Record Throws

A DATA TRACKING APP

MALLORY LEPPELMEIER

Table of Contents

1. Introduction

1.1 Motivation

Athletes work hard to make themselves better than the competition. Data tracking apps can help athletes improve their sport performance by working smarter as well. There are nutrition apps like MyFitnessPal (myfitnesspal, 2022) that allows athletes to track exactly how many calories and macronutrients they consumed throughout the day. This app helps athletes reach goals such as gaining, losing, or maintaining weight. Another app, called SleepScore (sleepscore, 2022), allows athletes to track sleep cycles. This can help athletes quantify their recovery overnight, which directly correlates to how hard they can push their next workout. There are even apps like Recover Athletics (recoverathletics, 2022) that can help athletes track soreness levels. It then recommends prehab and mobility exercises to help prevent injuries and recover faster. It's no secret that data tracking apps can help athletes be in prime shape for competition day. Imagine how much more valuable a data tracking app would be to athletes if it was on a very sports specific level.

In a sport where every centimeter counts, track and field throwers are always looking for the next thing that is going to help them gain that competitive edge. One of the things the throwers here at the University of Kentucky do is track practice throwing distances. It is common for an athlete to throw implements of different weights during a single event practice. As practice is unfolding, the athlete will flag the farthest throw with each implement. At the end of each practice, the flags are then measured, and distances are recorded in a shared Google Sheets (google, 2022) page. Google sheets stores all data on the cloud, protecting it from loss and allowing multiple users to edit the sheet on their own device. It also allows for an "offline" mode, allowing users to enter data without a network connection and syncing the data to the cloud later. This Google Sheets page is created by throws coach, Keith McBride (ukathletics, 2022), and shared with each thrower at the beginning of the year. Each thrower has one page for each event they throw containing a very large table. Across the top is every implement the athlete might train with for that event, and along the left side is every date in the

academic year. A small, zoomed out portion of the sheet can be seen from a phone view in the image below.

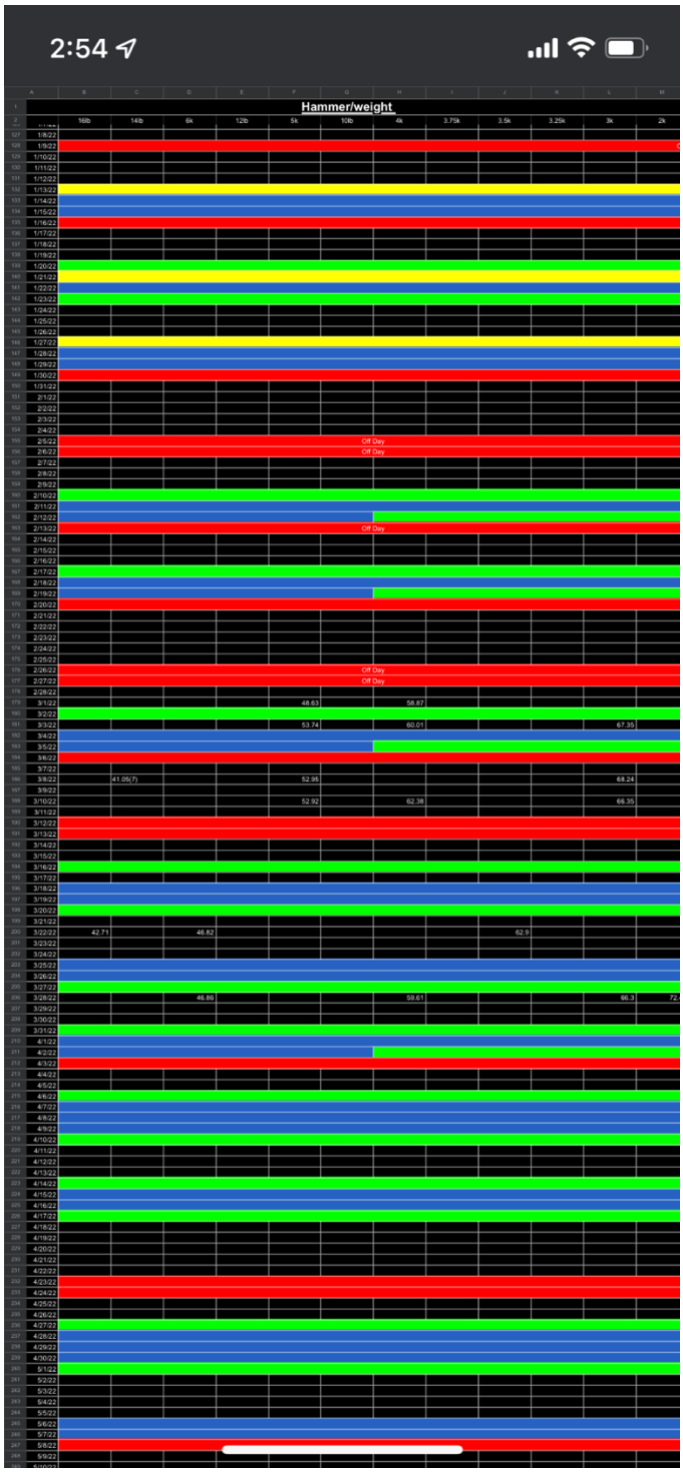


Figure 1.1 The University of Kentucky's current method of tracking data using Google Sheets

As displayed above, the sheet is both hard to see and navigate. The process of finding the correct athlete tab, date row, and implement column to correctly enter the distance is very slow. After a long practice, the last thing an athlete wants to do is hunt through this table on a little screen to record their distances for the day. Not only is this tracking system hard to navigate, but it also fails to present the data in useful ways to the athlete. Currently, the data is used to see improvements week to week and to track personal records with each implement. While this is beneficial, there is still so much untapped potential. The data could be displayed using graphs or tables to find trends in training. Looking at these displays, athletes can see best performances and correlate those performances to training variables. Training variables that would interest athletes include training volume, weight room load, strength base, and body weight. These displays could also help athletes compare current training to training at the same time in previous years to show improvement. As a thrower on University of Kentucky's track and field team, I have experienced the need myself and that motivated me to create Record Throws.

