



MealMaster

Final Paper

Andie Perreira & Tamsyn Evezard

Fall 2023

Executive Summary

MealMaster is a recipe repository, meal planner, and shopping list organizer all rolled into one sleek mobile application. In today's world where time is precious and meal planning is a struggle, MealMaster is the solution to cost and time effective nutritious eating. Log your favorite recipes, select your meals for the week, and watch as MealMaster transforms your choices into a precise shopping list, ensuring you have just what you need. Make your meals an enjoyable and hassle-free experience with MealMaster.

In the current society, many people find it challenging to coordinate meals, manage recipes, and create shopping lists. The market lacks a comprehensive and user-friendly solution that seamlessly unifies these essential tasks. MealMaster will be an innovative project that aims to empower users to effortlessly plan meals, save their favorite recipes, and generate precise shopping lists tailored to their needs.

MealMaster's primary objective is to be a user-centric mobile app that simplifies the meal planning process. Our application will be designed to be a platform for people for inputting their recipes, organizing weekly meal plans, and automatically generating shopping lists based on their chosen recipes.

This application can help individuals save money on buying groceries by planning exactly what to buy and when, thus also preventing excess food and groceries going to waste at the end of the week. Better planning and meal preparation (and the tracking of nutrients entering your body) can result in better nutrition and overall health. Overall, MealMaster aims to make nutrition and meal planning enjoyable and stress-free.

This will be our first attempt at developing this idea but this is something we have thought about for a while due to being athletes and struggling with eating healthy while saving money,

especially being student athletes in college. This application will help us plan our meals, and see what nutrients each meal contains for optimum nutrition. We are still in the design and planning part of the project, however, we have started to navigate React Native, the free application we are using to create our project, and have begun learning the parallels to the web development equivalents.

The biggest challenge for us would probably be attaining the skills to make a mobile application and use new resources like React Native and Expo Go. We have done neither of these before, however, with our background knowledge in programming languages, specifically Javascript, and resources available, we should pick up these new skills fairly quickly and learn along the way. We have had the opportunity to connect with individuals possessing valuable expertise in this particular domain with prior knowledge that will significantly enhance and support our journey.

Qualifications

As for our qualifications, we are both competitive water polo players and are currently seniors in our Computer Science degree. Our shared love of the sport developed the understanding of how important nutrition is in optimizing athletic performance and recovery. Although we cannot claim to be experts in the field of nutrition, this is a topic we are eager to explore and gain proficiency in.

The proficiency acquired through our Computer Science degree has endowed us with the precise technical expertise and problem-solving skills essential for the successful development of this mobile application. Our academic journey has provided us with skills such as Software Development, Programming, and Server-Side Development which will come in handy in our process, especially when we are further along down the road.

All in all, we both have about 3 years of programming experience through CUI courses in various programming languages including: C#, Python, C, C++, Java, HTML, CSS, and JavaScript. Therefore we are using our previous knowledge to learn the ropes of React Native and its coding equivalents.

As NCAA Division II Athletes and part of the CUI Women's Water Polo, our resilient, driven, and adaptable personalities are a great advantage for dealing with challenges along the way. Being a student athlete has taught us many valuable skills such as teamwork, communication, and time management which is a crucial part of this development.

Design Summary

The design process of MealMaster is driven by a commitment to user-centricity, beginning with extensive planning and research. Understanding the needs and preferences of the target audience is crucial, and we have employed informal surveys and interviews to gather valuable insights. These research efforts provide the foundation for designing a user interface (UI) and user experience (UX) that resonate with users and offer an intuitive and enjoyable interaction.

We recognize that the success of MealMaster hinges on its ability to simplify complex tasks, and this simplicity is reflected in the design philosophy. Prototyping plays a pivotal role in visualizing how the application will function. The iterative nature of the design process allows for continuous refinement based on usability testing throughout the application development. By involving potential users in testing, MealMaster aims to gather feedback that is instrumental in making adjustments and improvements to ensure a seamless and user-friendly experience.

Adopting the Agile methodology, the design phase is dynamic, allowing for flexibility and adaptability. Breaking the project into manageable sprints facilitates incremental progress and allows the creators to respond effectively to challenges. Regular collaboration and communication

within the team ensure alignment regarding project goals and foster a shared understanding of potential issues and challenges.

