Development of a GNSS Sports Tracking Application for Android Systems

Fábio Ramos
fabio.ramos@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

June 2023

Abstract

This project consists in the creation of an athletics mobile application, which will have different functionalities, where the main one will be the continuous location tracking during the race through the GNSS systems. Currently, there are already many applications of this kind, but there is none specific for athletics focused on specific features such as the calculation of decathlon points, save personal bests, training plans, ”Shadow Running” allowing you to repeat a previously done path and be notified if it is slower or faster, news from the world of athletics, or even data from professional athletes, and, most importantly, track constrained positioning, avoiding errors in distance and pace, as well as counting the number of laps and time per lap. Therefore, this dissertation aims to improve the lives of athletes by joining several features in a single application.
Keywords: GNSS; GPS; Android; Athletics

1. Introduction

The project ”Development of a GNSS Sports Tracking Application for Android Systems” is the result of a dissertation in Electrical and Computer Engineering at Instituto Superior Técnico. The purpose of the solution is to create a mobile application that incorporates unique features for athletes beyond the traditional tracking applications.

This includes incorporating athletics news to keep users updated on the latest events, combined events points calculator using official formulas available online by the IAAF to help athletes calculate their score in combined events such as the heptathlon and decathlon [1]. A database to save personal records of all official events. A pace calculator. A database to check professional athletes statistics by country, event and gender. Users can choose to receive sound notifications when certain thresholds (distance or time) are reached. ”Shadow Running” to rerun a previously run circuit against your ”shadow”. Training plans by event is also available on the application. Chat feature so that users can find other runners nearby and schedule a run. Ability to choose a favourite personal trainer. And last but not least, a section where users can make suggestions, donate and report bugs.

The app uses GNSS (Global Navigation Satellite System) technology to provide accurate and reliable location data, enabling athletes to track their performance in real time. The user interface is designed to be simple and easy to use.

The application was developed using Android Studio, with the help of various libraries and APIs. The testing phase involved a series of tests to ensure the application functions correctly, and user feedback was incorporated to make improvements.

Overall, the ”Development of a GNSS Sports Tracking Application for Android Systems” project has resulted in the creation of a unique mobile application that provides athletes with a range of features to help them improve their daily lives. The application represents a significant contribution to the field of sports technology.

2. State of art

2.1. The Android operating system’s history and evolution

Android is a popular open-source operating system designed for mobile devices, such as smartphones, tablets, and smartwatches. Originally founded in 2003 by Android Inc., the intent was to create smarter mobile devices that were more aware of their owners’ location and preferences [2]. In 2005, Google acquired Android Inc. to enter the mobile device market [2]. Since then, Android has grown significantly and is now the market leader in smartphones, occupying 72% of the global market share in 2021 [3, 4].

The Android operating system has evolved significantly since its initial release. The system has undergone several upgrades and improvements, with multiple versions released to date [5]. As of November 2021, around 70% of Android devices on the market have version 9.0 or newer [5]. This indicates that Android users are quick to update to the latest versions or purchase new phones that have the latest operating system.

Smartphones have become ubiquitous devices worldwide, with an increasing number of users every year. According to statistics, 1.54 billion smartphones were sold worldwide in 2021, with 1.11 billion running on the Android operating system [3, 6]. This highlights the massive market potential for Android-based mobile applications.

2.2. Sports tracking applications

There are currently numerous sports applications, as for running, but there is always room for improvement or additional features, such as the computation of decathlon points. The most popular running applications present in the market are Strava, NikeRun, ASICS Runkeeper, Adidas Running, PUMATRAC, MapMyRun, Couch to 5K, Zombies Run!, Charity Miles, Weight Loss Running, and Garmin Connect.

Each application has its own unique characteristics. In this project, some of the advantages of each were merged, like the coaches from "Couch to 5k" and the news feed from "Adidas Running". However, one of the most important factors when developing an application with a similar idea is adding unique features, which is no exception in this project, where it was incorporated track&field-specific elements such as combined event points calculation. Other features that no other program offers include the ability to chat with other runners nearby, shadow running, and training plans. The second most important factor is improving what is already done. One of the most significant enhancements that was done is path correction while running on the track. Although Garmin watches already do this, in order to increase accuracy by stabilising the route, you must run some reconnaissance laps, whereas in this application it's only be necessary to click a button on the start line.

3. Tools and systems used in the application development

3.1. Global Navigation Satellite Systems

GNSS is a satellite network that provides location and time information to compatible devices, allowing them to calculate their position, velocity, and time regardless of weather conditions. Global and regional constellations and Satellite-Based Augmentation Systems are used to improve the accuracy, reliability, and availability of the navigation system. The four currently known global constellations are GPS (Global Positioning System), GLONASS (Global Navigation Satellite Systems), Galileo, and BDS (BeiDou Navigation Satellite System), each with unique characteristics. Galileo is more accurate than GPS, but it is not yet 100% complete. GPS is more dependable in urban areas, while Galileo provides better coverage at higher latitudes [7]. Each system has its strengths and weaknesses depending on the specific use case and environment.

