G1 - Mach

ESOF 423-001

Spring 2023

Lukas Bernard, William Jordan, Kirsten Keblish,
Joey Knappenberger, & Bryce Lehnen

Section 1: Program

The source code for this course can be found at https://github.com/423s23/G1-Mach. The project we have been working on is a mobile app for a company called Mach. The app facilitates an incentive program for triathletes who sign up for the program. The main idea is to give out free rewards for completing tasks that help to showcase Mach and what they do. The app was developed using React Native and Expo. React made it easy to create an app that works for both Android and IOS, and Expo improved our ability to test the app. We later added in Firebase and Firestore to handle our database, and react native makes it easy to read and write to the database.

Section 2: Teamwork

Team Member 1 (35%): Generally, I was one to experiment with our codebase to see what I could actually accomplish with the project. Many of my hours spent on the project ended up being "fruitless" as the plugins I would try to add would end up not doing what we needed in respect to features. I developed the main homepage screen and the submit task screen. Both of which are the backbones to the project. Outside of the actual coding, I have been a pseudo Scrummaster by coordinating our daily meetings and organizing tasks for each spring backlog. The submit task page now has a helper page which explains what each type of task is required for approval. I also had a small part in setting up Firebase, and more so with how we use it to set up users and tickets which holds information about tasks that are submitted by users. I have also been the main team member creating and managing our various documents associated with the project like the portfolio, burndown chart, and slideshow for the final presentation. Overall, I would try to get my parts done before the others since they could then use anything I wrote as a reference.

Team Member 2 (16%): I've handled a lot of the architecture of the application - creating the react navigation stack the UI has been built in, creating the continuous integration system that tests and updates the full build every time someone pushes code, and building the backend through firebase. Early on in the project, there were a few paths that we went down that ended up being fruitless, so I have also spent a lot of time researching solutions and figuring out what will actually work. Getting CI working was especially challenging, as building and testing a full mobile application is resource intensive.

Team Member 3 (13%): I started my work on the project by laying the UI/UX design foundation. I designed and generated first a low-fidelity mockup for the project before advancing it into a working high-fidelity prototype. I used the general app layout created by Dylan, his styles, and his color schemes while also incorporating more effective UI/UX principles. Once the app was up and running, I began working on my designated screens which have been user info, settings, and the individual pages that are linked to the settings page. The individual settings pages included the contact info page, a notifications page, the help page, and the log out page. My job was to create these screens, put the necessary content on them, and style them accordingly. Following the creation of these pages, I worked to integrate them with the database and make them functional.

Team Member 4 (20%): Visibly on the front end I have contributed work mainly to the Rewards Screen in our app thus far. I'm still on the fence about a few stylistic choices as well as long term functionality of the rewards page and how I want it to behave. Behind the scenes I have worked to thoroughly plot out my ideas to see what I believe will be the most efficient use of my time with the most useful result. Additionally on the back end I have been the main contributor to the iOS section of the developer documentation and user documentation, setting a set of guidelines to reach a point where our app is openly accessible.

Follow-up: With the rewards page I was able to implement a few more features than originally planned, allowing the page to contain more information. Initially I wanted a cut and dry list of all rewards, however I have expanded upon that idea by adding a few key things. I added a pressable feature for each reward that enables a popup that contains additional information, and possibly images of the products in the future, to allow the user to get a better grasp of what is available. In addition to this, I was able to add more information and tiers to the visible list itself, including score requirements, titles, and more rewards than were originally implemented. The final major addition to the project I took responsibility for was editing multiple styles and pressables to bring more consistency to the individual pages throughout the app. For example, I brought consistent styling to the back buttons throughout the pages and allowed them to be in a place that made them functional for all types of devices.

Team Member 5 (16%): I primarily focused on building the admin approval page and designing the logic for how submissions will be viewed and approved or denied. Additionally I built the leaderboard page and worked on ensuring that the point system feels like a game and kept members engaged and wanting to earn points. Furthermore I have looked into how to upload images to a third party source so we don't have to store large files in our database. Finally I have built the about page and started on styling for common elements.

Section 3: Design Pattern

The design pattern used for our various screens is the facade pattern. Its main goal is to provide almost a level of abstraction when combining our various screens together. It allows us to have one controller that is in charge of everyscreen. This degree of separation has made it easier for everyone to work on their own separate screens and then have them all connected through the controller. It can easily be seen with our main file ‘App.js’ which then connects the various screens in the ‘/Screens’ directory. The only drawback of this pattern is that we now have a large amount of stylesheets since each screen gets its own. The dozen or so stylesheets we have right now could be condensed down to about four or five while still maintaining readability in the code.

