CS5010: Functional Programming Project: Recipe Ingredient Inventory & Suggestion App

While I'm not a self-proclaimed cooking enthusiast, I've often experienced the frustration of opening my fridge only to find that one key ingredient is missing - I usually make some ham,egg and cheese sandwiches and often find that I'm missing one of the three! This recurring inconvenience inspired me to create the **Recipe Ingredient Inventory & Suggestion App**. The application is designed to help everyday cooks manage their pantry inventory, automatically suggest recipes based on the ingredients on hand, and generate shopping lists for missing items. Built using React's functional components and Hooks—eschewing classes entirely—and leveraging Firestore for persistent storage, this project embodies a modern, modular approach to web development.

Business Requirements

- **Pantry Management:**
  Users can add, update, and delete pantry items, including details such as quantity, expiration date, and category.

- **Dynamic Recipe Suggestions:**
  The app will filter and suggest recipes that can be made with the current inventory. Users can also search and filter recipes based on cuisine, dietary preferences, or cooking time.

- **Automated Shopping List Generation:**
  When key ingredients are low or missing, the application automatically generates a shopping list for easy restocking.

- **Persistent Data Storage:**
  All data (pantry items, recipe preferences, shopping lists) is stored in Firestore, ensuring real-time updates and synchronization across devices.

- **User Interface:**
  The UI must be clean, intuitive, and responsive, supporting both desktop and mobile usage.

- **Modular and Maintainable Code:**
  The project will adhere to a functional programming paradigm with each component defined as a function (one export per file) and no global variables, ensuring a scalable and maintainable codebase.

Nouns & Verbs

Nouns:

- User
- Pantry
- Ingredient
- Recipe
- Shopping List
- Inventory
- ~~Preference~~
- ~~Notification~~
- ~~Firestore~~

Verbs:

- Add
- Update
- Delete
- Suggest
- Generate
- Save
- Retrieve
- Filter
- Sync
- ~~Organize~~

Target Audience

- **Everyday Home Cooks (Someone like me):**
  Individuals who seek a practical way to track pantry items and plan meals without being culinary experts.

- **Busy Professionals:**
  Users who require a quick and efficient tool to manage ingredients and generate shopping lists on the go.

- **Budget-Conscious Shoppers:**
  Those aiming to reduce food waste and avoid unnecessary purchases by utilizing what they already have.

Use Cases / User Stories

1. **Pantry Inventory Management:**
   The user should be able to use the app to add and edit their ingredients to always maintain an updated inventory. *As a user, I want to add and update pantry items so that I can always know what ingredients I have available.*

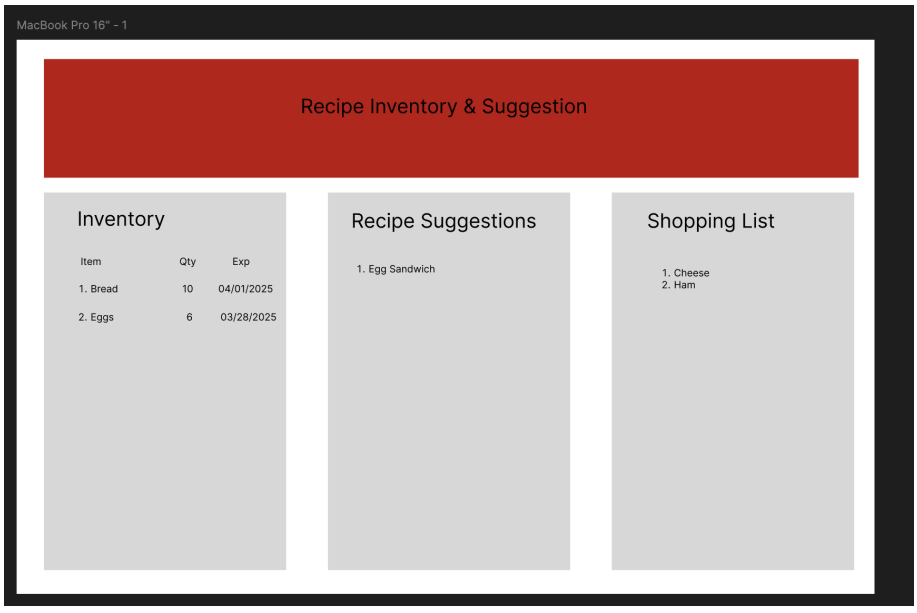
2. **Recipe Suggestions:**
   As a user, I want the app to suggest recipes based on my current pantry inventory, enabling me to decide what to cook without additional shopping.

