_time> <end_time> <contact>	<store_name> <business_type> <start_time> <end_time> <contact> Eg : ShoppersDrugMart commercial 10 12 9896677898
Enter <locationName> <zipcode> <province> <country>	<locationName> <zipcode> <province> <country> Eg : SouthParkStreet B33GHJ NS CA
** Added store will be displayed**	

```
You have logged in successfully
****MERCHANT MENU****

Choose one of the following
products, store, analytics
store
****STORE MENU****

enter one of the following - add or delete or update
enter a_store to add store
enter d_store to delete store
enter u_store to update stores
enter v_store to view stores
enter exit to exit store menu
a_store
Enter <store_name> <business_type> <start_time> <end_time> <contact>
AsianMart local 11 20 8527419632
Enter <locationName> <zipcode> <province> <country>
Inglsist BFG852 NS CA
Location Inglsist is added
Store AsianMart is added
```

Figure 28: Merchant add store

7.2.4. View store

On successful login, the merchant is redirected to the merchant menu screen. The merchant should choose “store” to go to the stores screen. The merchant should then choose “v_store” to view the stores. The console will show the list of products available and their corresponding stores.

Table 24: Merchant View Store

Application Prompt	Merchant Input
Choose one of the following: products, store, analytics	store
enter one of the following - add or delete or update enter a_store to add store enter d_store to delete store enter u_store to update stores enter v_store to view stores enter exit to exit store menu	v_store
** Stores belonging to the merchant will be displayed **	

```
Choose one of the following
products, store, analytics
store
****STORE MENU****

enter one of the following - add or delete or update
enter a_store to add store
enter d_store to delete store
enter u_store to update stores
enter v_store to view stores
enter exit to exit store menu
v_store
Store id: 3, name: AsianMart startTime: 11:00:00, endTime: 20:00:00, contact: 8527419632
Choose one of the following
products, store, analytics
█
```

Figure 29: Merchant view store

7.2.5. Update store

When the merchant wants to update store details, he can do it by entering the u_store command. Then the merchant would be asked for the id of the store that needs to be updated. The merchant

needs to pass the details that he wants to change in the form of key-value pairs. After updating the required fields, the merchant has to write ‘done’ to execute the changes.

Table 25: Merchant Update Store

Application Prompt	Merchant Input
Choose one of the following: products, store, analytics	store
enter one of the following - add or delete or update enter a_store to add store enter d_store to delete store enter u_store to update stores enter v_store to view stores enter exit to exit store menu	u_store
** Stores belonging to the merchant will be displayed **	
Enter the id to be updated	<store_id>
Pass the attributes to be updated. Enter in <key>:<value> format Acceptable keys are : name, type, startTime, endTime, contact Enter done after updating	<key>:<value> Eg : name:dollaramastore done
** Updated values of the store will be displayed **	

```

Choose one of the following
products, store, analytics
store
*****STORE MENU****

enter one of the following - add or delete or update
enter a_store to add store
enter d_store to delete store
enter u_store to update stores
enter v_store to view stores
enter exit to exit store menu
u_store
Store id: 3, name: AsianMart
Enter the id to be updated
3
Pass the attributes to be updated. Enter in <key>:<value> format
Acceptable keys are : name, type, startTime, endTime, contact
Enter done after updating
contact:7894561231
done
Store AsianMart is updated
Choose one of the following
products, store, analytics

```

Figure 30: Merchant update store

7.2.6. Delete store

The merchant can choose to delete a store. For this option, the merchant has to enter `d_store`. On entering this option, the console will display all the stores belonging to the merchant. Then the merchant will be prompted to enter the corresponding ID of the store to be deleted. On entering the correct ID of the store, the particular store will be deleted, and the merchant will be displayed a message that the Store is deleted.

