Software Requirements Specification

for

Superprice Application

Version 1.0 approved

Prepared by Group P7_Team_5

RMIT University

19/08/2023

Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
First Draft	9/8/23	First Iteration of the document, taking notes of things to write	1
intro added	13/8/23	Introduction added	2
description/ interfaces added	16/8/23	Added the description and external interfaces	3
Non functional req	18/8/23	Added non functional requirements	4
Added system features	19/8/23	Added the system features with supported wireframes	5
Final Draft	20/8/23	Added the Architectural diagrams and the spring plannings	6

1. Introduction

Purpose

The product whose software requirements are specified in this document is called Super Price, it is an application that is designed to help its users save money by comparing the grocery prices from various popular grocery stores and displaying the cheapest option to the user.

This SRS document will give a detailed view of the interfaces used and functional and non-functional requirements of SuperPrice. We will display the wireframes and the architectural diagrams that showcase our vision of the website.

We plan on describing the systems used to make the website functional and how they are interconnected, we will also summarize the hardware and software interfaces used.

Document Conventions

We used the font Arial with the font size of 11. The headings are written in Times New Roman with the font size 14 and are highlighted by using bold.

Intended Audience and Reading Suggestions

This document is intended for developers and testers. This SRS document contains the product description and functions, along with the user documentation early in the document.For the generic user, we have given the overall description that gives them a low level understanding of how the website works.

We also go in-depth for the more advanced user for whom this document is intended, we explain the external interface requirements that are used in making this project possible, we also give a list of the system features along with its functional and non-functional requirements. We provide a brief overview of the security features used by the website as well.
I would suggest that less experienced users just read the overall description part of this report and for the high-level users to read through the entirety of the document. If someone is just interested in the security, recommend them to read the other non-functional requirement section.

Product Scope

The purpose of the product is very simple, the product’s main objective is to search and compare pricing data from various grocery stores using a database that gets updated regularly, we believe that this website will help its users to save money by allowing them to compare and choose where to buy their groceries from

References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

2. Overall Description

Product Perspective

The SuperPrice application represents a paradigm shift in the modern grocery shopping experience. While it's a new and self-contained product, its development is inspired by the rise of digital shopping solutions and the need for real-time price comparisons.

The larger system consists of:

- Super Price App (Central Node)
- Supermarket API integrations
- Payment Gateways
- User Database
- Notification Services

Product Functions

- **Product Search and Categorization:** Allows users to easily find specific products or browse through different categories.
- **Price comparison:** Facilitates efficient and flexible organization of grocery deliveries.
- **Delivery Organization**: Facilitates efficient and flexible organization of grocery deliveries.
- **Notifications and Alerts**: Users are constantly in the loop with real-time notifications about their delivery order.
- **User friendly interface**: Designed to be intuitive and easy, ensuring users of all tech-savviness levels can navigate and operate the application without hitches.

User Classes and Characteristics

- **Casual shoppers:** Infrequent users, primarily looking for deals and may occasionally use the delivery service.
- **Regular shoppers:** Often use the app to compare prices and order groceries for delivery on a weekly basis.
- **Tech-savvy shoppers:** Users comfortable with advanced features, who may dive deeper into price analytics.

Operating Environment

The Superprice application will operate on any web browsing application with their latest version. The backend end will reside on MySQL with the latest shopping data.

Design and Implementation Constraints

- Real-time data requires robust server architecture.
- Secure payment gateways are essential, with encryption standards.
- Design should be compliant with accessibility standards.
- SuperPrice must adhere to data privacy laws, payment processing standards, and e-commerce regulations.

User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

Assumptions and Dependencies

- Assumes supermarkets will provide accurate and up-to-date APIs for product listings and prices.
- Assumes consistent internet connectivity for users to access real-time data.
- Depends on the continued operation and support of updated databases.

3. External Interface Requirements

User Interfaces

The primary interaction of users with the SuperPrice application will be through its graphical user interface (GUI).

- **Dashboard/Home Screen**: Here, users will be greeted with a dynamic mix of products on special offers, trending products, and category highlights. There'll be an emphasis on visual appeal with high-quality product images, brief descriptions, and clear pricing details.

- **Search Functionality**: An intuitive search bar will always be within reach, allowing users to quickly type in products they're interested in. As they type, auto-suggestions based on popular searches and available products will aid their search process

- **Product Listings**: When users browse or search for products, they're presented with a streamlined list. Each product card would show a clear image, the product name, its price at different stores, and a quick "Add to Cart" button. Detailed views will provide more in-depth information, including nutritional facts, expiry dates, and store-specific offers.

- **Shopping Cart & Checkout**: The shopping cart will allow users to review their product selection, see an aggregated total, choose their preferred store, and then proceed to a secure checkout.

- **Profile page:** Users will be able to see a profile with all their details such as name, address and payment methods used. This information would be stored in the database after they signed up, and can be edited using an edit button, which allows users to change the information in their profile.

Hardware Interfaces

The SuperPrice application will need to function across a variety of devices to ensure accessibility and usability for its target audience.