3. **Shopping List Generation:**
   As a user, I want the app to generate a shopping list automatically for ingredients that are low or missing, ensuring my kitchen stays stocked.

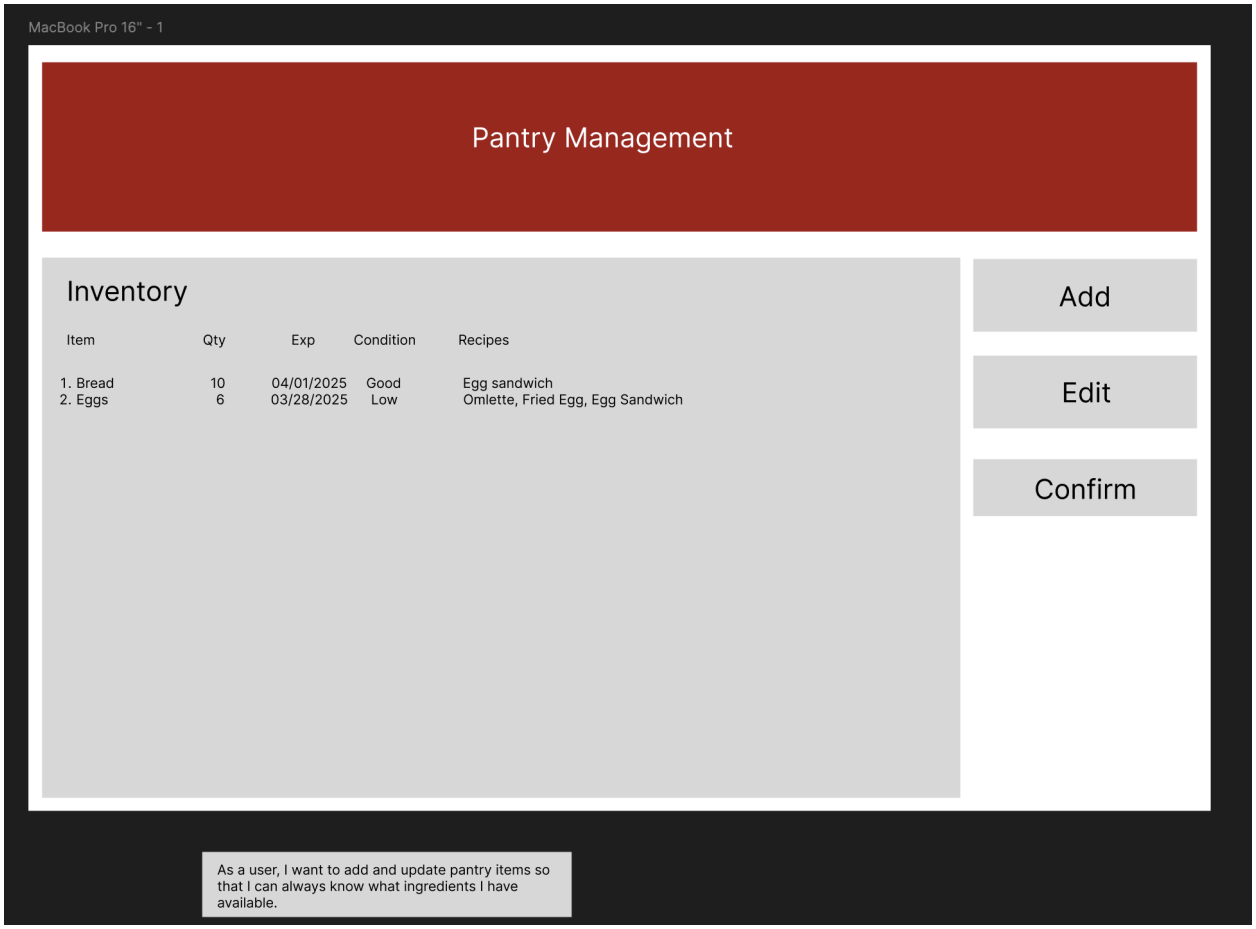
4. **Real-Time Data Synchronization:**
   As a user, I want my pantry data, recipe preferences, and shopping lists saved so that I can access them from any device.

