Software Requirements Specification

for

SuperPrice

Version 1.1 approved

Prepared by P1-03

RMIT

11/08/2023

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

3. External Interface Requirements 3

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

4. System Features 4

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

5. Other Nonfunctional Requirements 4

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

6. Other Requirements 5

Appendix A: Glossary 5

Appendix B: Analysis Models 5

Appendix C: To Be Determined List 6

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Updated architecture diagram | 17/9 | To better represent what the program design architecture compared to before | 1.1 |
| Updated database engine | 17/9 | Now using H2 instead of mySQL for our databases | 1.1 |

# Introduction

## Purpose

The objective of the SuperPrice project is to develop a user-friendly website for listing and comparing the prices of products between brands and supermarkets. The platform will offer product search and filtering functions, which will allow users to conveniently locate products by name, category, brand and other criteria. Users will also be able place orders for delivery directly through the site. This will be achieved with a cart system, where Users will be able to add products to their order while they navigate through the application. The interface will provide clear information about specific products which includes their images, descriptions, reviews and any deals. It will also allow users to compare the price of the product across different supermarkets. Users will also have the option to create an account for the app. Registered users will be able to write and edit reviews, view their purchase and reviews history and save their delivery details. The design of the website will prioritize responsiveness, accessibility, and security to optimise the user experience. Overall, the project aims to create an efficient platform for informed product exploration, comparison and discovery.

## Document Conventions

The following typographical conventions were followed:

* Sections: 18 Times Bold
* Subsections: 14 Times Bold
* Descriptions: 11 Arial Italic
* Links: 11 Arial Italic Underlined Blue

## Intended Audience and Reading Suggestions

The intended audience for this Software Requirement Specification (SRS) document includes stakeholders such as developers, testers, product owners and scrum masters. This document details information about the project’s functional and non-functional requirements, use cases, database schema, technical specifications, system architectures and interface design. The development team will use this document to understand the functionalities and scope of application through the listed functional and non-functional requirements and use case diagram. They will also need the document to understand how they should structure the project according to the outlined system architecture. Front end developers in particular will use interface designs to recognize the aesthetic and the user experiences the project aims to provide. Whereas backend developers will use the ER diagrams to understand how their relational database should be structured. Testers of different backgrounds will use this document to identify security vulnerabilities, edge cases which can cause unintended behaviors and what changes are required to improve the user experience. Product owners will use this document to determine what tasks need to be registered into the product backlog and ensure that the project will truly meet the needs of its intended audience. The Scrum master uses this document to ensure that Functional requirements are prioritized first and prevent scope creep – where many features are introduced at once.

## Product Scope

*The vision of SuperPrice is to provide a platform where users can compare the prices of products across a wide selection of supermarkets, be well informed about new deals and specials and them to schedule the delivery of their orders during convenient time slots. SuperPrice will work towards this vision by providing a clean and intuitive interface design which will help users to achieve their goals as efficiently as possible, with the least amount of input. The user experience will allow users to effortlessly locate products through minimal searching and filtering. A practical way of comparing prices between several supermarkets. Furthermore, the user experience will incorporate real-time pop-up notifications of new deals and offers, which the user can effortlessly click to be taken to the discounted product’s page. Lastly, the user experience will also include a cart system where users can add to at any time as they explore the wide range of products the platform has to offer. When checking out, users can also select a preferred time slot for when they would like the order to be delivered. This ensures that our customers will always be available to receive their order at their doorstep!*

## References

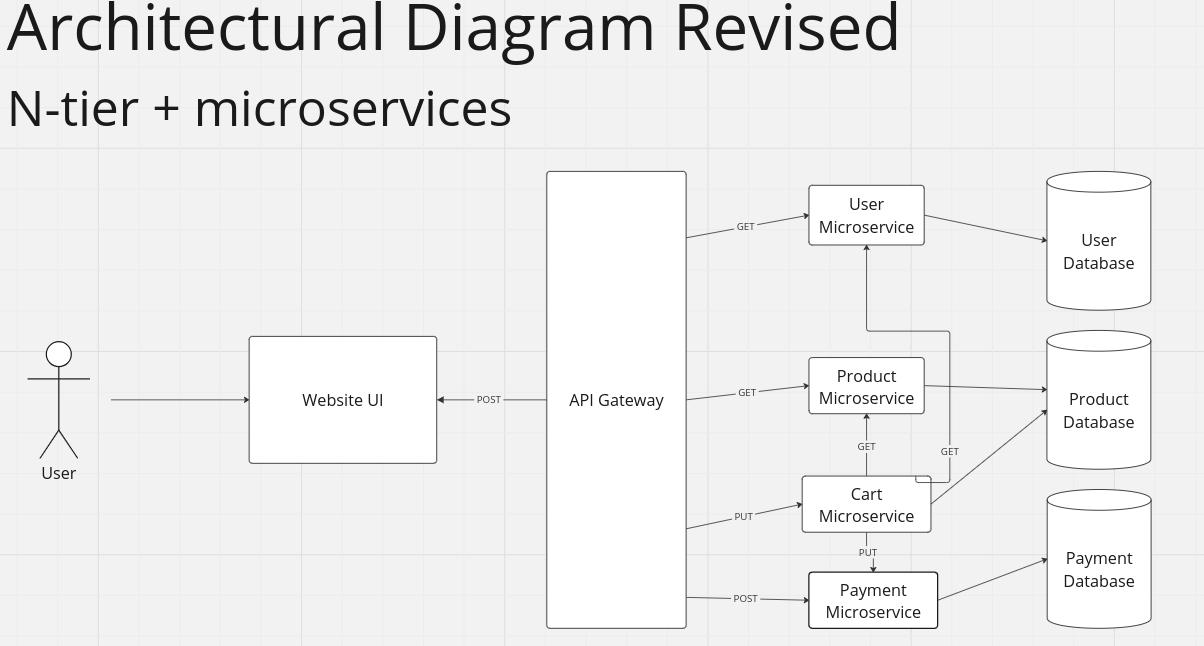
<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference,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

# Overall Description

## Product Perspective

The SuperPrice application is a new web application that aims to provide users with a user-friendly platform that enables them to find the lowest prices for groceries and facilitates efficient delivery arrangements. By streamlining the shopping process, the SuperPrice application saves time and money while enhancing the overall shopping experience for consumers.

## Product Functions



The SuperPrice project should provide many capabilities and functionalities. Such capabilities and functionalities consist of:

* Item Search
* Item Filter (by Brand, Category, Supermarket, etc.)
* *Sort Item (by Name, Price, etc.)*
* Add items to cart
* *Remove items from cart*
* *View cart*
* *Register for an account*
* *Login into an account*
* *Logout from an account*
* Write Reviews
* *Edit Reviews*
* *View Review History*
* Save delivery details to account
* *Online payment for orders*
* *Checkout an order for delivery*
* *Select a time slot for delivery to arrive*
* *View purchase history*
* *Receive notifications about new deals and offers*

## User Classes and Characteristics

One anticipated subset of users using this application are those with little experience with technology. They will likely by using limited functionality such as the search by name, add, remove, view cart, and checkout functionality. These users will likely make up a large portion of our user base and it is of best importance that the platform’s interface is best optimise for the user experience.

