Product Requirements

**Project**: Go-to Grocery App

**Team**: D.O.M.E

**Revision History**

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| 09-17-2025 | 1.0 | Software requirement | D.O.M.E |
| 10-14-2025 | 1.5 | Small additions were made to reflect features we plan to implement | D’yanna |

**Brief Problem Statement**

Rising food costs, dietary restrictions, and the difficulty of eating healthy while staying within limits have made grocery shopping a challenge for many people. Food waste is also becoming a bigger problem since consumers often ignore expiration dates or fail to keep track of what they’ve purchased. These issues lead to wasted money, poor eating habits, and less environmentally friendly lifestyles.

The motivation for the Go-to Grocery app is to help users shop smarter and healthier by using technology to solve these everyday challenges. The app allows users to save money by comparing local prices, avoid allergen foods by filtering ingredients, and find healthier grocery options based on personal goals. By scanning groceries, tracking calories, and receiving expiration reminders, users can reduce food waste and support better physical and mental health. This project is designed to provide an easy-to-use tool that makes grocery shopping more efficient, affordable, and aligned with healthier lifestyles.

**Stakeholders**

**“Software Engineering Team (Team D.O.M.E)**

The software engineering team consists of six team members who are responsible for the complete development of the Go-to Grocery app. Team D.O.M.E manages every stage of the project, from planning, design/implementation to testing and deployment.

**Responsibilities include:**

* Outlining the project requirements.
* Developing and implementing various features: login, scanning, notifications and health tracking.
* Conduct software testing and debugging.
* Dividing tasks and monitoring progression.
* Resolving any issues that occur during project lifestyle.

**Team D.O.M.E- Product Owner**

Since Go-to Grocery app is a student led capstone project with no external funding or sponsors. The Product Owner is shared among all the six team members of Team D.O.M.E.

**End Users- Students, Families, Teachers and Users**

These are the primary users of the Go-to Grocery app. The users are individuals aged 13 and above who want to shop smarter, save money, avoid allergens, and manage healthier lifestyles. To ensure project success, their needs must be gathered through surveys, interviews, and feedback sections. Selected users may also test early versions of the app for testing and accuracy of the features.

**User expectations include:**

* **User friendly interface**
* **Ability to avoid allergen foods.**
* **Healthier grocery options.**
* **Notifications for expiry dates and reminders.**
* **Track calorie intake and ability to compare local prices.**
* **AI food and recipe suggestions.**



**Fig: Stakeholders relation**

# **System Requirements**

The application is a centralized, cross-platform system for managing grocery lists and finding the lowest prices; the requirements are as follows:

* The project shall use JavaScript, HTML and CSS as the main programming languages.
* The project shall use React Native 0.81.x as a Framework.
* The project shall use a NoSQL database- Supabase as a Backend database.
* The project shall be compatible with the most recent web browsers versions.

# **Functional requirements (user stories)**

# List the Priority as 1 (High Priority - Critical) to 3 (Low Priority – Would be nice if we have time)

| **No** | **User Story Name** | **Description** | **Priority** |
| --- | --- | --- | --- |
| 1 | Registration | The user will be able to register with name, email, password, and profile details (e.g., food allergies/sensitivities and shopping goals such as budget). After registration, credentials are created and associated with the user’s profile. | R1 |
| 2 | Login | The user will be able to sign in using an email/password or Google single sign-on (OAuth). | R1 |
| 3 | Search | The user will be able to search for a grocery item by keyword and view retailers with current prices, including the lowest price and a recommended retailer. | R1 |
| 4 | Search w/ Filters | The user will be able to apply filters (e.g., price range, vegetarian or vegan, in-stock/on-sale) to refine search result | R1 |
| 5 | Scan | The user will be able to scan a product barcode in store to mark the item as purchased and automatically cross it off the shopping list; price data may be updated when available. | R1 |
| 6 | Shopping List | The user will be able to build and manage a shopping list from selected items, stores, and price points, including quantities, a running total, and links to store/location details. | R1 |
| 7 | Demographics/ Metrics | The user will be able to view monthly spending against a budget goal set during registration and review frequently purchased items to streamline future lists. | R2 |
| 8 | Location Services | The user will be able to enable location services to surface nearby stores and location-specific prices and deals. | R2 |
| 9 | AI Buddy | The user will be able to chat with the LLM to get insight on what recipes to use based on the food in their pantry, and if an item is out of stock, they will receive a suggestion for an alternative. | R2 |
| 10 | Pantry | Once a user confirms a purchase using the scan feature, the item will be added to the pantry, where its best-by date will be automatically tracked. | R1 |
| 11 | Settings | The user will be able to turn push notifications on or off and change information entered during the initial registration. | R3 |

**Non-Functional Requirements**

**Security**

NF-1: User login and data storage must be secured with encryption.