1.2 Related Work

For many sports, there are sports specific apps that are custom designed to help athletes excel. For example, Homecourt (homecourt, 2022) is an app that is designed for basketball players to track progress using workout history, in-depth statistics, and complete skill ratings. Using AI powered training, Homecourt will give athletes feedback to perfect form and suggest drills to improve weak points in their game. For golfers, there are countless apps on the market that allow athletes to track anything from scores to their golf swing. In the realm of track and field, there are apps for runners like ASICS Runkeeper (runkeeper, 2022) or Nike Run Club (nike, 2022). These are designed to track pace, time, distance, and even calories burned during a workout. Although there are many different sport specific apps, there are currently no equivalent apps on the market for throwers.

While there isn't an app specifically for throwers, there are apps that can be utilized by throwers to reach peak performance in other ways. These apps would include weightlifting apps like JEFIT (jefit, 2022) that allow athletes to track lifting progress. Throwing is a strength-based sport, so improvements seen in the weight room typically correlate to further distances. While strength is important to throwing far, good form is what converts power to distance. The University of Kentucky uses an app called Technique (hudl, 2022) to help break down throwing form. With Technique, coaches can record videos of athletes to review, analyze, and share later. In this app, videos can be replayed in slow motion and marked up to further highlight specific details of a throw. It also allows for side-by-side video comparison, which can be useful to find weaknesses in form. These apps can help; however, they do not harness the benefits a sport specific app would have for throwers.

The concept of tracking distances is not new to the throwing community. As mentioned before, the University of Kentucky uses Google Sheets for tracking. There are many throwers, however, that are using old fashioned pen and paper. They track their distances in what is called a throwing journal. This journal may include several things, like how they felt that day, technique cues being used, distances thrown, number of throws taken, and the weather. This method is equally frustrating when trying to look back and track improvement because the distances are not displayed. Professional discus athlete, Micaela Hazelwood (teamusa, 2022), is tracking distances using Excel (microsoft, 2022). She manually enters meet distances at the end of each season, and calculates statistics like average best throw, average throw, seasons best, seasons worst, total number of fouls (illegal throws that are not measured) and fouls per meet. Both the throwing journal and the Excel Spreadsheet fail to allow both the coach and the athlete to access the information.

1.3 Objectives

The goal of this app is to create a platform independent app that allows track and field throwers to record, store, and track throwing distances. The app will allow coaches and

athletes to have their own accounts that can be linked. This way, both the coaches and athletes can see the athlete's information and add new data. Distances can be added for either practices or meets. Stored data will be displayed in multiple forms including graphs, tables, and calendars.

2. Design

2.1 Initial Design

I started my design with the list of objectives I mentioned in Section 1.3. I divided each feature into different screens. I tried to deliberately think through the details of the data displays, user navigation, user information, and required input variables. This process took my idea and blew it up. One idea quickly became many, and I had to realistically think through what I wanted the app to accomplish. I realized that four of the five throwing events would track and display very similar information. Javelin throwers, however, typically do not record information about distances. Like pitching a baseball, throwing the javelin takes an incredible toll on an athlete's shoulder and elbow, requiring them to spend most of their time doing drills. I decided that the information required to be helpful for javelin throwers would be too different and I wouldn't have time to include that event in my design. Additionally, I have little experience and knowledge about the javelin and what exactly would be helpful to them. I have never thrown the javelin or had a teammate that trains it. The further research and additional coding required to add a useful javelin section to this app would stretch myself too thin, and this put adding this event outside the scope of my project.

2.2 Screen Layouts

I started designing the screens for this project with the login and sign-up screens. I decided that name, email, password, coach/athlete role, events, and gender would be the most important information to gather from each user. I included coach/athlete role,

events thrown, and gender to give the user a customized display. This led me to design the profile page with the user's information. I also added a place for coaches and athletes to link accounts, along with notifications that would be required to have an athlete or coach accept the account link.

I then focused on displaying the distance information using calendars, tables, and graphs. I specifically wanted to display practices on a calendar so coaches and athletes could quickly see an athlete's workload and keep track of what events are being thrown on which days. Being a four-event thrower, this is something that I would find helpful. Then, I worked on the graph screen, allowing athletes to see trends in distances with each implement. This information can be very helpful for coaches and athletes to decide what they need to be working on. If heavy implements aren't in an upward trend, the athlete might need to work on strength. If the light implements aren't in an upward trend, the athlete might need to work on speed. Training trends can be overlayed with competition trends, which would give valuable information on how to peak an athlete at the right time of the season. Lastly, I designed the table screen to display the athletes' personal records with each implement. This screen's inspiration came from a giant white board we have in our implement shed that holds all our personal records along with the date it was thrown. This table helps to show improvement and lets us know how long it's been since we hit those numbers.

Next, I designed the home screen, navigation, and overview screen. I designed the home and navigation screen to be clean and simple, so users could quickly access the other screens listed above. The overview was designed for users to see all the information listed for an event on one screen. A meet overview is also provided so a user can specifically see trends on competition days, because those are ultimately the days that matter most.