Section 4: Technical Writing

User Instructions:

Login Page: The login page allows new users to create a new account, and for current members to login with their associated email and password that they used when first creating their account. There is also a recovery option if a user forgets their password.

Home Page: The homepage is the main source of navigation for the entire app. It allows a user to access every other page from one convenient location. The page also displays quick information about the user's progress like their rank, level, and points needed to get to the next level. The user can also press on the rank box to learn about all of the different ranks. The user can pull down from the top of the screen to refresh the page with any new data that may have been updated while they were in the app.

Task Submission Page: The task submission page is the place where users will go to let the administrators of the app know when they have completed a task. The user can select a type of task ranging from buying their first Mach kit to a new blog post that they coordinated with the athletic director. If a user has questions about a certain task, they can go to the help page by pressing the button located in the top right. This page will explain in detail what is expected of each task and any limitations on submitting tasks. Below the task selection dropdown is a large comment window where the user can place any links that are required for a task or anything else that is required for a task to be approved.

Rewards Page: The rewards page shows all the various rewards a user can earn by completing tasks and getting points. Each reward is separated into categories based on the rank a user will have when earning a reward. Next to every reward is the points needed to earn that reward, and if a user has questions about a reward, they can press on it to learn more.

Leaderboard Page: The leaderboard page shows how users stack up next to each other in terms of overall points earned. The top three users get a podium spot and the current user can see their place compared to the rest.

Admin Approval Page: The admin approval page is for administrators only. The page will display the oldest task that has been submitted including information about the type of task, the comment from the user, and a timestamp. The admin can then approve or deny the task which, if approved, will add points to that user.

User Info Page: The user info page is a more in-depth look at what a user has earned by completing tasks and earning rewards. They can see their total progress to the 50,000pt reward, their current progress to the next level, and tasks that have been approved.

About Page: The about page simply displays the mission statement of Mach Apparel.

Settings Page: The settings page is where a user can go to see who they are logged in as, and is home to a few ways to change their info or the app. Under Contact Info, a user can change their name, email, phone number, or username if needed. Under Notifications, a user can quickly turn on or off push notifications from the app. And finally, there is a logout button for logging out of the current user.

User Documentation:
To try out the pre-release
- Download the Expo Go app on your phone - [Android]https://play.google.com/store/apps/details?id=host.exp.exponent and [iOS]https://apps.apple.com/us/app/expo-go/id982107779 - Go to the Expo Snack hosted https://snack.expo.dev/@grayturtlejoey/github.com-423s23-g1-mach:mach_rewards_expo@demo and select "My Device" in the panel on the right hand side

- Scan the QR code with your camera app on iOS or the Expo Go app on Android

Congratulations! The app is now streaming to your mobile device through Expo Go, and you're demoing the latest release of Mach Rewards!

Feel free to explore the layout of the screens, and see what features will be added in the future. If you encounter any bugs (including ones related to UI i.e. buttons not working or text overflowing off the screen) then please report your issues here: https://github.com/423s23/G1-Mach/issues
Reporting bugs lets us as developers resolve your issue(s) directly or point you to an already resolved issue that can help!

Developer Documentation:
To develop and run this app natively on your machine, you will need the following:
- Prerequisites
  - node.js, installed from your favorite package manager*
  - yarn, again from your package manager* of choice
  - (Optional) Android Studio, if you want to run the android version in an emulator
  - (Optional) xCode, if you want to run the iOS version in an emulator - this only works on mac

*Homebrew package manager for macOS and chocolatey on Windows are highly recommended

- Run
  - An Android or iOS device with the Expo Go app installed - [Android]https://play.google.com/store/apps/details?id=host.exp.exponent and [iOS]https://apps.apple.com/us/app/expo-go/id982107779
  - (Optional) An Android emulator created through the devices manager of Android Studio, if you want to run in an Android emulator
  - (Optional) An iOS virtual device created in xCode settings, if you want to run in an iOS emulator

Once you have these prerequisites, in your terminal:
1. Clone the GitHub repository
2. Navigate to the Mach_Rewards_Expo directory in the cloned repo
3. run `yarn`
4. run `yarn expo start`

If things have gone correctly, after everything has been installed and built for the first time, you should be presented with a QR code and a menu in your terminal. Scanning the QR code with your camera app on iOS or the Expo Go app on Android will open the app on your physical device, and selecting an option from the menu will launch in your pre-configured emulator of choice.

