

PATRICK'S MEAL PLANNER

App Project Documentation

Patrick Lenis

Year II, Computer Science in English

Programming for mobile devices

Supervisor: PhD. Lecturer Liviu Octavian Mafteiu-Scai

17.05.2021

1. Abstract

For this course, I developed a mobile application aiming to help you plan and organize your meals for the week. This report contains a detailed description of my app, overviews of similar apps, development plan as well as goals and target audience.

2.Introduction

Patrick's Meal Planner is an Android app that will help you plan and organize your meals for the week. The app will allow you to create custom meals along with the needed recipe and ingredient list and then add the m to the weekly plan.

For the visual interface of the app I used a minimal design style with a simple color scheme aiming to have a good user experience. I also implemented support for both light and dark color schemes to match users device theme.

The user should be able to navigate between the screens : Main plan organizer screen, recipe searching screen, and also settings/account screen.

3.Goals and Users

The main goal of this app will be to help people such as myself to organize their meals and thus optimize their shopping list for a better time and financial management. The app can double as a recipe book and ingredient tracker as it will also have support for these features.

The target audience will be pretty large as this is a feature that can improve the lives of most people. However, this app will bring the most value to people who find it difficult to decide what they want to eat and often find themselves missing or overspending on ingredients, going for groceries more than desirable, and generally feeling a need for more organization in this department.

4. My Contributions

Some apps that are similar in concept to my project are: Paprika and Mealime

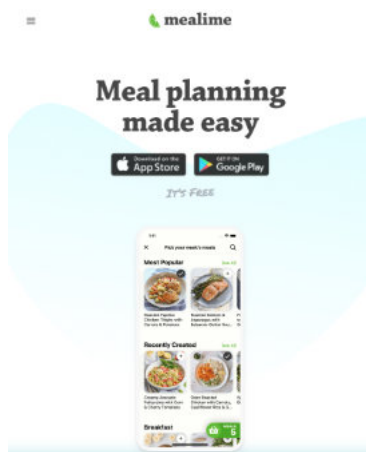
- Paprika

“Paprika is an app that helps you organize your recipes, make meal plans, and create grocery lists. Using Paprika's built-in browser, you can save recipes from anywhere on the web.”



- Mealime

“Eat better, enjoy stress-free evenings, and feel great about yourself. With over 200 personalization options, eat exactly how you want to eat.”



While the apps I mentioned before do a great job at what they are intended for I personally think they take the problem in a whole different spectrum. Those apps focus more on things like actually finding recipes, calorie consumption, and so on. For my app, I want to keep it simple and efficient and so the app will allow the user to add their own meals rather than browse for recipes and select them to form a plan for the upcoming week. This is meant to simply help you decide in advance what you want to eat so that you can save both time and money when you go shopping.