Table 26: Merchant Delete Store

Application Prompt	Merchant Input
Choose one of the following: products, store, analytics	store
enter one of the following - add or delete or update enter a_store to add store enter d_store to delete store enter u_store to update stores enter v_store to view stores enter exit to exit store menu	<code>d_store</code>
** Stores belonging to the merchant will be displayed **	
Enter the id to be deleted	<code><store_id></code>

** Store id <id> deleted **	
-----------------------------	--

```

Choose one of the following
products, store, analytics
store
*****STORE MENU****

enter one of the following - add or delete or update
enter a_store to add store
enter d_store to delete store
enter u_store to update stores
enter v_store to view stores
enter exit to exit store menu
d_store
Store id: 3, name: AsianMart
Store id: 5, name: KoreanStore
Enter the id to be deleted
5
Store ID : 5 deleted
Choose one of the following
products, store, analytics

```

Figure 31: Merchant delete store

7.2.7. Add products

The merchant can choose to add products. For this, the merchant should select the products option, followed by entering a_product. Now the merchant will be prompted to enter the name of the product, name of the store, along with the price and stock if the product. On entering the corresponding details successfully, the merchant will be prompted with a message that all the details of the product have been added.

Table 27: Merchant Add Products

Application Prompt	Merchant Input
Choose one of the following: products, store, analytics	products
enter one of the following - add or delete or update enter a_product to add product enter d_product to delete product enter u_product to update product enter v_product to view product enter exit to exit store menu	a_product
Enter product_name, store_name, price, stock Example : <product_name> <store_name> <price> <stock>	<product_name> <store_name> <price> <stock> Eg : iPad walmart 100.99 30
** Product <product_name> with id <product_id> added **	

** Product <product_name> from store <store_name> is added **	
---------------------------------------------------------------	--

```

Choose one of the following
products, store, analytics
products
**** PRODUCTS SCREEN****

Enter a_product to add products to inventory
Enter u_product to update products
Enter exit to exit the merchant products menu
a_product
Enter product_name, store_name, price, stock
Example : <product_name> <store_name> <price> <stock>
Ketchup AsianMart 8.99 20
Product Ketchup with id 8 added
Product Ketchup from store AsianMart is added
Enter a_product to add products to inventory
Enter u_product to update products
Enter exit to exit the merchant products menu

```

Figure 32: Merchant add store

7.2.8. Update products

7.2.8.1. Update price

The merchant has the option to update the price of a product. For this option, the merchant has to enter u_product. Then the merchant will be shown a bunch of options to update about the product. In that the merchant must select u_price to update the price. Now, the stores belonging to the merchant will get displayed. Then the merchant has to enter the corresponding store_id, which in follow will show all the available products in that particular store. Further, the merchant must enter the ID of the product and the price associated with it. On successfully entering the details, the merchant will be displayed that the price for the product is updated.

Table 28: Merchant Update Product price

Application Prompt	Merchant Input
Choose one of the following: products, store, analytics	products
enter one of the following - add or delete or update enter a_product to add product enter d_product to delete product enter u_product to update product enter v_product to view product enter exit to exit store menu	u_product

Enter u_price to update product price Enter u_stock to update product stock Enter u_categoryname to update product category name Enter u_categorydesc to update product category description Enter u_tags to update product tags	u_price
** Stores belonging to the merchant will be displayed **	
Enter the store id ** List of available products in the store will be displayed **	<store_id>
Enter the product id Enter the price to be updated	<product_id> <price>
** Updated price of the selected product will be displayed **	

```

Choose one of the following
products, store, analytics
products
**** PRODUCTS SCREEN****

Enter a_product to add products to inventory
Enter u_product to update products
Enter exit to exit the merchant products menu
u_product
Enter u_price to update product price
Enter u_stock to update product stock
Enter u_categoryname to update product category name
Enter u_categorydesc to update product category description
Enter u_tags to update product tags
u_price
***** List of available stores *****
Store ID 3 Store Name AsianMart
Enter the store id
3
***** List of available products *****
Product ID - 8 Product Name - Ketchup Price - 8.99
Product ID - 9 Product Name - Chilli Price - 5.99
Enter the product id
9
Enter the price to be added
6.99
updated price : 6.99
Price of Chilli is updated to 6.99
Enter a_product to add products to inventory
Enter u_product to update products
Enter exit to exit the merchant products menu

```

Figure 33: Merchant update price

7.2.8.2. Update stock

This option lets the merchant update the stock in the inventory. The merchant must enter `u_product` for this option. The merchant will then be presented with a number of alternatives for updating information about the product. The next option to select would be `u_stock`. In addition, the merchant must input the store ID as well as the product ID. After correctly entering the information, the merchant will see that the information about the updated stock will be shown on the screen.