Inversely, a small subset of this platform’s users will be highly advanced with technology. They will likely be using the application’s more advanced capabilities, which includes filtering and sorting products. Since these users only make up a small portion of the entire user base, it might be more beneficial to hide away complex functionalities to keep the interface clean to ensure less technical users can have a good experience. However, this will decrease the experience of the more technical users as they will need to do more button presses to achieve their goals.

Another subset of users includes those who are new to the website. New users may not be familiar with the website's complete functionality and thus the website should apply standard user interface design choices, such as an intuitive user interface, easy navigation through the pages and a consistent navigation bar. New users are a very important user class as they could become returning customers if they like using the website, so it is important to have the new user experience as smooth as possible.

Users who have already been on the website and made an account should already be familiar with the website so there should be no significant changes to the website once it’s made. These users will vary in technical expertise however it is likely that they will use the website’s core functionalities such as searching, filtering and purchasing items. It is important to satisfy these customers through methods such as deals and the ability to leave reviews to ensure that they stay as a customer.

## Operating Environment

The software application is required to operate in a web browser environment. Web browsers with full support should include the latest versions of Google Chrome, Firefox, Microsoft Edge and Safari. The software solution is not expected to operate in deprecated web browsers such as Internet Explorer. The software system is required to work on a large variety of hardware platforms. It should be able to adapt to the different screen sizes and performance requirements of phones, laptops and desktops. Lastly, the software product should be able to run on the most recent versions of popular operating systems such as Windows, Mac and Linux.

## Design and Implementation Constraints

Several constraints that may be felt when developing SuperPrice could include:

* The inflexibility of relational databases where making changes can be more difficult (compared to document-based databases)
* *Knowledge of how to use the tools required to develop the project. Many members of the development teams have few or no experience using technologies such as React, H2 databases and Spring Boot*
* *Experience. The development team has near to no experience developing real world software solutions. Therefore, they don’t have knowledge of common practices which could reduce confusion and save time.*
* *Security. The development team has no experience sanitizing malicious user inputs.*
* *Time. Having more time for development could meaning a higher quality product*
* *Financial. No money means, people cannot be paid to test the product, which means less people providing feedback*

## User Documentation

Tutorials and resources to learn the technologies relevant to the development of the SuperPrice platform includes the following:

Front end:

* JavaScript: <https://www.w3schools.com/js/>
* *TypeScript:* [*https://www.typescriptlang.org/*](https://www.typescriptlang.org/)
* *HTML:* [*https://www.codecademy.com/learn/learn-html*](https://www.codecademy.com/learn/learn-html)
* *CSS:* [*https://www.w3schools.com/css/*](https://www.w3schools.com/css/)
* *PostCSS:* [*https://postcss.org/*](https://postcss.org/)
* node.js: <https://nodejs.org/docs/latest-v20.x/api/>
* *Next.js:* [*https://nextjs.org/docs*](https://nextjs.org/docs)
* react: <https://legacy.reactjs.org/docs/getting-started.html>
* *tailwind:* [*https://tailwindcss.com/docs/installation*](https://tailwindcss.com/docs/installation)
* *Material UI:* [*https://mui.com/*](https://mui.com/)

*Back end:*

* *Java:* [*https://www.w3schools.com/java/*](https://www.w3schools.com/java/)
* *Maven:* [*https://maven.apache.org/index.html*](https://maven.apache.org/index.html)
* *SQL:* [*https://www.w3schools.com/java/*](https://www.w3schools.com/java/)
* *H2 Database Engine:* [*https://h2database.com/html/main.html*](https://h2database.com/html/main.html)
* *Spring Boot:* [*https://spring.io/projects/spring-boot*](https://spring.io/projects/spring-boot)

## Assumptions and Dependencies

The assumptions that were made while developing this document include:

* User reviews will not be for the general product but will be for the product from specific supermarket chains (not specific store locations). This is because the price of the product could influence the review.
* *Since this is not a solution for the real world, we are not expected to consider the following:*
  + *Refunds and Cancelation of Orders (How does the user ask for an order to be refunded?)*
  + *Missing stock (How do we refund a portion of the order?)*
  + *Orders and Items lost in delivery (Again, how do we refund a portion of the order?)*
  + *Complaints about our services (Where will customers be able to file their complaints?)*
  + *Email verification when registering for an account (Not part of the scope)*
  + *Swearing/Censorship in reviews (Not part of the scope)*
  + *Inappropriate usernames and profile images (Not part of the scope)*
* *The data schema in the ER Diagram in Appendix B truly reflects what the data looks like in the real world.*
* *Users will not have their payment details saved to their account, due to security concerns (Someone else on the already logged in account making purchases)*