Lastly, I designed the add data screens where users enter meet or practice data. For both entries, users need to enter the event and date. Entries include an optional notes section, such as how sore they were that day, the weather, ques they were working on,

and other information they deem useful. The ability to write notes gives the feel of a throwing journal and gives a space to store qualitative data. Practice data additionally requires users to declare how many implements were thrown, along with the implements weight and furthest distances thrown. Meet data additionally requires users to declare how many throws were taken and the distances for each throw. Lastly, I designed a data info screen that would display all the information entered for that data entry.

2.3 Creating a Logo

I created my logo using Canva (canva, 2022), a free online tool that allows users to design custom visual graphics. I found a basic logo I liked, a modern design that was clean and simple. Using the logo as a template, I started playing with color and font options. I wanted my app to have a grayscale color palette with pops of bright color that would stand out. I also wanted a font that was fun but easy to read. After some trial and error, I finally created my logo.

Figure 2.3 Record Throws Logo

2.3 Final Design

Using both the page layouts and the Record Throws Logo, I created more elaborate designs using BuilderX (builderx, 2022). BuilderX is a browser-based design tool which helps developers take their ideas and put them on a screen. I laid out my user interface

into a design that was clean and functional. Then, I added the aesthetic of the logo to each page. I used the color from the logo to create my color palette and made all the text the same font as my logo. Each screen in the app was designed to highlight the data, not distract from it. The final design can be seen in the figures below.

Figure 2.3.1 Final Designs for Authentication, Home, Overview and Navigation Screens.

Figure 1.3.2 Final Designs for Data Entry and Display Screens

3. Set Up

3.1 React Native

I chose to implement the app using the React Native (reactnative, 2022) platform for many reasons. First, React Native is compatible on both IOS and Android devices. This increased accessibility maximizing the potential target audience. Secondly, React Native combines native development with React's JavaScript library. This gives the app

the look and feel of a native app. Lastly, React Native was released 8 years ago, meaning plenty of documentation and tutorials exist to help me create my app. Additionally, I have had a little experience working with React Native for my independent study project last semester. This familiarity gave me confidence that I could implement my design in this platform.

3.2 Firebase

I chose to use Google Firebase's Cloud Firestore (firebase, 2022) for my database and user authentication. One reason I choose Google Firebase is because it allows offline support for Apple, Android, and web clients. This is important for users, because track and field is predominantly an outdoor sport. The app needs to have the ability to add data in poor cell service. Another reason I chose to work with Google Firebase is its free multi-platform authentication feature. Firebase Authentication requires users to sign into the app. The credentials are then verified through the backend services and a response is sent to the client. After a successful login, that user's basic profile information can be accessed. Firebase Authentication handled the security involved in logging which gave me more time to work on other elements of my app.

3.3 Redux

I chose to implement my project using Redux (redux, 2022). Redux is a JavaScript library that helps developers organize and manage the state of their application. It does this by saving and modifying the state in a centralized place called a store. This allows the state to be shared among multiple screens while still allowing the developer to know exactly where and how the state is being modified. This is very useful in large applications where the developer might not know the dataflow. Redux easily allows user information to be passed across every screen and component in this app. It will also keep the data organized as the app grows.

4. Implementation

4.1 Database Structure

Firebase's cloud firestore is a hierarchical database that is structured in collections, subcollections and documents. My project contains 9 collections: Users, Hammer, Shot Put, Discus, Weight, HammerMeet, Shot PutMeet, DiscusMeet, WeightMeet. The user's UID from authentication is used as the key for each document in the User collection. The fields stored in the document are email, events, gender, and name. The other eight collections contain a subcollection for each user called userDistances. The user's UID is also the key for these subcollections. The user's subcollection contains a document for each entry they made pertaining to that collection. Each document is given a random identification key and contains the fields date, formatedDistances, and notes. I chose to structure the database this way, so I didn't have to pull excess information each time a document is updated. A picture of the structure can be seen below.

Figure 4.1 Database Structure

4.2 Redux Structure

I used Redux to structure the code that fetches data from the database. Redux uses actions, reducers, and a store to manage the application state. Redux actions

determine what data is being modified and where it is being modified. Redux reducers tell how the data is being modified. The store holds the state of the application, meaning it stores the current data of the application. In my application when an action method is called, it fetches the data from the database. There is an action method that pulls and formats the current user's data for each of the nine database collections mentioned in Section 4.1 Database Structure. Once the data is pulled and properly formatted, the action then dispatches the type of data and the formatted data to the reducer. This tells the reducer which aspect of the state needs to be updated. Then the reducer creates a new state with the updated info.

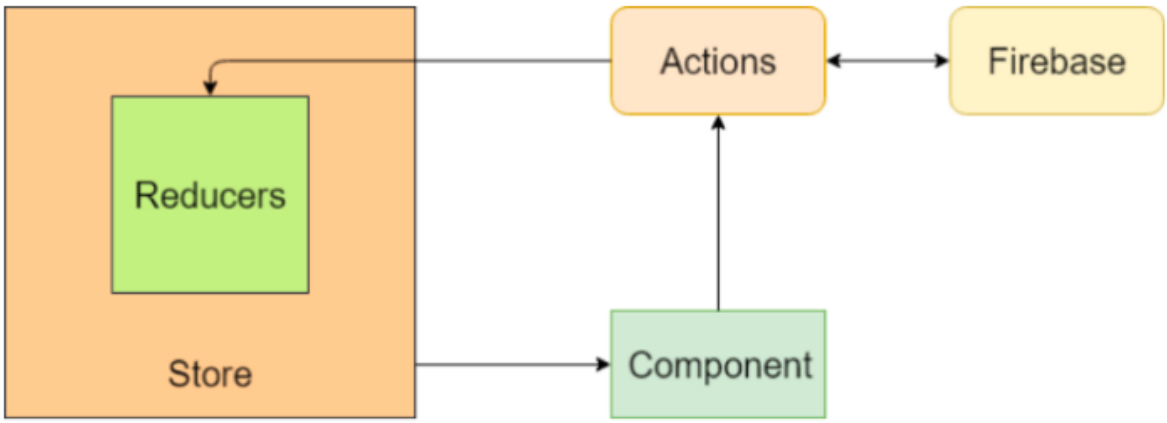


Figure 2.2 Visual Representation of how Redux and Firebase interact with components in Record Throws.