- **Web interface:** The application should also be accessible via a web portal for users who prefer shopping on their computers. This will require compatibility with common web browsers like Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge.

Software Interfaces

The SuperPrice application will interact with several software components:

- **Database Systems:** The application will need a robust and scalable database system to store product information, user details, prices from various stores, and transaction logs. MySQL will be used after consideration.
- **Operating Systems:** As the application will be available on multiple platforms. The application will be able to run on the latest web browsers.

Data Interaction:
Incoming Data:

Product searches by users
User registration and login details
Price updates from stores
Delivery requests
Outgoing Data:

Price comparison results
Notifications and alerts
Delivery confirmation and tracking details

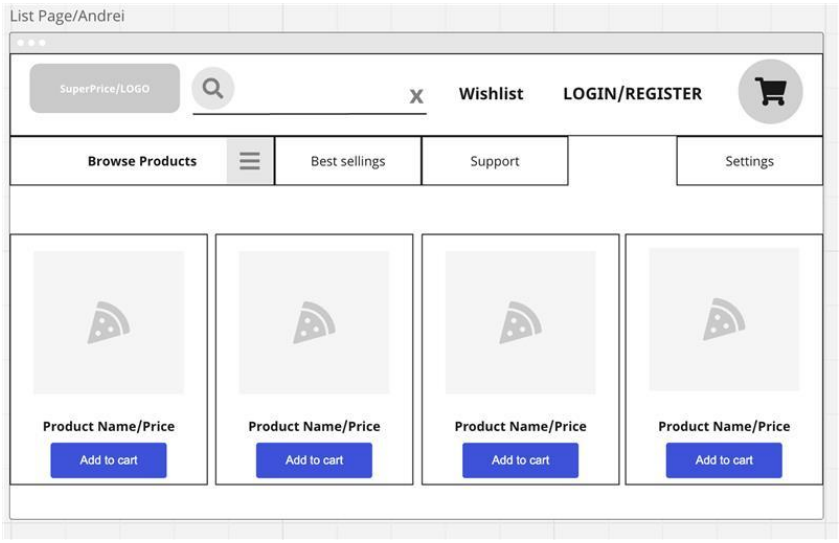
Communications Interfaces

Communication is key to the functionality and reliability of the SuperPrice application.

- **Web Browser:** Users should be able to access the platform through any modern web browser. The application should be designed following the latest web standards, ensuring compatibility with browsers like Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge.
- **Network Server Protocol:** Data transfer between the client and server should be secured using HTTPS protocols, ensuring data privacy and security.
- **Communication Standards:** FTP might be used for bulk data transfer, especially for updating large sets of product listings or prices. HTTP/HTTPS will be used for regular application access and data requests.
- **Synchronization Mechanisms:** Real-time synchronization is essential to keep product prices up-to-date. WebSockets or similar real-time communication protocols might be beneficial for this purpose.

4. System Features

System Feature 1 - Search Bar



4.1.1 Description and Priority

This is a basic but essential feature, the user will enter the name of the product that he wants and we will search for that product in our databases, we will then display the results in a tabular format. This is a high priority feature as without this feature

our website would not function as intended.

4.1.2 Stimulus/Response Sequences

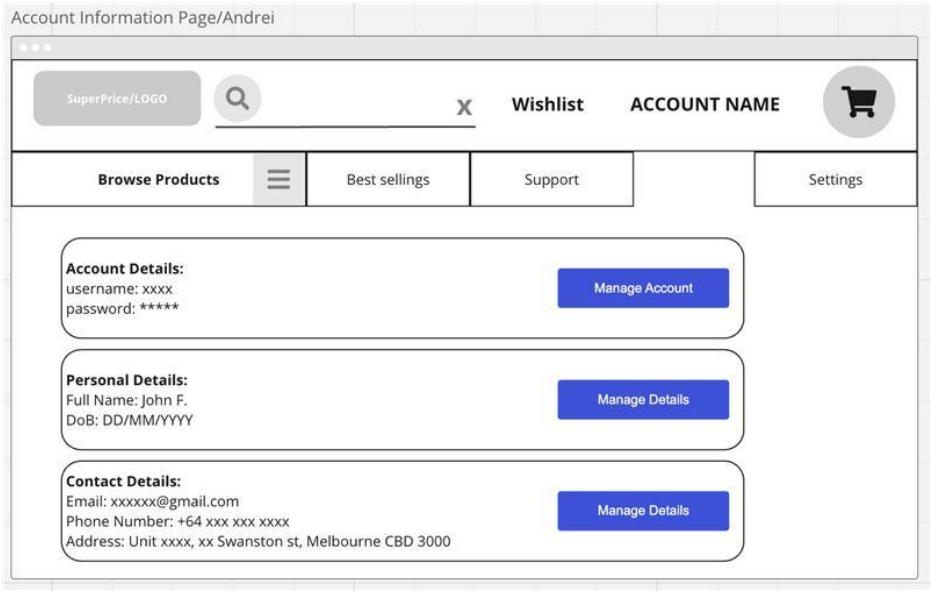
Action - The user types the name of the product in the search bar.
Response - The system displays a dropdown menu of the options who may match the users search

Action - The user selects a product.
Response - The system the product on a separate page with its details.