*The dependencies the project is expected to have after development include:*

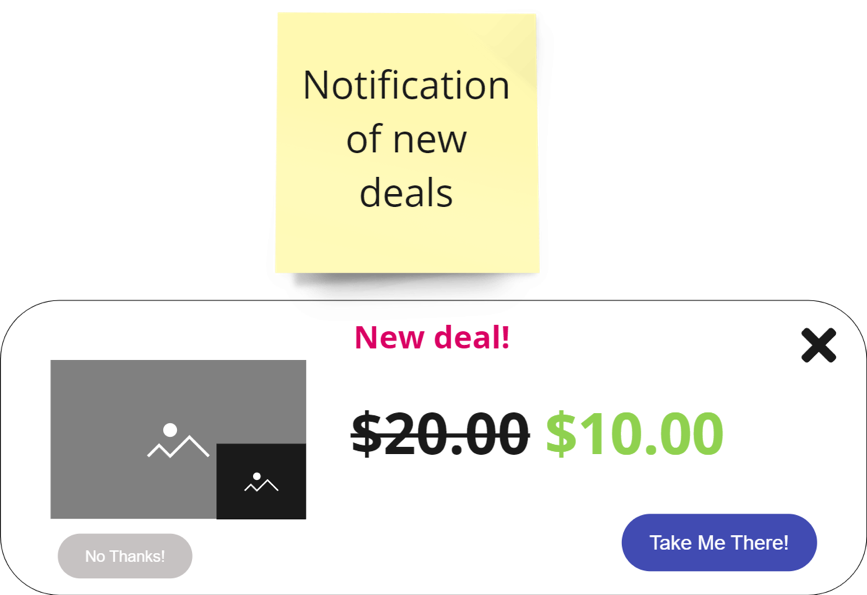
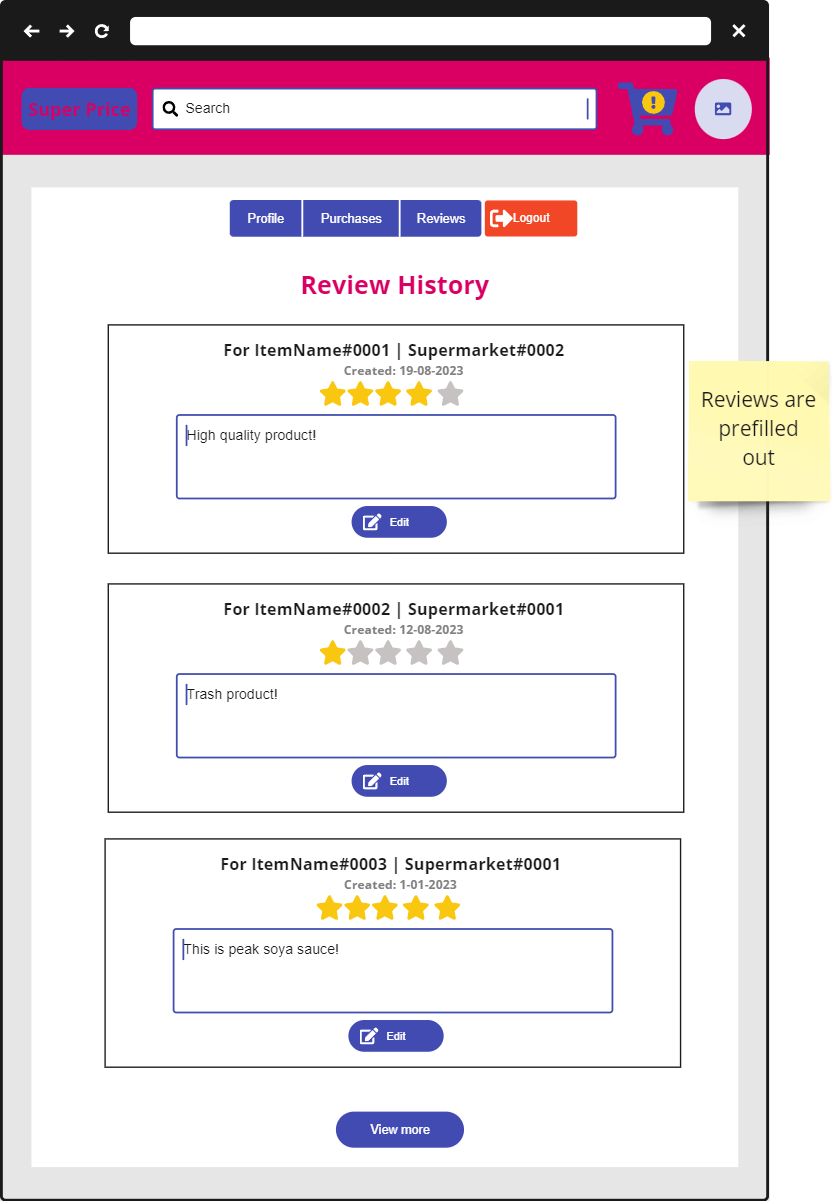
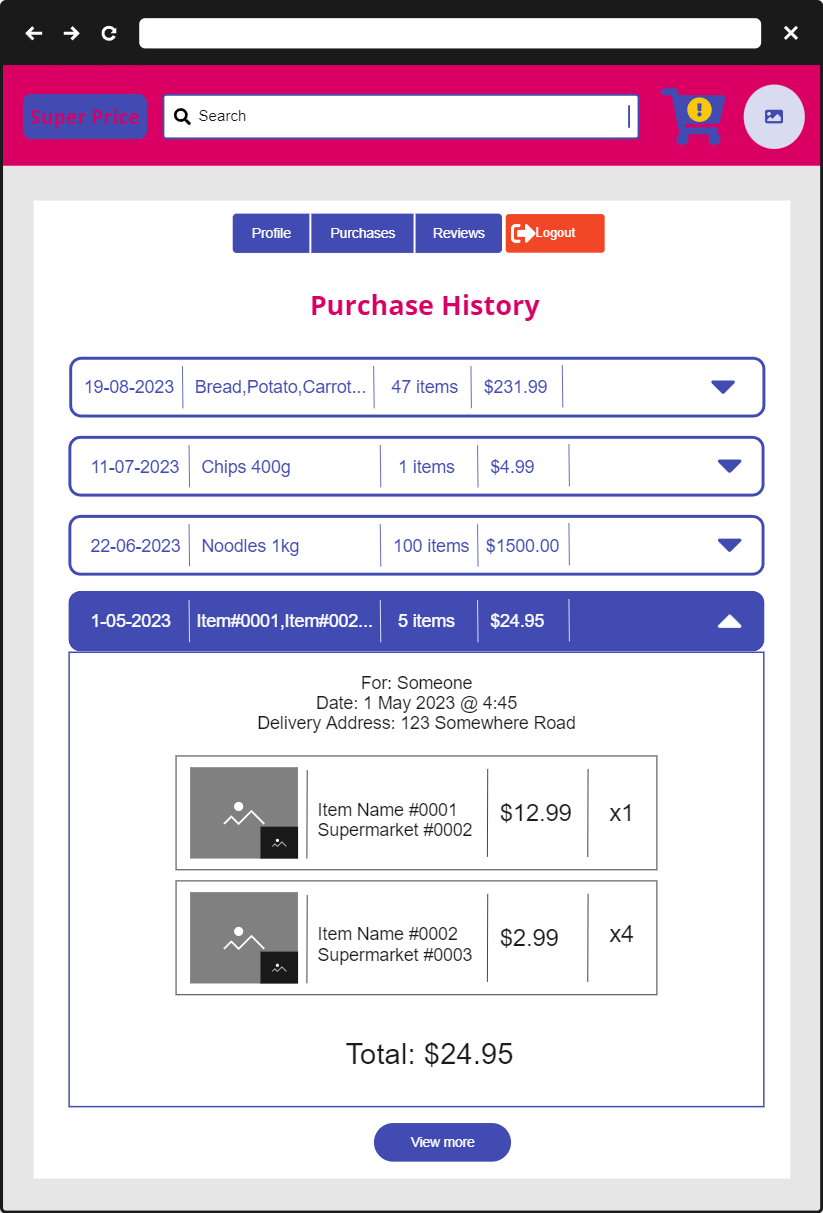
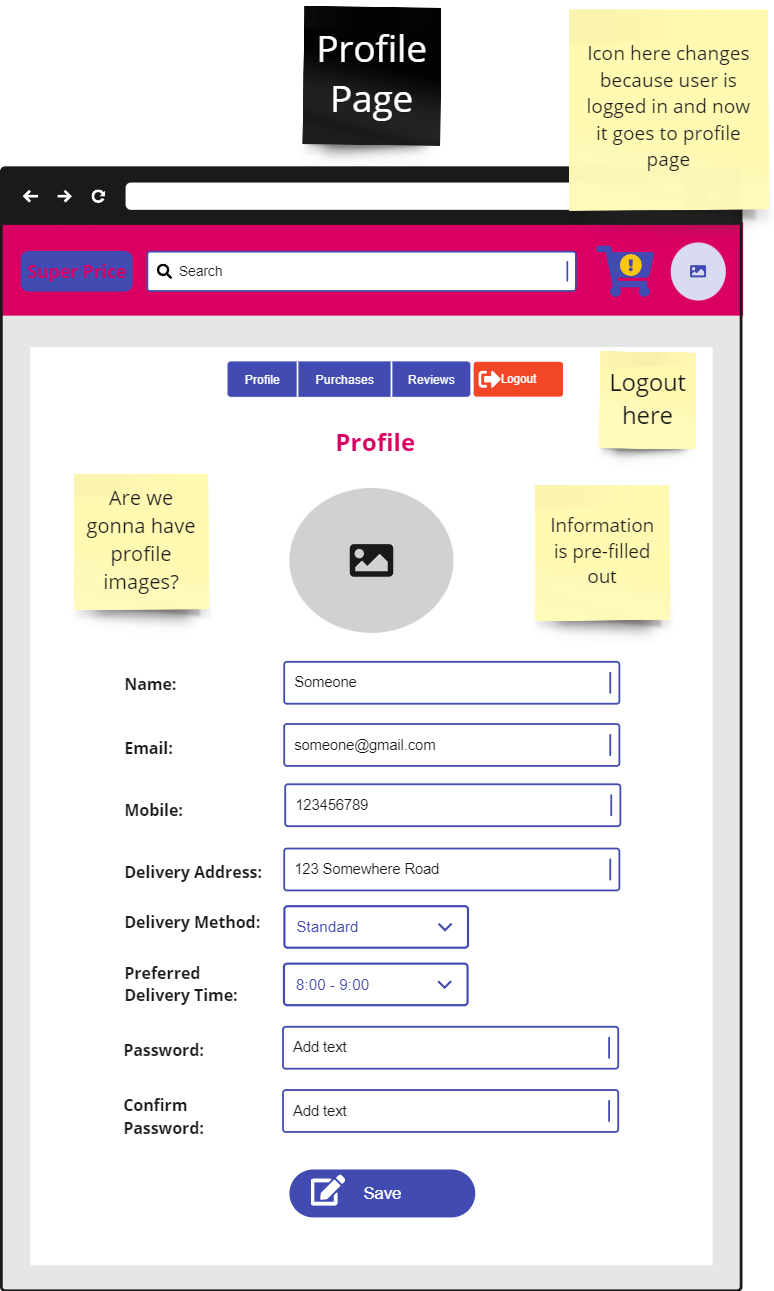
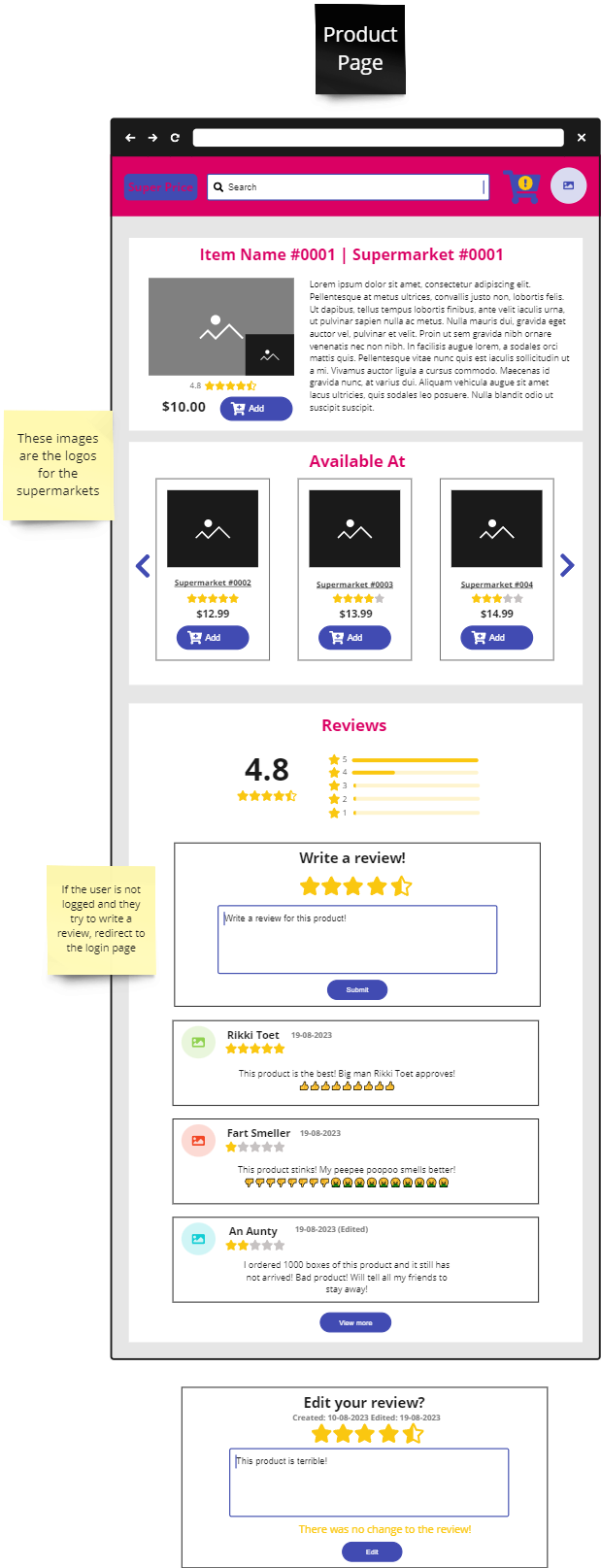
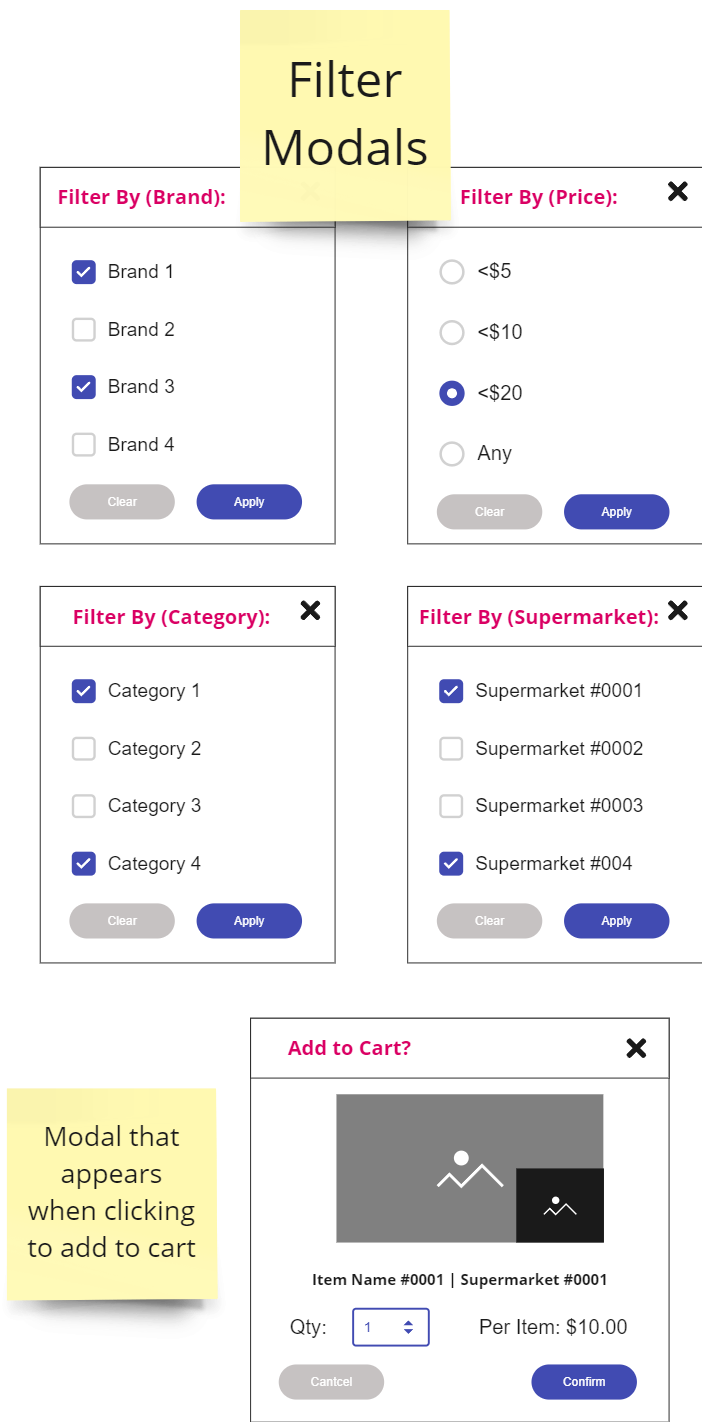