Table 29: Merchant Update Product stock

Application Prompt	Merchant Input
Choose one of the following: products, store, analytics	products
enter one of the following - add or delete or update enter a_product to add product enter d_product to delete product enter u_product to update product	u_product

enter v_product to view product enter exit to exit store menu	
Enter u_price to update product price Enter u_stock to update product stock Enter u_categoryname to update product category name Enter u_categorydesc to update product category description Enter u_tags to update product tags	u_stock
** Stores belonging to the merchant will be displayed **	
Enter the store id ** List of available products in the store will be displayed **	<store_id>
Enter the product id Enter the stock to be updated ** Updated stock of the selected product will be displayed **	<product_id> <price>

```

Choose one of the following
products, store, analytics
products
**** PRODUCTS SCREEN****

Enter a_product to add products to inventory
Enter u_product to update products
Enter exit to exit the merchant products menu
u_product
Enter u_price to update product price
Enter u_stock to update product stock
Enter u_categoryname to update product category name
Enter u_categorydesc to update product category description
Enter u_tags to update product tags
u_stock
***** List of available stores *****
Store ID 3 Store Name AsianMart
Enter the store id
3
***** List of available products *****
Product ID - 8 Product Name - Ketchup Stock - 30
Product ID - 9 Product Name - Chilli Stock - 20
Enter the product id
8
Enter the stock to be updated
25
Stock of Ketchup is updated to 25
Enter a_product to add products to inventory
Enter u_product to update products
Enter exit to exit the merchant products menu
]

```

Figure 34: Merchant Update stock

7.2.8.3. Update category name

The merchant has the option to update the category name of a product. For this option, the merchant has to enter `u_product`. Then the merchant will be shown a bunch of options to update about the product. In that the merchant must select `u_categoryname` to update the category name. Now, the merchant must enter the category ID of the product and the name of the product. On successfully entering the details, the merchant will be displayed that the category name for the product is updated

Table 30: Merchant Update Product category name

Application Prompt	Merchant Input
Choose one of the following: products, store, analytics products	
enter one of the following - add or delete or update enter a_product to add product enter d_product to delete product enter u_product to update product enter v_product to view product enter exit to exit store menu	u_product

Enter u_price to update product price Enter u_stock to update product stock Enter u_categoryname to update product category name Enter u_categorydesc to update product category description Enter u_tags to update product tags	u_categoryname
Enter the updated categoryname for the product Example : <product_category_id> <name>	<productcategory_id> <name>
** Updated category name of the product is displayed **	

```

Choose one of the following
products, store, analytics
products
***** PRODUCTS SCREEN****

Enter a_product to add products to inventory
Enter u_product to update products
Enter exit to exit the merchant products menu
u_product
Enter u_price to update product price
Enter u_stock to update product stock
Enter u_categoryname to update product category name
Enter u_categorydesc to update product category description
Enter u_tags to update product tags
u_categoryname
Enter the updated categoryname for the product
Example : <product_category_id> <name>
4 Chilli
Category name of 4 is updated to Chilli
Enter a_product to add products to inventory
Enter u_product to update products
Enter exit to exit the merchant products menu

```

Figure 35: Merchant Update category name

7.2.8.4. Update category description

The option lets the merchant to update the category description of a product. For this option, the merchant has to enter u_product. Then the merchant will be shown a bunch of options to update about the product. In that the merchant must select u_categorydesc to update the category description. Now, the merchant must enter the category ID of the product and the description of the product. On successfully entering the details, the merchant will be displayed that the category description for the product is updated.

