MealsMyWay Software Design Document Final

02/13/2025

Sponsor: Dr. Ana Paula Chaves Mentor: Paul Deasy

> Isaiah Swank Laura Guerrero Maximilian Poole Colin MacDonald



Table of Contents

Introduction	3
Implementation Overview	4
Architectural Overview	5
Design:	5
Discussion:	6
Module and Design Interface Descriptions	7
Module and Design Interface Discussion:	7
Class Diagrams:	8
Implementation Plan	10
Design:	9
Discussion:	10
Conclusion	12

Introduction

Meal prepping is a valuable way to maintain a healthy lifestyle, save time, and reduce food waste. However, it can quickly become overwhelming due to the effort involved in managing recipes, planning meals, and organizing grocery shopping. For people with busy schedules, meal prepping can help reduce stress and free up time, but the process itself can feel like a lot of work. Additionally, more and more people are looking to add a social aspect to meal prepping. Whether it's sharing recipes, meal plans, and grocery lists with friends or family; which only adds another layer of complexity.

While there are apps available that aim to support meal prepping, they often come with significant limitations. Many restrict users to use specific ingredients and/ or meal types, which can affect users with dietary restrictions or preferences. Additionally, these apps often fail to consider ingredients users already have at home when generating shopping lists or recipes, which leads to unnecessary purchases and food waste. Another common drawback is the lack of social sharing and collaboration on recipes. Most existing apps do not allow users to easily share recipes, meal plans, or grocery lists with friends or family members, which makes it difficult to coordinate and collaborate with others on meal prep. To address these issues, the development of MealsMyWay offers a solution that makes meal prepping not only more efficient but also more flexible and social. By utilizing both web applications and mobile platforms, the app provides users with a convenient way to organize and manage their meal prep. Users will be able to fully customize meal prep calendars, easily manage their recipes, and generate smart shopping lists based on the ingredients they already have on hand. The app will also feature robust sharing capabilities, allowing users to collaborate with others by sharing grocery lists, recipes, and meal plans. Additionally, as a bonus feature, artificial intelligence may be integrated to recommend meals based on user preferences and past meal history.

Meal planning is a growing need for convenient, waste-reducing meal solutions among busy families and individuals. MealsMyWay seeks to fill a unique niche in this market with family-centered, customizable, and social features. Dr. Ana Paula Chaves, the project sponsor and a mother of two, developed a manual meal prep system during the pandemic to balance her teaching career and family needs. Inspired by the benefits and challenges of this approach, she envisioned MealsMyWay as a flexible, collaborative platform to simplify meal prepping.

Implementation Overview

Our vision is to provide a free to use application that supplies users with recipes that they are free to choose and assign onto a given day, and recipe a recipe and prep list generated by the app so they know what they need and what they need to do to create multiple dishes at the same time. We are using Ionic to create the application as it supports tri-fold programming and gives us the ability to concurrently make Android, IoS, and Web versions of the same app. It is an XML based language that has a very similar appearance to the HTML language but has special tags that automatically converts the inside code to be compatible for Android's Java programs and IoS's Swift programs.

We will be using a variety of packages that are built into Angular for this ionic project to enhance the project functionalities. We will be using node packet manager to allow for easy installation of these Angular packages. Packages such as HTTP client will allow us to use online APIs to fill in the recipe database and platform to determine a mobile vs web view to ensure a good user interface. These packages are just a few examples of the many that will be used throughout the implementation process, some of which have yet to be determined. The packages used during the development process are subject to change as the need for new or updated ones is required.

The ionic framework and sprint style approach to development combined with GitHub for version control will ensure that work is able to be completed regularly, in a timely manner, and with coordination from the team. Ionic is a well documented framework that is similar to languages that we currently have experience in which will allow for an easier learning curve during the development process. AWS will be used to manage the database and allow for our client to continue use and management of this program after the project has been completed.

Architectural Overview

Architecture Design:



Architecture Discussion:

The architecture consists of multiple responsibilities and features of each component through a service oriented architecture. The User Interface facilitates interaction between users and the system, allowing them to sign in and navigate features. The Calendar Presentation and Calendar Service handle meal scheduling, allowing users to view and edit their meal plans. The Pantry/Freezer Presentation and Pantry/Freezer Service track stored ingredients and allow users to update their inventory. The Recipes Presentation and Recipes Service enables users to browse, edit, and retrieve recipes, ensuring ingredient availability checks. The Profile Presentation and Profile Service manage user profile settings. Finally, the Database stores all necessary data, including user meal plans, ingredient inventory, recipes, and profile information.