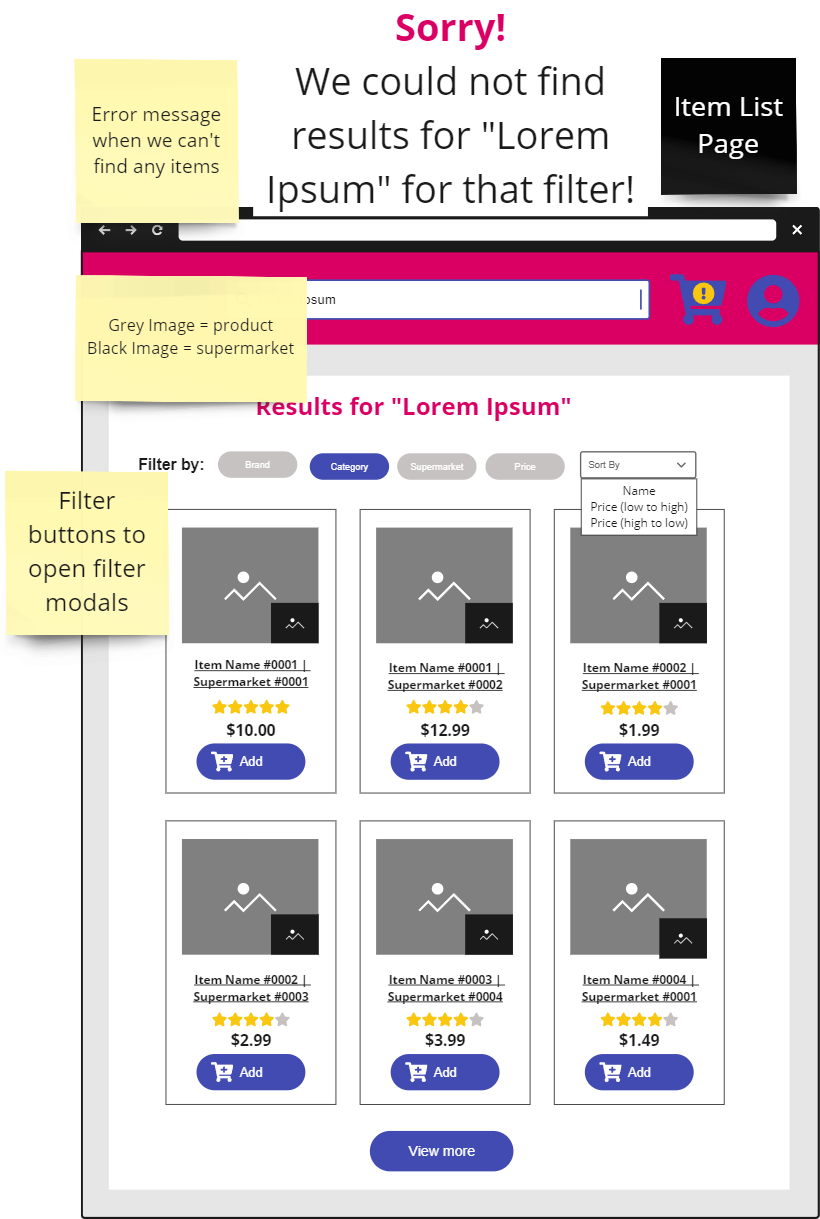
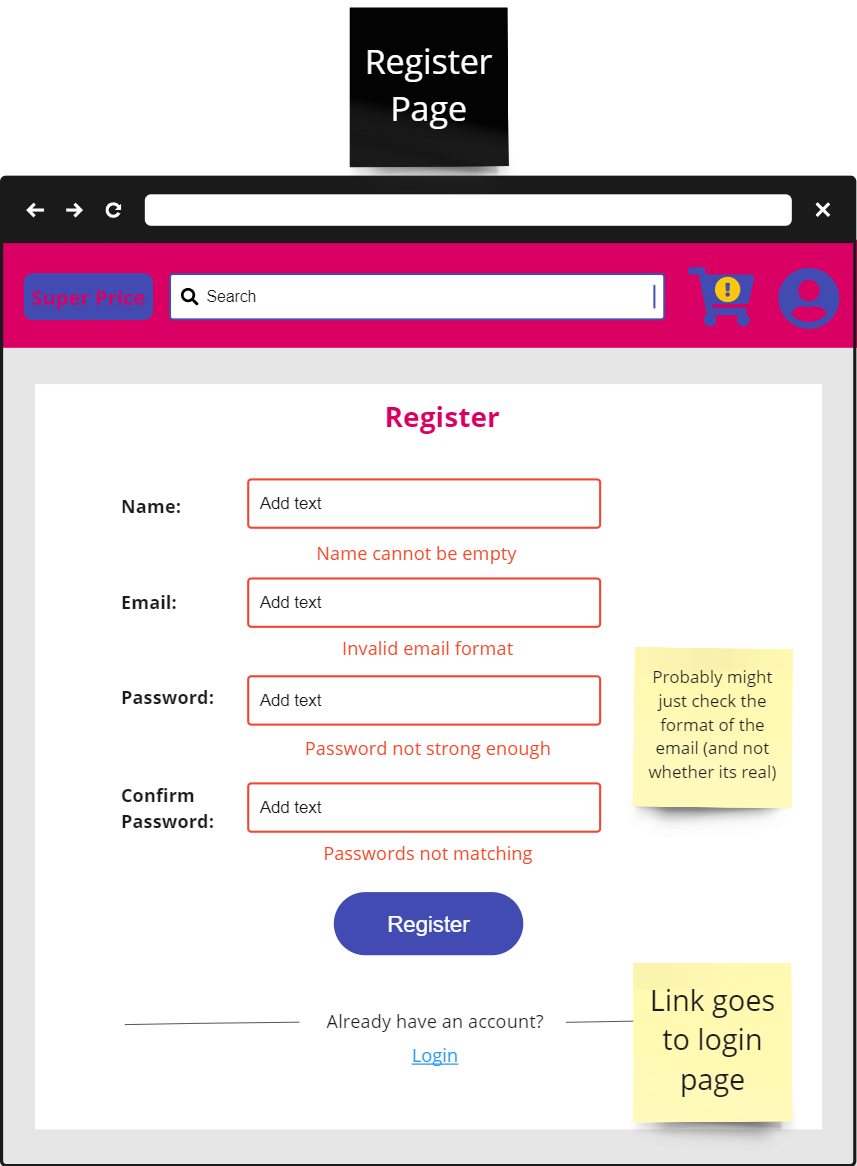
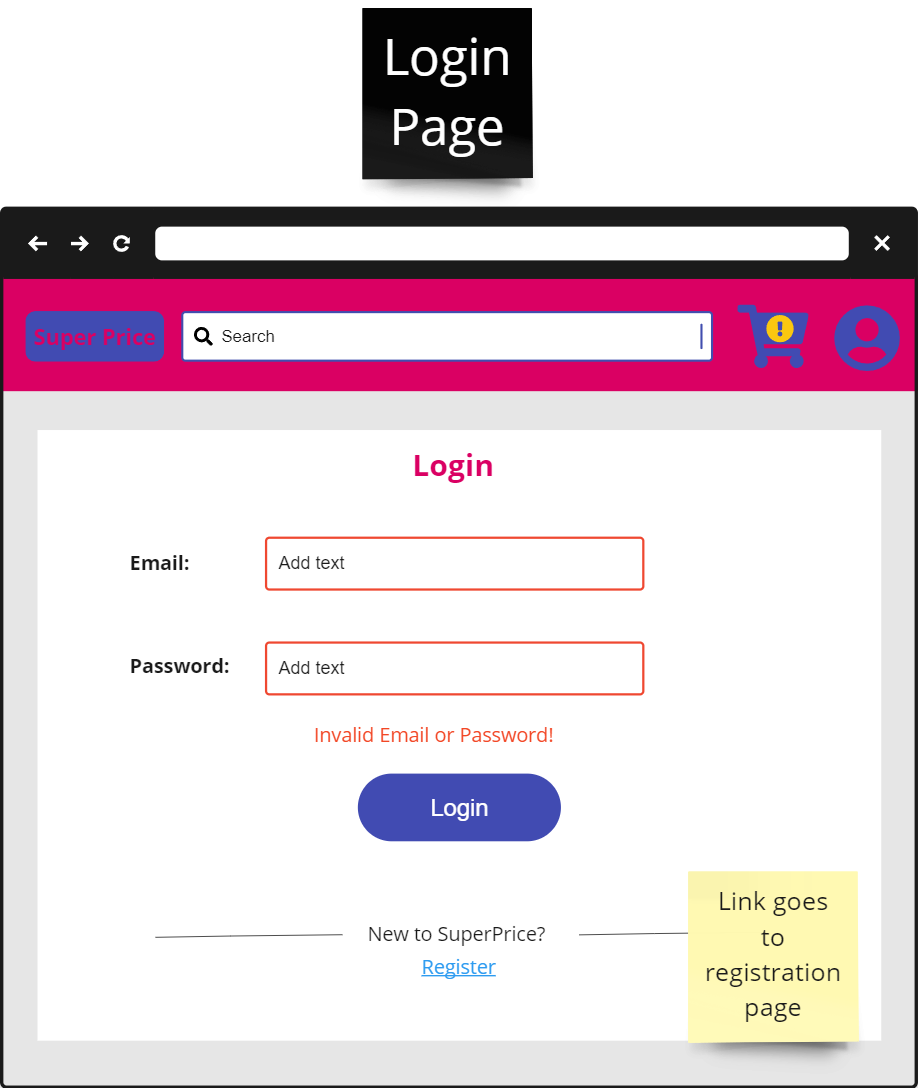
* *The Springboot backend server. If the server is not online, then the frontend cannot interact with the data stored in the databases since the backend acts as a middleman between the two. Also, if a different Java server environment is used, then the code for Springboot may to translate for new environment*
* *The H2 database. If format of the data is changed in the database, then errors may occur when the backend server queries the database*
* *react and react-router, they are needed during runtime, or the frontend app will not work*
* *CSS files (after compilation), or else the application will not have the aimed for aesthetic*
* *HTML files (after compilation), or else the application interface will have no structure*
* *JavaScript files (after compilation), or else the users will not be able to interact with the interface*

