

---

# Software Requirements Specification

for

**<SuperPrice - Price Matching and Delivery Application >**

Version 2.0 approved

**Prepared by** <Ibrahim Al-Ashhab, Rashik Raj, Abhijeet Kumar, Udit Pradeep Malshe, Vidyut Venkatesan>

<RMIT University>

<17/09/2023>

# Table of Contents

<b>Table of Contents</b>	<b>ii</b>
<b>Revision History</b>	<b>ii</b>
<b>1. Introduction</b>	<b>1</b>
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
<b>2. Overall Description</b>	<b>1</b>
2.1 Product Perspective	1
2.2 Product Functions / Functional Requirements	2
2.3 User Classes and Characteristics	2
2.4 Operating Environment	2
2.5 Design and Implementation Constraints	2
2.6 User Documentation	3
2.7 Assumptions and Dependencies	3
<b>3. External Interface Requirements</b>	<b>3</b>
3.1 User Interfaces	3
3.2 Hardware Interfaces	3
3.3 Software Interfaces	3
3.4 Communications Interfaces	4
<b>4. Other Nonfunctional Requirements</b>	<b>4</b>
4.1 Performance Requirements	4
4.2 Security Requirements	4
4.3 Software Quality Attributes	4
<b>5. Other Requirements</b>	<b>5</b>
• Database Requirements:	5
<b>Appendix A: Glossary</b>	<b>5</b>
<b>Appendix B: Analysis Models</b>	<b>5</b>
<b>Appendix C: To Be Determined List</b>	<b>5</b>

## Revision History

Name	Date	Reason For Changes	Version

# 1. Introduction

## 1.1 Purpose

*SuperPrice is a price matching and delivery application, which will help users to compare the prices of products in local stores. In this SRS document, the following will be discussed: functional and non-functional requirements, use cases, system architecture, user interface design, assumptions and constraints, dependencies, testing and acceptance criteria.*

## 1.2 Document Conventions

*Bold and underline - headings*

## 1.3 Intended Audience and Reading Suggestions

*This document is designed to guide developers, project managers, and testers while developing, managing, and testing the application.*

## 1.4 Product Scope

*The focus of this SuperPrice project is the development of an easy-to-use price matching and delivery application. The Super Price app will show you the best prices of items from different stores and supermarkets in your area. The app will allow you to shop easily, save time and money while offering a variety of delivery options. Realtime price drop and special offer alerts will also be provided by the app. To make it more reliable, the app will be based on user reviews and ratings. At the end of the day, SuperPrice's goal is to make shopping easier for consumers and local businesses alike.*

## 1.5 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

## 2.1 Product Perspective

*<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system,*

*relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>*

## 2.2 Product Functions / Functional Requirements

- **Product Search and Categorization:**
  - *Allow users to search for specific products or browse through different categories.*
  - *Provide a smooth and hassle-free shopping experience.*
- **Price Comparison:**
  - *Enable users to search for products and compare prices across various local supermarkets.*
  - *Help users identify the store offering the lowest price for their desired items.*
- **Delivery Organization:**
  - *Facilitate user-organized deliveries, offering multiple flexible delivery options.*
  - *Ensure convenient and efficient delivery arrangements.*
- **Notifications and Alerts:**
  - *Provide timely notifications and alerts about price drops and special offers.*
  - *Ensure users are informed of opportunities to save money.*
- **User Reviews and Ratings:**
  - *Allow users to leave reviews and ratings for supermarkets and products.*
  - *Enhance reliability and assist users in making informed purchasing decisions.*
- **User-Friendly Interface:**
  - *Prioritize simplicity and ease of use for both novice and experienced shoppers.*
  - *Ensure an intuitive and accessible application interface.*

## 2.3 User Classes and Characteristics

*<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>*

## 2.4 Operating Environment

*<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>*

## 2.5 Design and Implementation Constraints

*<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security*

*considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).>*

## **2.6 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.>*

## **2.7 Assumptions and Dependencies**

*<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>*

# **3. External Interface Requirements**

## **3.1 User Interfaces**

*<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>*

## **3.2 Hardware Interfaces**

*<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>*

## **3.3 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.>*

### 3.4 Communications Interfaces

*<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>*

## 4. Other Nonfunctional Requirements

### 4.1 Performance Requirements

- **Response Time:**
  - *The application should respond to user actions, such as product searches and price comparisons, within 2 seconds to ensure a smooth and seamless user experience.*
  - *Rapid response times will prevent user frustration and maintain engagement.*
- **Search and Comparison Speed:**
  - *The application must be capable of handling simultaneous product searches and price comparisons for multiple users without significant delays.*
  - *Searches and comparisons should complete within 1 second, even during peak usage.*

### 4.2 Security Requirements

- **User Data Protection:**
  - *User data, including personal details and payment information, must be encrypted both during transmission and storage to prevent unauthorized access.*
- **Authorization Levels:**
  - *Different user roles (users, administrators) should have appropriate levels of access and permissions to ensure data integrity and prevent unauthorized actions.*
- **Secure API Integration:**
  - *APIs connecting with supermarket systems and payment gateways should use secure communication protocols (e.g., HTTPS) and require API keys or tokens for access.*

