



MOBILE SOFTWARE ENGINEERING II

DLBCSEMSE02

TASK

Task 1: Develop your individual tracking app

COURSE OF STUDY

SOFTWARE DEVELOPMENT

MATRICULATION NUMBER

92107689

TUTOR

Christian Remfert

DATE

22nd August, 2024

Table of Contents

1. Introduction.....	3
1.1. Goal and Initial Motivation.....	3
1.2. Data Collection and Analysis.....	3
1.3. Approach taken in Data Collection.....	3
1.4. Problem Solving Process.....	4
1.5. Planning and Organizational Activities.....	4
1.6. Reason for Selecting This Method.....	4
2. Implementation.....	5
2.1. Project Management.....	5
2.1.1. Project Structure.....	5
2.1.2. Progress in the Project.....	6
2.1.3. Progress Report.....	6
2.1.4. Tools and software.....	6
2.2. Technical Implementation.....	7
2.2.1. Wireframe.....	7
2.2.2. App components and their Interaction.....	9
2.2.3. Important Functions of the app.....	10
3. Conclusion.....	17
4. References.....	19

1. Introduction

In modern football, data-driven decisions have become crucial in optimizing player performance and ensuring the success of a team. One of the key metrics for evaluating a player's fitness and readiness for a match is the distance covered during training sessions. This project focuses on developing a tracking application that monitors the distance covered by football players during their training sessions, evaluates their performance, and determines their eligibility for upcoming matches based on predefined criteria.

1.1. Goal and Initial Motivation

The main aim behind this project is to create a system that follows football players' movement during their practice. The reason behind this idea comes from demand of having more objective and measurable way of evaluating whether a player is fit or prepared to play in games. With technology being used to acquire physical data of players' performance, it means coaches including team managers can arrive at much better choices in relation to team selection. The project objective is therefore meant to ensure that only those with necessary fitness levels are chosen for matches thereby improving their teams' performances (Rampinini et al., 2007).

1.2. Data Collection and Analysis

The data collection is a very vital part of this project. Every player has a wearable technology device in this case it's a GPS enabled watch which keeps track on the distance they run during training sessions. These training session take place twice every week, on Tuesdays and Thursdays. The watches are collected after every session and the distance information extracted from them is noted down for analysis later on. By totaling the distances covered in both of these sessions, their average figures for each player are obtained. To establish if a player will be considered for selection or not, there is need to compare the above average with the minimum requirement of 12.5 km/hour that must be covered at least (Coutts & Duffield, 2010).

1.3. Approach Taken in Data Collection

This study's data collection approach is systematic as well as technologically oriented. Players' movements during training sessions are tracked using wearables that offer real-time information concerning distances covered by individual players at any given time. This method ensures that data collection remains accurate, consistent and objective at all times. Use of GPS watches removes manual recording errors and enables precise monitoring of individuals' performances throughout playing episodes. After which such details are compiled and their

means worked out to determine who among them would qualify to participate depending upon covering more kilometers than others did (Dellaserra et al., 2014).

1.4. Explanation of Problem-Solving Process

The process used to tackle this problem involved identifying the main performance indicators that could be employed to measure player fitness, in this case distance covered during training. The problem was how to build a system that would effectively collect, analyze and compare such data with an established norm. This challenge was resolved by making use of wearable technology for data collection and implementing an automated system for data analysis. Additionally, attention was paid to ensuring that the data processing workflow was smooth, precise, and dependable (Malone et al., 2017).

1.5. Planning and Organizational Activities

This project was conducted in several stages which commenced with identification of the problem and defining the scope of the project. A detailed plan specifying how information would be obtained from various sources including methods of data collection, analysis procedures as well as criteria for choosing players were developed. The wearable devices were sourced and adapted to suit those needed in the project. Creation of algorithms for processing data in tracking application development coupled with designing features for storing, calculating and comparing them against some predetermined benchmark occurred simultaneously. To ensure that it proceeded according to plans and met its objectives, there were regular meetings held at intervals while reviewing its progress from time to time (Kerzner, 2017).