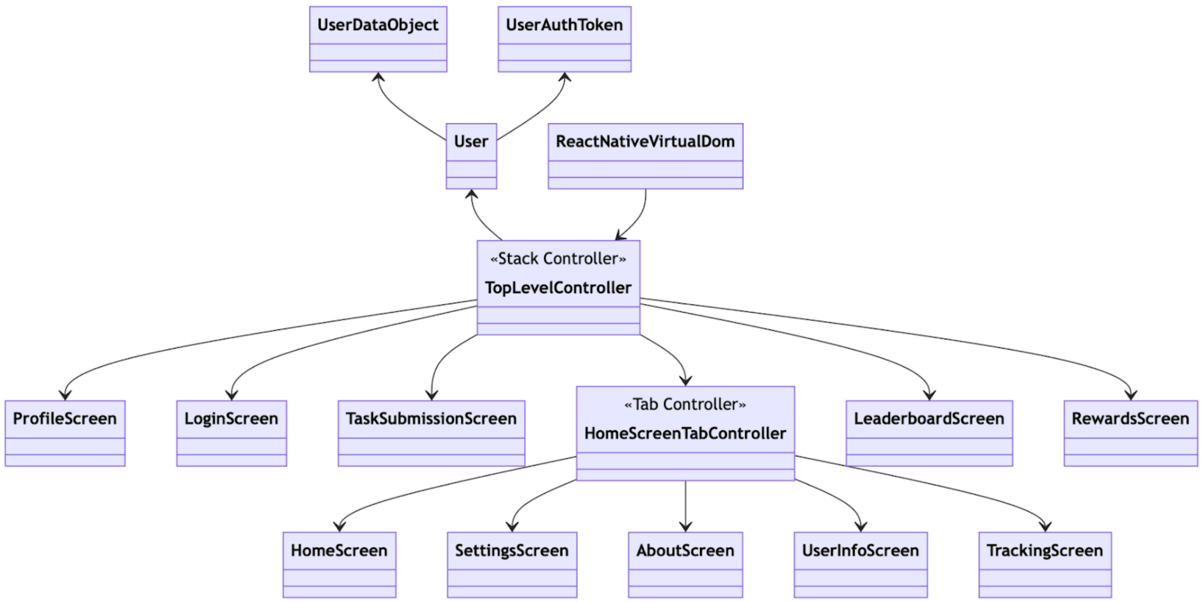
Continuous Integration:

Right now, continuous integration is being actively worked on. Due to the fact that jsx must be compiled before it can be run, and because our codebase compiles into a complete application for mobile devices, automated testing is difficult. We technically have a GitHub action set up to try and test and build our codebase, but it fails every time as it stands. We are working on setting this up the proper way, to have a BUILD branch that tries to build the code and publish the application files whenever we push to it, as well as updating the Expo Snack link automatically to one with the latest version.

User Testing:

User 1: The user generally liked how the app was organized, but felt as though some areas could use improvement. The user made the following suggestions. The homepage rank and star box looked like a button, so it should have a popup that displays all the ranks someone could have. The homepage's progress bar should be a different color as everything that is Mach blue is a button except the progress bar itself, and it should be clearer that the progress bar is displaying points. The rewards page should be more clear about which rewards have already been received or the point quota of those pages have been met. Many other pages just had styling issues or small content issues that needed to be addressed.

Section 5: UML



Section 6: Design Trade-Offs

Our largest design trade-off was deciding to make a mobile app rather than a web application. This decision has created a situation where we had to cut or lessen features due to the greater difficulties with how we are creating the app. However, having a mobile app was the best decision as it will be easier for users to ultimately use instead of a web app.

A smaller trade-off we have made is in regards to the stylesheets for our various screens. Each screen has its own stylesheet which has made it far easier for each member to work on the screens assigned to them without creating merge conflicts. However, this makes the app bulkier and more complex to manage. Towards the end, we will consolidate the various stylesheets into more manageable ones.

Section 7: Software Development Life Cycle Model

We have been using the agile development model. This has allowed us to set manageable goals for each sprint while maintaining communication throughout the whole process with our semi-daily scrum meetings. The meetings have been especially helpful in letting the team members know where everyone is at with their work and what blockers they are encountering. Having each sprint run only two weeks, however, has been challenging as it can be difficult to meet the entire goal for the sprint since this is not the only thing we have to work on as students. Overall though, the sprint goals have been a net positive as it has forced us to clearly lay out what tasks there are to do, and who is working on what tasks.