The frontend development of MealMaster was carried out using React Native and Expo Go, a framework known for its efficiency in developing cross-platform mobile applications. This



choice not only saves time and resources but also ensures a consistent user experience across both iOS and Android platforms. We are actively engaged in navigating React Native, leveraging our existing programming knowledge, particularly in Javascript, to learn and apply the necessary skills.

Within React Native, we used Yarn and Babel along with the React Native and React Navigation dependencies. Yarn is an advanced package manager for Javascript and it is used to efficiently ensure all operations are able to run on every machine the same way and also reduces bugs. Additionally, we used Babel which is a tool that is used to transform modern Javascript code into compatible versions so it is able to run smoothly on older browsers and environments.

This capability is essential for maximizing the compatibility and reach of our applications, allowing them to run seamlessly across diverse platforms and devices. Together, these tools form the backbone of our development environment, enhancing our ability to deliver robust, high-performance mobile applications with React Native.

The complete mobile application will be demonstrated using the Expo Go simulator app, showcasing the culmination of the design and development. We are driven by a commitment to providing a solution that not only meets the technical requirements but also delights users with its usability and aesthetic appeal.

Prototype

As we progressed, we learned the ropes of React Native and Expo go using resources including React Native and Expo documentation (<https://reactnative.dev/>, <https://docs.expo.dev/>), YouTube tutorials, and specific examples online (for example, from StackOverflow and ChatGPT). Our process of development and implementation adapted over time to follow this structure:

1. Basics of a single page Grocery List organizer

- Functions, Pressables, TextInput, Lists, Layout, Style, etc.