1.6. Reason for Selecting This Method

The decision to employ wearable technology for data collection was based on the requirement of accuracy, efficiency, and real-time data capturing. One way that minimized human error and made it feasible to analyze the data quickly was through this approach. Moreover, by using an ordered technique of finding average distance moved in order to compare them among players, it allowed the selection criteria which were objective and fair. Thus, this approach agrees with the project goal of leveraging available information in making prudent choices hence improving team performance by selecting only players who meet relevant fitness levels (Brocherie et al., 2014). This systematic way of tracking and analyzing player's performance ensures that the selection process is based on accurate information, accountable and consistent with the quest for excellence in team sports.

2. Implementation

2.1. Project Management

Efficient project management is crucial to ensuring a seamless and efficient development of the tracking app for football players, which meets all predefined objectives. The actions taken to manage and track the progress of this project are outlined below, with a focus on project structure, project progress tracking as well as reporting.

2.1.1. Project Structure

During initiation and planning, the project objectives was clearly defined, goals clearly states deliverables, and success criteria of the project. In the Scope Definition, identification was made on data collection features, analyses and reportage functions in the apps. In resource allocation, the Coaches was tasked with the responsibility of collected the wearable devices and make sure the players input their data.

i. Design and Development:

System Architecture: The wireframe of the app is developed which gave the IU predefined look and feel of the app

Technical Design: In this app, the technology are Java, and flrebase

Implementation: The app was build in three weeks sprints with regular code reviews and testing.

ii. Testing and Validation:

Unit Testing: The individual functional components are tested for accuracy.

Integration Testing: Ensured all app components work together seamlessly.

User Acceptance Testing: A selection of 20 players was selected to use the app and give reports.

iii. Deployment and Maintenance:

Deployment: The app was deployed to github.

Documentation: A detailed documentation is prepared covering user guides, technical specifications, system requirements.

Maintenance: As a developer of the app, I take notes of all the reports, Monitor the app, address any issues, and plan for future updates.

2.1.2. Progress in the project

The project was well-serviced by a combination of agile practices and regular monitoring mechanisms:

i. Agile Methodology:

Scrum: Agile management framework that focuses on iterative development and continuous reevaluation of progress.

Sprints: This project was divided into weekly sprints lasting three weeks in total. Completed features were reviewed at the end of each sprint before planning for the next.

Daily Stand-Ups: The routine stand-up meetings were held often to highlight how far we have gone, challenges we faced, adjust where necessary.

2.1.3. Progress Reports:

Weekly Reports: A weekly report was produced containing all tasks that were finished alongside those in progress and issues encountered. These reports were shared with stakeholders.

Milestone Reviews: End of phase milestone reviews were undertaken to measure progress against the project plan such as demonstration of completed features, feedback sessions, and planning for next phase among others

2.1.4. Tools and Software:

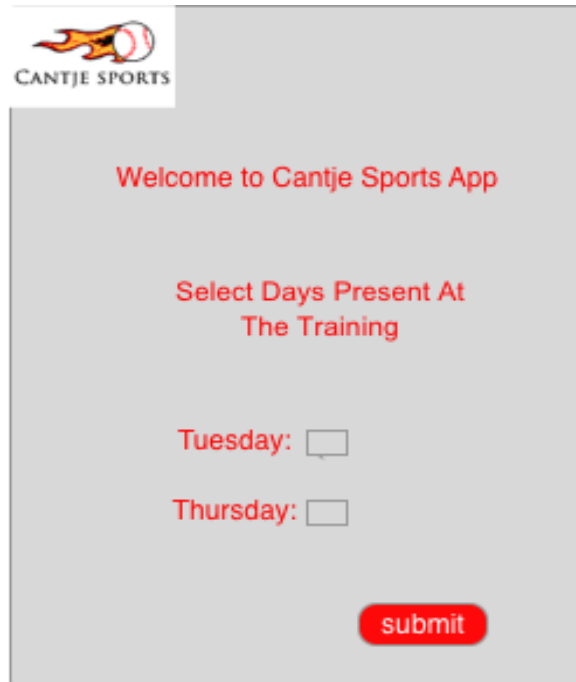
Project Management Tools: Jira was used to manage tasks, monitor the progress and log issues. This was done in order to keep a record of all activities related to the project.