### 4.3 Software Quality Attributes

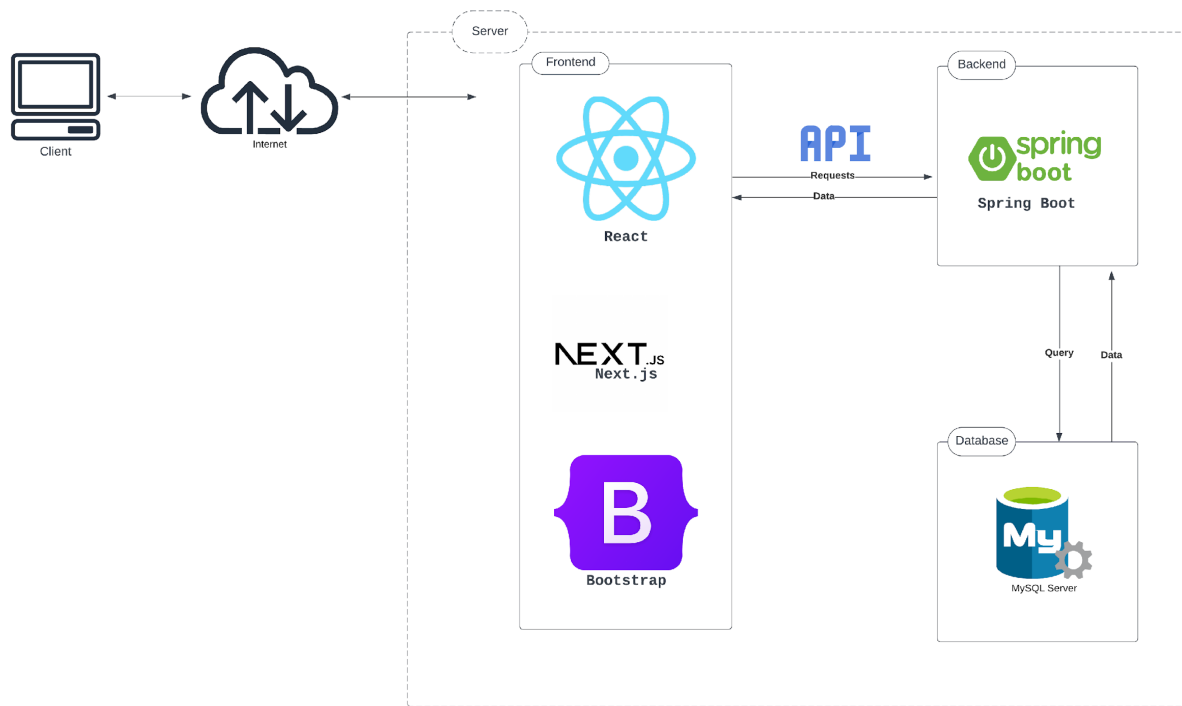
- **Usability:**
  - *The application's user interface should be intuitive and easy to navigate*
- **Availability:**
  - *The application's server infrastructure should provide high availability,*

## 5. Other Requirements

- **Database Requirements:**

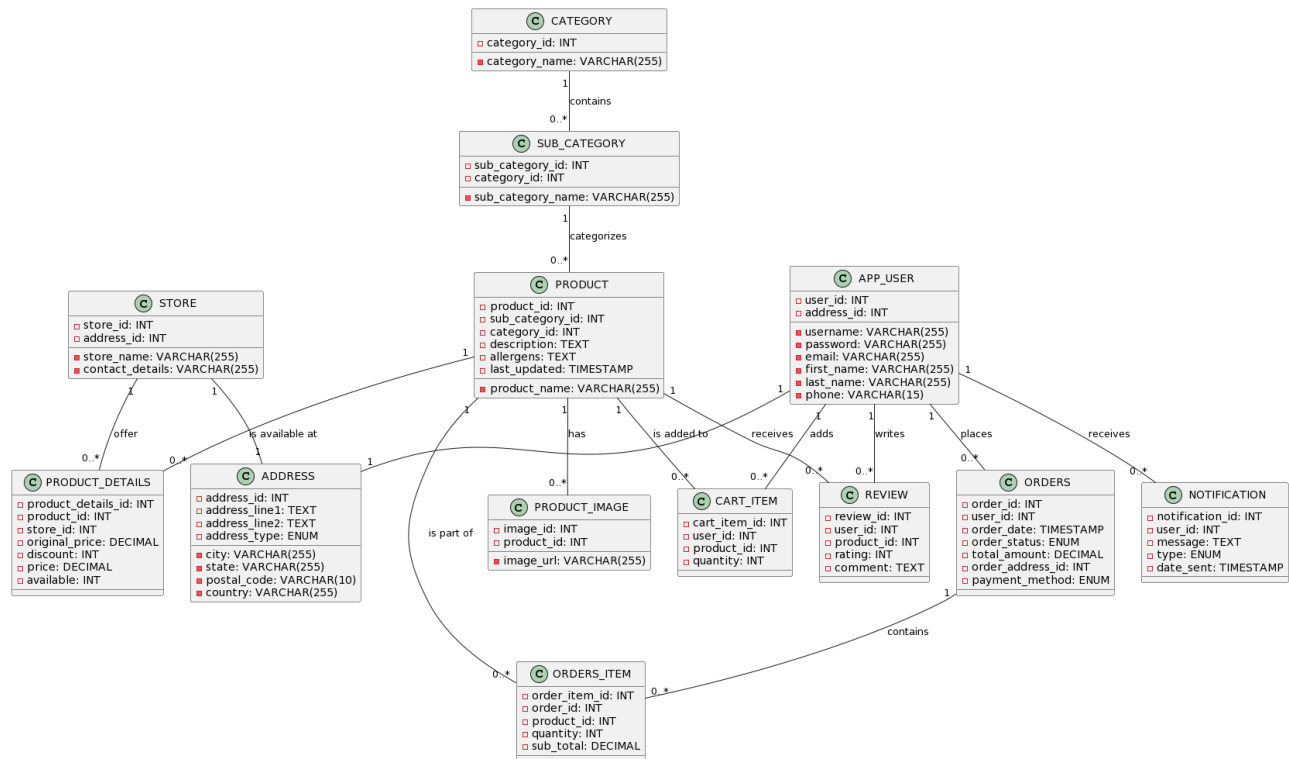
- *The application's database should support efficient storage and retrieval of product information, and reviews.*

## 6. System Architecture



- **Client:** Represents the user's web browser.
- **Internet:** Represents the internet through which the user accesses the application.
- **Frontend (Next.js/React):** The frontend framework where the user interface is built.
- **Backend (Spring Boot):** The backend framework handling business logic and data processing.
- **MySQL:** The database where data is stored.
- **API:** Represents the various APIs that the frontend might call.

## 7. Data Model



### ADDRESS:

Represents the address details.

- Attributes: address\_id, address\_line1, address\_line2, city, state, postal\_code, country, address\_type.
- Relationships:
  - One-to-Many with STORE: A store can have one address.
  - One-to-Many with APP\_USER: A user can have one address.
  - One-to-Many with ORDERS: An order can have one address.

### CATEGORY:

Represents different categories of products.

- Attributes: category\_id, category\_name.
- Relationships:
  - One-to-Many with SUB\_CATEGORY: A category can have multiple sub-categories.
  - One-to-Many with PRODUCT: A category can have multiple products.