3.2. Android Studio

Android Studio is an IDE (Integrated Development Environment) for mobile application development supported by Google [8]. The Android operating system consists of four layers, namely: the applications layer, the application framework layer, the libraries layer, and the Linux kernel layer [9].

The applications layer contains the pre-installed Java-based programs, such as email clients, SMS sending and receiving programs, calendars, Google Maps, and browsers, among others [10].

The application framework layer gives access to the APIs (Application Programming Interfaces) that allow the developer to access specific hardware functions of the device on which Android runs, such as GPS, camera, and location information, among others. The location manager, for instance, is important in the development of sports tracking applications to determine the user's location.

The libraries layer contains libraries written in C/C++, which are exposed to the programmer via the application framework [11]. The Android runtime contains the Java libraries that execute all Android applications, and it also includes the DVM (Dalvik Virtual Machine), which was developed to achieve minimal memory consumption [12]. Some of the libraries used include audio libraries for audio notifications and SQLite to save running data and personal records.

The Linux kernel layer handles critical functions, such as memory management, process management, and security. It contains the controller model for interacting with hardware, such as Wi-Fi, audio, energy management, and the camera, among others. It is also responsible for monitoring the several critical phases of starting blocks.

3.3. Java Programming Language

Java is one of the programming languages you can use for coding on Android Studio. In 2019 Kotlin has replaced Java as Google's preferred language for developing on Android [13]. Both are object-oriented programming languages, which is one of the main reasons they are used for android development [14]. The other main reason for Java being chosen for Android development is that it includes a large set of libraries that are easy to use and help in the creation of Android applications in a timely and effective manner [14]. Android is designed to work on a variety of hardware platforms, and this language includes platform-independent features [14]. The final major factor is its performance, which has a lower probability of crashing issues [14]. Java was chosen over Kotlin since it is a simple language but at the same time robust.

3.4. SQL Markup Language

SQL (Structured query language) is a programming language designed for working with relational databases. IBM developed this language to serve as an interface to the (Database Management Systems) [15]. It is a declarative language, which implies that the programmer specifies simply the intended goal and not how to achieve it [16]. An instruction passed to the DBMS is known as a "query" [17].

In this case, SQLite is the DBMS that will be used since it is the one integrated in Android Studio, and the initial idea was to store running data and personal records on the user's device because using a server database would require payment or compliance with privacy policies[18, 19].

3.5. XML Language

XML (Extensible Markup Language) is a markup language, which corresponds to a set of signals and codes applied to text or data to define its configuration [20]. XML is used in Android Studio as a bridge between humans and computers to define application component attributes such as size, style, and position in the layout, among others [21]. In this project, this language was used to design the user interface of the application.

3.6. Google Maps API

As previously stated, an API is a set of instructions that may be applied in a software developed by any user. When using an API, it abstracts away the implementation complexities of certain features and uses only the end result [22, 23, 24].

Google Maps makes use of GPS, GLONASS, and BDS, among other systems. It uses whichever GNSS receiver the phone has on board, and most of the time it uses at least two of them simultaneously[25]. Most smartphone manufacturers have included the ability to use multi-GNSS on their devices since 2016 [26].

In this work, the Google Maps API was used to build the maps that are embedded in the application. This API is a free service and allows not only representing maps, but also obtaining routes between points or even drawing shapes to represent areas on maps [27].

3.7. Cloud Firestore

Cloud Firestore is an online NoSQL real-time database provided by Firebase, which has a collection of many other services, all of them has a free tier pricing plan [28, 29, 30, 31]. This service is required to be possible to have the chat feature.

4. Architecture of the proposed solution

4.1. Framework for GNSS tracking

To accomplish the path correction feature, when the user decides to go for a run while on the running feature screen, he has the option to select if he is currently on an athletics track and which lane he will be running.

After that, he can run on the track without worrying about erroneous point locations, for example points obtained that aren't in the track where the user is running, since they will be corrected by identifying the most likely location of the user with an algorithm that will be discussed later in this chapter.

To obtain the user's location, as previously explained, the Google API, more specifically the Android Fused Location Provider API, which provides latitude, longitude, and altitude, will be used, and these coordinates will be converted to Cartesian Coordinates which are more suitable for distance and pace calculations. Finally a line is drawn on a map so that the user knows the path he took.

The images 1, 3, 7 and 8 depicts the general outline of the proposed architecture.

