Problem

In the world of running and athletics many amateur athletes do not have access to means of easily recording their trainings and records, and even access to a coach. My aim is to design an android application (using android studio) that allows amateur athletes to record their trainings and track their progress more easily than using a training booklet. I also aim to implement a training generator that acts as a coach and generates training programmes based on the athletes needs and ability.

Problem	Why
Recording trainings in a training booklet	Tracking progress will be more difficult flicking through a book than searching through a list of activities. Statistical analysis will be far easier if data is recorded on a database.
Real-life coaches giving training programmes	Coaches may not be available for athletes sometimes or even all the time. Coaches may not always be giving the most relevant and useful trainings tailored to the athlete as they usually deal with a group of athletes.
Remembering records for certain distances	Simply remembering records for various distances an athlete has run can be difficult. Saving records to a database allows athletes to more easily remember their records and use them for statistical analysis, for perhaps predicting records of other distances.
Recording trainings in detail	A few apps exist for recording trainings of athletes. However, they do not allow the athlete to record their trainings in detail and only allow them to record a single distance and time. Any statistical analysis of data recorded is also impossible without payment. For middle distance athletes and sprinters who consistently run interval training sessions, recording all the details is impossible.

Figure 1 - Problem Table

Users want a platform that allows them to record trainings and other activities (races) in detail, keeping all the data in one place and allowing it to be used for statistical analysis so the athlete can track their progress. They also want a 'training generator' that gives relevant and useful trainings tailored to themselves, acting like a coach. Finally, they want a way of recording their personal bests and using them to make predictions for performances of other distances.

Users

User	Role in current system	Role in new system
Coach	Gives training to athletes for completion	No role in new system. 'Training generator' acts as a coach.
Athlete	Completes trainings given by coach and participates in races to win or achieve personal records. Typically, records trainings, races and personal records to a training booklet or other system.	Records trainings, races and personal bests in detail to new system's database. Makes use of the system for statistical analysis, predictions and to receive trainings to complete.

Figure 2 - User Table

Context

Training Program – Stored as a training in the database, has a name, date, description, type and difficulty associated with it. A training program can have multiple training sets associated with it (distances run).

Training Set – A training set has a distance, time (or speed), number of repetitions and rest associated with it. The distance is the distance the athlete must run, the time or speed is how fast the athlete must run the distance. The number of repetitions is how many times the athlete must run the distance. Finally, the rest is how much rest the athlete has in-between running each distance. In a training program it is assumed that the athlete has 5 minutes rest in-between each training set to recover their heart rate to normal.

3x400m Time: 1:12, Rest: 1:30, 8x1000m Time 3:40, Rest 1:00.

A training program has this standard description describing each of the training sets in detail with the number of repetitions followed by distance, time and then the rest (time is given in the standard chronometer form).

12000m Pace 4:20/km.

A training set with only one repetition is described as shown. A distance followed by pace per km (in standard chronometer form). The end of a training program description is indicated by a full stop.

Interval Training – A training program that consists of multiple repetitions of a certain distance with intervals or rest in between them. Typically, an interval training program consists of multiple training sets with rests in between them as well. As oppose to a continuous training program where the athlete never stops running.

Active Training – A training program that has not yet been completed either inputted manually or assigned by the training generator. These do not have a difficulty associated with them as they have not been completed.

Race – An object in the database that stores the details about a race the athlete has participated in, name, date, distance, performance. The description of a race is generated as followed.

1500m Performance: 4:12.68

Personal Best – An object stored in the database similar to a race however it is the fastest instance of a race or performance of a certain distance. Does not necessarily have a race associated with it but often does. The description of a personal best is generated the same as a race. However, it has a prediction associated with it that will be generated within the application.

Research

Interview

Interview with Barry Allen a 1500m middle distance runner.

Q1: What are you looking for in a new application for amateur athletes?

A1: Being a middle-distance runner I tend to do a lot of interval training on the track in the summer. I find it hard to record all the details of my trainings unless I use a training booklet, because most systems out there do not allow you to add multiple distances and speeds for a particular training which I find bizarre. If I were to use an application to log all my trainings, it would have to have the ability to record all the fine details and how hard I felt the training was, to be worth while using.