Interface Mockups

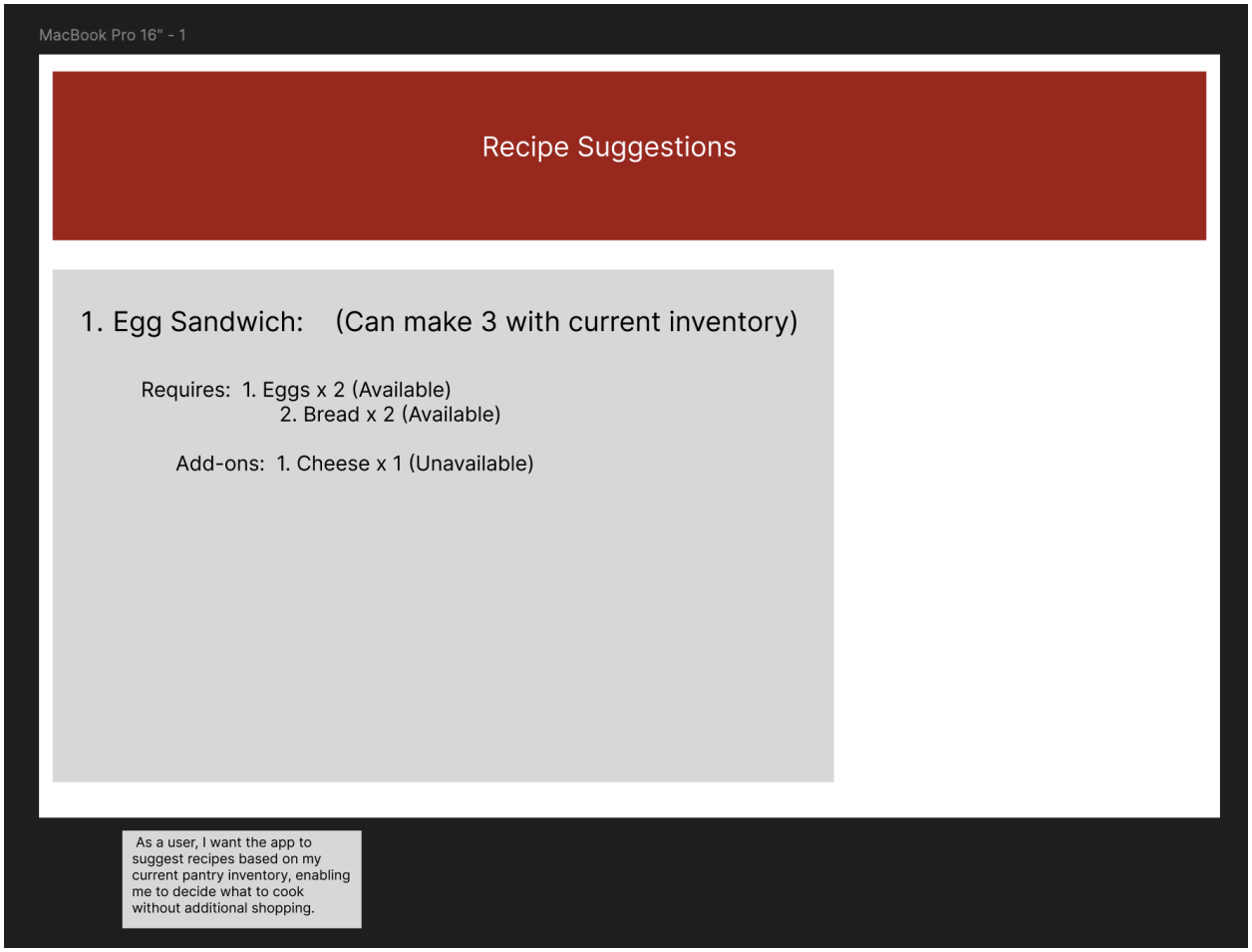
- **Dashboard Screen:**
  Displays an overview of the pantry inventory, a list of recipe suggestions, and a summary of the generated shopping list.



- **Pantry Management Screen:**
  A detailed interface for adding, editing, or removing ingredients. Includes fields for ingredient name, quantity, expiration date, and category. The design emphasizes ease of input and quick updates.