The redux actions format the data into objects depending on which database collection it's being pulled from. The action that pulls from the User collection, formats the document from the database into an object containing all the current user's information. The action that pulls from the four meet collections formats the data into two arrays of objects. The first array contains an object for each document in the user's subcollection. The second array collects the farthest throw from each meet. It looks through all the documents in the database and creates an object for each meet containing the farthest distance, the date, and the ID of that document. The action that pulls from the four practice collections also formats the data into two arrays of objects. The first array contains an object for each document in the user's subcollection. The

second array collects the personal record for each implement thrown. It looks through all the documents in the database and creates an object for the user's farthest distance with each implement. Each object contains the implement, distance, date, and ID of that document.

4.3 Database Safety

To keep the API Key for my database temporarily safe, I created a .env file using react-native-dotenv (github, 2022). This allows me to store the API Key as a constant variable and keep it secret. However, .env files are stored in plain-text and are at risk of being read by unauthorized users. In the future, I will need to make this more secure by creating my own backend that can save the database API Key. The client can then send a request to the backend for the data. The backend can process the request, make the API call, and send the data back to the client. This is the safest way to store API Keys.

4.4 External libraries

Picking an external library is something that takes a lot of time and research. While looking at libraries, I focused on both the function of the component and how the component could be styled. Once I found a library that I like, then I converted the data from the database into the correct format for the component. This manipulation was done using JavaScript and can be seen in the code for each screen. One of the biggest issues I had was converting dates to the correct format for each component. The calendar, graph, and table component all needed the date in a different format. It took time to figure out how to correctly do this in JavaScript. The rest of the time was used to style the components to match my designs. I tried to pick libraries that had good documentation available to help this process, however sometimes this wasn’t possible. In one instance, I had to completely change which library I was going to use because the component couldn’t be styled in a way that would match my other screens. Although this process was time consuming, it resulted in a more cohesive display.

4.5 Implementing pages

Once I set up my database structure, familiarized myself with redux, and figured out my external libraries, the screens were straightforward to implement. The remaining time was used to style each screen. I tried to make the app look as close to my original designs as possible. When doing this, attention to detail is very important. Small changes can make big differences in making screens look uniform and professional.

4.6 Changes to Design

Along the way I had to adjust my original design. Most of these changes were due to time restrictions. One of my original objectives was the ability to link coaches and athletes accounts. This feature was going to take a lot of work to implement it correctly. Not only would I need a way to search and select other users to link accounts, but I would also need to create a way for other users to accept the link request. All this information would need to be stored and updated in the database. On top of all of that, once the link is made, the coach would need to gain access into the athletes' data. To select which athlete's information would be displayed, additional dropdown menus would be needed on all the screens. This would not only take a long time to correctly implement, but it would also require a ton of testing. Instead of stretching myself too thin, I decided to not implement this feature and instead focus on making sure what I did implement was done well.

5. Results

5.1 Testing

I performed functional testing on each screen as they were implemented. The app has been tested on both IOS and Android phones. It has fast load and reaction times, allowing users to quickly log in and start entering data. The tracking screen displaying graphed data has the longest load time. The long load time is due to the external

graphing component used. This can be improved by replacing the component with a different external library or creating my own component. The app is also able to quickly push and pull data from the database allowing users to quickly see their input data. Having a fast load time is important to the user's experience and will allow the app to grow and be modified as it is used by future athletes.

Testing with large amounts of data was limited due to time constraints. I tested the app using some of my own data from this track season. Through the experience, I have some improvements I would like to see on the app. I would like to allow athletes to customize the implement list shown when adding data. This customization would be beneficial because not every thrower trains with the same implements. It would also be helpful to allow users to adjust the time frame being displayed. This would allow users to easily visualize data from a specific season or training block, whereas the current design displays all data at once. This can make the screen messy after more data has been entered. Another improvement that could be made, would be the ability to edit entries. As I was entering data, I realized there was no way to change mistakes. Likewise, I would like to see a way to edit and update profile information. These changes would make my app more functional for users.

5.2 User Testing

To test Record Throws, I gave it to my teammates, asking them to report convenient and cumbersome features. Overall, my teammates were really excited about the app. They all wanted to know how they could download it for themselves. One teammate specifically liked the clean and functional design of the app. He told me he liked that he could see his data being displayed after entry. Another teammate liked the ease of entering data. He told me that if he had this app, he would be more diligent at marking and measuring his throws. One of my teammates used the app during her practice. She loved how fast it was to input data in Record Throws. Specifically, she enjoyed the efficiency of the drop down implement option while entering her training data. My teammates also gave me some suggestions. One suggestion was to automatically

navigate to the data display screen after data is entered. This can give the user confirmation that the data was successfully entered. Another suggestion was to implement a tutorial for first time users so they can learn how to navigate the app. This suggestion came from a couple of my teammate's confusion on where to enter new data.