**SUB\_CATEGORY:**

*Represents sub-categories of products.*

- *Attributes: sub\_category\_id, sub\_category\_name, category\_id.*
- *Relationships:*
  - *Many-to-One with CATEGORY: Each sub-category belongs to one category.*
  - *One-to-Many with PRODUCT: A sub-category can have multiple products.*

**PRODUCT:**

*Represents individual grocery items.*

- *Attributes: product\_id, product\_name, sub\_category\_id, category\_id, description, allergens, last\_updated.*
- *Relationships:*
  - *Many-to-One with SUB\_CATEGORY: Each product belongs to one sub-category.*
  - *Many-to-One with CATEGORY: Each product belongs to one category.*
  - *One-to-Many with PRODUCT\_IMAGE: A product can have multiple images.*
  - *One-to-Many with REVIEW: A product can have multiple reviews.*
  - *One-to-Many with ORDERS\_ITEM: A product can be part of multiple orders.*
  - *One-to-Many with CART\_ITEM: A product can be added to multiple carts.*
  - *One-to-Many with PRODUCT\_DETAILS: A product can have multiple details.*

**PRODUCT\_IMAGE:**

*Represents images of products.*

- *Attributes: image\_id, product\_id, image\_url.*
- *Relationships:*
  - *Many-to-One with PRODUCT: Each image belongs to one product.*

**STORE:**

*Represents supermarkets and stores.*

- *Attributes: store\_id, store\_name, address\_id, contact\_details.*
- *Relationships:*
  - *One-to-Many with PRODUCT\_DETAILS: A store can have details of multiple products.*
  - *Many-to-One with ADDRESS: Each store is located at one address.*

### **APP\_USER:**

*Represents registered users.*

- *Attributes: user\_id, username, password, email, first\_name, last\_name, phone, address\_id.*
- *Relationships:*
  - *One-to-Many with REVIEW: A user can write multiple reviews.*
  - *One-to-Many with ORDERS: A user can make multiple orders.*
  - *One-to-Many with NOTIFICATION: A user can receive multiple notifications.*
  - *One-to-Many with CART\_ITEM: A user can have multiple items in their cart.*
  - *Many-to-One with ADDRESS: Each user can have one address.*

### **REVIEW:**

*Represents user reviews and ratings for products.*

- *Attributes: review\_id, user\_id, product\_id, rating, comment.*
- *Relationships:*
  - *Many-to-One with APP\_USER: Each review is written by one user.*
  - *Many-to-One with PRODUCT: Each review is about one product.*

### **ORDERS:**

*Represents a purchase made by a user.*

- *Attributes: order\_id, user\_id, order\_date, order\_status, total\_amount, order\_address\_id, payment\_method.*

*- Relationships:*

- One-to-Many with ORDERS\_ITEM: An order can consist of multiple items.*
- Many-to-One with APP\_USER: Each order is made by one user.*
- Many-to-One with ADDRESS: Each order can have one address.*

**ORDERS\_ITEM:**

*Represents individual items within an order.*

- Attributes: order\_item\_id, order\_id, product\_id, quantity, sub\_total.*
- Relationships:*
  - Many-to-One with ORDERS: Each order item is part of one order.*
  - Many-to-One with PRODUCT: Each order item represents one product.*

**NOTIFICATION:**

*Represents notifications sent to users.*

- Attributes: notification\_id, user\_id, message, type, date\_sent.*
- Relationships:*
  - Many-to-One with APP\_USER: Each notification is received by one user.*

**CART\_ITEM:**

*Represents items added to a user's shopping cart.*

- Attributes: cart\_item\_id, user\_id, product\_id, quantity.*
- Relationships:*
  - Many-to-One with APP\_USER: Each cart item belongs to one user.*
  - Many-to-One with PRODUCT: Each cart item represents one product.*

**PRODUCT\_DETAILS:**

*Represents details of products in stores.*

*- Attributes: product\_details\_id, product\_id, store\_id, original\_price, discount, price, available.*

*- Relationships:*

*- Many-to-One with PRODUCT: Each product detail is about one product.*

*- Many-to-One with STORE: Each product detail is available in one store.*

## **DataModel Changes**

### **1. Additions:**

*- **ADDRESS Table:** A new table to represent the address details.*

*- **CATEGORY Table:** While the old schema had a CATEGORY table, the new schema has added a unique constraint to the category\_name.*

*- **SUB\_CATEGORY Table:** A new table to represent sub-categories of products. This table also establishes a relationship with the CATEGORY table.*

*- **PRODUCT Table:***

*- Added sub\_category\_id to establish a relationship with the SUB\_CATEGORY table.*

*- Added allergens attribute.*

*- **PRODUCT\_IMAGE Table:** A new table to represent images of products.*

*- **STORE Table:***

*- The new schema has added an address\_id to establish a relationship with the ADDRESS table.*

*- **APP\_USER Table:***

*- Renamed from "USER" to "APP\_USER".*

*- Added attributes: first\_name, last\_name, and phone.*

*- Address attribute from the old schema has been replaced with address\_id to establish a relationship with the ADDRESS table.*

*- **PRODUCT\_DETAILS Table:** A new table to represent details of products in stores.*

### **2. Modifications:**

*- **PRODUCT Table:***

- Removed Price, Availability, and ImageURL attributes. These details are now in the PRODUCT\_DETAILS and PRODUCT\_IMAGE tables.

- **STORE Table:**

- Removed StoreLocation. The location details are now in the ADDRESS table.

- **REVIEW Table:** No significant changes.

- **ORDERS Table:**

- Renamed from "TRANSACTION" to "ORDERS".

- Added attributes: order\_status, order\_address\_id, and payment\_method.

- **ORDERS\_ITEM Table:**

- Renamed from "TRANSACTION\_ITEM" to "ORDERS\_ITEM".

- Removed SubTotal attribute.

**3. Removals:**

- **PRODUCT\_CATEGORY (Association Table):** This table is not present in the new schema. The many-to-many relationship between products and categories seems to be resolved using the SUB\_CATEGORY table in the new schema.

- **PRODUCT\_STORE (Association Table):** This table is not present in the new schema. The relationship between products and stores is now handled by the PRODUCT\_DETAILS table.

Actions

New workflow

All workflows

Team Group-P2-07 CI tasks

Management

Caches

Runners

Beta

✓ Merge pull request #95 from cosc2299-sept-2023/frontend/testi...  
Team Group-P2-07 CI tasks #11: Commit 7e447aa pushed by AbhijeetKS20  
main  
4 days ago  
43s  
...

✓ Merge pull request #71 from cosc2299-sept-2023/testing/frontend  
Team Group-P2-07 CI tasks #10: Commit 120d2d6 pushed by AbhijeetKS20  
main  
5 days ago  
36s  
...

✓ Merge pull request #94 from cosc2299-sept-2023/backend/order...  
Team Group-P2-07 CI tasks #9: Commit 16c5af3 pushed by udit710  
main  
5 days ago  
55s  
...