Table 31: Merchant Update Product category description

Application Prompt	Merchant Input
Choose one of the following: products, store, analytics	products
enter one of the following - add or delete or update enter a_product to add product enter d_product to delete product enter u_product to update product enter v_product to view product enter exit to exit store menu	u_product
Enter u_price to update product price Enter u_stock to update product stock Enter u_categoryname to update product category name Enter u_categorydesc to update product category description Enter u_tags to update product tags	u_categorydesc
Enter the updated categorydescription for the product Example : <product_category_id><description> <description>	<productcategory_id> <description>
** Updated category description of the product is displayed **	

```

Choose one of the following
products, store, analytics
products
**** PRODUCTS SCREEN****

Enter a_product to add products to inventory
Enter u_product to update products
Enter exit to exit the merchant products menu
u_product
Enter u_price to update product price
Enter u_stock to update product stock
Enter u_categoryname to update product category name
Enter u_categorydesc to update product category description
Enter u_tags to update product tags
u_categorydesc
Enter the updated categorydesc for the product
Example : <product_category_id> <description>
4 Spices
Category description of 4 is updated to Spices
Enter a_product to add products to inventory
Enter u_product to update products
Enter exit to exit the merchant products menu

```

Figure 36: Merchant Update category description

7.2.9. Update tags

The merchant has the option to update the tags associated with a product. For this option, the merchant has to enter `u_product`. Then the merchant will be shown a bunch of options to update about the product. In that the merchant must select `u_tags` to update tags related to the product. Now, the merchant must enter the name of the product. This option will display the list of available tags for the product. Now the merchant will be prompted to enter the ID of the tag and the updated tag name corresponding to the product. On successfully entering the details, the merchant will be displayed that the tag name for the given tag ID of the product.

Table 32: Merchant Update Product tags

Application Prompt	Merchant Input
Choose one of the following: products, store, analytics	products
enter one of the following - add or delete or update enter a_product to add product enter d_product to delete product enter u_product to update product enter v_product to view product enter exit to exit store menu	<code>u_product</code>
Enter u_price to update product price Enter u_stock to update product stock Enter u_categoryname to update product category name	<code>u_tags</code>

Enter u_categorydesc to update product category description Enter u_tags to update product tags	
Enter a product name	<product_name>
** List of available tags for the product will be displayed **	
Enter the tag id and updated tag name	<tag_id> <updated_tag_name>
** updated tag name for the given tag id will be displayed **	

```

Choose one of the following
products, store, analytics
products
***** PRODUCTS SCREEN****

Enter a_product to add products to inventory
Enter u_product to update products
Enter exit to exit the merchant products menu
u_product
Enter u_price to update product price
Enter u_stock to update product stock
Enter u_categoryname to update product category name
Enter u_categorydesc to update product category description
Enter u_tags to update product tags
u_tags
Enter a product name
Chilli
Available tags Chilli
Enter the tag id and the tag name to be updated
9 Spices
Tag name updated for Tag9
Enter a_product to add products to inventory
Enter u_product to update products
Enter exit to exit the merchant products menu

```

Figure 37: Merchant update tags

7.2.10. Analytics

The merchant can also view the statistics about a product. The merchant just has to enter the demand_trend option in order to view detailed analysis of any product for a given interval of time.

Table 33: Merchant demand trend analytics

Application Prompt	Merchant Input
--------------------	----------------

Choose one of the following: products, store, analytics	analytics
Enter demand_trend to view the demand trend of a product	demand_trend
** Path of the demand trend analytics charts generated is displayed **	

```

Choose one of the following
products, store, analytics
analytics
****ANALYTICS****

Enter demand_trend to view the demand trend of a product
demand_trend
****PRODUCT DEMAND TREND ANALYTICS****

*****
Product demand charts are generated in ./productDemandCharts/ folder
Choose one of the following
products, store, analytics
[redacted]