Version Control: Git and GitHub were employed for source code management. Code integrity was maintained through regular commits and branches, which enabled code changes over time.

Continuous Integration: CI/CD pipelines were established for automated testing and deployment purposes, where tests could be easily run to check and ensure quality app quickly.

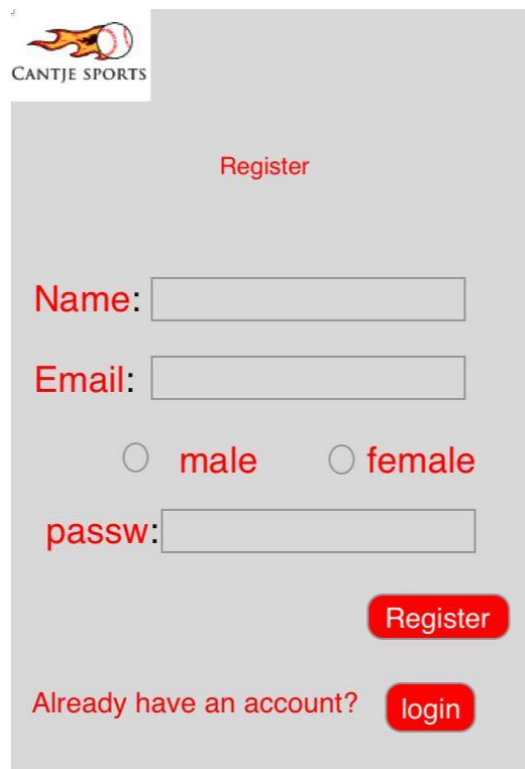
2.2. Technical Implementation

2.2.1. Wireframe




The wireframe shows a mobile app interface for 'Cantje Sports'. At the top left is the logo, which consists of a stylized orange and yellow flame above a white baseball with red stitching, with the text 'CANTJE SPORTS' below it. The main content area has a light gray background. It starts with the text 'Welcome to Cantje Sports App' in red. Below this is the text 'Select Days Present At The Training' in red. There are two lines of text: 'Tuesday:' followed by a small white square input box, and 'Thursday:' followed by a small white square input box. At the bottom right is a red rounded rectangle button with the word 'submit' in white.

(Chris, 2024a).



The wireframe shows a mobile app interface for 'Cantje Sports' with a registration form. At the top left is the same logo as the previous screen. The main content area has a light gray background. It starts with the text 'Register' in red. Below this are three input fields: 'Name:' followed by a white rectangular box, 'Email:' followed by a white rectangular box, and 'passw:' followed by a white rectangular box. Between the email and password fields are two radio button options: a radio button followed by the text 'male', and another radio button followed by the text 'female'. At the bottom right is a red rounded rectangle button with the word 'Register' in white. Below this button is the text 'Already have an account?' followed by a red rounded rectangle button with the word 'login' in white.

(Chris, 2024b).



CANTJE SPORTS

Welcome to Cantje Sports App


Enter the kilometre on this days

Tuesday: km

Thursday: km


submit

(Chris, 2024c).



CANTJE SPORTS

CONGRATULATION!



13.0 km

You have achieved all that is
necessary to be selected

(Chris, 2024d).



(Chris, 2024e).

2.2.2. App Components and their interaction

The dual-fragment approach on the Home screen of the app features a registration fragment and a login fragment. With this design, new users can easily sign up while for returning ones can quickly log in. Once one has logged in successfully either by signing up or logging in, they are directed straight to the Input Activity screen where they insert their training data that includes distance covered on Tuesday and Thursday.