The architecture follows a layered approach where the User Interface communicates with presentation components, which in turn interact with corresponding services. Services perform business logic operations and retrieve or modify data in the Database. Information flows between services, allowing cross-feature functionalities, such as checking recipe ingredients against the pantry or verifying scheduled meals in the calendar. The system enables bi-directional interactions, meaning updates made in one component (e.g., pantry inventory) can influence other components (e.g., recipes and calendar). The Database serves as the central repository ensuring data consistency across all modules.

This architecture embodies the Model-View-Controller style by separating presentation as the user interface, business logic as the services, and data storage as the database. The Client-Server Architecture is evident, as the user interface acts as a client requesting services from backend components. It also reflects a Layered Architecture, where interactions flow through structured layers, ensuring modularity and maintainability. Additionally, the system incorporates elements of a Microservices-like Approach, as each service is responsible for a specific function, promoting scalability and separation of concerns.

Module and Design Interface Descriptions

Module and Design Interface Discussion:

The systems detailed in this section, represented by the class diagrams, include the Calendar, Recipe, Pantry/Freezer, and Settings components. Each module plays a crucial role in managing meal planning, inventory tracking, and user preferences. The Calendar system organizes meal planning on a weekly basis, allowing users to assign multiple recipes to specific days. Before generating a grocery list, it cross-references the pantry and freezer inventory to determine which ingredients are already available, ensuring that only missing items are added to the list. As meals are prepared, the pantry is automatically updated to reflect ingredient usage. Users can modify meal plans, adjust meal placements, and share plans with others.

The Recipe system stores and manages recipes, each containing a list of required ingredients. Users can assign recipes to the calendar, search for recipes based on their pantry stock, and bookmark favorites for quick access. The system also supports meal portioning, allowing users to store leftovers in the freezer with designated portion counts for future meal planning. The Pantry/Freezer system keeps track of all ingredients available to the user, ensuring seamless integration with the calendar for meal planning and grocery management. The pantry is updated automatically when meals are prepared, and users can manually adjust inventory by adding, increasing, decreasing, or removing items. The freezer stores leftovers and frozen meals, which can be reassigned to future meal plans.

The Settings system allows users to manage their profile information, including name, email, and privacy preferences. It determines whether recipes and meal plans are shared with others or kept private and enables users to manage shared meal plans. Additionally, the settings module provides authentication features, allowing users to sign out securely and adjust privacy settings to control data visibility.

Class Diagrams:

Calendar: The Calendar component is responsible for managing meal plans while ensuring an efficient grocery shopping experience by checking pantry and freezer stock

before generating lists. This minimizes waste and optimizes planning.

- *CalendarPresentation*: This class provides the user interface functionality for displaying, updating, and managing meal plans. It allows users to:
 - View their meal plans.
 - Modify existing meal plans.
 - Update the calendar interface to reflect changes.
 - Display grocery lists based on missing ingredients.
 - Show preparation lists for upcoming meals.
- *CalendarService*: This backend class handles core meal-planning logic, ensuring a streamlined experience. It:
 - Generates grocery lists by comparing required ingredients with pantry/freezer stock.
 - Creates preparation lists for scheduled meals.
 - Supports meal plan adjustments, including changing dates and removing meals.
 - \circ $\,$ Allows users to edit meal placements within the plan.
 - \circ Provides functionality for sharing meal plans with others.

CalendarPresentation		CalanderService
	CalendarPresentation day: list name: string grocery: list prep: list mealPlan: list[list] displayMealPlan() updateMealPlan() updateMealPlan() displayGroceryList() displayPrepList()	+ day: list + name: string
+ day: list + name: string + grocery: list + prop: list		+ grocery: list + prep: list + mealPlan: list[list]
+ mealPlan: list[list]		+ generateGroceryList()
+ displayMealPlan() + updateMealPlan() + updateDayView() + displayGroceryList() + displayPrepList()		+ generatePrepList() + changeMealPlanDate() + removeMealFromPlan() + editMealPlacement() + shareMealPlan()

Recipe: The Recipe component enables users to browse, search, and manage their recipes, including favoriting frequently used ones. It integrates with the pantry system to help users filter recipes based on available ingredients.

- *RecipePresentation*: This class is responsible for the front-end user interaction related to recipe management. It allows users to:
 - View a list of recipes.
 - Search for recipes based on keywords or filters.
- *RecipeService*: This backend class supports recipe creation, storage, and searching, handling the following operations:
 - \circ Creating new recipes.
 - Adding recipes to the user's collection.