✓ Merge pull request #93 from cosc2299-sept-2023/origin/backen...  
Team Group-P2-07 CI tasks #8: Commit 20ac0df pushed by udit710  
main  
5 days ago  
50s  
...

✓ Merge pull request #92 from cosc2299-sept-2023/origin/backen...  
Team Group-P2-07 CI tasks #7: Commit 54c0c3f pushed by udit710  
main  
5 days ago  
44s  
...

✓ Merge pull request #88 from cosc2299-sept-2023/product-price-...  
Team Group-P2-07 CI tasks #6: Commit 03f00e1 pushed by Vidut03  
main  
5 days ago  
40s  
...

✓ Merge pull request #91 from cosc2299-sept-2023/origin/backen...  
Team Group-P2-07 CI tasks #5: Commit edaa0dc pushed by udit710  
main  
5 days ago  
33s  
...

✓ Merge pull request #89 from cosc2299-sept-2023/origin/backen...  
Team Group-P2-07 CI tasks #4: Commit 2ac3cda pushed by udit710  
main  
5 days ago  
31s  
...

✗ Merge pull request #53 from cosc2299-sept-2023/setup-workflow  
Team Group-P2-07 CI tasks #3: Commit a806739 pushed by AbhijeetKS20  
main  
last week  
5s  
...

✗ Merge pull request #52 from cosc2299-sept-2023/backend/setu...  
Team Group-P2-07 CI tasks #2: Commit 673e191 pushed by AbhijeetKS20  
main  
last week  
7s  
...

✗ Merge pull request #49 from cosc2299-sept-2023/setup-workflow  
Team Group-P2-07 CI tasks #1: Commit f143970 pushed by Vidut03  
main  
last week  
8s  
...

< Previous

1

2

Next >

## 8. User Interface Design / Wireframes

### 1. Checkout Page

SUPERPRICE

HOME

CATEGORY

CURRENT SALES

Search

Saved

KL

Order Summary

Coles - QV

\$20.00

✓

Nivea Moisturiser

\$10.00 × 2

Coles - Melbourne Central

\$14.50

✓

Subtotal

\$34.50

Select Payment Method

Method 1

Method 2

Method 3

Delivery Address

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim, ad minim veniam, quis nostrud exercitation

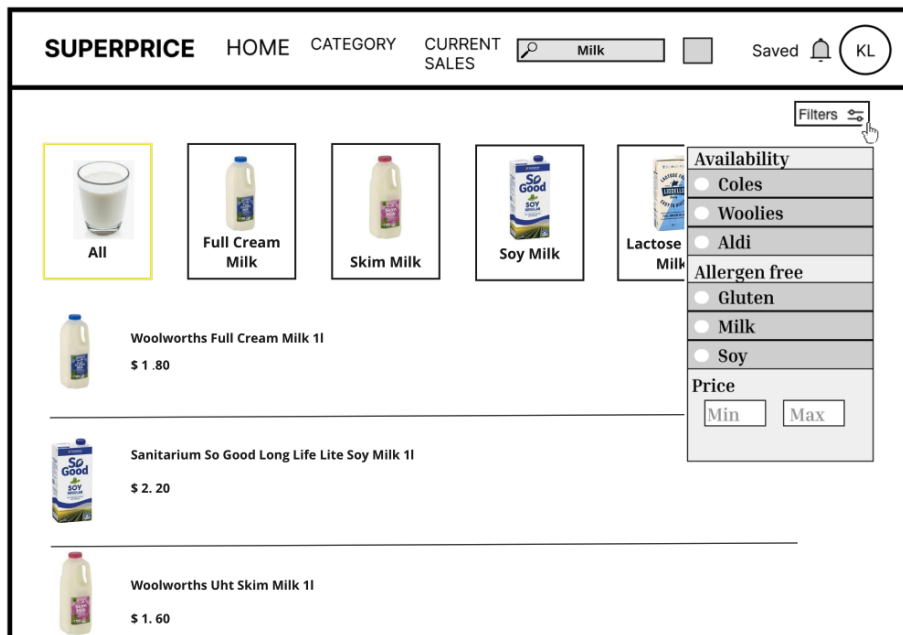
Place Order

Note: Total payment is required to place the order. The distributed payment(s) to all stores will be sent by our system separately.