Q2: Would you find a system that generates training for you useful? And do you think you would use it on a regular basis, bearing in mind you would have to record as much data as possible for it to be fully optimised?

A2: I do find it as quite a cool thing to have, but I don’t know that I would use it that much. I do have a personal coach and I do see this idea being useful for athletes that don’t have access to a personal coach or even a coach. But I would probably rarely use it. Most likely I would use it for getting interval track sessions when my coach is not around or hasn't given me anything to do. It would really have to learn the athlete's habits and distances they like running quickly to be useful for anything other than a track session.

Q3: Do you think the idea of a personal best predictor sounds useful?

A3: I don't think it would be particularly useful as such, but I think it would be a real unique addition to the system. I know a lot of athletes who are really good at some distances and bad at others, and some athletes that are good across the board. I think it would be really interesting to see what those athletes who aren't as consistent get predicted. Even for myself I would find it quite interesting to see what potential a computer thinks I have.

Q4: Do you think it is important that a system like this makes information readily available to the user? And what information would you recommend making readily available?

A4: Yes, I think the whole point of a system like this is for it to analyse data and make information and correlations readily available to the user. I know for sure something worth making available is how far people run and for how long. Athletes are constantly boasting to each other how far they run on a weekly basis, how many trainings they do and so on. I think it would be important to show how much running the athlete has done in the past 7 days, making that available. And it would be interesting also to see information on how far an athlete runs on a weekly basis. I suppose data like this could also be linked to the primary race distance of the athlete and interesting relationships could be made. But primarily for the athlete they want to see how far they run and how many trainings they do.

Modelling Research

Quantifying the difficulty of a training for an athlete will allow me to make my 'training generator' as optimised as possible. Instead of making loose inferences from feedback of previous trainings quantifying and modelling an athlete completing a training is the best approach. In 2 journals and 1 article I discovered the key to quantifying the difficulty of exercise in general was not modelling heart rate or breathing rate, but was in fact modelling lactate accumulation in the blood. Another website I discovered dedicated to measuring blood lactate levels for optimized training programmes and routines, outlined how to model the accumulation and dissipation of blood lactate throughout a training program.

Firstly, the article, "Use of Blood Lactate Measurements for Prediction of Exercise Performance and for Control of Training"[1] outlined some key values, their definitions and typical values they can take.
Blood Lactate – amount of lactate present in the blood per liter. Takes values from 1-15 mol/l.
Baseline Lactate – amount of lactate in the blood before the athletes starts exercise. Typically, 1-1.5 mol/l.
VO2Max – refers to the maximal oxygen uptake, or the maximum amount of oxygen and body can use over a given period per kilogram of the body. Typically takes values from 20 – 85 ml/l/kg.
Lactate Threshold – refers to the intensity of exercise which results in an accelerated production of lactate in the blood. Takes values from 3.5 – 5 mol/l.
Anaerobic Threshold – refers to the metabolic point at which the rate of lactate production is equal to the rate of lactate dissipation within the blood. At an exercise intensity above this threshold the amount of blood lactate increases gradually or in some cases rapidly but in a linear manner. Typically takes values from 2-7 mol/l blood lactate being produced.

---

[1] L. Véronique Billat, "Use Of Blood Lactate Measurements For Prediction Of Exercise Performance And For Control Of Training", *Sports Medicine* 22, no. 3 (1996): 157-175, accessed September 16, 2017, https://www.researchgate.net/publication/14324118_Use_of_Blood_Lactate_Measurements_for_Prediction_of_Ex

Maximal Steady State – refers to the intensity of exercise which results in the highest possible steady amount of blood lactate. Thus, being the intensity of exercise resulting in blood lactate equal to the anaerobic threshold.

On the website lactate.com “Lactate Testing and the Science of Running Performance”[2] multiple examples of exponential curves that model exercise intensity (speed in the context of running) against blood lactate level are given and available for analysis.