4.1.3 Functional Requirements

REQ-1: It should show search history.
REQ-2: It should recommend items as the user is typing.
REQ-3: It should be able to handle special characters and handle errors

System Feature 2 - User Profile



4.2.1 Description and Priority:

This feature will allow the user to create a profile that can store his address and preferred delivery method. This will allow the user to easily change addresses and delivery methods by switching to a different profile.

4.2.2 Stimulus/Response Sequences:

Action - The user clicks on his user profile.
Response - The system opens his profile page and displays the important details of the user.

Actions - The user clicks on the edit profile button.
Response - The system opens a window that allows the user to re-enter his profile details.

Action - The user clicks on log-out.
Response - The system gives a prompt to the user to confirm.

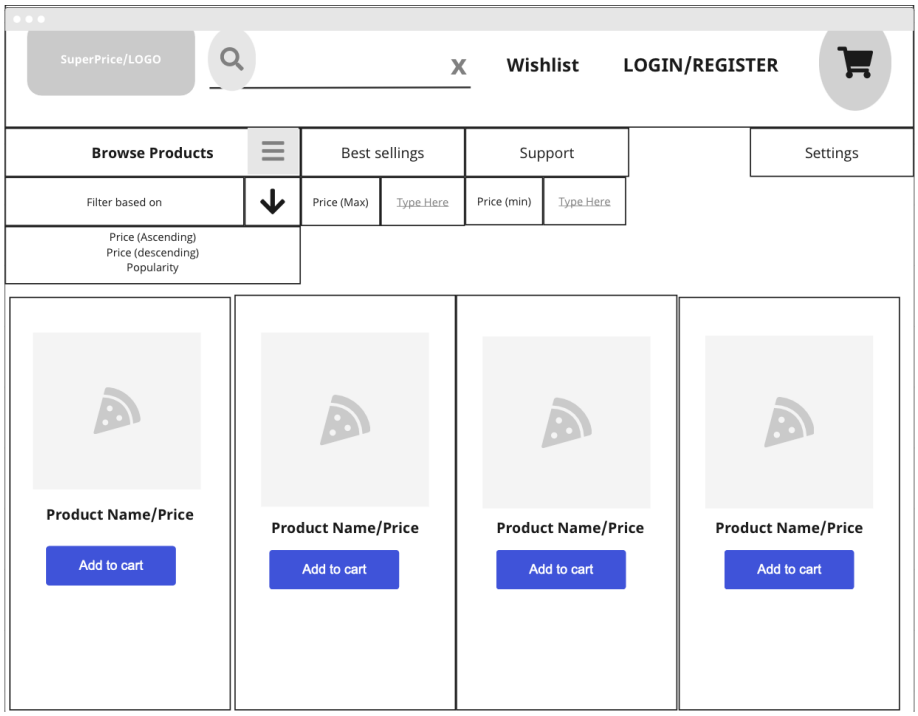
4.2.3 Functional Requirements:

REQ-1: The feature should be easy to read

REQ-2: The feature must be able to store multiple profiles

REQ-3: The feature must allow the user to easily delete/create new profiles.

System Feature 3 - Filter System



4.3.1 Description and Priority:

This feature will allow the user to filter the results of his search to match their preferences. Users will be able to select from fields such as Name, Price, and Popularity. This is a medium priority feature. This feature is not essential but is very useful regardless.

4.3.2 Stimulus/Response Sequences:

Action - The user clicks on the filter option.
Reaction - The system displays a drop down menu of the various filters.

Action - The user selects a filter.
Reaction - The system then chooses the items in the database according to the filters chosen.

Action - The user clicks to close the filter system.
Reaction - The system closes the drop-down menu and displays results to the user.

4.2.3 Functional Requirements:

REQ-1: The filters must be easy to use and understand.

REQ-2: The filters must not take too much time to filter the product.

System Feature 4 - Sort Products

4.4.1 Description and Priority:

This feature will allow users to sort their searches based on the information of the products. Users will be able to sort alphabetically by name, low to high by price, and vice versa. Furthermore, users may also be able to sort by the product's popularity/review score. This feature will be used in tandem with the filtering system to provide users with a granular searching system that allows them to find the best bargains according to their preferences. This feature will have medium priority.

4.4.2 Stimulus/Response Sequences:

Action - The user searches for a product they want to purchase.
Reaction - The search comes back successfully.

Action - The user clicks on the sort button.
Reaction - The menu displays a dropdown menu showing checkboxes.

Action - The user selects the sort they prefer and clicks confirm.
Reaction - The website reorders the products according to the sort option.

4.4.3 Functional Requirements:

REQ-1: The sort button should open the drop down menu.