NF-2: The system should not allow unauthorized access to user accounts.

NF-3: The system should log out users after a certain time of inactivity.

**Reliability**

NF-4: The system should remain available and responsive for most part, making sure reminders and notifications are delivered on time.

NF-5: The system should back up data.

**Usability**

NF-6: The UI should allow users to enter/add grocery items in under a certain set of time.

**User Interface**

NF-7: App should present a consistent design across all screens and must be user friendly.

NF-8: App should use colors and texts to communicate dietary preferences and grocery status.

NF-9: System should provide error messages in understandable language (plain text) when input is invalid.

**Accessibility**

NF-10: Users should be able to adjust font size.

NF-11: The text and background should have enough contrast for readability.

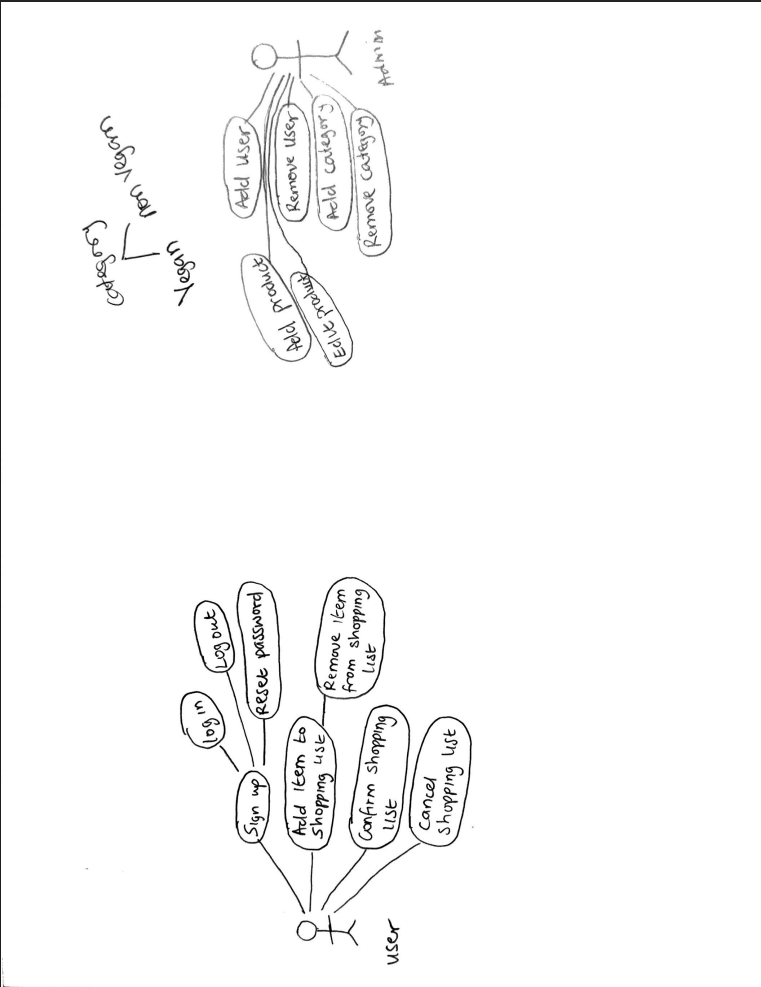
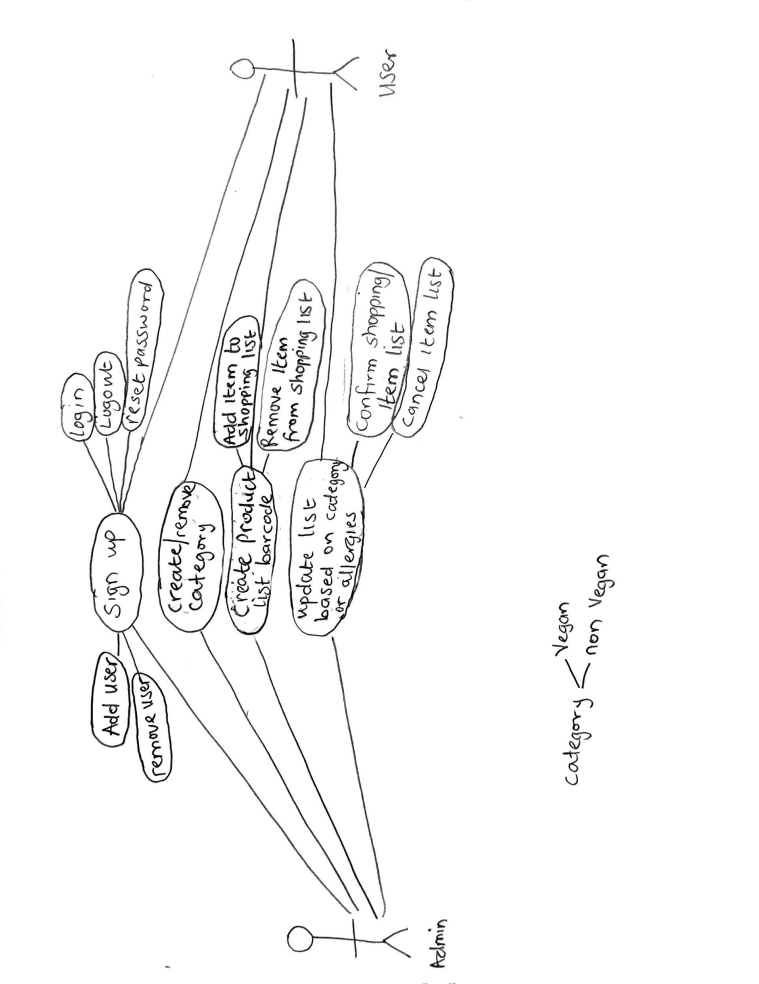
**Cross-Platform Compatibility**