# External Interface Requirements

## User Interfaces

Rules and things worth noting about the User Interfaces:

* Titles are bold and have the colour magenta
* All interactive controls are in blue
* All errors or anything requiring increased attention is in the colour red
* Warnings are in yellow
* Green is used when a task has been completed successfully or for a price drop
* Light Gray is used when a control is inactive or holds little importance
* For laptops or desktops, there could be text next to the cart and profile icons in the navigation bar, so it is made clear what their function is
* When the user logs in, the guest profile icon is changed to the user’s profile picture
* *The top navigation bar will be consistent throughout all pages*



Mockups can be viewed here: <https://miro.com/app/board/uXjVMuibjbY=/>

## Hardware Interfaces

There will be no hardware component of the SuperPrice application besides the workstation (or laptop) which will be used for its development and to test it locally.

If the project is deployed on services such as AWS (Amazon Web Services), then the physical servers at Amazon could be considered as a hardware component.

## Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

Software used:

* Spring boot 3.1.2
* *Java SDK 17*
* *H2 Database Engine 2.2.224*
* *Next 13.4.12*
* *PostCSS 8.4.27*
* *React 18.2.0*
* *React-DOM 18.2.0*
* *Tailwind 3.3.3*
* *TypeScript 5.1.6*

*Operating system server ran on (also the machine for development):*

* *Windows 10*
* *Mac*

## Communications Interfaces

To test the application locally, only the HTTP protocol will be used. The application will not stream any videos, send any files, send any emails, or establish any web socket connection. HTTP does not encrypt any data and therefore data sent between browser and server is not confidential. If the platform is to deploy on a web service, then HTTPS should be used instead, which encrypts sent data and therefore keeps it confidential.

To develop the product, Microsoft Teams will be used for formal communication between team members. For more informal conversations, Discord will be used instead. For version control, git and GitHub will be used. This technology will ensure that there will be backups for the platform’s code and changes can be easily reverted.

# System Features

## The Search Bar

4.1.1 Description and Priority

The search bar functionality is a crucial aspect of the website as it allows customers to quickly find specific items they are looking for. The search function will be a required functional requirement for the website, given its significance in enhancing user experience and enabling efficient product discovery.

4.1.2 Stimulus/Response Sequences

When a user enters their desired item into the search bar, the website's search functionality will initiate a search query based on the input. The search query will be processed by the back-end system, which will then retrieve relevant products from the database.

4.1.3 Functional Requirements

REQ-1: The application should return a list of products related to the customer’s search query.

REQ-2: If are no related products the application should return an error message informing the customer that there are no items related to their search.

## Checkout page

4.2.1 Description and Priority

The checkout page is a high priority as it is one of the core features of the application. The page will allow the user to input their payment details into the website and view the total amount to be paid, including the total price of items, delivery fee, service fee, etc. It will also allow the user to schedule the delivery for ASAP or for a later date or time.

4.2.2 Stimulus/Response Sequences

The checkout page will show all the items that are in the cart as well as their quantities individual price, the total price of all items in the cart and the complete total price of the items plus any extra fees. The actors will be the user who wants to check out their cart. Related use cases include the view cart page. The user must first be on the cart view page, then the user can click the checkout button which will direct the user to the checkout page. After the user has entered their payment details, the system will purchase the items, schedule a delivery and show the user a page thanking them for their purchase.

1. The user will click the checkout button from the view cart page
2. The system will show the user the checkout page
3. The user will fill out their payment details
4. The system will verify the payment details and once verified will show a page to select the delivery date and time
5. The user will select their preferred delivery date and time
6. The system will show the user a price summary
7. The user will click the purchase button
8. The system will purchase the items and schedule a delivery

If the user changes their mind midway through, the system will cancel the transaction and return the user to the home page. If the user enters invalid payment details the system will notify the user to check their details again. If the user selects a time that is unavailable for delivery, the system will notify the user and show times that are available for delivery

4.2.3 Functional Requirements

REQ-1: Invalid payment details should inform the user that the details are incorrect

REQ-2: System will correctly calculate the total price of the items + fees

REQ-3: Invalid delivery date will inform the user that the date and times are unavailable and will show valid dates and times for delivery

## Add to cart

4.3.1 Description and Priority

Add to cart functionality is a high priority function for our website. It will allow users to add items from the website into a virtual shopping cart to keep track of what users want to buy. It will be a required functionality for the website as it will enhance the typical user experiences and is the industry standard in comparison to similar websites.

4.3.2 Stimulus/Response Sequences

When the user finds an item, they want to buy, they will click on a button that says, “Add to Cart”, in which then the item will be put into a virtual shopping cart and the user will see a small notification and have confirmation that the desired item is in the cart.

4.3.3 Functional Requirements

REQ-1: Any items listed on the website should be able to be added to the virtual shopping cart.

REQ-2: User should receive confirmation that the item is in the shopping cart.

REQ-3: Users should be able to put any quantity of an item in the shopping cart.

## View cart (remove items)

4.4.1 Description and Priority

The view cart function is a high priority function as it is a core aspect of our website. It will allow users to view the items that they have already added to the virtual shopping cart and gives them the ability to remove any items they desire. When viewing the cart, they should be able to see the total cost of all the items combined and be able to view all the prices of each item on one page. It will be a required functionality for the website to enhance the user experience and is the industry standard in comparison to similar websites.

4.4.2 Stimulus/Response Sequences

When the user wants to check what is currently in their virtual shopping cart, they will click on a button which will give them a small window to view their items. Any item they wish not to purchase, the user will click on a button which would remove the item from the virtual shopping cart and will receive confirmation that the item is removed.

4.4.3 Functional Requirements

REQ-1: Users should be able to view all the items they have added into the shopping cart including price and quantity of the items added to the cart.

REQ-2: Users should be able to remove any item that is in the cart when they view it.

## Filter

4.5.1 Decription and Priority

The filter function is a low priority function as although it enhances the user experience it is not vital to the user experience. It will give users the chance to filter out their search options to find the item they want to find. It will enhance the user experience and is the industry standard for similar websites.

4.5.2 Stimulus/Response Sequences

The user will go onto the search bar and search for their item. Then the user will interact with the filter options from the side and apply any suitable filters which then give them the filter search results.

4.5.3 Functional Requirements

REQ-1: User should be able to see all applicable filters

REQ-2: User must search for an item before using the filters

## Notifications

4.6.1 Description and Priority

The notification function is a low priority function as it is not a core aspect of the website. It will allow for the website to notify the user when there are any appealing deals for the user. It will enhance the user experience notifying users so they can efficiently buy any desired item and it is the industry standard for similar websites.

4.6.2 Stimulus/Response Sequences

The user will be on the website, and then suddenly they will receive a pop-up about an item they may interest them. Users can now choose whether to go through the notification and buy the item or ignore it.

4.6.3 Functional Requirement

REQ-1: It will be a pop-up that should catch the user's attention but not too intrusive

## Alerts

4.1.1 Description and Priority

The alerts are a medium priority function. Although it is not a core function of the application, it is still necessary to alert the user of important information. Alerts can inform the user of important information that is happening in their current context. This will ensure that important information is properly given to the user when necessary.

4.1.2 Stimulus/Response Sequences

The user will be on the log in page and will input an incorrect username or password. The website will alert the user that the username and password are incorrect by highlighting the incorrect fields and give text that describes the issue and how to resolve it. The user is now alerted to the problem and can now check their details and try to log in again.

4.1.3 Functional Requirements

REQ-1: If the application has a problem that the user must handle, such as incorrect log-in information, the application will alert the user of where the issue is, and how to resolve it.

REQ-2: If the application has run into an error while doing a task, it will alert the user that it has failed in doing a task and prompt the user to try the task again.