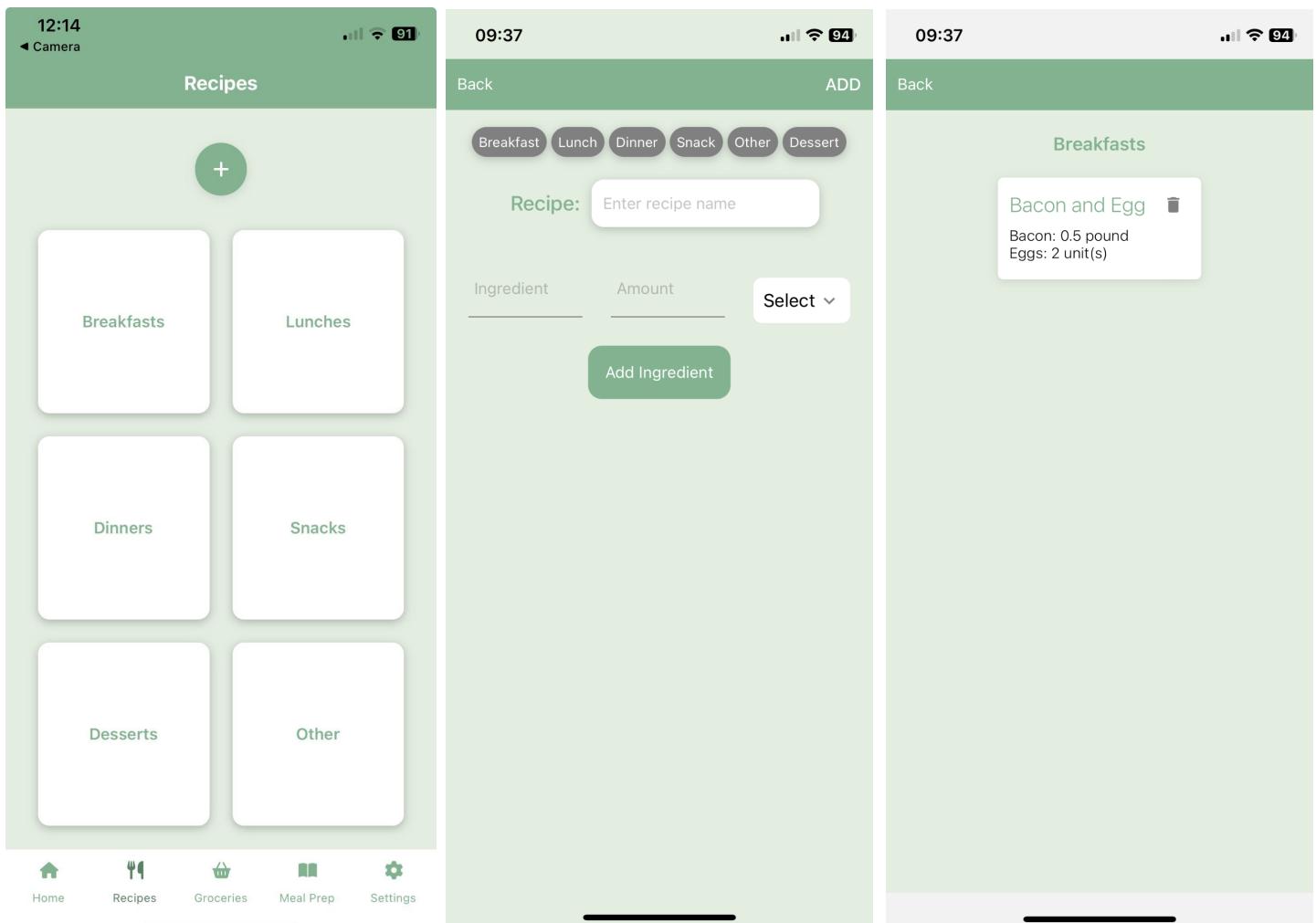
2. Stack and Tab Navigation integration



- Tabs: Home, Recipes, Groceries, Meal Plan, and Settings Tabs
- Stack: e.g. within Recipes: Add Recipes, Breakfasts, Lunches, Dinners, etc.

3. Saving & Loading data from one page to another: e.g.

- Saving recipe names, ingredient details, and description to SecureStore (expo storage).
- Loading data into specified meal categories from SecureStore.



4. Style & Format:

- adjusted throughout development - StyleSheet

Future features include the ability to select recipes from Breakfasts, Lunches, Dinners, Desserts, or Other to add to your Meal Plan calendar, which automatically adds the ingredients needed to your grocery list, user authentication and data storage.

In summary, the design of MealMaster reflects a meticulous and user-centric approach, incorporating Agile principles to ensure flexibility and responsiveness. The choice of React Native for frontend development, positions MealMaster to be a seamless and secure solution for users seeking an efficient and enjoyable meal planning experience. As the design phase progresses, the creators are excited about the potential impact MealMaster can have on simplifying the lives of individuals, making meal planning a stress-free and rewarding endeavor.

Project Summary

The project plan for MealMaster encompasses a systematic approach to developing a user-friendly recipe repository, meal planner, and shopping list organizer mobile application. The plan begins with an extensive research and requirements gathering phase for an understanding of user needs. Following this is the design and prototyping phases, laying the foundation for a well-structured user interface. Development will proceed iteratively, with a focus on backend infrastructure setup, frontend implementation, and algorithmic logic for meal planning and shopping list generation. Testing and quality assurance will be essential throughout the development process, leading to a carefully planned deployment.

The Planning Phase includes defining the app's purpose, target audience, and core features, researching competition and finding unique selling points, and creating a project plan, including timelines and milestones. We have opted to construct a chart to visually show the evolution of our

developments and the corresponding timeline. This will ensure we stay on track and prepare for the upcoming stages. See the Gantt Chart below for a detailed layout of the project plan:

MealMaster2000		Duration	Due	
				Milestone
Planning		2 weeks	1-Oct	
	Learn React Native	in progress		
	Research similar applications	in progress		
	Discuss target audience, purposes, and core features	in progress		
	Diagrams	in progress		
Design	Dependencies: Planning Completed	4 weeks	1-Nov	
	Design UI and UX			
	Prototypes			
Development	Dependencies: Design	4 weeks	1-Dec	
	Backend infrastructure for accounts, recipes, meal planning and shopping lists			
	Frontend development			
	User authentication and data storage			
Testing	Dependencies: Development	2 weeks	8-Dec	
	Unit testing for bugs			
	User testing for feedback			
	Testing on different devices and screen sizes			
Bonus Features	Dependencies: Testing	3 weeks	15-Feb	
Deployment	Dependencies: Testing	5 weeks	15-Mar	
	Configure app store listings			
	Publish on App Store/Play Store			
	Monitor launch & any issues			
Post-Launch	Dependencies: Deployment	5 weeks	1-May	
	Get user feedback and reviews			
	Address bugs & make improvements			
	Marketing and Promoting			