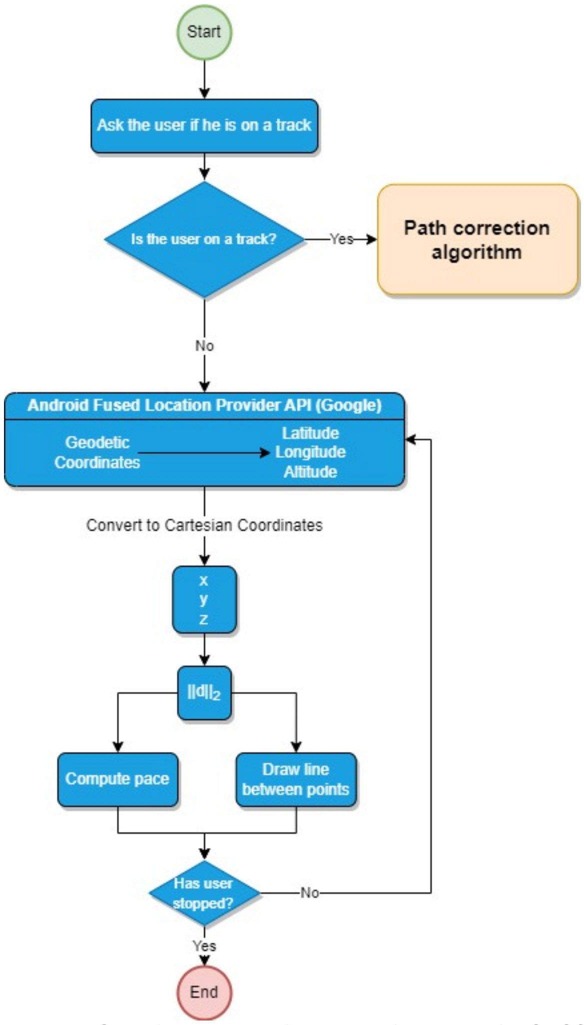

Figure 1: Simplified version of proposed framework for GNSS tracking

4.2. Geodetic to Cartesian coordinates conversion

It is necessary to convert the geodetic coordinates returned by the API into Cartesian coordinates in order to compute the distance and pace using the elapsed time.

In Figure 2 it's represented the ellipsoidal model of the Earth. The value of the semi-axes varies according to the geodetic datum considered (WGS-84 - used by GPS, PZ-90 - used by GLONASS, GRS-80 - used by BDS) [32].

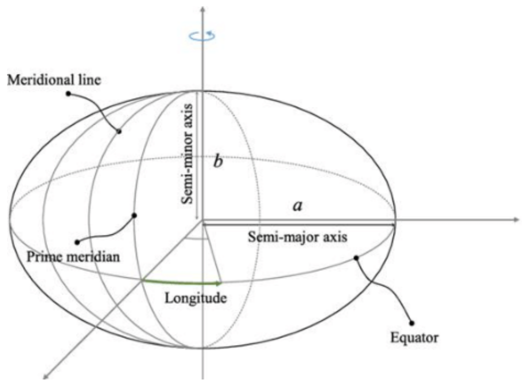

Figure 2: Ellipsoidal model of the Earth

Equation 1 shows the relation between the flattening factor and the semi-axes.

$$f = \frac{a - b}{a} \quad (1)$$

To properly classify an ellipsoid, two parameters are required, typically the semi-major axis (a) and the flattening factor (f), which can differ between different models and result in variations in the semi-axes and other characteristics. The several values for the reciprocal of flattening for each model are represented in Table 1.

Table 1: Inverse flattening of the most commonly used ellipsoids [33, 32]

Reference Ellipsoid Name	Reciprocal of Flattening 1/f
WGS-84 by GPS	298.257223563
PZ-90 by GLONASS	298.257839303
GRS-80 by BDS	298.257222101

The equations for converting to Cartesian coordinates are presented in system of equations 2[34], where ϕ is latitude, λ is longitude, h is height (above the ellipsoid), and R_N is the radius of curvature in the prime vertical, as given by equation 3.

$$\begin{cases} x = (R_N + h)\cos(\phi)\cos(\lambda) \\ y = (R_N + h)\cos(\phi)\sin(\lambda) \\ z = \left((1 - f)^2 R_N + h\right)\sin(\phi) \end{cases} \quad (2)$$

$$R_N = \frac{a}{\sqrt{(1 - f(2 - f)(\sin(\phi)^2)}} \quad (3)$$

4.3. Framework for path correction

Figure 3 shows a simplified algorithm of how the path correction when running on a track was implemented. The best approach suggested to replace the erroneous points with points from the track is to model the track by mathematical equations and check which point from the track is closest to the point retrieved from the Android Fused Location Provider API.

