

# Design doc for Sprint 1

## Accomplishments and Task Contributions

During this sprint, our team focused on the design phase of the Go-To Grocery List app. We completed the Design Rationale, User Interface Wireframes, ER Diagram, Design Summary, and UML Diagram. The design defines a three-layer structure—data, feature services, and user interface—to support future features like AI recommendations and allergy checks. Users can navigate key tabs such as Inventory, Search, Lists, Analytics, and Settings, covering core features like barcode scanning, filters, and expiry reminders. We discussed using React Native for cross-platform development and Firebase for data storage. These deliverables established the app's architecture, workflow, and foundation for development in the next sprint.

### Contribution:

- Subol Dhital — Wrote the Design Rationale section, explaining the reasoning behind key design choices.
- D'yanna Grey — Designed and developed the User Interface wireframes to visualize the app's layout and navigation.
- Olajumoke Kupoluyi — Created the ER diagram to illustrate the database structure and entity relationships.
- Elsa Joy — Wrote the Design Summary, outlining the overall structure and purpose of the system.
- Mikal Debesay — Created the UML diagram for the project, ensuring relationships and components were accurately represented.
- Saubhagya Bhandari — Combined all team documents into the final report and handled the submission of the assignment.

## Design Rationale

Our team is in the early process of designing and creating the Go-To Grocery App. The fundamental goal behind this software is to help individuals shop smarter by comparing grocery prices at several local supermarkets. We also encourage people to identify items that are suitable for their health needs, budget, and allergies. Since the app is still in its early stages, this section discusses the concepts we've discussed so far and why we're thinking in that direction.

At first, we considered creating a simple app that simply displayed pricing from nearby stores. However, we eventually discovered that people had various goals when purchasing. Some people value low-cost things, while others choose healthful or organic meals. There are also users who have allergies and choose to avoid ingredients of their choice. As a result, we decided to include filters that allow users to select between inexpensive, nutritious, or allergy-friendly solutions. We as a team decided that this would help our app to be more user-friendly.

We also discussed how to gather grocery price data. One option was to allow customers to

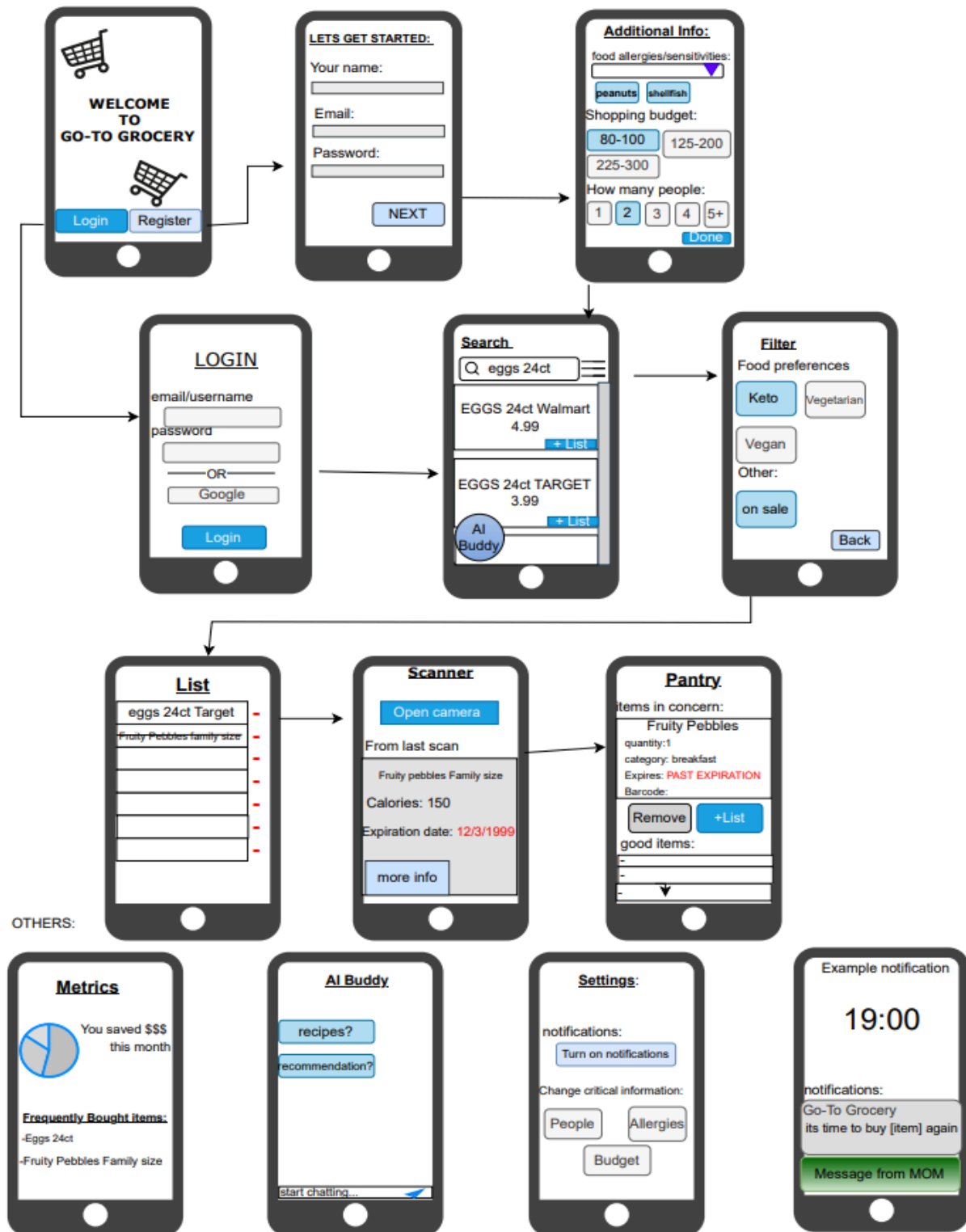
manually enter prices, but this might be time-consuming and inaccurate. Another suggestion was to allow location access so that the app could automatically display nearby stores and prices. This is more useful and saves time, but we also understand that location access requires us to maintain user privacy.