Notably, the Home screen implements an `onStart()` method, leveraging Firebase to automatically detect any previously authenticated users who have not explicitly signed out. This ensures that returning users are not prompted to log in again, thereby providing a smoother and more efficient user experience.

On the Input Activity screen, users input various details related to their weekly training sessions. After submitting both Tuesday's and Thursday's entries, the application moves them to Result Activity view. It is dynamically generated using input information such as these and gives ideas about how well someone performed and whether they qualify for an upcoming game or not.

The application also has an all-embracing menu with a number of crucial functions. Even from any screen, users can also log out by closing their session and returning to the Home

screen. Besides, the menu also offers them an option to navigate back to previous activity screen which ensures flexibility and user-friendliness.

In general, the app structure highlights smooth sailing across menus, dynamic content rendering as well as other user-oriented features including session persistence that support access to major services.

2.2.3. Important functions of the app

1. ActionBar:

It is important to maintain a professional and cohesive look throughout the application. Branding elements such as the logo and a consistent color theme establish familiarity and make it appear as if the app was designed in a manner that pleases users too. Moreover, ActionBar title enhances usability by offering clear cues for navigation hence making it easier for users to determine where they are within the app. The UI design's meticulousness contributes to both functional and user experience oriented efforts of the app.

i. **ActionBar Initialization:** ``actionBar = getSupportActionBar();``

The `getSupportActionBar()` method retrieves the ActionBar associated with the activity, allowing it to be customized.

ii. **Title Setup:** ``actionBar.setTitle(R.string.title);``

This line sets the title of the ActionBar using a string resource (`R.string.title`). The title helps users identify the current screen or context within the app.

iii. **Logo Display:** ``actionBar.setDisplayHomeAsUpEnabled(true);``

`actionBar.setLogo(R.mipmap.cantje_logo);`

`actionBar.setDisplayUseLogoEnabled(true);``

These lines enable the display of a logo (`R.mipmap.cantje_logo`) on the ActionBar, reinforcing brand identity. `setDisplayHomeAsUpEnabled(true)` and `setDisplayUseLogoEnabled(true)` ensure that the logo is visible alongside the title.

iv. **ActionBar Background Color:** ``actionBar.setBackgroundDrawable(new
ColorDrawable(getResources()`

`.getColor(R.color.sky_blue)));``

The background color of the ActionBar is customized to sky_blue, aligning with the app's color scheme and improving visual consistency.

2. onStart Method

To ensure a smooth user interface, this is the best implementation needed by the app which will take log in users directly into its core functionality (kilometer tracking) without requesting them to login again. The onStart() method, on the other hand, ensures seamless session continuation improving user comfort and responsiveness of the application. This approach also enhances security since particular parts of the app are only made available to authenticated users alone (Google, 2024).

The code snippet provided below was developed by the author as part of the project implementation. It demonstrates the functionality for user authentication and session management in the Android application.

```
@Override
protected void onStart() {
    super.onStart();
    FirebaseUser currentUser= auth.getCurrentUser();
    if(currentUser !=null){
        Toast.makeText(this, "user exist", Toast.LENGTH_SHORT).show();
        gotoCalculation = new Intent(this, kilometerCover.class);
        startActivity(gotoCalculation);
        finish();
    }else{
        Toast.makeText(this, "no user", Toast.LENGTH_SHORT).show();
    }
}

;
```

To include the custom behavior that runs when the activity just started being created and is about to appear on the screen, it overrides onStart() method. In order for this to happen, it first executes base class's implementation of onStart() using super.onStart(). Firebase authentication session is checked by the auth.getCurrentUser() method to know whether someone already logged in. If a user has been found (currentUser != null) then a Toast message shows that there is a user. The app will then navigate to kilometerCover activity using an intent and finish() closes the current activity hence the back button cannot be pressed again

to get back to login screen. But if no user is found (`currentUser == null`), this toast should come up indicating there's no authenticated user at this point so that they remain on the same screen.