I also asked my coach to review Record Throws. He was very excited about the app and encouraged me to continue working on it. He wants to see Record Throws published in the app store so he can use it with his athletes. My coach is not very tech savvy, so he liked the simplicity of the design. He said the app had all the basic components he desired. The biggest improvement he would make to the app would include weight room tracking. This would allow athletes to directly see how changes in the weight room affect the distances they are throwing.

5.3 Future Features

There are many features that could be added to further improve Record Throws. One useful feature would be adding both an import and an export option. The export option would allow throwers to export their data into an Excel file. This would allow throws to keep their data when they are done throwing and no longer using the app. It would also allow coaches to have data from their past athletes, allowing them to compare trends to future athletes. The import option would allow athletes to import throwing data from the Track and Field Results Reporting System Website (tfrrs, 2022). The website is a database that stores results from every collegiate meet all the way back to 2009. This would allow throwers to instantly track their meet data without manual input. This feature could also potentially allow users to compare their meet data with the meet data of competitors. This feature could help coaches and athletes predict how far competitors are going to throw at the next meet. Having a better idea of what to expect from competitors allows an athlete to mentally prepare for the meet.

Another feature I would like to add to this app would be to include tracking for javelin throwers. Including javelin would be very important if I were to publish the app because it would be marketable to all throwers. One matrix that would be important for javelin throwers would be volume tracking. This feature would allow javelin throwers to enter the number of throws they took in a practice, helping prevent over training injuries. Volume tracking would be beneficial to the other four events as well. Not only is it important for injury prevention, but it's also important for establishing a training base. Throwers can only handle so much training volume. It is important to periodize volume to have an athlete throw their farthest at the right time in the season. Having volume displayed along with distances would allow coaches to find an athlete's optimal training volume.

Another helpful feature to implement would be adding optional fields to each entry. This would organize information that a user might want to enter in the notes section of the current app. One optional field that could be added is weather. Since track and field is an outdoor sport for most of the year, weather plays a big role in performance. It would be helpful to look back on a bad practice and see if cold or rainy weather was a factor. Discus is an event that is highly affected by wind speed and direction. It is not unheard of for an athlete to have a massive jump in their personal best if they catch a good wind. It would be helpful to record wind in those outlier performances. To do this without bogging down the data entry process, weather data could automatically be entered based on the user's location. Another optional field could be a section for users to upload videos from that day's practice or meet. This would allow users to look back not only at how they threw that day but also would allow them to see their form. This could potentially be a way for athletes and coaches to share videos after practices or meets, since both would have access through their linked accounts.

The next feature I would like to include is a goals section of the app. This would allow users to set goals for the season and could notify both a coach and an athlete when a goal has been reached. Goals are very important to athletes and this app could encourage athletes to set season goals. Additionally, this would allow an athlete to

have their goals with them, constantly reminding them what they are working towards. A related feature I would like to implement is a pop-up notification that informs an athlete they hit a new personal record after the data is entered. Personal records are very important in the throwing community, and a celebratory notification would be fun to include.

One feature that was suggested is an option for a coach to enter throwing workouts for their athletes. This would allow coaches to plan workouts for the week and publish them to their athletes, giving athletes time to prepare. There are so many features that could be included in this app. It makes me very excited about what this app could turn into and how many athletes it can help.

6. Conclusion

6.1 Next Steps

To continue this project, first I would like to implement the coaching view that was in my original design. I think it is a very valuable feature for this app. Next, I would like to make the changes mentioned in section 5.1 Testing to improve the apps functionality. After that I would like to extensively test the app on more types of devices like iPads. Many throws coaches use iPads to record and analyze videos, so functionality on these devices is important. Once complete, I would like to beta test with my current teammates using TestFlight (developer.apple, 2022) for IOS and Google play (developers.google, 2022) for Android. These platforms would allow the app to experience real world scenarios and be tested using large amounts of data. Eventually, I would like to see Record Throws get on the IOS and Android app stores. To have my app be published would be a long process that includes an app review. Both IOS and Android have high standards to ensure their users are installing safe, high-quality apps. The review process requires meeting testing, safety, performance, and design standards. Publishing Record Throws on the app stores would be a huge undertaking, but it would be very rewarding to see it used by Kentucky throwers.

6.2 Accessing project

My final code can be found on GitHub at https://github.com/mnle239/RecordThrows. The app contains 12 screens, 14 shared components and 4 redux files totaling just under 4,000 lines of code. To download and run the code on a device, use the following terminal commands.

git clone https://github.com/mnle239/RecordThrows.git
cd RecordThrows
npm install
expo start -c

The first and second command will download the repository from GitHub and enter the downloaded folder. The third command will download Expo CLI (docs.expo , 2022) for the project. Expo CLI is a React Native developer tool that aids in creating projects, viewing logs, and opening simulators. The last command will start Expo CLI. A Web Page GUI will open and show logs along with options to open simulators.

6.3 Final Thoughts

I really enjoyed working on Record Throws this semester. It was fun to see a need and create a product based on that need. It was especially rewarding creating an app about something I am so passionate about and would have used during my track and field career. I had the privilege of bouncing ideas off my coach, teammates and family members who all were big contributors in helping me narrow down features and designs. Along the way I learned a lot about design and how to display a lot of information in clean and organized ways. I learned a lot about the react native environment and had ample opportunity to utilize its many features. I also learned a lot about databases and how to properly structure a database. I am very grateful for the opportunity to create this app and I am very proud of the finished product. I hope to continue working on the app and maybe one day see it on the app store.