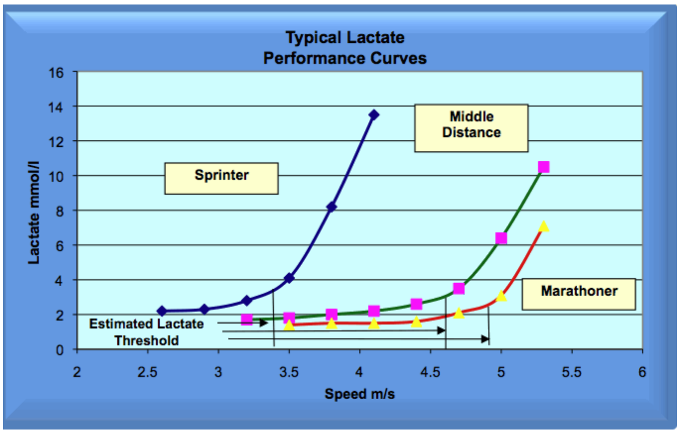


Figure 3 - Lactate Performance Curves

In the context of running, the estimated lactate threshold is the speed at which the body generates blood lactate that surpasses the body’s lactate threshold. This is also known as V4Speed and *Figure 3* shows that this is the speed at which the gradient of the curve is exactly 1 (scale distorts the gradient). Also outlined on the website, a marathon runner has a higher VO2Max meaning he can use more oxygen over a period of time and hence why he is able to run faster for longer. So, in terms of VO2Max Marathoner>Middle Distance>Sprinter. From *Figure 3* the gradient of the exponential curve or how quickly it ascends upwards is clearly correlated to the value of the athletes VO2Max in some way. This is because sprinters being more powerful make use of more of their anaerobic system when running at high speeds than marathon runners. Using this information, it is possible to model a lactate performance curve as an exponential model using the value of the athlete’s V4Speed, VO2Max and the standard baseline lactate level.

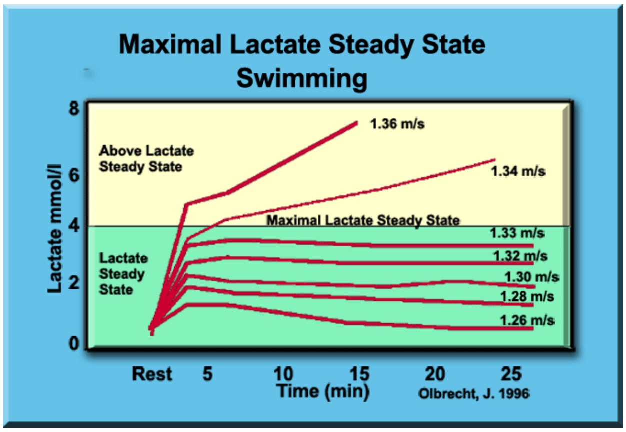


Figure 4 – Lactate Increase Over Time

Figure 4 is a diagram which helps us model the training program as a whole. Using the ideas from *Figure 3* an estimate of the expected blood lactate level the body reaches running at a particular speed can be achieved. *Figure 4* helps us understand how the lactate levels in the blood change against time. When we are exercising at a certain intensity that produces a level of lactate below the anaerobic threshold or maximal lactate steady state *Figure 4* clearly shows the lactate level remains the same no matter how long we are exercising for. This is because the body is removing lactate from the blood at the same rate it is are accumulating it. If the body is exercising at an intensity above the anaerobic threshold *Figure 4* indicates the lactate levels increase at a rate directly proportional to (expected lactate level – anaerobic threshold). Making note that the time it takes for the blood to reach the expected lactate level, is constant. It is now possible using this information to model an athlete’s blood lactate against time when they are running a distance at a particular speed.

[2] "Lactate, Science And Running". *Lactate.Com*. Last modified 2017. Accessed September 15, 2017. http://www.lactate.com/running/index.html.