REQ-2: The search is sorted according to the user input

5. Other Nonfunctional Requirements

Performance Requirements

Responsive User Interface: The application is quick and responsive, allowing users to easily navigate the software interface without delay; the button should activate the next page within half a second. This will increase user satisfaction while navigating the application.

Search speed:
Users are able to quickly search for their desired product within one second, even with a large database of products. This will help avoid user frustration.

Reliability:
The application is expected to work 99.9% of the time, so that users who rely on it will be assured that the application will never crash.

Notifications:
The application should send notifications of product recommendations, hot bargains, and order confirmations after their triggering event without delay, so that users can be assured that the application is working. These notifications should arrive to the user within 5 seconds of doing an action such as placing an order.

Safety Requirements

- User data shall always be encrypted and never be stored on the database as plain text. All sensitive data such as users' payment information should always be encrypted to ensure user privacy.
- Before placing an order or making a purchase, users should always be asked to confirm their actions to avoid placing accidental or unauthorized transactions. The application may ask the user to confirm the identity with 2 factor authentication before placing their order.

Security Requirements

- User profiles are protected by strong password requirements and users can also opt to turn on 2 factor authentication to further increase the security on their account. Users may be asked to enter their password to place an order or edit their profiles.
- All communication between the front end and back end are encrypted, as well as any sensitive user information that is stored, by encryption algorithms that are industry-standard.

Software Quality Attributes

Availability:
The application must have a system uptime of at least 99.9%, ensuring that it is available and accessible to users almost all the time.

Usability:

The application's user interface should achieve a usability score of at least 85% in user testing evaluations, indicating high user satisfaction and ease of use. These user testing evaluations will be detailed in the test cases accompanying the user stories.

Testability:
The application can be easily tested for its features, allowing developers to quickly identify bugs within certain features and fix them accordingly.

Business Rules

- Only users that are registered with the application and authenticated can place orders and review their own profile information.
- The product prices and availability must be accurate and up-to-date so that the price comparison feature can work effectively.
- Only users that have bought the product can leave a review for it, ensuring that the reviews for a product are reliable and trustworthy.

6. Other Requirements

USER STORIES AND TEST CASES

User story #1:

As a budget-conscious user,
I want to set up price drop alerts for my most frequently purchased or favorite items,
So that I am instantly notified when there's a better deal available.

Test category: Proactive Price Monitoring and Alerts

Test Category ID	1
Test Case ID	1.3
Test Case Summary	Allow users to set up price drop alerts for their most frequently purchased or favorite items
Related Requirement	Website instantly notified users when there's a better deal available, enabling them to save money
Prerequisites	1. Users have access to internet 2. Users is familiar with website 3. Users have a compatible device
Test Procedure	1. User access to website 2. Click on search bar and type keywords 3. Choose desired items and tick the ‘Alerts’

	box 4. After that, user will get notification when that particular item's price dropped
Test Data	Key words for items' search, Different stores' items, Website
Expected Result	Website will take into consideration whether there is a price drop or bundle deal for that particular items, then will notify users immediately through mails/push-noti/etc.
Actual Result	Website will take into consideration whether there is a price drop or bundle deal for that particular items, then will notify users immediately through mails/push-noti/etc.
Created By	Xuan Tuan Kiet Trinh
Date of Creation	19 Aug 23
Status	Pass

User story #2:

As an old person who has to take different medicines and re-order.
I want a system that remembers my order history.
and helps me to re-order more medicines when I am running out and get it delivered to my doorsteps.

Test category: Re-ordering

Test Category ID	2
Test Case ID	2.2
Test Case Summary	Allows the user to re-order from the order history of the user.
Related Requirement	
Prerequisites	1. The user has a pre-existing account on the website. 2. The user has a pre-existing shopping history on the website. 3. The user has a compatible device, strong internet connection and is comfortable with web-based applications.

Test Procedure	1. Log onto the website. 2. Log into the user account on the website. 3. Click on the user account option > order history. 4. Click on the “add to cart” button on the order
Test Data	Order history
Expected Result	The user is able to access his/her history and use it to re-order that particular order.
Actual result	The user was successful in re-ordering using his/her order history.
Created By	Vishwas Aggarwal.
Date of Creation	19 August 2023
Status	Pass

User story #3:

As a busy salaryman,
I want the website to have the option of either delivery or takeaway,
So that I can either collect my products on my own or have them delivered to my doorstep.

Test category: Selecting product collection method

Test Category ID	2
Test Case ID	2.4
Test Case Summary	Allows the user to select a collection method at the time of checkout.
Related Requirement	
Prerequisites	1. A compatible system, good internet connection and prior information of using shopping websites. 2. The user has already added the desired product to the cart.