7. References

https://www.myfitnesspal.com/. Accessed on April 4, 2022.

https://www.sleepscore.com/. Accessed on April 4, 2022.

https://recoverathletics.com/. Accessed on April 4, 2022.

https://ukathletics.com/sports/mens-track-and-field/roster/coaches/keith-mcbride/5039. Accessed on April 4, 2022.

https://www.google.com/sheets/about/. Accessed on April 4, 2022.

https://www.homecourt.ai/. Accessed on April 5, 2022.

https://runkeeper.com/cms/. Accessed on April 5, 2022.

https://www.nike.com/nrc-app. Accessed on April 5, 2022.

https://www.jefit.com/. Accessed on April 5, 2022.

https://www.hudl.com/products/hudl/professional. Accessed on April 5, 2022.

https://www.teamusa.org/usa-track-and-field/athletes/Micaela-Hazlewood. Accessed on April 5, 2022.

https://www.microsoft.com/en-us/microsoft-365/excel. Accessed on April 5, 2022.

https://www.canva.com/about/. Accessed on April 6, 2022.

https://builderx.io/. Accessed on April 6, 2022.

https://reactnative.dev/. Accessed on April 6, 2022.

https://firebase.google.com/. Accessed on April 6, 2022.

https://redux.js.org/. Accessed on April 6, 2022.

https://github.com/goatandsheep/react-native-dotenv. Accessed April 7, 2022.

https://www.tfrrs.org/. Accessed April 7, 2022.

https://developer.apple.com/testflight/. Accessed April 7, 2022.

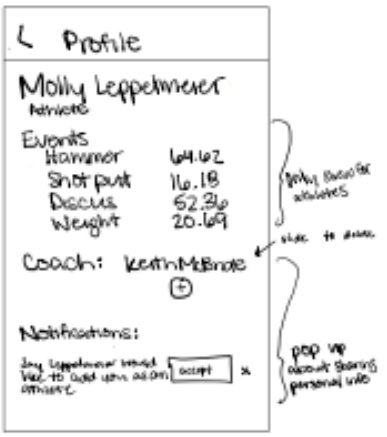
https://developers.google.com/android/guides/beta-program. Accessed April 7, 2022.

https://docs.expo.dev/. Accessed April 7, 2022.

8. Appendix


LOGO
username
password
Log In
New here? Sign Up

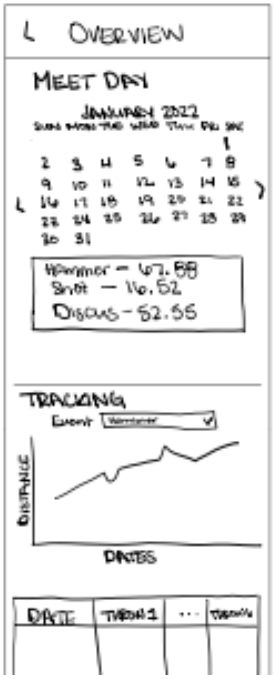

CREATE Account
name
email
password
confirm
Drop down section on Athlete selection
SIGN Up
Have account? Log in


Profile
Molly Leppelmeier
Athlete
Events
Hammer 64.62
Shot put 16.18
Discus 52.36
Weight 20.69
Coach: KeithMcBride
Notifications:
accept
pop up


HAMMER
DISCUS
SHOT PUT
WEIGHT
MEET INFO


Overview
HAMMER
Athlete
(if coaching view)
Only time Select multiple athletes
JANUARY 2022
4K — Distance
5K — Distance
selection goes to view data screen
TRACKING
DISTANCE
DATES
DATE
4K
5K


Molly Leppelmeier
Coach: KeithMcBride
PROFILE
CALENDAR
PERSONAL RECORDS
TRACKING
FD


Overview
MEET DAY
JANUARY 2022
Hammer — 67.88
Shot — 16.52
Discus - 52.55
TRACKING
Event
DISTANCE
DATES
DATE


PERSONAL RECORDS
HAMMER
Coach select one athlete
SHOT PUT


(if coaching view)
Athlete
Change dates on selection
Tracking
HAMMER
4K
5K
TRACKING
DISTANCE
DATES
SHOT PUT
4K
5K
TRACKING
DISTANCE
DATES