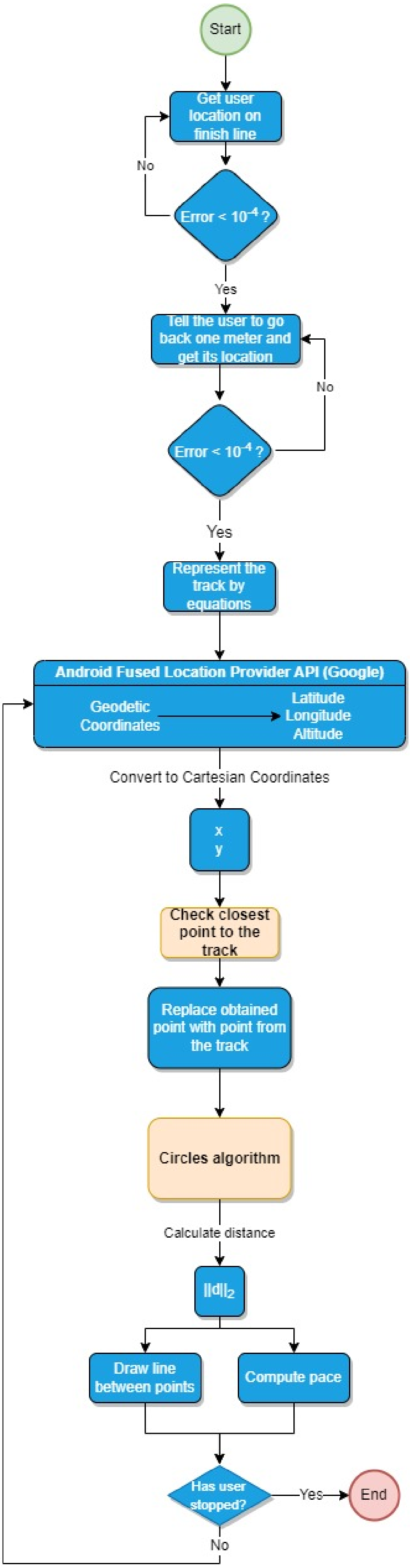

Figure 3: Simplified version of proposed framework for path correction

Representing the track mathematically requires the use of two points in order to depict a straight line in the Cartesian plane and after that knowing the official track measurements it is possible to obtain the other straight line and circles as well [35, 36]. Since these equations must be accurate, the solution found was to request that the user place two markers on the track, one at the finish line and one behind it, as shown in Figure 4.

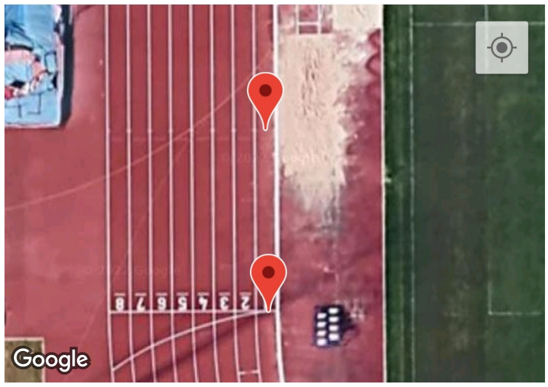

Figure 4: Example of how user must place markers on the track

4.4. ECEF to ENU coordinate system

To mathematically depict the track, it is necessary to convert from geodetic to Cartesian coordinates, and then from ECEF (Earth-centered, Earth-fixed coordinate system) to ENU (East-North-Up) coordinate system. However, for this purpose it's only going to be used East and North coordinates instead of East, North and Up, because there are no altitude variations on a track and it's easier to represent the track in 2 dimensions. Figure 5 shows a model of the relation between local ENU coordinates and ECEF coordinates.

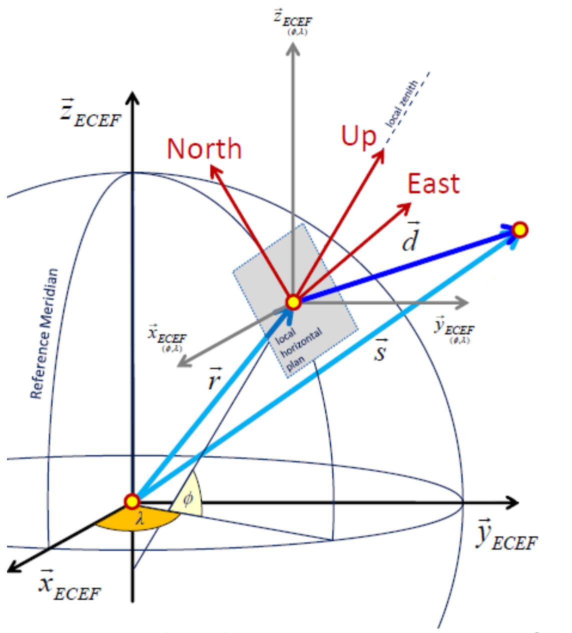

Figure 5: Model of transformations between ENU and ECEF coordinates.