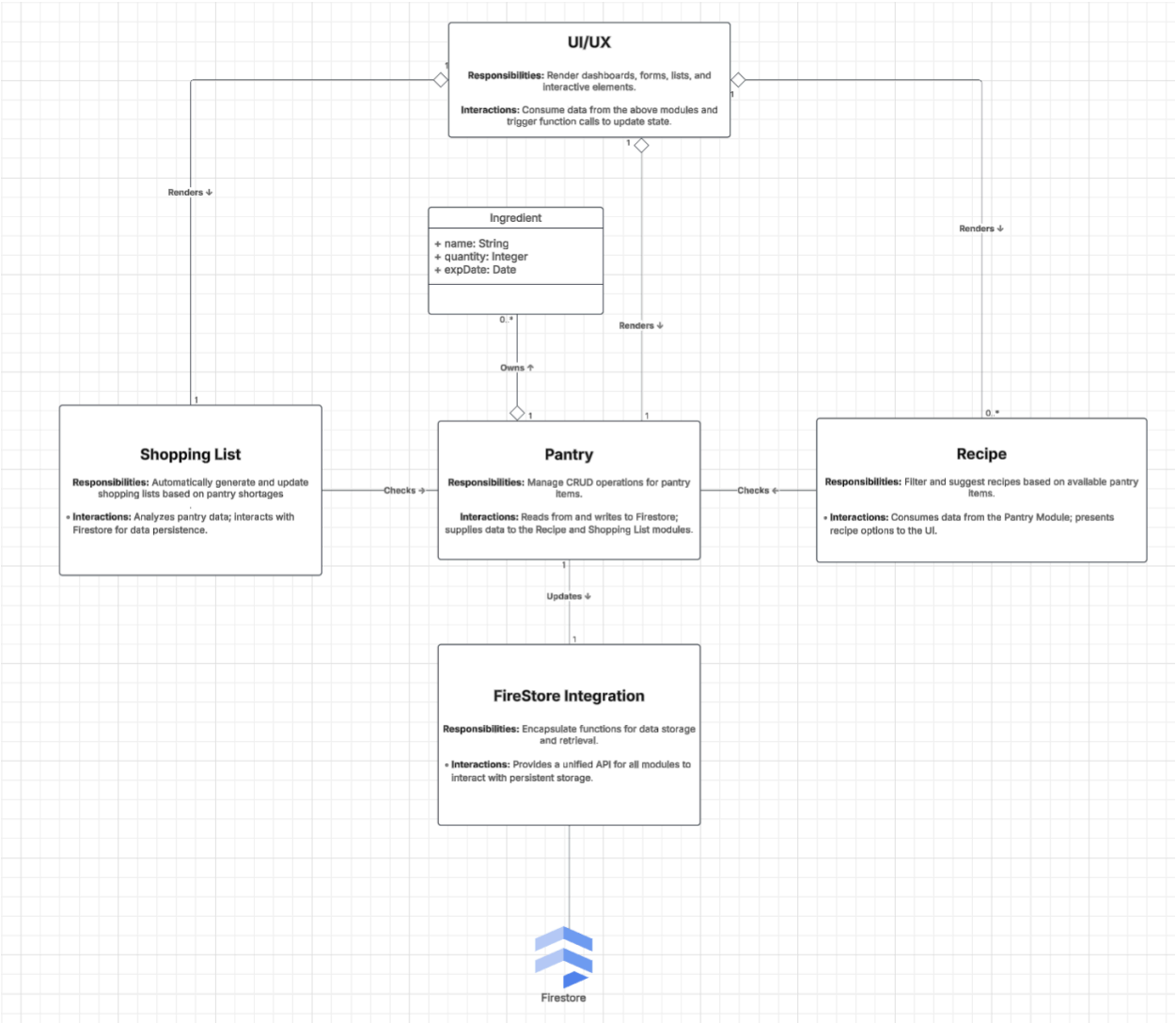


- **Recipe Suggestion Screen:**
  Presents a curated list of recipes that can be prepared with the current ingredients. Users can click on a recipe to view detailed instructions, nutritional information, and preparation time.



- **Shopping List Screen:**
  Organizes missing or low-stock ingredients into a checklist format, allowing users to mark items as purchased and update the inventory accordingly.

Modules Diagram



- **Pantry Module:**
    - **Responsibilities:** Manage CRUD operations for pantry items.
    - **Interactions:** Reads from and writes to Firestore; supplies data to the Recipe and Shopping List modules.
- **Recipe Module:**

    - **Responsibilities:** Filter and suggest recipes based on available pantry items.
    - **Interactions:** Consumes data from the Pantry Module; presents recipe options to the UI.
- **Shopping List Module:**
    - **Responsibilities:** Automatically generate and update shopping lists based on pantry shortages.
    - **Interactions:** Analyzes pantry data; interacts with Firestore for data persistence.
- **FireStore Integration Module:**
    - **Responsibilities:** Encapsulate functions for data storage and retrieval.
    - **Interactions:** Provides a unified API for all modules to interact with persistent storage.
- **UI/UX Components:**
    - **Responsibilities:** Render dashboards, forms, lists, and interactive elements.
    - **Interactions:** Consume data from the above modules and trigger function calls to update state.