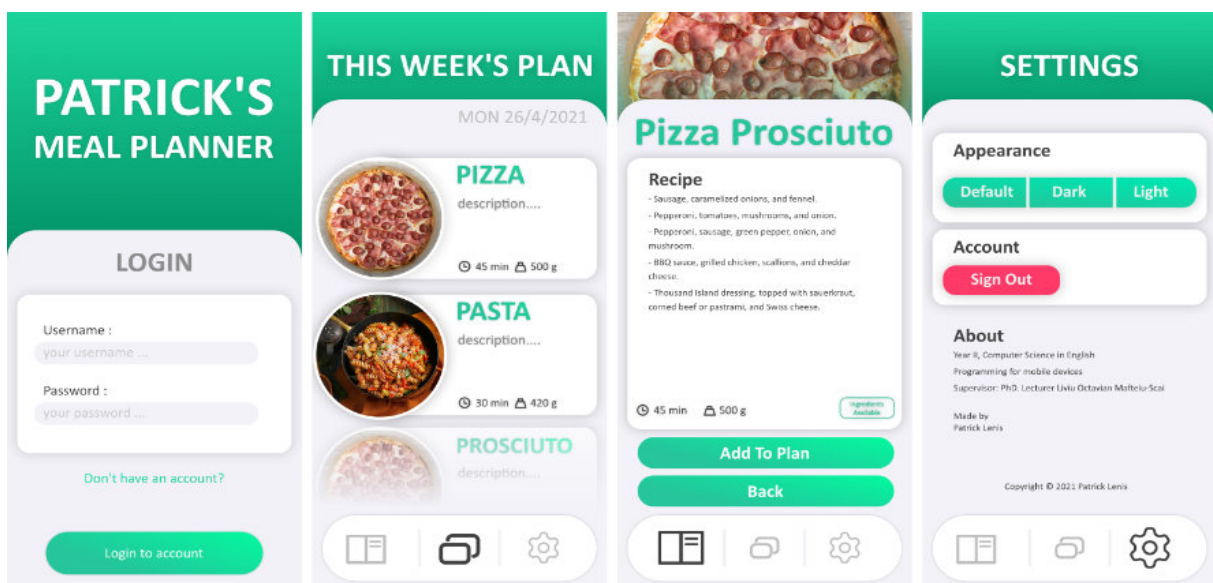
5. Functionality

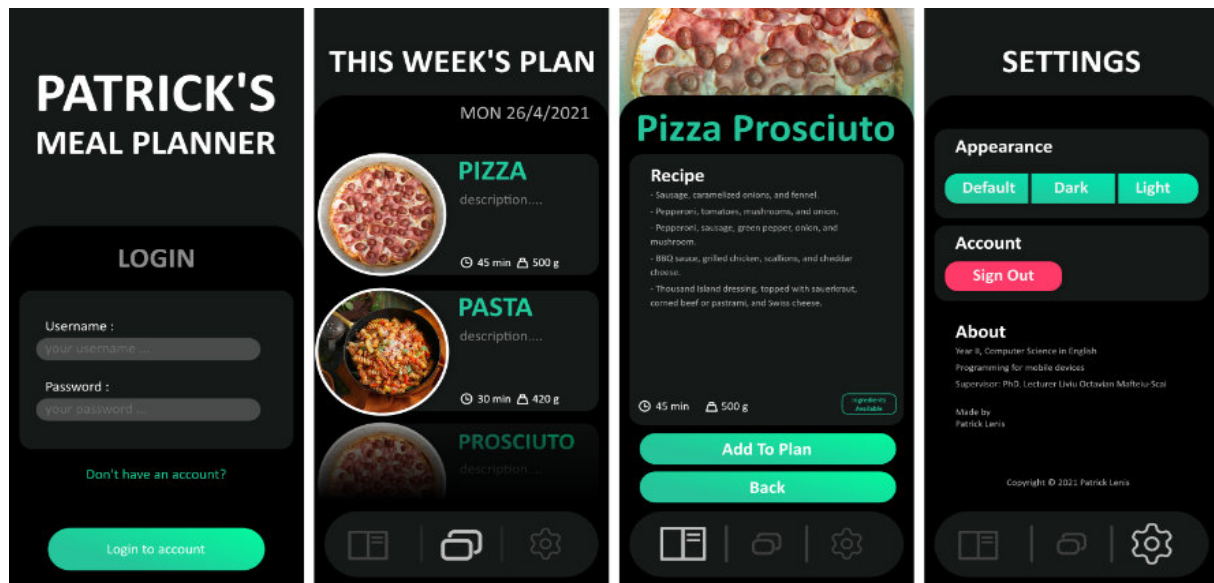
My app is designed to provide a simple yet practical and helpful user experience. The main functionality of the app can be seen in the UML diagram below.



6. User Interface

Here is the initial UI diagram created for my app in both light and dark mode. The final app ended up being a little different than this original UI concept but the core design was preserved and I am very satisfied with what I have achieved.





7. User's Manual

When the user first opens the app he will be granted with a login screen. Here he can type in his credentials and get access to the app. Similarly an registration screen can be accessed to create a new account. After the login, the user will interact with three main pages which will be accessible via a bottom navigation bar. The pages are "Meal Plan Page", which will include all recipes selected by the user for his meal plan, "Recipe List" page, which will include recipe suggestions as well as a useful search criteria panel, and finally a "Settings" page to manage the theme and account settings.

8. App's structure (technical manual)

For the development of my application I used Android Studio as an IDE and Kotlin as a programming language. First I created a Login and Register view which appears when the application is started and then I linked those with a main view which contains a bottom navigation bar with three fragments (recipes, meal plan and settings). I also needed to create some additional views for some functionality which include a recipe row format view for the recipe cards appearing on the screen as well as a recipes details view which appears when one such card is clicked by the user. For a programming point of view I implemented an API called Spoonacular API which provides a large database of existing recipes. I created various network kotlin classes to access and retrieve data from this API using a query map. Besides from the API, my application includes a Room local database which stores the most recently searched

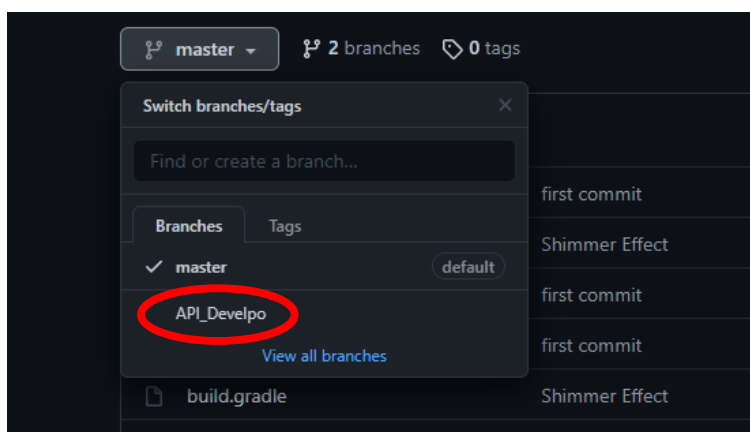
recipes and the recipes that were added to the meal plan. I opted for this method in order to limit the number of calls to my API and preserve the app running even when offline. To make this work I created a repository class which gets information from my database classes first and calls my networking API classes only when new data is required. The data from my repository is then linked with my card views by kotlin data binding so the actual recipe data may be displayed. Here I also used a coil library to load images from a given URL. Next up I used the classes associated with my views mentioned above to provide all the logic for my application to run.

9. Conclusions and Future work

As a conclusion I really enjoyed working on the project and I had so much to learn about app design and Android Studio. As everybody knows there is no finished software only abandoned software and this project will not be abandoned yet. In the future I plan to improve the login part of my app to use a Firebase entry and actually store my users accounts on an external database. This Firebase addition will also mean syncing my local databases with Firebase to provide users with their data on multiple devices. Other additions I have planned are further improvements of the UI and UX as well as adding more functionality to the recipe search and meal plan organisation.

10. References

For the full source code of my app please access my GitHub repository at the link “github.com/PatrickLenis/Meal_Planner_App” and please make sure to select the API_Development branch for the latest version.



Other references include all the libraries and API I used for this project:

- Dagger hilt
- Navigation safeargs
- Kotlin extentions
- Room database
- Data binding
- Data Store
- Shimmer
- RecyclerView
- Retrofit
- Coil Image Loading
- Gson
- Jsoup