## Log-In

4.1.1 Description and Priority

The log-in function is a high priority function. This is because the ability for users to log in is one of the main functions of the application. A user with an account has more functions available to them such as the ability to leave a review on a product, view their purchase history and save their details.

4.1.2 Stimulus/Response Sequences

The user will be on the website's main page and wants to log in. They will click on the log-in button on the top navigation bar and will be prompted to enter their log-in details. After entering their details, the system will verify their details and if they are correct, will log the user in and direct them to the previous page they were on.

4.1.3 Functional Requirements

REQ-1: If the user is on a product page before clicking on the log in button, the system will direct the user back to the product page after successfully logging in.

REQ-2: The user is able to log in whenever they wish and if there are features such as leaving a review which require an account, the system will prompt the user to log in before doing any actions.

## Register

4.1.1 Description and Priority

The register function is a high priority function as users who do not have an account will want to register an account. The register function should be available to all users who visit the page and are not currently logged in.

4.1.2 Stimulus/Response Sequences

The user will be on the website's main page and wants to create an account. The user will click the log in page and then click the link to redirect them to the register an account page. After entering their details, the system will create an account using the details and the user will then log in with the details and be directed to their account page

4.1.3 Functional Requirements

REQ-1: If the user is prompted to log in, there should also be an option to create an account if they do not have one.

REQ-2: The account should be immediately created so that the user can use their account features as soon as possible.

# Other Nonfunctional Requirements

## Performance Requirements

As SuperPrice’s main feature is being able to access and display to a user any item from its massive catalogue of supermarket products, this functionality responding to requests almost instantly is essential. Waiting over 5 seconds for a database query when searching for products would be unacceptable for users, however it is important this requirement is not confused with a webpage not loading the query due to poor internet connectivity on the users' end. Similarly, database queries for user details should also not tolerate a timing of more than 5 seconds, as users experiencing delays when trying to access their profiles would not be admissible.

## Safety Requirements

The main legislature that would apply to the SuperPrice app regarding safety is the Online Safety Act (2021, federal). It governs any online service provider, including owners of apps like SuperPrice, ensuring their product is free from harmful or explicit content that could affect users. By being compliant with this act, the SuperPrice app will guarantee that users are safe by the standards of the Australian government whilst on the site.

## Security Requirements

As we are storing user data when profiles are created, many federal and state cybersecurity & data protection laws will apply to the SuperPrice product. These include the Privacy Acy (1988, federal) and the Privacy and Data Protection Act (2014, Victoria). These two acts specify the standards required for properly protecting and using user data, violation of which is punishable by law, and so will be strictly obeyed for the implementation of the SuperPrice app.

## Software Quality Attributes

Whilst all software quality attributes are highly important for SuperPrice, ease of use is a high priority for the product to fulfill the requirement of the software having a “User-friendly interface”. As this was specifically requested by the customer, this quality is taking very high priority. Whilst this is difficult to verify as it is subjective to each user, a general idea of the product’s ease of use can be gathered by having users attempt to find certain features in the application, as is part of our testing plan. Additionally, correctness is essential as one of the main draws of SuperPrice is to help users find the best price for their shopping. Displaying incorrect information that misleads users would go directly against the platform’s primary directive, hence this quality also having a very high priority.

## Business Rules

Users can only manage their personal information after verifying their login credentials to ensure only the actual user themselves can view and modify their personal information.

# Other Requirements

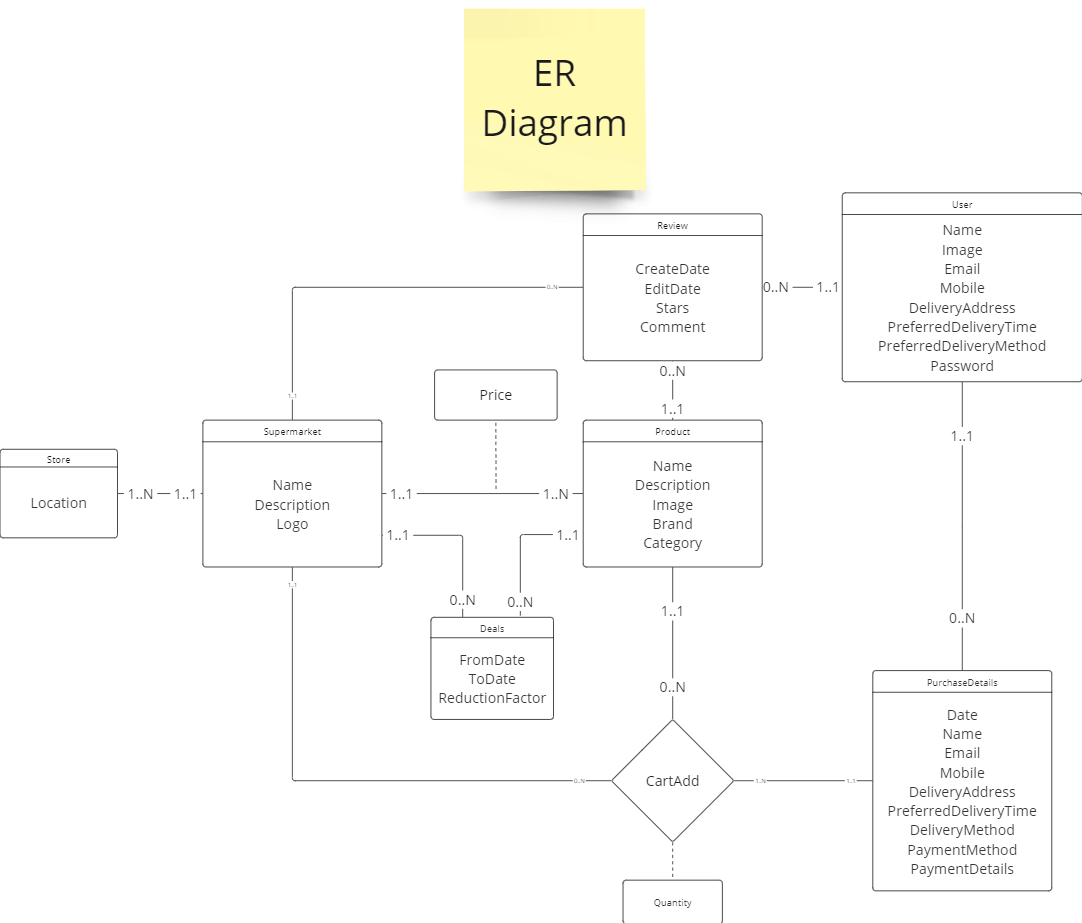
*The Privacy Act 1988 states that SuperPrice needs to ensure that personal data is not mismanaged*

*Consumer Laws also applies because SuperPrice is selling goods and services.*

Appendix A: Glossary

*N/A*

Appendix B: Analysis Models



ER diagram can be viewed here: <https://miro.com/app/board/uXjVMuibjbY=/>

Appendix C: To Be Determined List

N/A