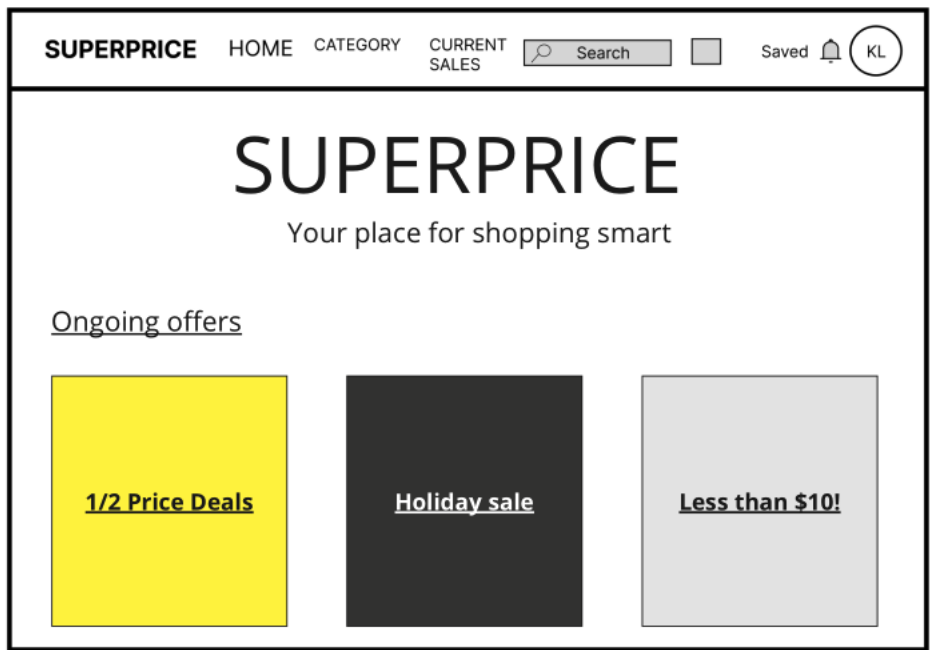
## 2. Category Hover view



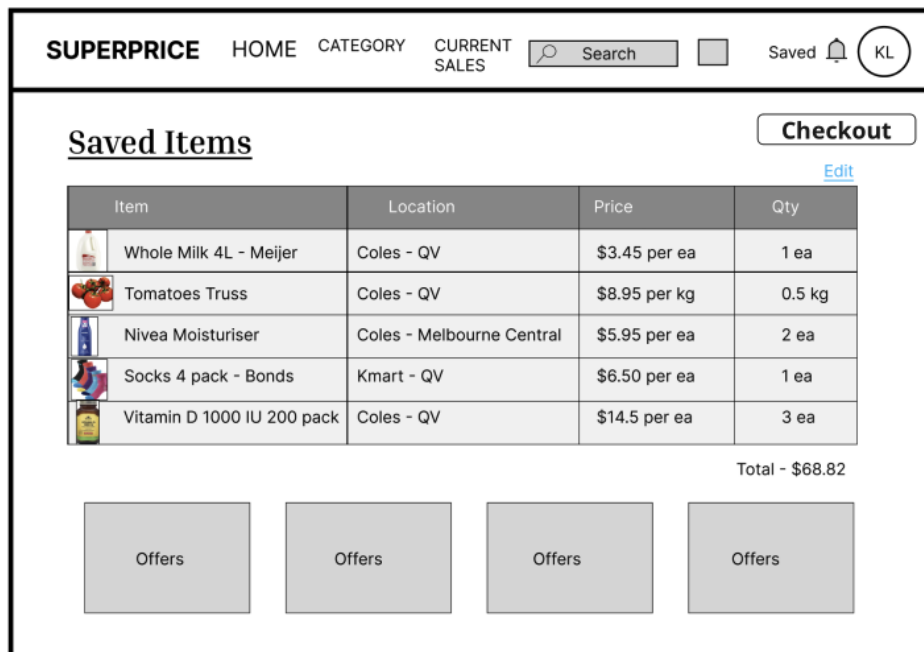
## 3. Filter View



## 4. Home Page




## 5. Saved Items Page






## 6. Product Details View

**SUPERPRICE**
HOME
CATEGORY
CURRENT SALES
☐
Saved




Kmart - QV
→
\$3.95
[Compare more...](#)

### Product Details

Whole cream Milk in a 4L container.

### Allergens

Milk

### Ingredients

Whole cream Milk, emulsifiers 690, Salt.

### Storage Instructions

Keep refrigerated at or below 5°C. Do not freeze. Once opened, consume within 5 days.

### Reviews & Ratings

Person\_1

3.3

Ea ipsum commodi aut rerum doloreque qui dolor ipsum vel magni quia. Est illum odit quo natus possimus et similique excepturi qui internos praesentium sit galisum quos nem nihil reprehenderit.


Person\_2


2.6


Ea ipsum commodi aut rerum doloreque qui dolor ipsum vel magni quia. Est illum odit quo natus possimus et similique excepturi qui internos praesentium sit galisum quos nem nihil reprehenderit.

[Read more...](#)

### Related



A2 Full cream milk 2L



Pura Full cream milk 2.2L



Devondale Full cream milk 2L


## 7. Search Results Page


**SUPERPRICE**
HOME
CATEGORY
CURRENT SALES
☐
Saved



All



Full Cream Milk


Skim Milk



Soy Milk


Lactose Free Milk



Long Life Milk


Woolworths Full Cream Milk 1L  
\$ 1.80

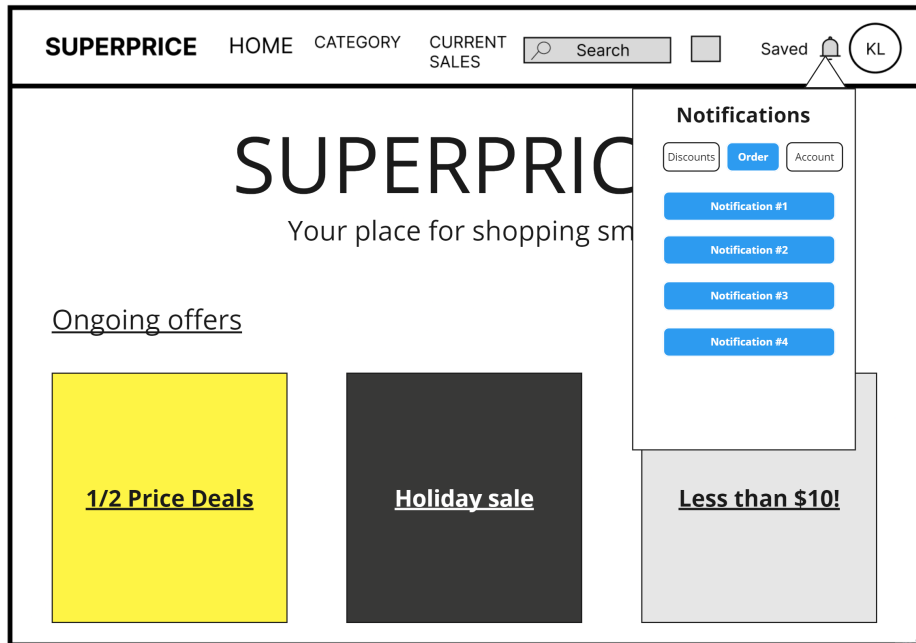
---


Sanitarium So Good Long Life Lite Soy Milk 1L  
\$ 2.20

---


Woolworths Uht Skim Milk 1L  
\$ 1.60





## 10. User Login Page

**SUPERPRICE**

HOME

CATEGORY

CURRENT SALES

☐

Saved

Login

Login

No Account? Sign up here

## 11. User Sign up Page

The wireframe shows a web page layout. At the top is a navigation bar with the text 'SUPERPRICE' on the left, followed by links 'HOME', 'CATEGORY', and 'CURRENT SALES'. To the right of these links is a search bar with a magnifying glass icon and the word 'Search'. Further right are a small square icon, the text 'Saved', a bell icon, and a circular profile icon containing the letters 'KL'. Below the navigation bar is a large rectangular area containing a 'Create Account' form. The form is a vertical rectangle with the title 'Create Account' at the top center. Inside the form are three input fields stacked vertically, labeled 'email', 'password', and 're-enter password'. To the right of the 'password' and 're-enter password' fields is a 'Create Account' button. Below the input fields and button, at the bottom left of the form, is the text 'Already member? Log in here'.