3. Login Function

The code is very important; it helps in controlling user authentication within the system. By using success and failure listeners, the application informs users about how their login attempts have been successful or have failed. The latter ensures that authenticated users can be taken directly to their main functionalities whereas the former enables them get error messages regarding login failure in order to improve their user experience. This enhances reliability and usability of the app's authenticating process.

The code snippet provided below was developed by the author as part of the project implementation. It demonstrates the functionality for user login

```
private void login(){
//    using firebaseAuth email and password signin

auth.signInWithEmailAndPassword(email.getText().toString(),password.getText().toString())
//    adding firebase success listener
    .addOnSuccessListener(new OnSuccessListener<AuthResult>() {
        @Override
        public void onSuccess(AuthResult authResult) {
            Toast.makeText(requireContext(), "Login sucessful",
Toast.LENGTH_SHORT).show();

            gotoCalculation = new Intent(requireContext(), kilometerCover.class);
            startActivity(gotoCalculation);
            getActivity().finish();
        }
    })
//    Firebase failure listener
    .addOnFailureListener(new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception e) {
            Toast.makeText(requireContext(), "failed " +e.getMessage(),
Toast.LENGTH_SHORT).show();
        }
    })
}
```

```
});  
  
}
```

In an Android app, this code snippet shows how to log in a user with the help of Firebase Authentication. It features both listeners for success and failure that respond to these authentication processes. The method `signInWithEmailAndPassword()` is used to authenticate users via Firebase using the email address and password supplied. To obtain strings for email and password from text fields in `EditText`, `getText().toString()` methods are invoked.

`addOnSuccessListener` is used as a listener which triggers when sign-in has been successful, this shows a `Toast` message on the screen for successful login.

An `Intent` is created that takes the user to `kilometerCover` activity.

`startActivity(gotoCalculation)` initiates another activity.

`getActivity().finish()` kills the parent activity of this current fragment so that if a user tried to go back by pressing his/her back button he/she would not be redirected back on the login page.

`addOnFailureListener` function adds a listener which triggers if the sign-in fails. In case of failed authentication, A `Toast` message is displayed with error message returned by Firebase (`e.getMessage()`) that tells what went wrong (e.g., incorrect password or email).

4. Registration Function

This code handles all of the user registration part, merging Firebase Authentication and Firestore. The success listener guarantees that after a user has registered, their information is immediately stored in Firestore and navigated to the next activity. Moreover, the failure listener gives a better experience in case the Firestore operation fails by offering informative feedback; thus, improving robustness and usability of the application. This example illustrates good management of two backend services (Firebase Authentication and Firestore) in an Android app.

The code snippet provided below was developed by the author as part of the project implementation. It demonstrates the functionality for user registration in the Android application.

```
auth.createUserWithEmailAndPassword(playerEmail,playerPassword).addOnSuccessListener(new OnSuccessListener<AuthResult>() {  
    @Override  
    public void onSuccess(AuthResult authResult) {  
        Map<String, Object> users= new HashMap<>();  
        users.put("name",playerName);  
        users.put("email", playerEmail);  
        users.put("sex", playerSex);  
        db.collection("users")  
            .add(users)  
            .addOnSuccessListener(new OnSuccessListener<DocumentReference>() {  
                @Override  
                public void onSuccess(DocumentReference documentReference) {  
                    Toast.makeText(requireContext(), "Registration Successful",  
Toast.LENGTH_LONG).show();  
                    gotoSexSelectionIntent= new Intent(requireContext(), kilometerCover.class);  
                    startActivity(gotoSexSelectionIntent);  
                    getActivity().finish();  
                }  
            })  
            .addOnFailureListener(new OnFailureListener() {  
                @Override  
                public void onFailure(@NonNull Exception e) {  
                    Toast.makeText(requireContext(), "failed! "+e.getMessage(),  
Toast.LENGTH_SHORT).show();  
                }  
            });  
    }  
});
```

This code segment is for registering a new user in Firebase Authentication, creating a Firestore document to save user information, and managing both success and failure

scenarios. The method, `createUserWithEmailAndPassword()`, is used to create a new user with the provided email (`playerEmail`) and password (`playerPassword`). The `addOnSuccessListener` responds when registration was successful. A map is created for storing name, email and sex of the player. These key-value pairs will be stored in a firestore document.