We haven't finalized the layout or tools yet, but we're considering using React Native to enable the app to operate on Android. We could store food and user data in a database such as Firebase or Superbase. In the future, we intend to include an AI recipe tool that may provide users with meal suggestions and ingredient lists based on their likes and budget.

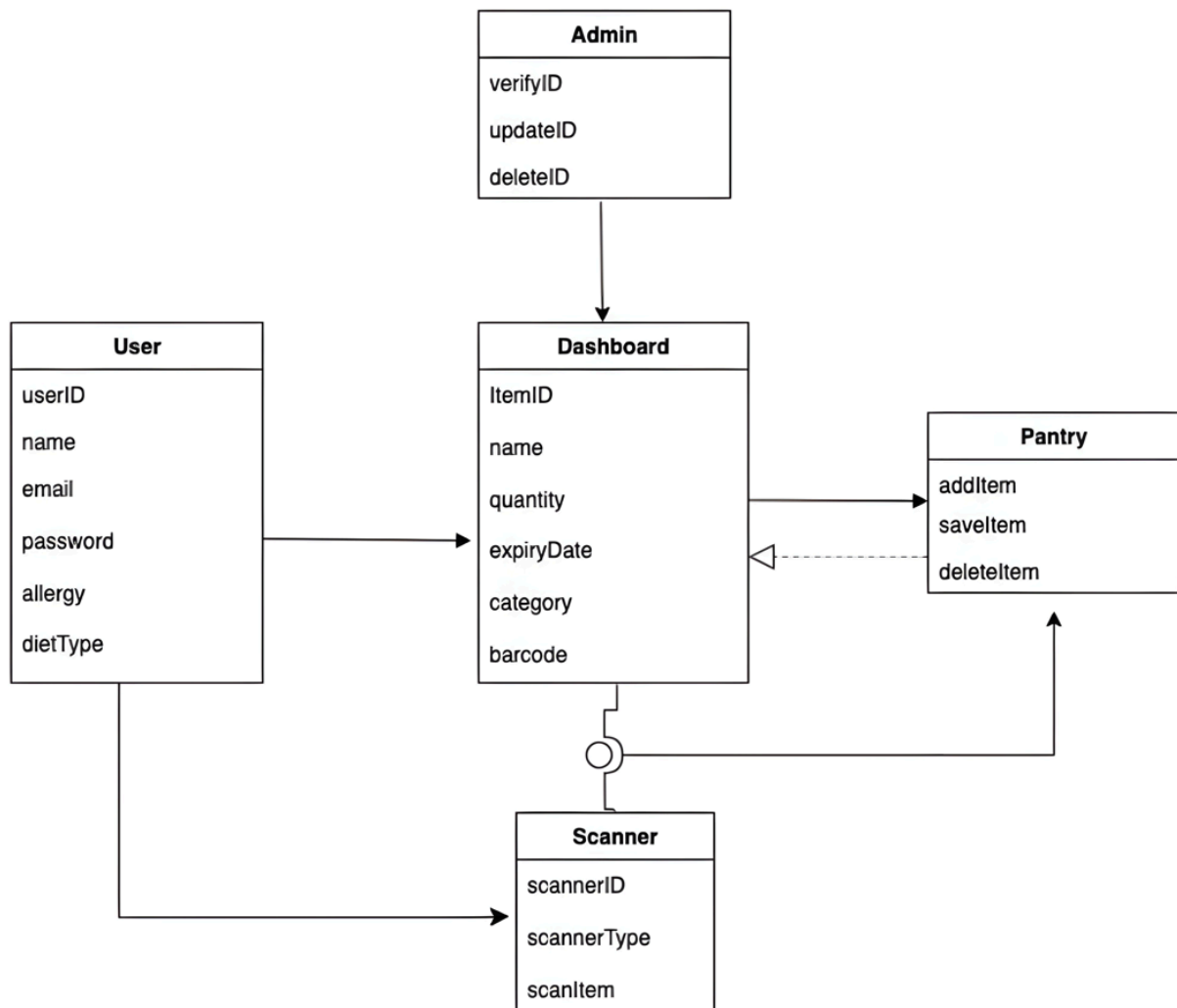
<b>Design Topic</b>	<b>Ideas We Talked About</b>	<b>Current Plan</b>	<b>Reasoning</b>
Getting Prices	Manual typing	Location based	Faster and easier, but it needs permission
Platform	Website or mobile	Mobile app	Easier to use while shopping
Health/Allergy	Optional	Core part of the app	Helps more types of users
Extra Feature	Grocery list only	AI recipe helper	Makes the app more creative and fun

We know our design will change as we move forward, so we will keep updating this section with new ideas and reasons for every change.

# User Interface Wireframes



## ER Diagram



## Design Summary

Three layers make up our application: data, feature services, and user interface screens. This will make it simple to add future AI, allergy checks, and pricing to the code. By comparing local prices and filtering for budgets, allergies, and health needs, the main purpose of the app is to assist consumers in making more informed purchasing decisions.

### UI & navigation.

Users are presented with the Inventory, Search, Lists, Analytics, and Settings tabs on Home after registering or logging in. This covers our top R1 stories (Search, Filters, Scan, Shopping List) and allows users to scan items, search and filter results, add to lists, and receive expiry reminders.