```

Figure 38: Merchant analytics

Figure 18 shows the demand trend analytics of a sample product - iphone

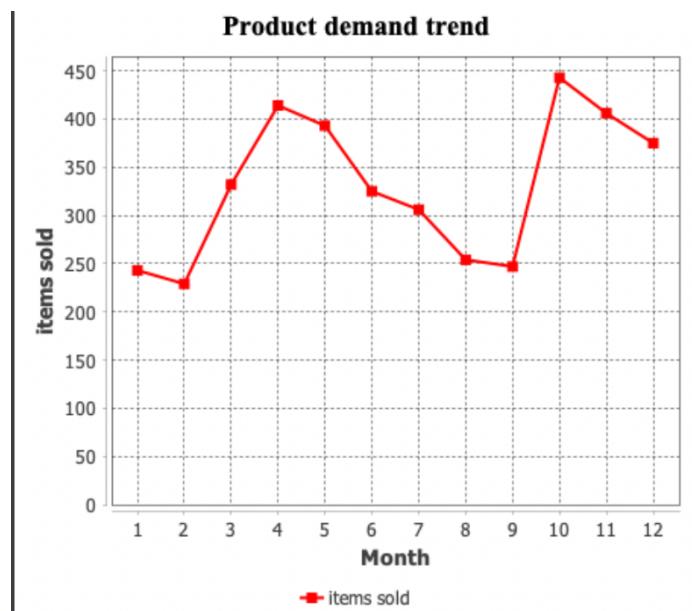


Figure 39: Demand trend analytics

7.3. Common features

The user can hit ":" at any point, followed by "logout" to logout

```
Choose one of the following
Use v_, s_, c_ to view, share, checkout followed by cart
eg: v_cart, s_cart, c_cart
:
For additional navigation enter ":" 
Enter the page name
Possible pages: logout, cart, subscriptions, userProducts, profile, user_menu, price_analytics
logout
****LOGGED OUT****
****LOGIN SCREEN****
use : for additional navigation
Are you a merchant or a user?
```

Figure 40: logout

The user can at hit ":" at any point in the application to navigate between inputs

```
eg: v_cart, s_cart, c_cart
:
For additional navigation enter ":" 
Enter the page name
Possible pages: logout, cart, subscriptions, userProducts, profile, user_menu, price_analytics
userProducts
****PRODUCTS SCREEN****

To add product, enter a_product.
To filter products, enter f_zipcode, f_seller, f_zipcode_seller
To search by tags , enter s_tags
To view all products, enter v_product.
To toggle product availability based on location, enter t_product.
To compare the product prices, enter c_product
Enter "exit" to exit the product menu
```

Figure 41: additional navigations

The user can hit ":" followed by "help" to check all the available screens

```
Enter "exit" to exit the product menu -
:
For additional navigation enter ":" 
Enter the page name
Possible pages: logout, cart, subscriptions, userProducts, profile, user_menu, price_analytics
help
Use ":" for additional naviagtion on any page
Following is the map
User menu screen -> Products menu
                  -> Cart menu
                  -> Analytics menu
                  -> Subscriptions menu
                  -> Profile menu
Merchant menu screen -> store menu
                      -> products menu
                      -> analytics menu
For additional navigation enter ":" 
Enter the page name
Possible pages: logout, cart, subscriptions, userProducts, profile, user_menu, price_analytics
```

Figure 42: help

8. Analytics

An algorithm is created to find the lowest pricing for a product over a year's time. The local minimum notion is used to accomplish this. Every month, the local minimum price for each product is calculated by collecting prior pricing history. Graphs will be generated based on these local minimum values whenever a user requests the lowest possible price for a product.

9. Design Pattern

We implemented the factory pattern to generate the various screens shown to the user. Screen is an interface which has many concrete implementations like LoginScreen, RegistrationScreen, UserMenuScreen, etc. ScreenFactory class is responsible for generating these screens based on the input received. If the input is “login”, ScreenFactory will generate an instance of class LoginScreen

10. Smell Analysis Summary

We refactored the project after 80% of the development to reduce the smells. Table 34 shows insights on how many smells were reduced. The screenshot of the output of running designate against the application before and after refactoring is also shown.

Table 34: Smell Analysis

Smells Category	Smell Name	Pre-refactoring	Post-refactoring
Architecture	Feature Concentration	7	0
Design	Unnecessary Abstraction	3	0
	Feature Envy	1	0
	Insufficient modularization	2	0
	Unutilized Abstraction	56	51
	Complex conditional	4	0
Implementation Smells	Long Identifier	2	0
	Long method	1	0
	Long parameter list	12	6