- Searching for recipes by name.
- Searching for recipes based on available pantry ingredients.
- Filtering recipes by favorited status.
- Allowing users to mark recipes as favorites for quick access.

RecipePresentation		RecipeService
		+ name: string + recipe: list + favorite: bool
+ name: string + recipe: list + favorite: bool		
+ displayRecipes() + searchRecipe()		+ createRecipe() + addRecipe() + searchRecipeName() + searchRecipePantry() + serachRecipeFavorite() + favoriteRecipe()

Pantry/Freezer: The Pantry/Freezer component provides inventory management, allowing users to track, update, and manage food stock levels. It ensures that users can efficiently monitor their available ingredients when planning meals.

- *PantryFreezerPresentation*: This class offers the user interface for managing pantry and freezer inventories. It allows users to:
 - Display current pantry and freezer stock.
 - Update pantry and freezer inventory manually.
 - Modify the quantities of stored items.
 - Adjust the user interface dynamically based on stock changes.
- *PantryFreezerService*: This backend class manages inventory data retrieval, updates, and adjustments. It provides:
 - Methods to retrieve pantry and freezer stock lists.
 - Functions to increase or decrease item quantities as needed.
 - Methods for adding new items to inventory.
 - The ability to remove items that are no longer available.

PantryFreezerDrecentation	PantryFreezerService	
+ pantryItem: string + pantryItemQuantity: int + freezerItem: string + freezerItemQuantity: int	t	+ pantryItem: string + pantryItemQuantity: int + freezerItem: string + freezerItemQuantity: int + pantryList: list + freezerList: list
+ pantryList: list + freezerList: list + displayPantry() + displayFreezer() + updatePantry() + updateFreezer() + updateQuantityView()		+ getPantryList() + getFreezerList() + incrementItemQuantity() + decrementItemQuantity() + addItem() + removeItem()

Settings: The Settings component enables users to manage their profile, privacy settings, shared meal plans, and authentication status.

- *SettingsPresentation*: This class provides the front-end interface for user settings management. It allows users to:
 - View and update their profile information (name, email, password).
 - Manage shared meal plans.
 - Modify privacy preferences.
 - Sign out of the application.
- *SettingsService*: This backend class handles authentication and user account settings. It provides:
 - Methods for retrieving user profile data.
 - Functions for updating shared plans.
 - Privacy setting management to control data visibility.



Implementation Plan

Gantt chart Design:



Gantt Chart Discussion:

Going through the gantt chart above, our plan starts in January with database organization which is almost completed. The database itself is propagated with some recipes and is organized and is in the middle of being transitioned to an AWS Database so our client can take it over when the capstone period ends which our team member Max is handling as of writing this. The next goal of our project is Architecture layout which should be completed by mid February. The group has already planned out the basis of the architecture layout and will be split on implementing placeholders for all the other services for their respective assignments. Moving onto Recipe and Menu creation which goes into March, Laura is working on handling recipes and connecting it to the menu/calendar that Isaiah and Max will work on at around the same time. While that is happening, Colin will be working on the pantry and freezer systems to be able to see what the user has at the time as well as sending that back to the menu and recipe systems.

When those systems are finished or if someone finishes their part before the others, they will start working on the ability to share menus and calendars to other users and being able to actively change it and being able to update for the other people who have access to the given calendar. Stretch goals will be worked towards in any time we may have left over after completing the social sharing features. The stretch goals don't have any particular assigned person to them but the list currently includes Allergy filtering, AI Integration to recipe suggestion, and a friends system to expand upon the social features. All the remaining time will be used to test the limits of the application, error handling, and ensuring it works as intended so it can be ready to be delivered to the client.

Conclusion

MealsMyWay is a meal prep application that makes it easier to compile dishes, designate them to a given time and day, generate a list that totals up all the needed ingredients, and the steps required to cook all the dishes concurrently. MealsMyWay will be a free to use widely applicable app that can assist anyone who wants to save time in their day to day lives. However, not sacrificing the quality of the dishes prepared. We are implementing the application at this moment by writing using several packages like HTTP and using software like AWS to support the backend of the database.

Our architecture layout will be our guideline for how to implement systems while having them interact with each other in the correct ways, such as the calendar accessing both the recipes and pantry freezer systems. The class diagrams will further help explain what exactly we need to put into each system and class to make it function how we want it to and help remove extraneous or unnecessary resource usage. Our gantt chart is ongoing and subject to change but will hopefully keep us on track to being able to deliver a viable product to our client, and more importantly, one both parties will be happy with. MealsMyWay will change the way the meal prep world thinks with these design additions.