Equations of how to convert from ECEF (X, Y, Z) to ENU and from ENU to ECEF are shown below [37], where ϕ_0 denotes the latitude of the origin of the local reference system, in this case it will be the track finish line and λ_0 denotes the longitude of the origin as well. $\theta_1 = \lambda_0 + \pi/2$, $\theta_2 = \pi/2 \quad \phi_0$

$$[e\ n\ u]_{enu} = [x\ y\ z]_{ecef}\ \ Rot_z(\theta_1)\ \ Rot_x(\theta_2) \quad (4)$$

$$[x\ y\ z]_{ecef} = [e\ n\ u]_{enu}\ Rot_x(\quad \theta_2)\ Rot_z(\quad \theta_1) \quad (5)$$

Rot_x and Rot_z are rotation matrices over the X-axis and over the Z-axis, respectively [38].

$$Rot_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\alpha) & sin(\alpha) \\ 0 & sin(\alpha) & cos(\alpha) \end{bmatrix} \quad (6)$$

$$Rot_z(\gamma) = \begin{bmatrix} cos(\gamma) & sin(\gamma) & 0 \\ sin(\gamma) & cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (7)$$

4.5. Check closest point to the track

Figure 6 depicts the model of the athletics track. As seen in the image, the lines are infinite, and the curved section of the track is represented by circles. The simplest solution found to check the closest point was to depict two lines perpendicular to the straights of the track where the curves start, and if the provided point from the Google API is between those two perpendicular lines (represented by orange points in the figure), it is replaced by the closest point from the straights of the track, and if not (represented by green points in the figure), it is replaced by the closest point from the semicircles of the track.

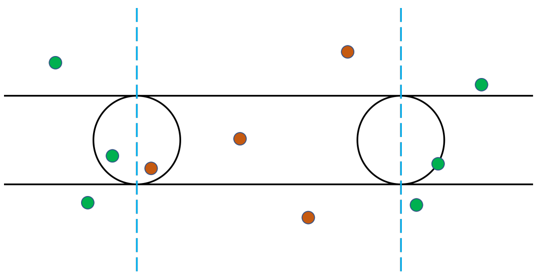

Figure 6: Athletics track represented by mathematically figures

The algorithm explained above is shown in Figure 7. This algorithm is essential because, for example, the leftmost orange point in Figure 6, it is closest to the circle, but it should be relocated to a point along the straights. It will also be described in the following chapter how to determine the closest point to the straight lines and circles.

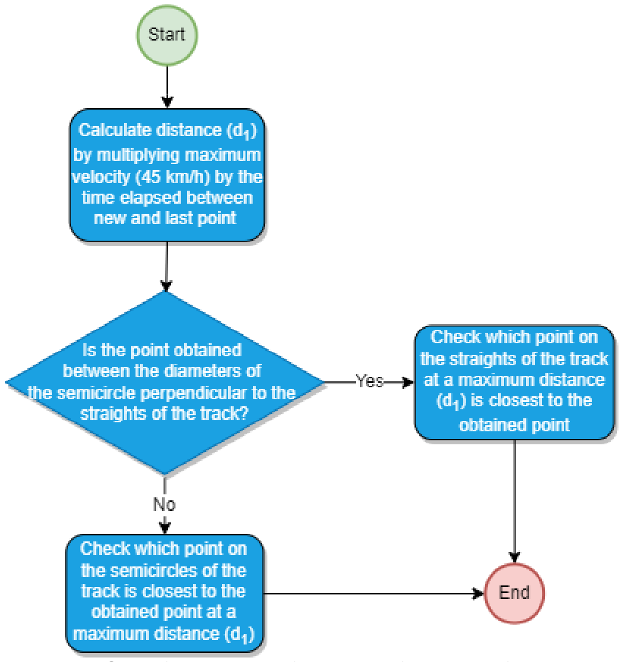

Figure 7: Simplified version of proposed framework for checking the closest point from the track

It is necessary to prevent GNSS from failing and providing impossible locations, such as behind the last obtained point (since the athlete must simply run ahead) or too far ahead of the last obtained point. To accomplish this the maximum distance that the athlete could have run since the last obtained point will be calculated by multiplying the time elapsed between the last obtained point and the new one by a maximum velocity of 45 km/h, which is equal to 12.5 m/s, which would be an impossible speed to achieve since Usain Bolt himself reached 44 km/h in a 100 meters run [39].

Figure 8 shows circles algorithm presented on algorithm from Figure 3. If the obtained point is in one line from the track and the last point was in the other line it means that GNSS failed and didn't detect any point on the semicircle and so 30 points evenly spaced apart will be added. While running on the semi-circles, 10 points will also be added between each new point and the last one to prevent the points from not looking like a curve when connected.