Test Procedure	1. On the cart page the user clicks on the “buy now” button. 2. On the check-out page the user chooses a collection method (takeaway or delivery) 3. If the user selects the ‘take-away’ option, the user selects the nearest store(based on their post-code) and proceeds to the payment page. Alternatively, if the user selects the ‘delivery’ option the user proceeds to the address page to enter their address and the proceeds to the payment page.
Test Data	1. Product the user wants to buy. 2. User address. 3. User postcode.
Expected Result	The user is able to select a collection option.
Actual result	The user was successfully able to select a collection method.
Created By	Vishwas Aggarwal.
Date of Creation	19 August 2023
Status	Pass

User story #4:

As a budget-conscious shopper,
I want to easily compare prices of a specific grocery item across various supermarkets in my area,
So that I can purchase the product from the store offering the lowest price, saving money on my shopping.

Test category: Price comparison.

Test Category ID	3
Test Case ID	3.1
Test Case Summary	Allows user to compare the price of a particular product in different supermarkets.
Related Requirement	Filtering and sorting results

Prerequisites	1. The user is comfortable with web-based applications. 2. The user has a compatible device, good internet connection.
Test Procedure	1. The user enters the website by clicking the website link on their browser. 2. Search for the desired product. 3. Select compare button. 4. See the different prices available for the selected item.
Test Data	1. Keyword for searched item.
Expected Result	The website displays the different prices at which the selected product is being sold in different supermarkets.
Actual result	The website is successful to display the different prices at which the selected product is being sold in different supermarkets.
Created By	Ahnaf Tausif.
Date of Creation	19 August 2023
Status	Pass

User story #5:

As a student,
I want to update and edit my profile information at any time,
so that I can keep my preferences and interests up to date.

Test category: Profile information change.

Test Category ID	3
Test Case ID	3.2
Test Case Summary	Allows users to change/update their profile information.
Related Requirement	The user must be able to access their profile settings to change information.

Prerequisites	1. The user is comfortable with web-based applications. 2. The user has a compatible device, good internet connection. 3. The user already has an active account.
Test Procedure	1. The user enters the website by clicking the website link on their browser. 2. User clicks on settings. 3. The user clicks on the profile option. 4. User selects the “change” button next to the information that he/she wants to change. 5. The user saves the changes once satisfied by clicking the ‘save’ button.
Test Data	1. The user’s login credentials.
Expected Result	The information on the user’s profile changes to what the user desired.
Actual result	The website is successful to change the user’s profile information.
Created By	Ahnaf Tausif.
Date of Creation	19 August 2023
Status	Pass

User story #6:

As a senior citizen,
I want the SuperPrice app to have a user-friendly interface with clear and easily readable product information,
so that I can comfortably browse and shop for groceries online.

Test category: Menu browse.

Test Category ID	3
Test Case ID	3.3
Test Case Summary	Allows users to scroll through the homepage and select an item of their choosing.

Related Requirement	Users must be able to use the basic functionalities of a website.
Prerequisites	1. The user is comfortable with web-based applications. 2. The user has a compatible device, good internet connection
Test Procedure	1. The user opens the website by clicking the link on their browser. 2. The user scrolls through the main menu. 3. The user selects an item of his/her choosing.
Test Data	Any item available on the homepage.
Expected Result	The user views an item from the homepage without logging in.
Actual result	The user is able to select and view an item from the homepage.
Created By	Ahnaf Tausif.
Date of Creation	19 August 2023
Status	Pass

User story #7:

As a busy working professional,
I want to organize the delivery of my groceries through the SuperPrice application at a convenient time,
So that I can have my shopping done without needing to visit the store, and receive my items at a time that fits my schedule.

Test category: Delivery time setting

Test Category ID	3
Test Case ID	3.4
Test Case Summary	Allows users to preset the time at which they want to have item(s) delivered.
Related Requirement	Users must have the list of items added to cart and checked out.
Prerequisites	1. The user is comfortable with web-based applications. 2. The user has a compatible device, good internet connection. 3. The user must already have an account. 4. The user must have a few items added to cart that he/she preset.

	5. The user needs to have sufficient balance in his/her account. 6. The user is logged in.
Test Procedure	1. The user enters the website by clicking the link on their browser. 2. The user selects the setting button. 3. The user selects the “set delivery time and day” button. 4. The user then selects the day and time and saves it by pressing the save button.
Test Data	Time and date
Expected Result	The user presets the date and time in which he/she wants the items to be delivered.
Actual result	The user is able to set a date and time to have the selected items delivered.
Created By	Ahnaf Tausif.
Date of Creation	19 August 2023
Status	Pass

User story #8:

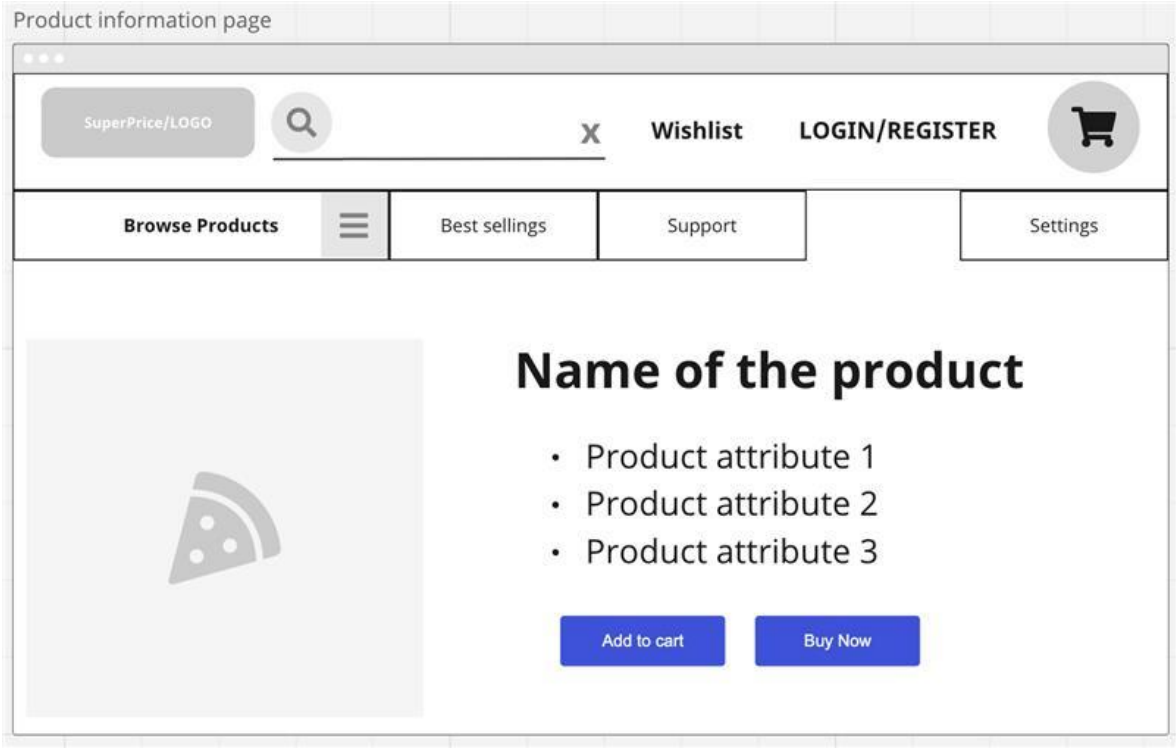
As a savvy and frugal shopper,
I want to be able to easily refine and organize my search results on the website,
so I can quickly identify the best product options that match my preferences.