NF-12: The app should run on Android and scale properly on both phones and tablets without loss of functionality.

**Accuracy**

NF-13: The barcode scanning feature shall accurately identify products at least 95% of the time.

NF-14: Notifications/reminders for expiry should be delivered at least 90% of the time within a set minute of the scheduled alert.



**Use case description**

| Use Case Number | UC‑01 |
| --- | --- |
| Use Case Name | Registration |
| Overview | The user enters personal details, creates an account, then sets preferences |
| Actor(s) | New user |
| Pre-condition(s) | * Not signed in * Internet available |
| Scenario Flow | Main Steps:   * 1) Enter name, email, password (+ confirmation). * 2) Accept Terms/Privacy. * 3) Submit. * 4) Account + profile created in Supabase. * 5) Onboarding (set allergies/sensitivities and determine a budget goal (can be skipped and changed later in Profile). * 6) Navigate to Home. |
| Alternate Flows | * Email already in use → “sign in” * Weak password → show rules * Offline → error * Server error → retry/cancel |
| Post Condition | * The system will store the user information when user clicks on   “Register”   * Account created * Home Page shown |

| Use Case Number | UC - 02 |
| --- | --- |
| Use Case Name | Login |
| Overview | The user shall enter his UserName and password to be able to log in |
| Actor(s) | Signed‑in user |
| Pre-condition(s) | * Account exists * internet available |
| Scenario Flow | Main steps   * Open app ->Sign in * 2) Enter username /password (or auto‑login / Google -> Auth success ) * 3) On success, go home. |

| Use Case Number | UC‑03 |
| --- | --- |
| Use Case Name | Search |
| Overview | The user can find a grocery item and see current prices by retailer |
| Actor(s) | Signed‑in user |
| Pre-condition(s) | * Catalog/price service available * internet available |
| Scenario Flow | Main steps   * Type keyword * The system returns products with retailers and prices. * System shows the items * User taps a result to view details or add to the list |
| Alternate Flows | * No results → empty state with tips * network error → retry |
| Post Condition | Results shown and users list created |

| Use Case Number | UC‑04 |
| --- | --- |
| Use Case Name | Search with Filters |
| Overview | The user will be able to apply filters to search relevant items |
| Actor(s) | Signed‑in user |
| Pre-condition(s) | * Search context active * Internet available |
| Scenario Flow | Main steps   * Choose Filters -> Set price range, in-stock/on-sale * Apply * Results update * Clear filters, update -> resets. |
| Alternate Flows | * Filters yield zero results -> show “No matches. * Offline → show last results |
| Post Condition | Filtered results shown; user can proceed to item details or add to list |

| Use Case Number | UC-05 |
| --- | --- |
| Use Case Name | Receive Expiry Notification |
| Overview | To reduce food waste, the system alerts the user before an item expires. |
| Actor(s) | Shopper(any individual managing groceries) |
| Pre-condition(s) | Items with expiration dates are kept by the user in the pantry. Notifications are activated. |
| Scenario Flow | * Every day, the system verifies expiration dates. * A notification is sent out if an item is inside the user's designated reminder window, such as two days before expiration. * The user sees the reminder. |
| Alternate Flows | Reminders aren't sent if the user disables notifications or if the expiry date is missing. |
| Post Condition | Users receive timely reminders about items that are about to expire, allowing them to take action. |