CALENDAR

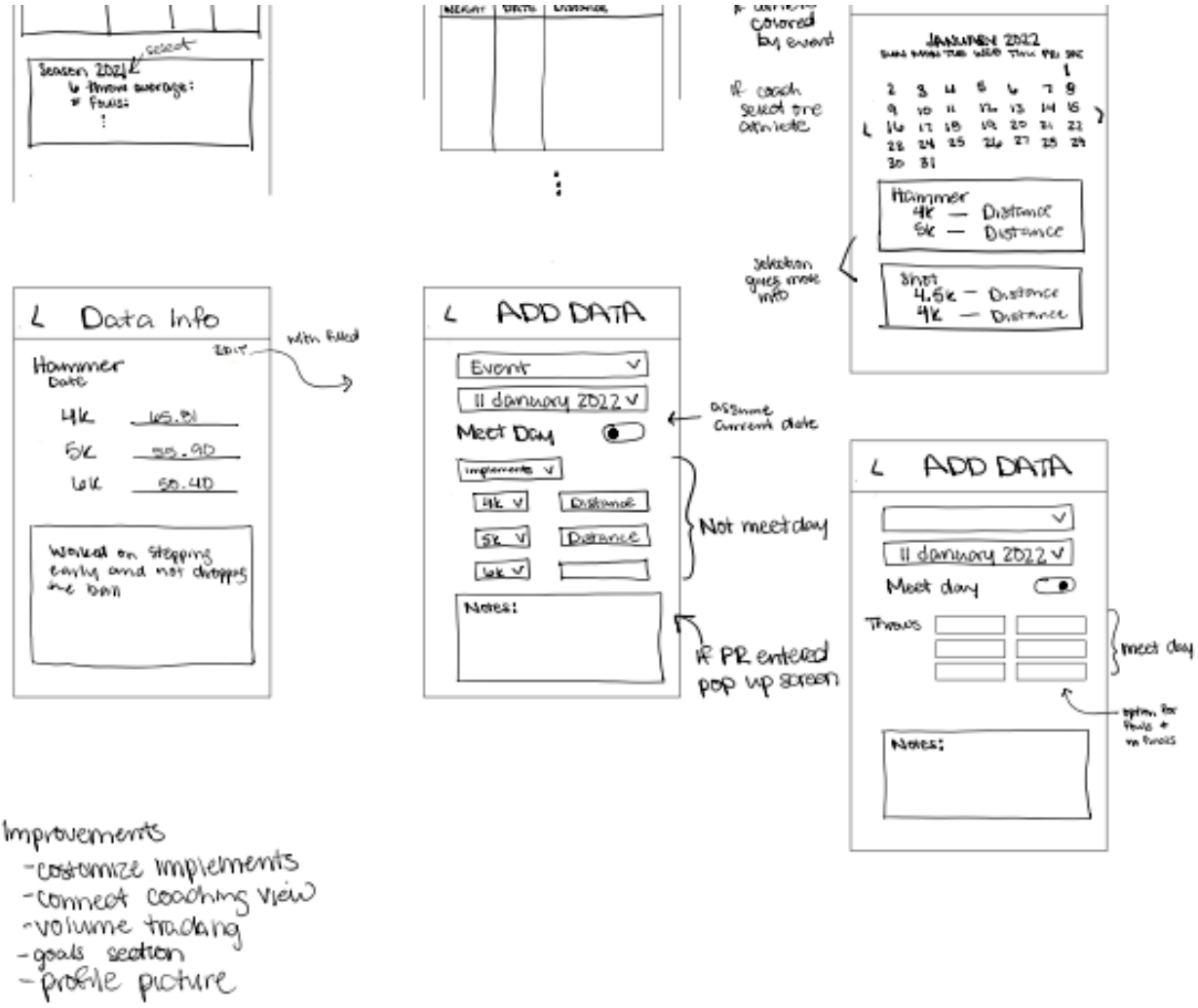
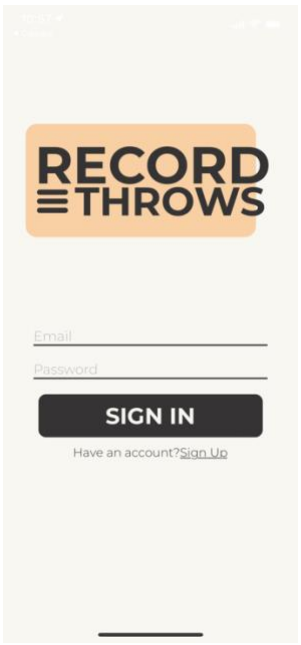
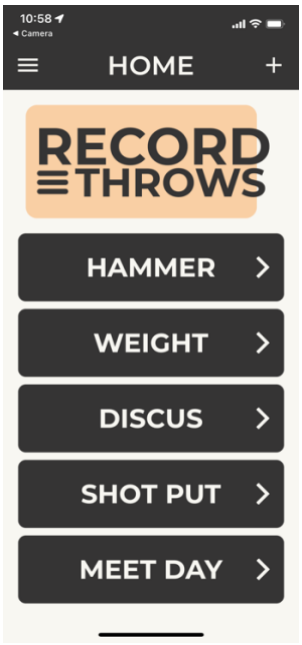
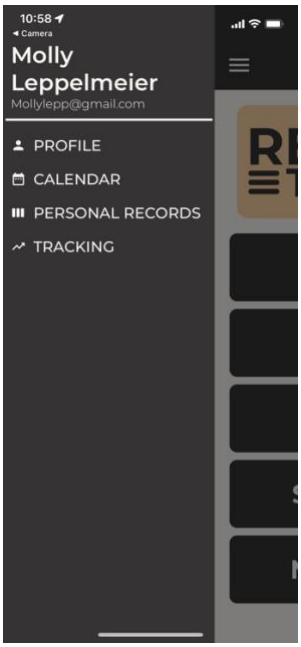


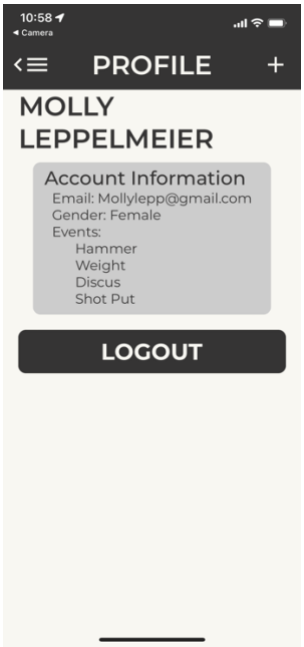
Figure 8.1 Original Screen Designs


RECORD THROWS
Email
Password
SIGN IN
Have an account?Sign Up

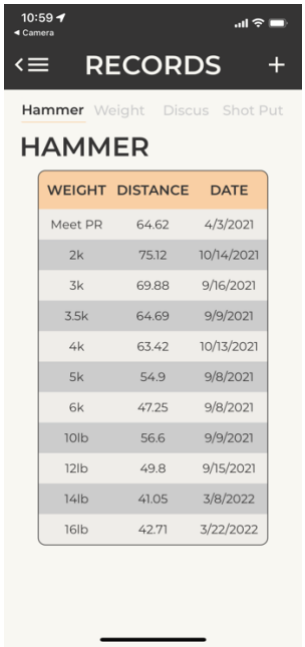

RECORD THROWS
CREATE ACCOUNT
Name
Email
Password
Confirm
Female
Male
Hammer
Weight
Shot Put
Discus
SIGN UP
Have an account? Sign In