## 9. Assumptions and Constraints

### Assumptions:

1. *Single Database: The architecture assumes a single MySQL database instance. In a real-world scenario, there might be multiple databases or replicas for scalability and redundancy.*
2. *Stateless Backend: The backend is assumed to be stateless, meaning it doesn't maintain any user session information. Any state information required is either stored in the database or on the client side.*
3. *Single Backend Instance: The diagram shows a single backend instance. In reality, there might be multiple instances for load balancing and failover.*
4. *Monolithic Backend: The backend is assumed to be monolithic, even though there's a mention of services. In a microservices architecture, each service could be a separate entity with its own database.*
5. *Static Pages: The pages (Homepage, ProductPage, CheckoutPage) are assumed to be static and served by Next.js. Dynamic content is fetched via API calls.*
6. *Direct Database Access: The backend has direct access to the database. In some architectures, there might be a separate data access layer or service.*

#### Constraints:

1. *Scalability: With a single database and backend instance, scalability might be a concern. Horizontal scaling strategies would need to be considered.*
2. *Performance: Direct database access from the backend can introduce latency, especially if the database is not optimized or if complex queries are used.*
3. *Security: Direct exposure of API to the frontend can be a security risk. Proper authentication and authorization mechanisms need to be in place.*
4. *Maintenance: A monolithic backend can become complex over time, making it harder to maintain and deploy.*
5. *Database Dependency: The system is heavily dependent on the MySQL database. Any downtime or performance issues with the database will directly impact the application.*
6. *Limited Integration Points: The diagram shows a single "Services" integration point. In reality, there might be multiple third-party services (e.g., payment gateways, recommendation engines) that the backend interacts with.*

## 10. Dependencies

- Spring Boot
- React

## 7. Testing, User Stories, and Acceptance Criteria

Story #1: Item Checkout		Priority	
		Effort	
As a	user		
I want	to be able to review my order summary,		
So that	I can select a payment method, and review your shipping address during checkout.		
Acceptance criteria	Given that I am checking out, I should be able to see my order summary and total, address and be able to select the payment method.		
Test Case	<p>Test Case : Select Payment Method</p> <p>Description: Ensure the user can select a payment method during the checkout process.</p> <p>Preconditions: User is on the checkout page with an order summary displayed.</p> <p>Steps:</p> <ul style="list-style-type: none"> <li>• Review the order summary.</li> <li>• Select a payment method (e.g., credit card, PayPal).</li> <li>• Provide necessary payment information.</li> <li>• Continue to the next step.</li> <li>• Expected Result: The user successfully selects a payment method and proceeds to the next step of the checkout process</li> </ul>		

Story #2: Notifications		Priority	
		Effort	
As a	user		
I want	I would like to be notified of any new discounts		
So that	I can keep track of all the discounts.		
Acceptance criteria	1. I should get notified of any new discounts as soon as they are released 2. There is option to get notified via app notification, email or mobile 3. There is a view to filter the discounts		
Test Case	<b>Quick Access to Latest Deals</b> <ul style="list-style-type: none"> <li>Description: As a deal-seeking shopper, I want quick access to the latest deals, discounts, and special offers on the website's home page to save money and take advantage of limited-time promotions.</li> </ul> <b>Test Case: Display Latest Deals on Home Page</b> <ul style="list-style-type: none"> <li>Scenario: The user visits the website's home page.</li> <li>Steps to Reproduce:               <ol style="list-style-type: none"> <li>Open the web browser.</li> <li>Navigate to the website's home page.</li> </ol> </li> <li>Expected Outcome: The home page should prominently display the latest deals, discounts, and special offers.</li> </ul>		

Story #3: Availability of an item		Priority	
		Effort	
As a	budget-conscious shopper,		
I want	to see if a specific product is currently in stock at nearby supermarkets,		
So that	I can decide where to buy it without making an unnecessary trip.		
Acceptance criteria	Given I am a budget-conscious shopper, When I search for a specific product, Then I should see a list of supermarkets that have the product in stock near me.		
Test Case	<b>Description: To check availability of product</b>  <b>Steps:</b> <ol style="list-style-type: none"> <li>Search for product</li> <li>View item page</li> </ol> <b>Expected Outcome:</b> Availability of product on different stores is seen		



Story #4: Delivery Tracking		Priority	
		Effort	
As a	online shopper		
I want	to be able to track my order's delivery status in real-time,		
So that	I can be informed of its whereabouts and estimated time of arrival, ensuring I'm available to receive it.		
Acceptance criteria	<p>Given I have placed an order and it's ready for delivery, When I go to the 'My Orders' section in the application, Then I should see a 'Track Delivery' option for my recent order.</p> <p>Given I click on the 'Track Delivery' option, When the tracking page loads, Then I should see the current status of my delivery (e.g., "Out for delivery", "Arriving soon", "Delayed").</p> <p>Given I'm on the delivery tracking page, When I look at the delivery details, Then I should see an estimated time of arrival (ETA) for my order.</p>		
Test Case	<p>Test Case: Track delivery Preconditions: Logged into an account. Steps 1. Make a mock purchase to be delivered 2. Track the delivery drive Expected result: Delivery driver is accurately tracked. •</p>		

Story #5: Product Search and Categorization		Priority	
		Effort	
As a	online user		
I want	to easily search for products and view them categorized,		
So that	I can quickly find what I'm looking for without browsing through unrelated items.		
Acceptance criteria	<p>Given I'm on the app homepage, When I use the search bar, Then I should see relevant product results based on my search query.</p> <p>Given I'm browsing products, When I look at the navigation or filter options, Then I should see products categorized (e.g., "Dairy", "Electronics").</p>		
Test Case	<p>Test Case: Search for items Steps: 1. Input an item into the search bar 2. Check if system returns related items Expected output: System only shows items related to the search term. •</p>		

Story #6: Price Comparison		Priority	
		Effort	
As a	Online Shopper		
I want	to be able to compare the price of a product at different supermarkets		
So that	I can save on spending		
Acceptance criteria	<p>Given I am an online shopper, when I click on a searched product, Then I get a list of all prices for that product in supermarkets around me.</p>		
Test Case	<p>Test Case: Compare prices for items from different stores</p> <p>Steps:</p> <ol style="list-style-type: none"> <li>1. Search for an item</li> <li>2. Click the item from the search page</li> </ol> <ul style="list-style-type: none"> <li>Expected result: The prices for the item from different stores are displayed.</li> </ul>		