| Use Case Number | UC-06 |
| --- | --- |
| Use Case Name | View Reports & Insights |
| Overview | The software gives data on users' buying patterns, eating habits, and food consumption over time. |
| Actor(s) | Shopper(any individual managing groceries) |
| Pre-condition(s) | The user's buying history and pantry are saved. |
| Scenario Flow | * Users can navigate to the Reports section through the Settings menu. * System compiles and displays reports and charts on food waste, spending patterns, and nutritional habits. |
| Alternate Flows | Reports will be partial if the user has a limited history or filters data to view only spending or nutrition. |
| Post Condition | The user's buying history and pantry are saved. |

| Use Case Number | UC-07 |
| --- | --- |
| Use Case Name | Custom Reminder Settings |
| Overview | Users can customize their expiry notifications, choosing when they want to receive reminders. (eg. 1 day before, 1 week ago) |
| Actor(s) | Shopper(any individual managing groceries) |
| Pre-condition(s) | Users can track pantry items and their expiry dates. |
| Scenario Flow | * User opens Notifications through the Settings menu. * User chooses a reminder window (e.g., 2 days before expiry). * Notifications follow updated rules. |
| Alternate Flows | Users can select numerous reminder points if they want to receive expiration notifications, but if they disable all reminders, no notifications will be delivered. |
| Post Condition | Notifications match the user's selected preferences. |

| Use Case Number | UC-08 |
| --- | --- |
| Use Case Name | Barcode Scan for Quick Add |
| Overview | Users can quickly add product details to their pantry by scanning an item's barcode, eliminating the need for manual entry. |
| Actor(s) | Shopper(any individual managing groceries) |
| Pre-condition(s) | Rear camera enabled |
| Scenario Flow | * User scans the barcode of the item(s). * The system retrieves product information (brand, nutrition, and expiration date). * The item is saved. |
| Alternate Flows | If the barcode isn't recognized, the system will prompt for manual entry. |
| Post Condition | The item has been added with accurate details. |

| Use Case Number | UC-09 |
| --- | --- |
| Use Case Name | Pantry Management and Food Status Tracking |
| Overview | Users can view and manage all pantry items, including their freshness status—such as *Good to Eat*, *Expiring (date)*, *Spoilage Approaching (date)*, and *Expired (date)*—and take actions like removing items or adding them to the shopping list for repurchase. Items are automatically sorted into categories based on their condition, prioritizing those that require attention. |
| Actor(s) | Shopper(any individual managing groceries) |
| Pre-condition(s) | User has added food items to the pantry through the barcode scan feature or manual entry. |
| Scenario Flow | * User opens the Pantry feature. * The system displays all items sorted by freshness status. * User views each item’s condition and expiration date. * User selects an item to either remove it or add it to the shopping list. * The system updates the pantry and shopping list accordingly. |
| Alternate Flows | If no items are available in the pantry, the system will prompt the user to scan or add new products. |
| Post Condition | The pantry is updated with accurate item statuses, and any selected items are removed or added to the shopping list. |

| Use Case Number | UC-10 |
| --- | --- |
| Use Case Name | AI Buddy for Recipe and Ingredient Assistance |
| Overview | Users can interact with the AI Buddy to generate personalized recipe suggestions based on the items currently available in their pantry. If a searched ingredient or pantry item is out of stock, the AI Buddy will provide suitable alternatives to help the user complete their recipe without interruption. |
| Actor(s) | Shopper(any individual managing groceries) |
| Pre-condition(s) | User has an active pantry with items logged through the scan feature or manual entry, and the AI Buddy feature is enabled. |
| Scenario Flow | * User opens the AI Buddy feature. * User asks for recipe ideas using items available in their pantry. * The system analyzes pantry contents and compiles relevant recipe options. * If an ingredient is missing or out of stock, the system suggests an appropriate substitute. * The user reviews the recipe and may choose to save, modify, or add missing items to their shopping list. |
| Alternate Flows | If no sufficient ingredients are available in the pantry, the system will recommend simple recipes requiring minimal additional ingredients or prompt the user to restock specific items. |
| Post Condition | The user receives tailored recipe recommendations and substitution suggestions, enhancing meal planning and reducing food waste. |