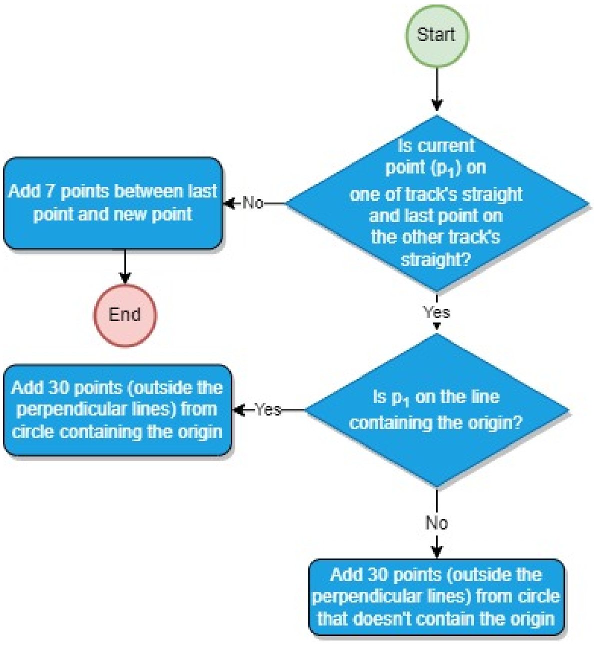

Figure 8: Proposed framework for circles algorithm presented in Figure 3

As seen in Figure 9 in the case where the last point is in one straight line and the new one in the other line and considering the worst case scenario (user running in the 8th track), adding 30 points manages to have the precision intended since connecting all points in straight lines looks like a curve and the whole lines are inside the track. When considering running in the 8th track, semi-circles' length of 142.44 m, each point is spaced apart by 4.748 m.

Figure 9: The track's semi-circle divided into 30 evenly spaced points

Every time a new point in the curves is obtained, 10 points are added between the new point and the previous one. As seen in Figure 10, when considering the worst case scenario, where the user is running in the 8th track and GNSS only detects two points in the entire curve, one at the beginning and the other at the end, it is possible to see that when connecting all of the 12 points, all lines are still inside the correspondent track and it looks like a curve, securing the minimum precision desired. In this illustrative scenario, each point is approximately 11.9 m apart.

Figure 10: The track's semi-circle divided into 10 evenly spaced points

The equations below compute the arc lengths between points on a circle, which are required to compute the travelled distance on the semi-circles [40].

$$\theta = cos^{-1}\left(1 \quad \frac{d^2}{2r^2}\right) \ rad \tag{8}$$

$$arc\ length = r\theta \tag{9}$$

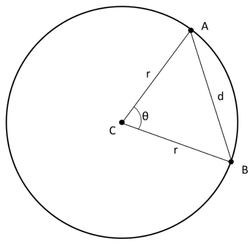

Figure 11: Calculating the arc length of a circle segment

As shown in Figure 12, if tracking fails, the path might look like the one in blue, and the distance and pace would be inaccurate. The goal is to make it look like the white line.

Figure 12: Example of a path with and without correction on a track

5. Implementation details

5.1. Representing the track mathematically

As previously stated, the track is modelled by two lines, two perpendicular lines to those lines, and two circles. Figure 13 shows the flowchart for mathematically modeling the track, where the track orientation is known by the slope of the line produced by the two pins inserted by the user.

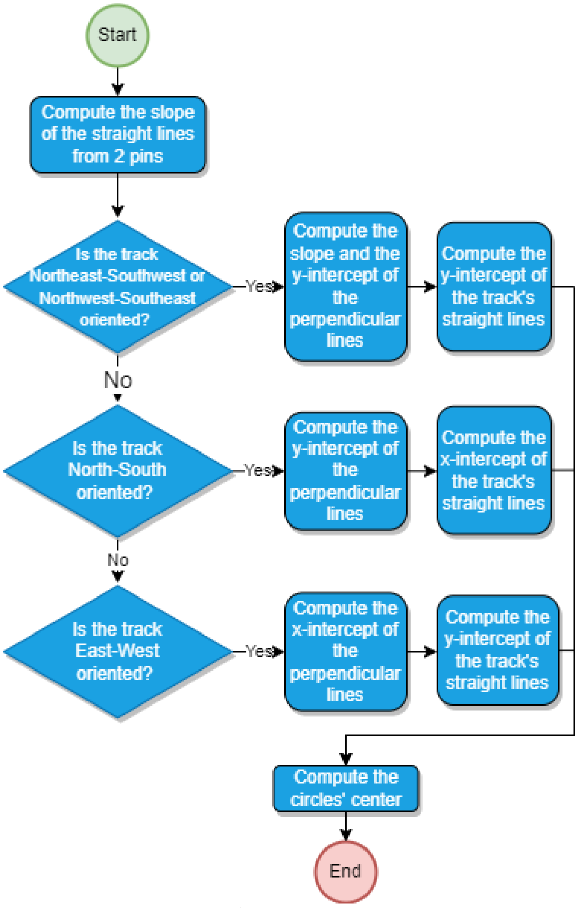

Figure 13: Flowchart of modelling the track mathematically

5.2. Implemented features

This section will go into more detail about the features implemented.

5.2.1 Home page and menu

The application has a home page and a menu where the user can navigate through the main features.