The Design Phase encompasses designing the user interface (UI) and user experience (UX), and developing prototypes. During this phase, this is where we not only map out the potential layouts of our app, but our intentions for this app and how we are going to make it appealing towards our intended audience. Following extensive collaboration through brainstorming sessions, this has led us to believe that the most effective solution to engage with users in this current and adjacent generations is through the development of a mobile app. Considering the widespread accessibility of smartphones within the majority of the population, investing in a mobile app presents more opportunities to create meaningful connections with the users.

While engaged in the execution of the Design Phase, we made the decision to construct a prototype that emulates the functionalities and user experience anticipated in the final mobile application. Consequently, we adjusted our project timeline, allocating additional time to the

prototype phase. This adjustment, in turn, led to a corresponding shift in the schedule for the Development Phase and later stages of the project.

Development of the application entails building the backend infrastructure for the user, incorporating components such as accounts, recipes, meal planning, and shopping lists, developing the frontend for various platforms (iOS, Android), and in the future, implementing user authentication and data storage AND....

The algorithms that we will be using are broken down in the diagram below:



Testing, as touched on earlier, would be conducting unit testing and integration testing for bugs, user testing to gather feedback on usability, and ensuring the app works smoothly on different devices and screen sizes. Finally, the Deployment Phase includes preparing the app for release by configuring app store listings, publishing the app on the Apple App Store and Google Play Store, monitoring any launch issues. After launching the app, we would gather user feedback

and reviews for future improvements or bug fixing, and plan marketing and promotion strategies to attract users.

Testing Summary

Building & Compiling

Open the MealMaster folder in Visual Studio Code and ensure all required files are in the folder. Navigate to App.js, open your terminal in VS Code and run the following command:

```
npx expo start
```

This should pull up an options menu and you will be choose either a, i, or w to test the application on your preferred platform - see an example from another folder called *my-react-app* below:

```
○ (base) tamsynevezard@Tamsyns-Air my-react-app % npx expo start
Starting project at /Users/tamsynevezard/Documents/CUI/FALL 2023/Project/my-react-app
Starting Metro Bundler

> Metro waiting on exp://192.168.0.20:8081
> Scan the QR code above with Expo Go (Android) or the Camera app (iOS)

> Using Expo Go
> Press s | switch to development build

> Press a | open Android
> Press i | open iOS simulator
> Press w | open web

> Press j | open debugger
> Press r | reload app
> Press m | toggle menu
> Press o | open project code in your editor

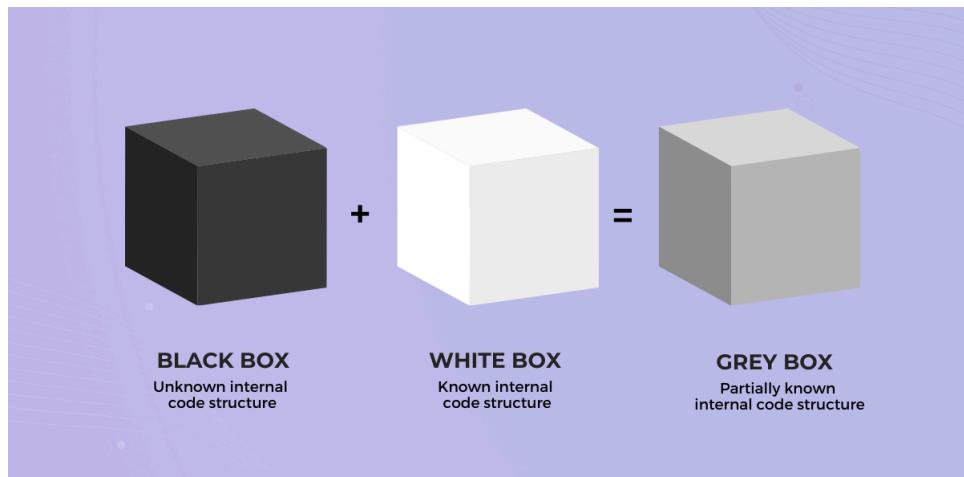
> Press ? | show all commands

Logs for your project will appear below. Press Ctrl+C to exit.
```

Installing Expo Go on your mobile device will allow you to run the mobile application from there from scanning the QR Code shown above. Some software testing that will be done includes unit testing, integration testing, system testing, regression testing, and beta testing. Some

non-functional-testing may include performance, security, install and usability testing. We would also need to explore black-box,white-box, gorilla and grey-box testing.