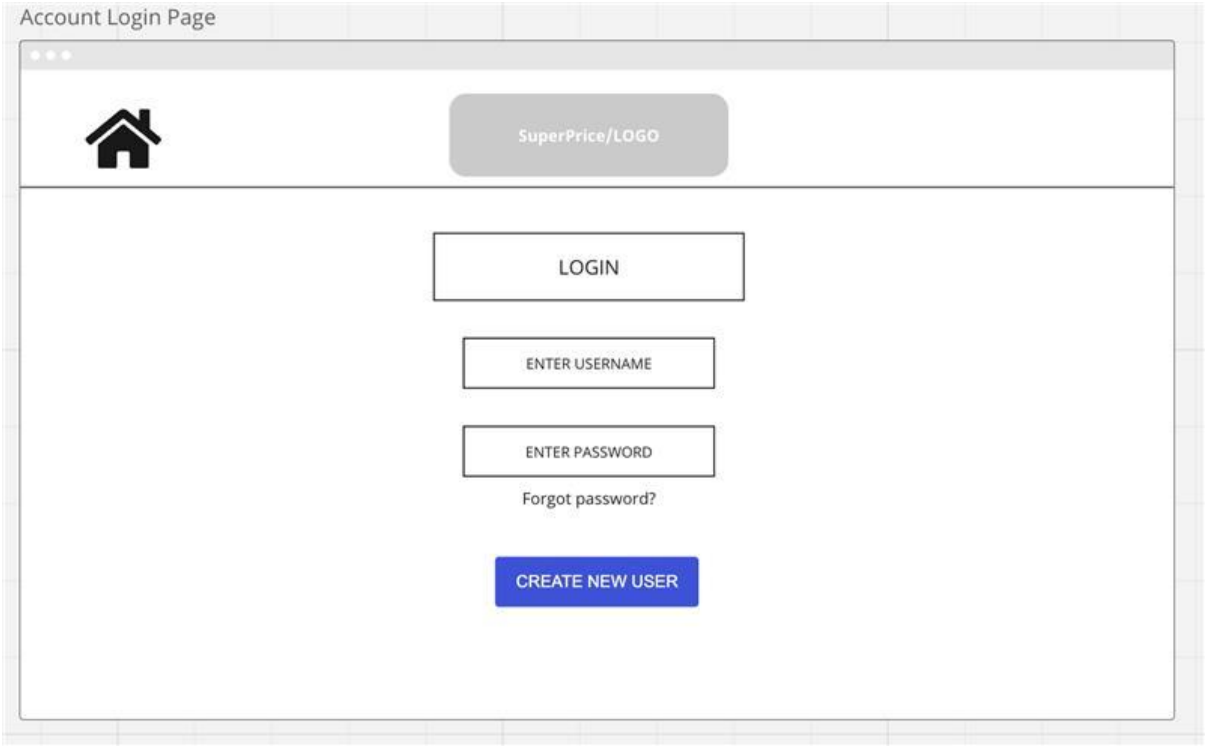
Test category: Filtered search

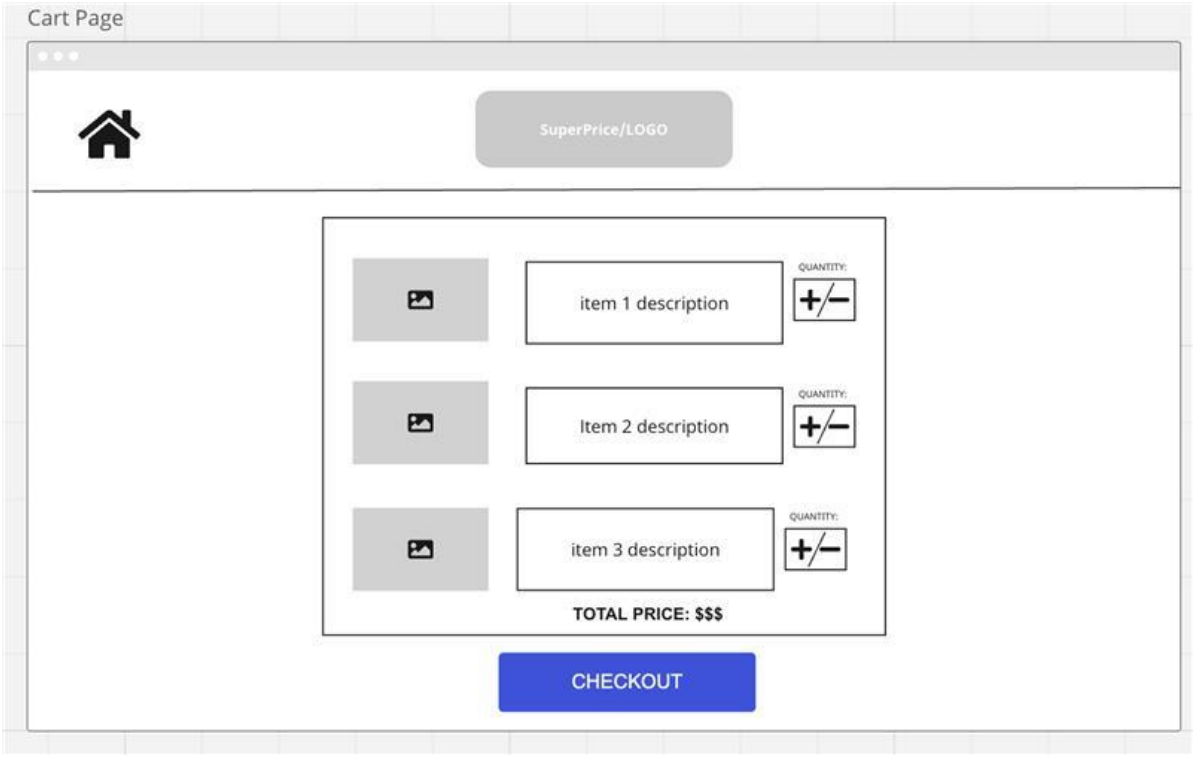
Test Category ID	3
Test Case ID	3.5
Test Case Summary	Allows users to filter searched results.
Related Requirement	Users must be able to use the basic functionalities of a website.
Prerequisites	1. The user is comfortable with web-based applications. 2. The user has a compatible device, good internet connection.

Test Procedure	1. The user enters the website by clicking on the website link on their browser. 2. The user then clicks on the search bar and inputs the keyword for the item they want to search. 3. The user then clicks on the filter button and sets the appropriate filters.
Test Data	Keyword for search and filter options.
Expected Result	The user sees the search result according to their filters.
Actual result	The system filters the search result and shows the findings to the user.
Created By	Ahnaf Tausif.
Date of Creation	19 August 2023
Status	Pass