5.2.2 News feed

The user can use this feature to stay up to date on the latest athletics news. This was accomplished using the RSS (Really Simple Syndication) feed, which uses an XML (Extensible Markup Language) file with the information (title, synopsis, and date) to automatically update news in real time from the official World of Athletics website[41, 42]. If the user clicks on the XML file, it opens the website with the entirely updated information.

5.2.3 Training plans

With the help of this tool, users can follow detailed training routines to improve their 5 km, 10 km, or half-marathon times. Any link clicked by the user gives access to a PDF that the application can read on its own by utilising a library that is available on GitHub [43].

5.2.4 Combined events points calculation

With this feature, athletes will no longer have to compute points during combined events competitions by themselves. Having the event mark the application allows them to see the points earned instantly, where the application applies official IAAF (International Amateur Athletic Federation) formulas [1].

5.2.5 Pace calculator

This feature allows the user to calculate a pace, time or distance by inserting the two variables that are known in order to determine the third unknown. The formula to compute the pace is the inverse of speed since the units come in minutes per kilometre or minutes per mile.

5.2.6 Personal bests database

Saving jumps, throws and any run distance marks and having all of them in one place where they can be sorted by name and date and searched by name is not a problem anymore. This application also has the ability for the user to insert his records and it saves them in a local database, in other words on the phone.

5.2.7 Professional Athletes Statistics

This feature is a way to get inspired by professional athletes' results in every official track and field event by gender. These are databases that come with the application and they must be manually updated and then release new versions.

5.2.8 Chat

Finding athletes nearby to run with is not a problem anymore. This feature makes it easy to search for other athletes who are also using the application sorted by distance. Users can then chat with each other using the application's chat tool to arrange a time to run together or to simply get training tips to get better faster.

To have this functionality available in the application it was already required to have an online database. The database used was Cloud Firestore since it is well-known, free and easy to implement it on the application.

In order to have user accounts and save the chats on the database, it was required to also develop a sign-up and login page.

5.2.9 Running

This is the main feature of this project. Allowing the user to use the phone to track his runs, either on the road or on track, without worrying about wrong data when running on a track.

5.2.10 Shadow running

With this tool, which can be accessed before start running by clicking on the icon with a ghost, the user can pick his own pace and the intervals (distance or time) at which he wishes to hear an audio alert if things are going well or poorly. This functionality may also be accessed by clicking the same icon, but on the past runs menu, where you will no longer be able to pick the shadow's pace once you are competing against your shadow from a previous run.

5.2.11 Settings

On settings tool the user can insert his body information for calories calculation while running, select a coach to join him in his runs, change the application language and enable or disable the music feature on the app.

5.2.12 Report bugs

A report bugs option was created so that users could either report bugs they noticed or offer suggestions to make the application better. When users hit the "send" button, the message is sent using the phone's embedded mail app.

6. Evaluation

6.1. Test run on the road

A test run was performed using this application and a Garmin watch was also used in tandem, yielding only minor differences in altitude and calories.

6.2. Test run on the track

Figure 14 represents a run done on a track.

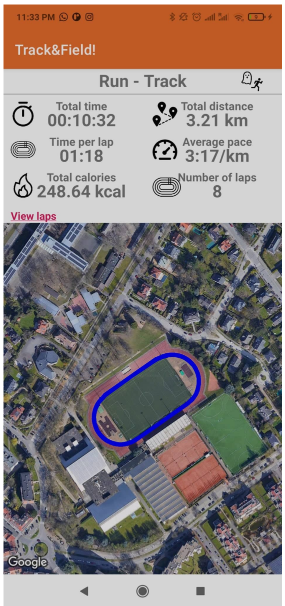

Figure 14: Run test on the track print screen

Figure 15 depicts the same race, but using the Garmin watch. It is possible to see that it has minor flaws, flaws that did not exist with the use of this thesis' application.

Figure 15: Run test on the track using Garmin watch

The distance with the Garmin watch was of 3.35 km, which results on an error of 4.36% using equation 10:

$$Error(\%) = \frac{|Measured_{Value} - True_{Value}|}{|True_{Value}|} \times 100 \quad (10)$$

this error is less than 5% which is not significant. Nevertheless, when on a longer race with more laps, it could potentially lead to a more substantial disparity.

7. Conclusions

This thesis focused on creating a unique application to improve athletes' daily routines. A custom algorithm was used in the implementation to accurately depict and correct the user's path on the track. The application's performance was validated by rigorous testing involving athletes from various clubs. Although it was decided not to release the app on Google Play due to potential limitations, alternative distribution channels such as Google Drive were deemed to be more convenient [44].

References

[1] Bojidar Spiriev and Attila Spiriev. *IAAF scoring tables of athletics*. International Association of Athletics Federations Monaco, 2011.

[2] Iggy Krajci and Darren Cummings. History and evolution of the android os. In *Android on x86*, page 1. Springer, 2013.

[3] Android statistics (2022) - business of apps. Accessed on Mar. 21, 2022.