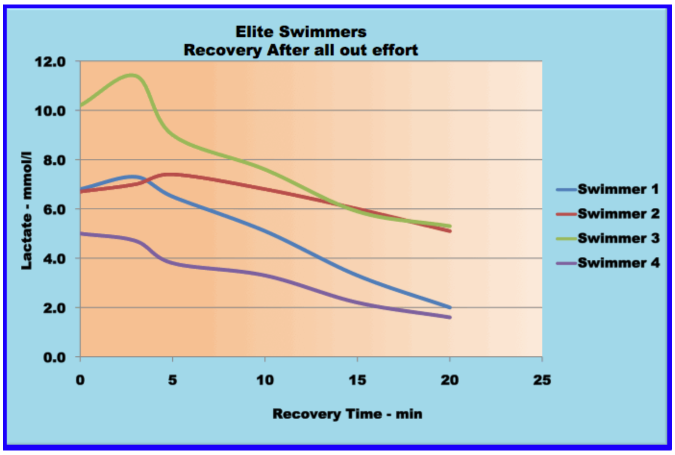


Figure 5 – Lactate Recovery Model

The final piece of the puzzle is the recovery period, and how blood lactate dissipates during rest. *Figure 5* shows the rate of dissipation of blood lactate is not the same for all athletes. So, the recovery rate must be variable and not constant for different athletes. *Figure 5* initially indicates an increase in blood lactate before dissipating. This increase is directly proportional to the blood lactate level after exercise. Using these ideas, it is possible to model the recovery period as an increase and then decrease in blood lactate given to us by the athlete's recovery rate.

To conclude, I will take away all of these ideas and will model each of these graphs to quantify the difficulty of a training program for a particular athlete. The calculation will take part in a separate thread from the main application as the time complexity of the algorithm could be significant. I will use the average blood lactate level of the whole training program as an indicator on the difficulty of the training program. To make these models individual to each athlete I will use 4 variables that will be stored in the database.

V4Speed – value of x (speed) at which the lactate curve has a gradient of 1.
VO2Max – value that is directly proportional to the gradient of the lactate curve (higher VO2Max less steep the lactate curve is).
Lactate Threshold – previously referred to as the anaerobic threshold, the blood lactate level at which the rate of production of blood lactate is equal to the rate of dissipation. Blood lactate values above this result in an increase of blood lactate over time.
Recovery Rate – rate at which blood lactate dissipates during a rest or recovery period.

I also included 4 constants that I will consider changing to optimize the function.

Constant k – constant that changes the effect of VO2Max on the gradient of the lactate curve.
Baseline Lactate – constant that refers to the amount of lactate in the blood before exercise.
Lactate Time – time it takes for the expected blood lactate level to be reached during exercise.
Constant z – constant that changes how rapidly blood lactate increase per second when exercising at a level above the lactate threshold.

Extraction of Data

Sourcing real life data as initial data values in my database, from a variety of different athletes will make sure my prediction models work and behave realistically. When a new user creates an account, initial value sin the database will help generate the new user initial realistic values based on their ability. This will allow the 'training generator' to work effectively before the user has even recorded any trainings. For my prediction models to work I only need data about the athlete's gender, primary race distance, personal bests and statistics (V4Speed, VO2Max etc.). I will use the website thepowerof10.info[3] to source real life personal bests of athletes. The website records all the personal bests and records of all British athletes (professional and amateur). This means, I will be able to get a wide range of data that will make my prediction models applicable to all types of athletes. I will not be able to get a massive amount of data because extracting the data from the website manually will

3 "Power Of 10", *Thepowerof10.Info*, last modified 2017, accessed October 23, 2017, http://www.thepowerof10.info/.

be tedious, so I will only select the best data using my judgment to decide which data is useful and which is not. I will also have to estimate the values (V4Speed, VO2Max, Lactate Threshold and Recovery Rate) for each of the athletes I include in my initial data. This should not be a problem if the values I choose are consistent with each other, because I can modify some of the constants in the training generator function to optimize or re-calibrate the function.

Best known performances

Event	PB	2017	2016	2015	2014	2013	2012	2011	2010	2009	2008	2007	2006
100	14.31/13.9						14.31/13.9						
200	29.67							29.67					
400	50.70i	50.70i				62.31							
800	1:51.76	1:51.76	1:54.94	2:00.46									
1500	4:49.46					4:49.46	4:53.46	5:11.67					
4K	14:35				14:35								
5K	17:28	17:28											

Figure 6 - The Power of 10 Dataset

Figure 6 an example of the data set that I can extract from the website; personal bests from a variety of different distances. Some of the personal bests are from a long time ago, so I won't include them because I only want the best data. I aim to include around 100 athletes, their personal bests and statistics in the database initially.