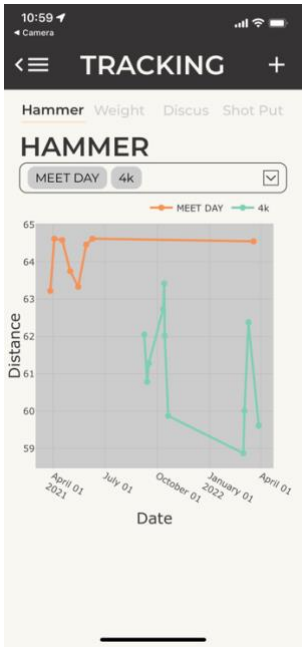

10:58
Camera
HOME
RECORD THROWS
HAMMER
WEIGHT
DISCUS
SHOT PUT
MEET DAY


10:58
Camera
Molly Leppelmeier
Mollylepp@gmail.com
PROFILE
CALENDAR
PERSONAL RECORDS
TRACKING


10:58
Camera
PROFILE
MOLLY LEPPELMEIER
Account Information
Email: Mollylepp@gmail.com
Gender: Female
Events:
Hammer
Weight
Discus
Shot Put
LOGOUT


10:59
Camera
CALENDAR
March 2022
Sun Mon Tue Wed Thu Fri Sat
Hammer Weight Discus Shot Put
HAMMER MEET
1: 62.07
2: 62.18
3: 64.55
4: 61.25
5: 60.76
6: 62.45
DISCUS MEET
1: 44.66
2: Foul
3: 46.84
4: Foul
5: 47.15
6: Foul


10:59
Camera
RECORDS
Hammer Weight Discus Shot Put
HAMMER
WEIGHT DISTANCE DATE
Meet PR 64.62 4/3/2021
2k 75.12 10/14/2021
3k 69.88 9/16/2021
3.5k 64.69 9/9/2021
4k 63.42 10/13/2021
5k 54.9 9/8/2021
6k 47.25 9/8/2021
10lb 56.6 9/9/2021
12lb 49.8 9/15/2021
14lb 41.05 3/8/2022
16lb 42.71 3/22/2022


10:59
Camera
TRACKING
Hammer Weight Discus Shot Put
HAMMER
MEET DAY 4k
MEET DAY 4k
Distance
Date

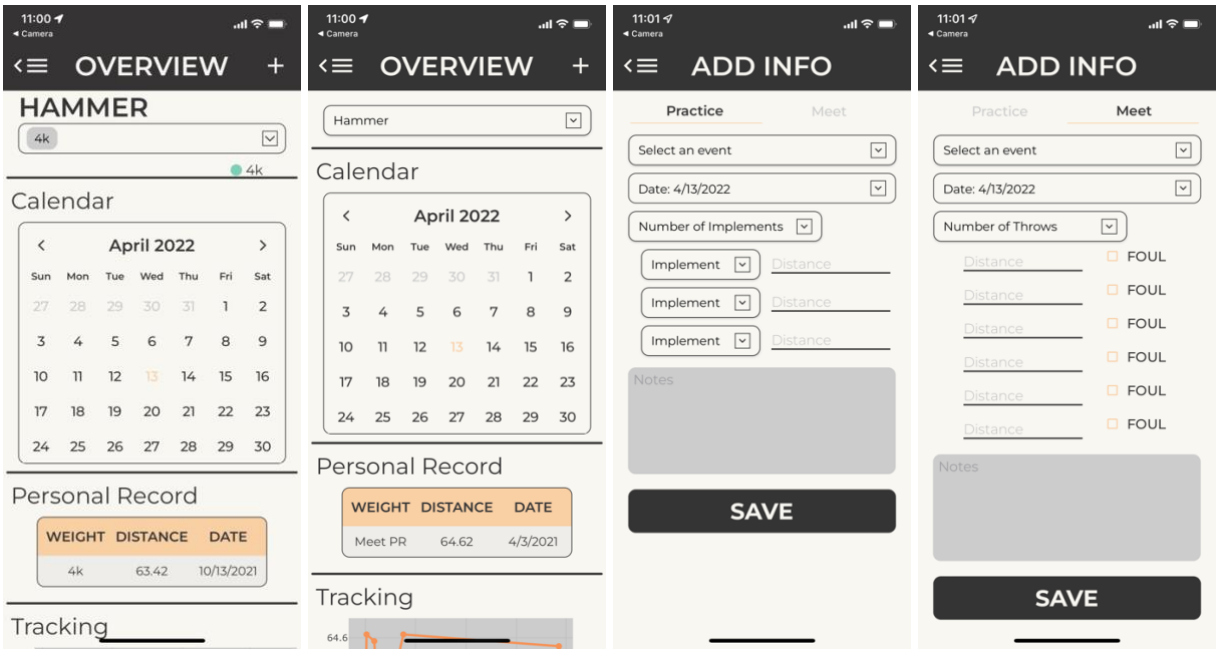


Figure 8.2 Screenshots of final implementation