[4] Mobile operating system market share worldwide — statcounter global stats. Accessed on Mar. 21, 2022.

[5] Distribution data for android. Accessed on Mar. 23, 2022.

[6] • smartphone sales worldwide — statista. Accessed on Mar. 21, 2022.

[7] Gps vs. glonass vs. galileo: What's the best gnss? expert world travel. Accessed on Mar. 18, 2022.

[8] Desenvolvedores android — android developers. Accessed on Mar. 24, 2022.

[9] Platform architecture — android developers. Accessed on Apr. 8, 2022.

[10] Parmjit Kaur and Sumit Sharma. Google android a mobile platform: A review. pages 1–5, 03 2014.

[11] Chao Wang, Wei Duan, Jianzhang Ma, and Chenhui Wang. The research of android system architecture and application programming. In *Proceedings of 2011 International Conference on Computer Science and Network Technology*, volume 2, pages 785–790, Dec 2011.

[12] David Ehringer. The dalvik virtual machine architecture. *Techn. report (March 2010)*, 4(8):72, 2010.

[13] Kotlin is now google's preferred language for android app development — techcrunch. Accessed on Mar. 24, 2022.

[14] 8 top advantages of choosing java app development in 2021. Accessed on Apr. 9, 2022.

[15] Donald D. Chamberlin. Early history of sql. *IEEE Annals of the History of Computing*, 34(4):78–82, 2012.

[16] What is declarative programming? - stack overflow. Accessed on Apr. 11, 2022.

[17] What is a query? database query explained. Accessed on Apr. 11, 2022.

[18] Types of databases - rad studio. Accessed on Apr. 11, 2022.

[19] Cloud vs local servers: Weighing up the pros and cons - evolvit. Accessed on Apr. 11, 2022.

[20] Tim Bray, Jean Paoli, C Michael Sperberg-McQueen, Eve Maler, François Yergeau, and John Cowan. Extensible markup language (xml) 1.0, 2000.

[21] Layouts — android developers. Accessed on Apr. 11, 2022.

[22] Joshua Ofoeda, Richard Boateng, and John Effah. Application programming interface (api) research: A review of the past to inform the future. *https://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/IJEIS.2019070105*, 15:76–95, 1 1. Accessed on Mar. 23, 2022.

[23] Application programming interface — computerworld. Accessed on Mar. 23, 2022.

[24] Application program interface from foldoc. Accessed on Mar. 23, 2022.

[25] What is glonass and how it is different from gps. Accessed on Mar. 23, 2022.

[26] Marco Fortunato, Joshua Critchley-Marrows, Malgorzata Siutkowska, Maria Loredana Ivanovici, Elisa Benedetti, and William Roberts. Enabling high accuracy dynamic applications in urban environments using ppp and rtk on android multi-frequency and multi-gnss smartphones. *European Navigation Conference, ENC 2019*, 5 2019.

[27] Plataforma google maps — google developers. Accessed on Mar. 23, 2022.

[28] Laurence Moroney. *Firebase Cloud Messaging*, pages 163–188. Apress, Berkeley, CA, 2017.

[29] Firebase — firebase.google.com. https://firebase.google.com/. [Accessed 23-Sep-2022].

[30] Usage and billing dashboard - Firebase Help — support.google.com. https://support.google.com/firebase/answer/9628313?hl=en. [Accessed 23-Sep-2022].

[31] Firestore documentation — Google Cloud — cloud.google.com. https://cloud.google.com/firestore/docs. [Accessed 23-Sep-2022].

[32] Volker Janssen. Understanding coordinate reference systems, datums and transformations. *International Journal of Geoinformatics*, 5, 01 2009.

[33] Pz-90. Accessed on Mar. 29, 2022.

[34] Ellipsoidal and cartesian coordinates conversion - navipedia. Accessed on Mar. 29, 2022.

[35] Running track geometry. Accessed on May. 23, 2022.

[36] Iaaf track and field facilities manual. Accessed on May. 23, 2022.

[37] Transformations between ecef and enu coordinates - navipedia. Accessed on May. 22, 2022.

[38] Rotation matrix – from wolfram mathworld. Accessed on May. 23, 2022.

[39] How fast is the world's fastest human? — britannica. Accessed on May. 23, 2022.

[40] Calculating the arc length of a circle segment - mathematics stack exchange. Accessed on May. 23, 2022.

[41] Davide. How do rss feeds work?, Jul 2022. Accessed on Sep. 20, 2022.

[42] Iaaf launches new rss feed service for subscribers and syndication to other websites: News: World athletics. Accessed on Sep. 20, 2022.

[43] GitHub - barteksc/AndroidPdfViewer: Android view for displaying PDFs rendered with PdfiumAndroid — github.com. https://github.com/barteksc/AndroidPdfViewer. [Accessed 20-Sep-2022].

[44] Link to download the app through google drive.