Story #7: Saved Items Page		Priority	
		Effort	
As a	logged in online shopper		
I want	to be able to view a list of my saved items		
So that	I can make changes to my selection		
Acceptance criteria	<p>Given I am a logged in online shopper, when I click on the saved option on the navigation bar, Then I get a list of all products I have added to my list along with their respective locations, prices and quantities.</p>		
Test Case	<p>Test Case:</p> <p>Verify a logged-in online shopper can access and view their list of saved items, including product details.</p> <p>Preconditions: User is logged in to an account</p> <p>Steps:</p> <ol style="list-style-type: none"> <li>1. Visit the saved items page</li> </ol> <p>Expected result: System returns all the items saved by this account.</p> <ol style="list-style-type: none"> <li>1. Item Checkout</li> </ol>		

Story #8: Home Page		Priority	
		Effort	
As a	deal-seeking shopper		
I want	to have quick access to the latest deals, discounts, and special offers on the website's home page		
So that	I can save money and take advantage of limited-time promotions		
Acceptance criteria	Given I open the website's home page, When I scroll or navigate to the "Ongoing offers" section, Then I should be presented with a curated list of current deals and offers.		
Test Case	Description: Home page has good design for it  Steps: 1. Open the website  Expected outcome: Good designed home page that clearly shows all the features.		

Story #9: Filter Search		Priority	
		Effort	
As a	price-conscious shopper		
I want	to filter all the search results for an item to less than a certain amount		
So that	I can stay in my budget.		
Acceptance criteria	Given that I am on the Search results page, When I click on the the "Filters" option Then I can see the option to filter the item by price		
Test Case	<b>Filter Search Results by Price</b> <ul style="list-style-type: none"><li>Description: As a price-conscious shopper, I want to filter search results for an item to less than a certain amount to stay in my budget.</li></ul> <b>Test Case: Filter Search Results by Price</b> <ul style="list-style-type: none"><li>Scenario: The user searches for a specific item and applies a price filter.</li><li>Steps to Reproduce:<ul style="list-style-type: none"><li>a. Enter the item's name in the search bar.</li><li>b. Apply a price filter of less than \$50.</li></ul></li><li>Expected Outcome: The search results should only display items with a price less than \$50.</li></ul>		

Story # 10 Filter Search		Priority	
		Effort	
As a	online shopper who does not like the quality of one brand,		
I want	to filter all the search results for an item to show the item in all brands except that brand,		
So that	I can save time while searching.		
Acceptance criteria	<p>Given that I am on the Search results page, When I click on the the "Filters" option Then I can see the option to filter the item by brand.</p>		
Test Case	<p><b>Filter Search Results by Brand</b></p> <ul style="list-style-type: none"> <li>Description: As an online shopper who does not like the quality of one brand, I want to filter search results for an item to show the item in all brands except that brand to save time while searching.</li> </ul> <p><b>Test Case: Filter Search Results by Excluding Brand</b></p> <ul style="list-style-type: none"> <li>Scenario: The user searches for a specific item and excludes a particular brand.</li> <li>Steps to Reproduce:             <ol style="list-style-type: none"> <li>Enter the item's name in the search bar.</li> <li>Apply a filter to exclude a specific brand.</li> </ol> </li> <li>Expected Outcome: The search results should not display items from the excluded brand.</li> </ul>		

Story # 11 Category search		Priority	
		Effort	
As a	user looking for specific products		
I want	to navigate through a category dropdown on the website's home page to easily filter and explore different product categories and subcategories,		
So that	I can quickly find the items I need.		
Acceptance criteria	<p>Given I open the app's home page, When I interact with the category dropdown menu, Then I should see a list of main categories, including "All," "Specials," "Grocery," "Meat," "Dairy," and more.</p>		
Test Case	<p>Description: Verify that a user can view items organized by their respective categories on the category page and using filters.</p> <p>Preconditions: User is on the category page.</p> <p>Steps:</p> <ol style="list-style-type: none"> <li>Navigate to a specific category page (e.g., "Electronics," "Clothing," "Home Decor").</li> <li>Observe the displayed items within the chosen category.</li> </ol> <p>Expected Result: The user can view a list of items that belong to the selected category, and only items that match the filters selected.</p>		

Story # 12		Category search	Priority	
			Effort	
As a	health-conscious shopper			
I want	to be able to easily filter and explore organic products within specific categories using the app's category dropdown,			
So that	I can make healthier choices while shopping.			
Acceptance criteria	Given I open the app's home page, When I interact with the category dropdown menu, Then I should see an option to filter for "Organic" products within each main category.			
	Description: Check that the category section is working.  Preconditions: User is on the category page.  Steps:  1. Choose category from menu 2. Get all items in that category  Expected Result: User get items in that category.			

Story # 13    Product Details Section		Priority	
		Effort	
As a	user allergic to certain allergens,		
I want	to check the ingredients list and get information on the allergens contained in a product		
So that	I can buy the safe product for myself.		
Acceptance criteria	<p>Given that I can see a searched product, When I click on the product and scroll to the "Product Details" section, Then I can see the entire ingredients list and allergen warnings for the same.</p>		
Test Case	<p>Description: Show details of a product to the user</p> <p>Steps:</p> <ol style="list-style-type: none"> <li>1. Click on product to visit its product page.</li> <li>2. Observe the displayed product details, including:</li> </ol> <p>Expected Result: The user is able to view the Product name, Description, Images of the product, Price, Shipping information, Ingredients/Contents and Customer reviews specified product information on the product detail page.</p>		

Story # 14    Product Details Section		Priority	
		Effort	
As a	shopper confused about the storage of various products,		
I want	to know the storage instructions for a product		
So that	I can store it in the correct manner.		
Acceptance criteria	<p>Given that I am on the search results page, When I click on a product and scroll to the "Product Details" section, Then I can see the storage instructions provided by the seller for that product.</p>		
Test Case	<p><b>Test Case: Display Storage Instructions for a Product</b></p> <ul style="list-style-type: none"> <li>• Scenario: The user selects a specific product and seeks its storage instructions.</li> <li>• Steps to Reproduce:             <ol style="list-style-type: none"> <li>a. Browse the product catalog.</li> <li>b. Select a product of interest.</li> <li>c. Look for storage instructions on the product details page.</li> </ol> </li> <li>• Expected Outcome: The product details page should clearly display the storage instructions for the selected product.</li> </ul>		

Story # 15      Item Checkout Page		Priority	
		Effort	
As a	online shopper,		
I want	to be able to view my selected items one last time during checkout,		
So that	I know which items are from which store(s).		
Acceptance criteria	<p>Given that I am on the "Saved Items" page, When I click on the the "Checkout" option Then I am taken to the checkout page where I can see all the items and their price(s) listed store wise.</p>		
Test Case	<p>Description: To buy the item required</p> <p>Preconditions: User is on the checkout page with an item selected.</p> <p>Steps:</p> <ol style="list-style-type: none"> <li>1. Select "Purchase" button.</li> <li>2. Enter address.</li> <li>3. Choose payment method</li> <li>4. Confirm payment details</li> </ol> <p>Expected Result: The user's order will go through and be delivered to them.</p>		