Pre-refactoring with 80% of the development done:

```

E:\Winter22\ASDC\assignment\assignment2\problem1>java -jar DesigniteJava.jar -i E:\Winter22\ASDC\wiseshopping -o wiseshopping_smells
Searching classpath folders ...
Parsing the source code ...
Resolving symbols...
Computing metrics...
Detecting code smells...
Exporting analysis results...
wrapping up ...
--Analysis summary--
    Total LOC analyzed: 3890      Number of packages: 13
    Number of classes: 97      Number of methods: 391
-Total architecture smell instances detected-
    Cyclic dependency: 0      God component: 0
    Ambiguous interface: 0      Feature concentration: 7
    Unstable dependency: 0      Scattered functionality: 0
    Dense structure: 0
-Totals design smell instances detected-
    Imperative abstraction: 0      Multifaceted abstraction: 0
    Unnecessary abstraction: 3      Unutilized abstraction: 56
    Feature envy: 1      Deficient encapsulation: 1
    Unexploited encapsulation: 0      Broken modularization: 1
    Cyclically-dependent modularization: 0      Hub-like modularization: 0
    Insufficient modularization: 2      Broken hierarchy: 0
    Cyclic hierarchy: 0      Deep hierarchy: 0
    Missing hierarchy: 0      Multipath hierarchy: 0
    Rebellious hierarchy: 0      Wide hierarchy: 0
-Totals implementation smell instances detected-
    Abstract function call from constructor: 0      Complex conditional: 4
    Complex method: 12      Empty catch clause: 0
    Long identifier: 2      Long method: 1
    Long parameter list: 10      Long statement: 107
    Magic number: 57      Missing default: 3
-----
Done.

```

We refactored and removed the feature concentration smells, thus eliminating all the architecture smells. Among the design smells, we removed unnecessary abstraction and feature envy. We also removed complex conditional and long identifier smells.

Post-refactoring:

```

E:\Winter22\ASDC\assignment\assignment2\problem1>java -jar DesigniteJava.jar -i E:\Winter22\ASDC\wiseshopping -o wiseshopp
Searching classpath folders ...
Parsing the source code ...
Resolving symbols...
Computing metrics...
Detecting code smells...
Exporting analysis results...
wrapping up ...
--Analysis summary--
    Total LOC analyzed: 5547      Number of packages: 38
    Number of classes: 96      Number of methods: 421
-Total architecture smell instances detected-
    Cyclic dependency: 0      God component: 0
    Ambiguous interface: 0      Feature concentration: 0
    Unstable dependency: 0      Scattered functionality: 0
    Dense structure: 0
-Totals design smell instances detected-
    Imperative abstraction: 0      Multifaceted abstraction: 0
    Unnecessary abstraction: 0      Unutilized abstraction: 51
    Feature envy: 0      Deficient encapsulation: 1
    Unexploited encapsulation: 0      Broken modularization: 1
    Cyclically-dependent modularization: 0      Hub-like modularization: 0
    Insufficient modularization: 0      Broken hierarchy: 0
    Cyclic hierarchy: 0      Deep hierarchy: 0
    Missing hierarchy: 0      Multipath hierarchy: 0
    Rebellious hierarchy: 0      Wide hierarchy: 0
-Totals implementation smell instances detected-
    Abstract function call from constructor: 0      Complex conditional: 0
    Complex method: 12      Empty catch clause: 0
    Long identifier: 0      Long method: 0
    Long parameter list: 6      Long statement: 62
    Magic number: 76      Missing default: 3
-----
Done.

E:\Winter22\ASDC\assignment\assignment2\problem1>

```

Justification of existing smells:

Unutilized abstraction

This smell is a false positive because all the classes listed in the excel sheet generated by designate is used in the project

Broken Modularization and Deficient Encapsulation

This smell is a false positive too. This has arised due to the Constants class. This class contains all our constants that are used across the project multiple times. We cannot break down this class further as the constants declared in the class are related to each other. This class contains only declarations and no methods because this is a utility class, created for holding constants alone. We cannot restrict its access to protected and move it a separate package because these constants are used across multiple packages. Doing so will violate the principle of maintainability and reusability.

Complex Method

This smell has arised in the render () method of multiple screens. This cannot be avoided as a screen is responsible for multiple tasks and hence all these if statements are combined in a single method. We are, however, keeping in mind the principle of single responsibility while doing so.

Long statement

This smells arised due to using Mockito.when(scanner.next()).thenReturn(<some_input>).thenReturn(<some_input>) .thenReturn(<some_input>) .thenReturn(<some_input>)...