Use `add()` function to append users map to Firestore's users collection. In case it's added successfully, On successful registration, display a Toast message. Create an Intent for navigating to `kilometerCover` activity. New activity is started using `startActivity()` method and finally close the current fragment activity by `getActivity().finish()`;

In case adding the user data into Firestore fails that is `addOnFailureListener` triggers showing a Toast message with error message.

5. KmCalculation Function

The importance of this technique is to decide if the footballer has reached the minimum distance, which allows him or her to participate in a game. In this method, distances are weighted and averaged so that there is an even appraisal of how well the player did in two training sessions. The employment of this keyword makes it obvious that the method is operating on object-specific instance variables, making things easier to understand logically.

The code snippet provided below was developed by the author as part of the project implementation.

```
public double calculateTotalKilometre() {  
    double tues= this.tuesday_distance * 1.5;  
    double thur= this.thursday_distance *1.5;  
    this.totalKilometre = (tues + thur)/2;  
    return totalKilometre;  
}
```

A technique has been developed for calculating the average total distance that a footballer covers during his or her routine practice on Tuesday and Thursday each week. The calculation involves weighting the distances and then averaging them out.

For example, `tuesday_distance` gets multiplied by 1.5 to give it the weight it deserves. This is stored in variable name 'tues'. Similarly, a weight of 1.5 is used to multiply `thursday_distance`, storing its result in the variable called 'thur'.

Total weighted distance is obtained by adding up weighted Tuesday and Thursday distances (tues + thur) together then dividing the sum by two. It represents the average for both training sessions put together. Result is saved in the field totalKilometre of that class.

The method returns the calculated average total kilometre

6. Dynamic Result Rendering

The code is a necessary element of the feedback loop in the tracking application. The outcome of this for players is shown by it and gives visual as well as written hints when their performance falls below par. By using Toast, ImageView, and TextView, feedback is made instant and interesting so that players can see how much they have improved or what they need to do more. This improves user experience by making it lively and educative.

```
if (result <12.5){  
    imageView = findViewById(R.id.imageView);  
    imageView.setImageResource(R.mipmap.maguire);  
    congratText.setText(R.string.failuretext1);  
    congratText2.setText(R.string.failure_advice);  
}
```

The code snippet below is the one that displays distance covered by a player during training as a result of performance. It checks whether the player meets the required threshold (12.5 km/hr) and provides feedback accordingly. The calculated average distance covered, referred to as the output, is revealed to the user through a Toast message. This message appears briefly on the screen to inform the user of their performance. This piece of code checks if a result is less than 12.5. In case any particular athlete fails to meet this minimum requirement.

Updating an ImageView element with an image called “maguire” which aims at visually symbolizing failure and giving immediate feedback to the user. Two TextView elements (congratText and congratText2) receive messages about failure. The first message indicates that the player has not met requirements while the second one gives some advice or motivation for improvement.

3. Conclusion

All these functionalities were successfully achieved in the development of this football player tracking app, hence it was a victorious quest. The aim behind coming up with the app was to control and measure how far players go during their trainings and thus determine if they are fit for upcoming matches. By properly monitoring training kilometers vis-a-vis established reference points, the application achieves its goal by assisting coaches make more rational decisions on whom to pick.

Although robust in its current state, there is room for future improvement in implementing the app. One possible enhancement might involve adding more functionality to improve players' performance. As an example, an AI-driven module that considers historical data can provide individualized training recommendations which will help the athletes improve their speed, stamina as well as general fitness. Furthermore, live feedback during practice sessions or performance-based motivational tips would not only motivate but also keep them engaged while at it improving their workout outcomes.