Our prototype will undergo mainly black-box, white-box, and grey-box testing (see diagram below). Black-box testing involves someone testing the prototype who does not know the inner workings of it, while white-box testing is done by one of the developers. Grey-box testing is the median between these two: the tester knows and understands some of the code structure of the prototype.



Testing the prototype in these ways aids us in understanding challenges, connections, and overall the final design concepts to implement in the final product as a mobile application.

Unit testing will include testing valid, invalid and null inputs for the recipe repository additions, and selection of recipes for the grocery list generation. Integration testing will be conducted to ensure that each page of the application is linked seamlessly and interactions between them produce the expected results (for example, after adding a recipe, the selection of that recipe adds the ingredients accurately to the grocery list). Throughout the development process, regression testing will be used to ensure that any changes or updates in our code have not adversely

affected existing functionalities. System testing will be performed periodically after each milestone is achieved to ensure the whole application works effectively as a whole.

Once the application is at a consumer-usable-level, we will make it available to a group of potential consumers for beta testing and feedback - this goes hand in hand with usability testing.

To optimize our mobile application, performance testing can be done to investigate how MealMaster does under certain conditions. Performance testing may include stress testing, load testing, and endurance testing, in other words, it pushes the application to its limits and sees if it breaks. During the deployment phase, install and security testing will be essential for consumer use.

Conclusion

We are excited to work on this project because this is something that could be really helpful for not just college students, but also anyone wanting to plan ahead and be prepared for the week and have as little waste as possible. Because nutrition, fitness, and athletic performance are all crucial aspects of our lives, we see working on combining these aspects with computer science in the future as a career as an option.

React Native is free so it will not cost anything to develop. As for right now, we are not doing this for monetary reasons but in order to help others going through the same problems we are. If the fully developed application is successful, we may put a price on the app.

In conclusion, MealMaster represents a creative solution to the modern-day challenge of meal planning, recipe organization, and efficient grocery shopping. As a mobile application, MealMaster aims to simplify the lives of individuals by integrating these essential tasks into a single, easy-to-use resource. With our team's background in computer science and our dedication to optimizing nutrition and athletic performance, we are committed to developing a comprehensive

and user-friendly solution. While we expect challenges along the way with learning new technologies like React Native, our programming knowledge and dedication to this project will help us succeed. As the work on MealMaster continues, we are excited about its potential to benefit college students and anyone else looking for a more organized and cost-effective approach to meal planning, better nutrition and overall health. Our goal is to make MealMaster a valuable tool for individuals. We look forward to the opportunity to bring this vision to life and positively impact individuals' lives, one meal at a time.

Additional Documents

Resources

Getting Started: <https://docs.expo.dev/get-started/installation/>

Updating SDK version: <https://docs.expo.dev/workflow/upgrading-expo-sdk-walkthrough/>

React Native Documentation: <https://reactnative.dev/>

React Navigation: <https://reactnavigation.org/docs/getting-started>

Core Components and APIs: <https://reactnative.dev/docs/components-and-apis>

Expo Go Installation: <https://docs.expo.dev/get-started/installation/>

ChatGPT: for small bug fixes and SecureStore integration instead of MongoDB (experienced errors with MongoDB).

- React Navigation in React Native Expo 2024: <https://www.youtube.com/watch?v=G9eSe9uupo0>
(Part 1 - Part 7)
- Nested React Navigation in React Native and Expo Apps | Stack Navigation Inside Bottom Tab |
Code <https://www.youtube.com/watch?v=PO5P0EscvFI>
- Let's build a full Stack Meal Planner App with React Native using MongoDB
https://www.youtube.com/watch?v=QcpYWc_19no