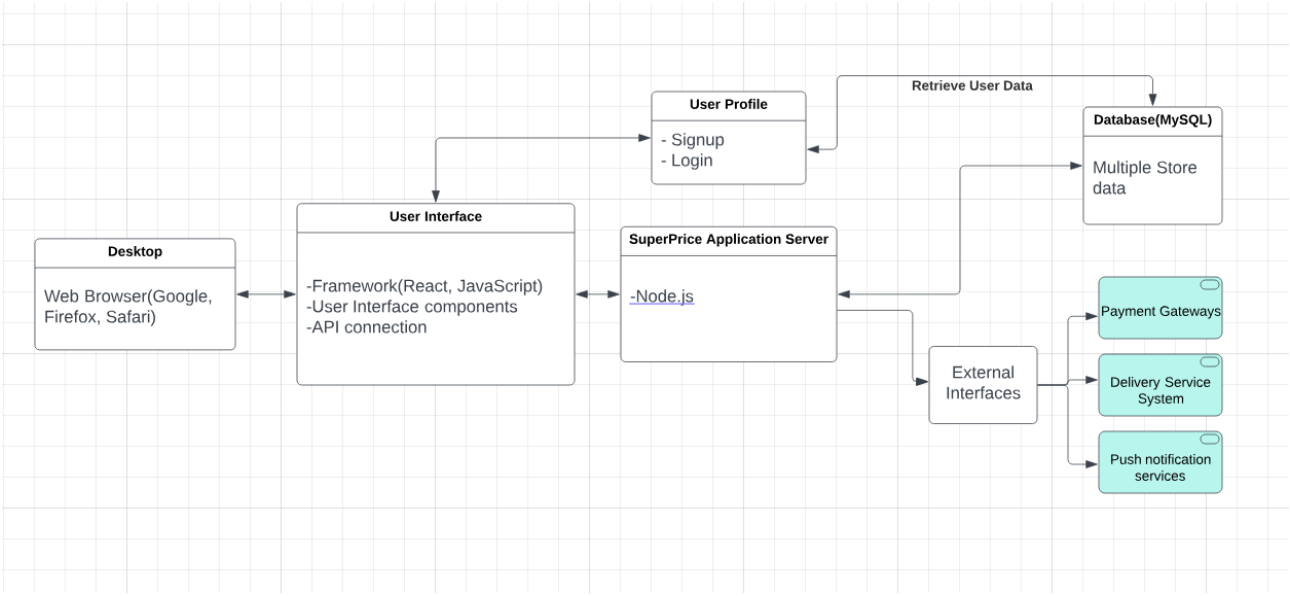
WIREFRAMES:




Account Login Page
SuperPrice/LOGO
LOGIN
ENTER USERNAME
ENTER PASSWORD
Forgot password?
CREATE NEW USER


Cart Page
SuperPrice/LOGO
item 1 description
QUANTITY:
+/−
Item 2 description
QUANTITY:
+/−
item 3 description
QUANTITY:
+/−
TOTAL PRICE: $$$
CHECKOUT

ARCHITECTURAL DIAGRAM



PRODUCT BACKLOG

SPRINT 0 RETRO

Team: 5

Date: 9th August

Attended:

Scrum Master: Kushagra Baghel
Product Owner: Jyoti Kundu
Development team:
Kushagra Baghel
Ahnaf Tausif
Vishwas Aggarwal
Lakindu Dissanayake
Xuan Tuan Kiet Trinh
Huynh Pham

1. Things That Went Well

We had good team communication, and the planning of the sprint was done without many issues.

2. Things That Could Have Gone Better

We needed to have a better meeting schedule as during our meeting times, some of the members were absent as they had to go to work.

3. Things That Surprised Us

Since all of us were used to using VS code, the shift to IntelliJ was a bit tricky as its setup had a few things which we had not encountered such as the issues with GitHub pushing and pulling which we solved by creating tokens.

4. Lessons Learned

We learned that we needed to communicate regularly to have a better understanding of what was being done in the project. Furthermore, we learned that planning ahead for a particular task helped cut down the total time to do the task by a lot.

5. Final Thoughts

We will be improving our meeting schedule timing along. Furthermore, we will be having regular meets to keep up with what everyone has done till that point.

SPRINT PLANNING

Team: Team_P7_G5

Sprint: 1
Date: 13/08/2023

Attended: Everyone
Scrum Master: Kushagra Baghel
Product Owner: Jyoti Kundu
Development team:
Vishwas Aggarwal
Xuan Tuan Kiet Trinh
Ahnaf Tausif
Huynh Pham
Lakindu Bandara Dissanayake
Kushagra Baghel

1. Goal

The aim of this sprint is to set the foundation for the development and fundamentals for our application, in our application users will be able to shop for products by checking their prices on different outlets simultaneously, which would help save time and efforts for our users.

2. Duration of the sprint

2 weeks

3. What is the team's vision for this sprint?

The team has decided to add the following features:

Creating a user profile for a new user and a sign in feature for an existing user.

A product information page which allows users to view the product details.

A list page where the user can see all the products with their prices and a Cart page where the user can see all the products that were selected for checkout.

At the end of sprint 1 the web application will be functional with basic features such as creating/signing to an account, price comparison functionality for viewing products and checking out.

4. Estimation in story points

User Story	Story Points	Reasoning
1 – Price comparison shopping	5	We need to establish a database, define API endpoints, and address architectural challenges.
2 – Editing and updating on profile details	3	As previously mentioned, we're storing user data, but since it's supplementary, it will require minimal effort.

4 – Easy to use interface	1	This requires smart web design and the use of conventional web page design language.
5 – Shopping history and reordering	2	It will just be a display of the fetched data from the database.
10 – Product search	2	Same as above
13 – Filtering and sorting results	3	While filtering and sorting are common features, their implementation can vary in complexity based on criteria and data structure.

Appendix A: Glossary

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

Appendix B: Analysis Models

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

Appendix C: To Be Determined List

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