This project gave me an opportunity to expand my knowledge of software development processes, especially in setting and attaining sprint objectives. To successfully accomplish this task, I had to plan properly, communicate clearly with various stakeholders and put down every detail in writing. All those activities were necessary for the maintenance of the project's course as well as satisfaction of its requirements.

The technical benefit that I gained was a better comprehension when it comes to various Android components. I became good at handling fragments which was vital in developing a dynamic user interface that could deal with multiple activities seamlessly. Additionally, I discovered how menus can be designed and implemented to provide users with intuitive navigation throughout the app. Moreover, for efficient recording and management of training data, I exploited Android's SQLite database thereby ensuring that data persistence is maintained without any performance problems.

The project also used Firebase extensively, an external service which has significantly boosted the app's capabilities. With this technology, the application was able to handle user authentication, update in real time and be cloud-based; hence is more powerful and friendlier

to users. This made it possible for me to expedite the backend while concentrating on providing a seamless user experience.

The project also used Firebase extensively, an external service which has significantly boosted the app's capabilities. With this technology, the application was able to handle user authentication, update in real time and be cloud-based; hence is more powerful and friendlier to users. This made it possible for me to expedite the backend while concentrating on providing a seamless user experience.

In summary, this work has been a holistic learning process involving technical expertise combined with project administration practices. The success of implementing this app and finding new ways through various difficulties have laid a foundation for my next undertaking that will have similar expectations. In anticipation of future opportunities for growth, I am certain that skills and knowledge gained during this exercise will form strong basis for addressing even more intricate as well as innovative software solutions.

GITHUB REPOSITORY

https://github.com/Etiti27/iu_tracking_app

4. References

1. Brocherie, F., Girard, O., Forchino, F., Al Haddad, H., Dos Santos, G. A., & Millet, G. P. (2014). Relationships between anthropometric measures and athletic performance, with special reference to repeated-sprint ability in young soccer players. *The Journal of Sports Medicine and Physical Fitness, 54*(1), 14-21.
2. Chris (2024a). Login screen wireframe for tracking app [Screenshot]. Sketch. <https://www.sketch.com>.
3. Chris (2024b). Registration screen wireframe for tracking app [Screenshot]. Sketch. <https://www.sketch.com>
4. Chris (2024c). User Input screen wireframe for tracking app [Screenshot]. Sketch. <https://www.sketch.com>.
5. Chris (2024d). KM not met screen wireframe for tracking app [Screenshot]. Sketch. <https://www.sketch.com>.
6. Chris (2024e). KM met screen wireframe for tracking app [Screenshot]. Sketch. <https://www.sketch.com>.
7. Coutts, A. J., & Duffield, R. (2010). Validity and reliability of GPS devices for measuring movement demands of team sports. *Journal of Science and Medicine in Sport, 13*(1), 133-135. <https://doi.org/10.1016/j.jsams.2008.09.009>.
8. Dellaserra, C. L., Gao, Y., & Ransdell, L. (2014). Use of integrated technology in team sports: A review of opportunities, challenges, and future directions for athletes. *Journal of Strength and Conditioning Research, 28*(2), 556-573.
9. Kerzner, H. (2017). *Project management: A systems approach to planning, scheduling, and controlling*. John Wiley & Sons.

10. Malone, J. J., Lovell, R., Varley, M. C., Coutts, A. J., & Caffrey, S. (2017). Unpacking the black box: Applications and considerations for using GPS devices in sport. *International Journal of Sports Physiology and Performance, 12*(S2), S2-18-S2-26.
11. Rampinini, E., Coutts, A. J., Castagna, C., Sassi, R., & Impellizzeri, F. M. (2007). Variation in top-level soccer match performance. *International Journal of Sports Medicine, 28*(12), 1018-1024. <https://doi.org/10.1055/s-2007-965158>.