Story # 17      User Signup		Priority	
		Effort	
As a	user who wants to make purchases		
I want	to create a new account		
So that	I can save items to order		
Acceptance criteria	<p>Given that I am on the sign up page When I input details and hit submit Then I my account is created and added to database.</p>		
Test Case	<p><b>Check if adding properly in backend</b></p> <ul style="list-style-type: none"> <li>• Check if the repository is adding to the database using a mock service</li> </ul> <p><b>Check if form exist on frontend</b></p> <ul style="list-style-type: none"> <li>• Check if the form element can post data in frontend</li> </ul>		

<b>Story # 16</b>		<b>Priority</b>	
<b>User login</b>		<b>Effort</b>	
<b>As a</b>	user who wants to purchase items for cheap		
<b>I want</b>	to log into my account		
<b>So that</b>	i can make orders		
<b>Acceptance criteria</b>	<p>Given that I am on the login page, When I enter my details and hit submit Then I can see the option to filter the item by brand.</p>		
<b>Test Case</b>	<p><b>Get user account information in backed</b></p> <ul style="list-style-type: none"> <li>• Use a mock user to see if the login page returns the same user as used for the mock.</li> </ul> <p>Check if form works on frontend</p> <ul style="list-style-type: none"> <li>• Check if the form element exists and posts data in frontend</li> </ul>		



cosc2299-sept-2023 / Projects / SEPT Feature Project

Q Type to search

+ -

🔄

🔍

📄

⚙️

SEPT Feature Project

Home Sprint 0 Next iteration Planning + New View

Filter by keyword or by field

Discard Save

Title	Assignees	Status	Iteration	Estimate
1 Price comparison #1	udit710	Done	Iteration	-
2 Comparison view design/wireframe #2	udit710	Done	Iteration	-
3 Item Checkout #3	ra5h1k	Backlog	Iteration	-
4 Navbar #5	Vidyut03	Backlog	Iteration	-
5 Saved Items List #6	udit710	Done	Iteration 2	-
6 Search Items #13	IbrahimAshhab	Done	Iteration	-
7 Product Detail #7	ra5h1k	Backlog	Iteration	-
8 Saved Items design/wireframe #8	udit710	Done	Iteration	-
9 Item Checkout/wireframe #10	ra5h1k	Backlog	Iteration	-
10 Price Comparison #9	udit710	Done	Iteration	-
11 Product Detail/wireframe #11	ra5h1k	Backlog	Iteration	-
12 Home page Wireframe #12	Vidyut03	Done	Iteration 2	-
13 Architecture #14	AbhijeetK520	Backlog	Iteration	-
14 Search Items API for search button #15	IbrahimAshhab	Backlog	Iteration	-
15 Navigation Bar #16	IbrahimAshhab	Done	Iteration	-
16 Category Selection Wireframe #18	Vidyut03	Done	Iteration 2	-

+ You can use Control + Space to add an item

## New Additions in Milestone 2

### User Stories:

Story #16		User login	Priority	
			Effort	
As a	user who wants to purchase items for cheap			
I want	to log into my account			
So that	i can make orders			
Acceptance criteria	Given that I am on the login page, When I enter my details and hit submit Then I can see the option to filter the item by brand.			
Test Case	<b>Get user account information in backed</b> <ul style="list-style-type: none"><li>• Use a mock user to see if the login page returns the same user as used for the mock.</li></ul> <b>Check if form works on frontend</b> <ul style="list-style-type: none"><li>• Check if the form element exists and posts data in frontend</li></ul>			

Story #17		User Signup	Priority	
			Effort	
As a	user who wants to make purchases			
I want	to create a new account			
So that	I can save items to order			
Acceptance criteria	Given that I am on the sign up page When I input details and hit submit Then I my account is created and added to database.			
Test Case	<b>Check if adding properly in backend</b> <ul style="list-style-type: none"><li>• Check if the repository is adding to the database using a mock service</li></ul> <b>Check if form exist on frontend</b> <ul style="list-style-type: none"><li>• Check if the form element can post data in frontend</li></ul>			

miro

### Wireframes:

SUPERPRICE

HOME

CATEGORY

CURRENT SALES

Search

☐

Saved

Login

email

password

Login

No Account? Sign up here

The screenshot shows a web application interface for 'SUPERPRICE'. The top navigation bar includes links for 'HOME', 'CATEGORY', and 'CURRENT SALES', along with a search bar, a 'Saved' button, and a user profile icon labeled 'KL'. The main content area features a 'Create Account' form with three input fields for 'email', 'password', and 're-enter password', followed by a 'Create Account' button. A link for 'Already member? Log in here' is located at the bottom of the form.

Summary of new requirements/code/tests:

1. Frontend:

- React init files (#39 PR)
- Navigation Bar (#40 PR)
- Search Results page (#43 PR)
- Product details page components (#44 PR)
- Price comparison components (#45 PR)
- Routing for search page (#47 PR)
- Footer (#50 PR)
- Home Page (#51 PR)
- Routing for Home Page (#54 PR)
- Integrate product API data (#108 PR)
- Integrate search product data (#129 PR)
- Product review components (#124 PR)
- Offers based on discounts in Home Page (#138 PR)
- Product Details routing (#137 PR)
- Updated the UI for product Details Page (#137 PR)

2. Backend:

- Product API endpoints (#48 PR)
- Image Data (#48 PR)
- Review API endpoints (#86 PR)

- Orders API endpoints (#89, 91, 92 PR)
- Data loading for product details page from backend to frontend (#108 PR)
- Data loading for search product page from backend to frontend (#129 PR)
- User API endpoints (#111 PR)
- Category API (#122 PR)
- Product Search API endpoints (#118 PR)

3. Model:

- Updated database from MySQL to H2 (#52 PR)
- Product price Database (#88 PR)
- Created SubCategory Model (#123 PR)
- Updated Product Model (#123 PR)
- Created Category Model( #126 PR)

4. Testing:

- Added frontend testing library (#71 PR)
- Orders API testing (#94 PR)
- Testing search results page (#95 PR)
- Testing for Review API (#86 PR)
- Testing for User API (#111 PR)