We need to do this in our integration test cases to mimic user behaviour and test our application

Missing default

This is a false positive. We have declared a default statement in all our 3 switch cases.

Long Parameter list

This is also a false positive. Constructor of user and methods related to register user do have more parameters because it has more fields.

Magic Number

This smell has arised because we define multiple product related attributes like price, quantity, etc in our test cases. We cannot avoid this because we need to create these objects for our test cases. All the smells tagged with magic number as this same reason.

In-details justification is mentioned in the excel sheet.

11.CI CD status

Following shows the ci cd pipeline.

Dev branch is our default branch

Merge branch 'RefactorToggleProduct' into 'dev'

Refactor long method in toggle product availability

See merge request !106

⌚ 4 jobs for dev in 6 minutes and 18 seconds (queued for 3 seconds)

→ d6dbbe9f 🏁

湓 1 related merge request: !108 Dev

Pipeline Needs Jobs 4 Tests 0

Code_quality

✓ code_quality

Test

✓ test

Build

✓ build

Develop

✓ develop

Main branch status:

Merge branch 'dev' into 'main'

Dev

See merge request !108

⌚ 3 jobs for main in 1 minute and 3 seconds (queued for 3 minutes and 5 seconds)

➡ latest

→ f5f9010c 🏁

湓 No related merge requests found.

Pipeline Needs Jobs 3 Tests 0

Test

✓ test

Build

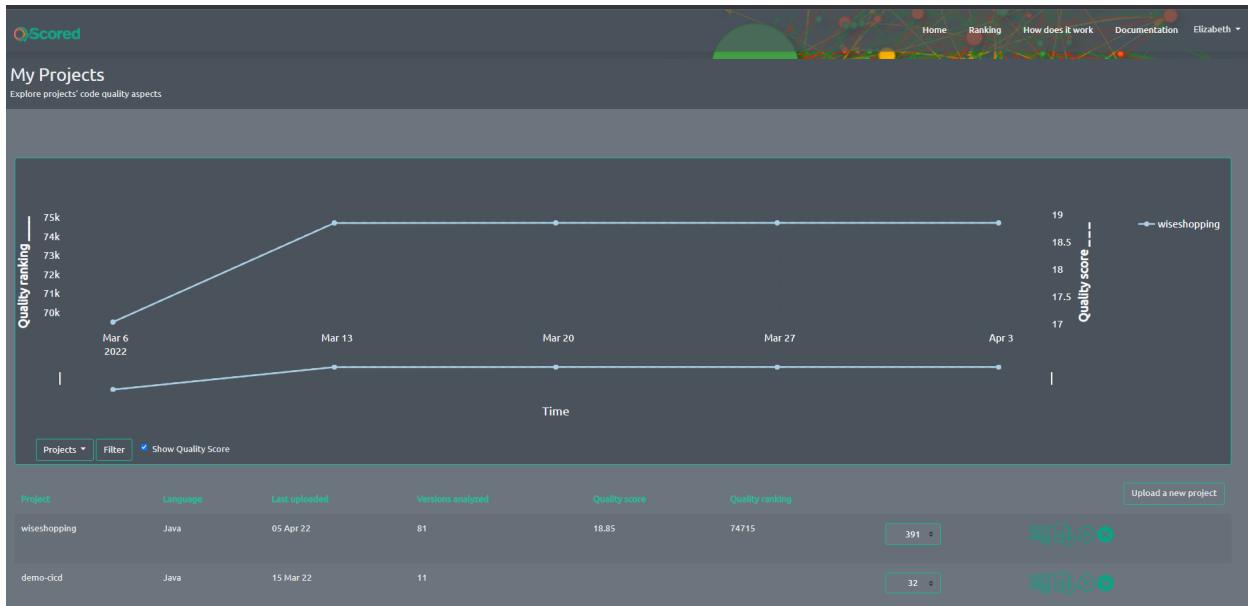
✓ build

Production

✓ production

Qscored status:

We had uploaded a demo dummy project called demo-cicd. We cannot filter it out of the graph. Please ignore that.