### Feature services (business logic)

- For the app to prefill a new pantry item after adding to the list and purchasing, the scanner service scans a barcode and puts it in the inventory.
- Each pantry item's name, quantity, category, expiration date, and barcode are managed by inventory.
- They are reminders, alerts, or notifications before expiry.
- Product availability and price range are narrowed down by Search & Filters; price comparison identifies the best deal.
- AllergyFilter saves user preferences and allergens, and the user doesn't get any product with the mentioned allergens. The other features can check items quickly.
- The purpose of AIRecommendation (a service) is to provide recipes or substitutes based on budget, allergies, and pantry.

### Data model (persistent objects)

As indicated in the class UML, there are User (profile), PantryItem, Notification/Reminder, and AllergyFilter. To support "lowest price" views while maintaining privacy, we link items with store data (location-based pricing when permission is given).

Platform & quality notes. The app is planned as a mobile app, so it's easy to use while shopping; cross-platform tech is under consideration to primarily reach Android. We will keep permissions minimal (Camera, Location optional, reward optional, and Notifications) and align with our NFRs for usability, accessibility, and accuracy of barcode scan/alerts.

This structure we have created now cleanly supports our R1 features (Registration/Login, Search & Filters, Scan, Shopping List, Reminders) and leaves room for R2 (Analytics, location-aware pricing, AI suggestions) without major rewrites.

## UML Diagram