12. Limitations and future scope

Wiseshopping is built as a console-based application focused on the core functionalities. Some of the noteworthy features of Wiseshopping are to find the cheapest cart, share cart with other users, price comparison of the products with other merchants/store, best time to buy a product over a period of a year (lowest price analytics), demand trend of a product over a year (demand trend analytics). Our team worked on a proof-of-concept to implement the User Interface Design, however owing to time restrictions and our resources' limited expertise of UI technology, we were unable to create an interactive UI. But the console application developed provides easy navigation which provides user with seamless experience. The project's future goals include creating an interactive user interface and using analytics. If given the chance, we'd like to improve the efficiency of our analytics algorithms in order to gain more insights and make Wiseshopping a more trustworthy tool. On top of the analytics, a recommendation system might be constructed to offer ideas to the user.

13. Member Contribution

13.1. Features

Authentication Module

- User registration – Harsh Hariramani
- User login - Pavithra Gunasekaran

- User reset password – Adarsh Kannan
- Merchant registration – Elizabeth James
- Merchant login - Pavithra Gunasekaran
- Merchant reset password – Adarsh Kannan

Core Merchant Functionalities

- Add/Update/Delete store – Elizabeth James
- Add products – Harsh Hariramani
- Update/Delete products – Nilesh Gupta (80), Pavithra Gunasekaran (20)
- Maintain product inventory – Harsh Hariramani (50), Pavithra Gunasekaran (50)

Core User functionalities

- View products
 - Search by product tags - Pavithra Gunasekaran
 - Filter by merchant and/or store location zip code - Nilesh Gupta (80), Elizabeth James (20)
 - Search feature by merchant and/or store location and zip code – Adarsh Kannan
- Add products to cart – Elizabeth James
- View cart – Elizabeth James
- Share carts with other users – Adarsh Kannan
- Find the store selling the products for the cheapest price – Adarsh Kannan
- Find whether a product is available at any store in a given location - Pavithra Gunasekaran
- Add/Update details like Name, Address and so on. – Nilesh Gupta

Other features

- Compare prices of products sold by different seller – Harsh Hariramani (40%), Pavithra Gunasekaran (60%)
- Find the store that sells the in-cart products for the cheapest price – Analytics
- Find and cleaning data - Pavithra Gunasekaran
- User: Price trend analytics - Pavithra Gunasekaran
- Merchant: Demand Trend Analytics – Elizabeth James

Subscription

- User: subscribe for price alerts for a product – Elizabeth James
- Application should send mails when product price changes – Elizabeth James

UX:

- Authentication screens
 - Login – Elizabeth James
 - Registration – Elizabeth James

- Reset Password – Pavithra Gunasekaran
 - Logout – Elizabeth James
- User functionality screens
 - User menu screen – Elizabeth James
 - User profile screen – Harsh Hariramani
 - Products screen - Elizabeth James (70%), Pavithra Gunasekaran (30%)
 - Cart screen – Adarsh Kannan
 - Subscription screens – Elizabeth James
 - Price trend analytics - Pavithra Gunasekaran
- Merchant functionality screens
 - Merchant Menu Screen – Pavithra Gunasekaran (80%), Elizabeth James (20%)
 - Products Screen – Pavithra Gunasekaran (80%), Nilesh Gupta (20%)
 - Store screen – Elizabeth James
 - Demand trend analytics - Pavithra Gunasekaran
- Ease of navigation between screens – Elizabeth James

13.2. Other tasks

Testing - All

Ensuring Test Coverage - Pavithra Gunasekaran (50%), Elizabeth James (50%)

Refactoring - Elizabeth James (80%), Pavithra Gunasekaran (20%)

CI CD – Elizabeth James

Database set up – Nilesh Gupta (70%), Elizabeth James (30%)

Design pattern - Implemented to Factory pattern to render screens – Elizabeth James

Documentation – Adarsh Kannan (40%), Harsh Hariramani (40%), Pavithra Gunasekaran (20%)

PPT presentation -- Adarsh Kannan (50%), Harsh Hariramani (50%)

UI – Adarsh Kannan (40%), Harsh Hariramani (40%), Pavithra Gunasekaran (20%)

Login screen testing – Harsh Hariramani

Heroku Deployment – Nilesh Gupta

User menu screen tests – Nilesh Gupta