UI Research

To design my User Interface (UI) I evaluated the UI of a popular running app STRAVA. This is an outline of all the ideas I took from the UI of STRAVA and how I will implement these ideas into my application.

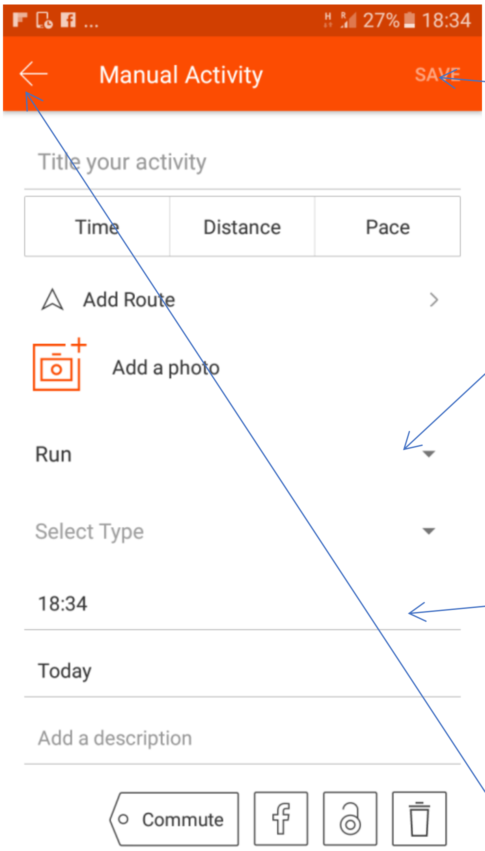


Figure 7 - STRAVA Manual Activity

'Save' button lights up white when all the fields have been filled. If data fields are invalid a text dialog pops up notifying user. This can be done with the *setAplha* method. A dialog can be created with *AlertDialog.Builder* class.

Spinner widget is used to select activity type and sport. This could be used to select a race distance or training type by loading in the string array from the resources directory.

Simple *EditTextView* widget is used for the date and time. However, when the view is focused on a pop-up dialog with a custom view allows the user to select a date or time. This event can be triggered by overriding the *onFocus* method in an instance of the *TextWatcher* class attached to the *EditTextView* widget. A custom view dialog can be created with the *AlertDialog.Builder* class and the *setView* Method.

The Back button can be created with the *setDisplayHomeUpAsEnabled* method and overriding the *onOptionsItemSelected* method.

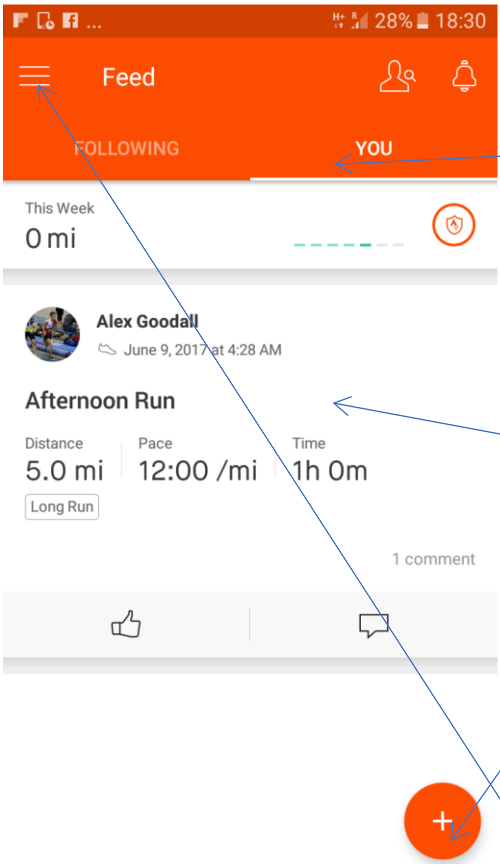


Figure 8 - STRAVA Feed

Tab-widget used to switch between 2 different fragment activities. This could be done using the deprecated way of hosting different child layouts for each tab (less complicated for simple activities). Or could be done by hosting different fragment activities

RecyclerView widget used to display items in the list. A class extending *RecyclerViewAdapter* would need to be written to display customs layout for different objects in the same *RecyclerView*. Small class extending *RecyclerView.ViewHolder* must be written for every object or custom view that will be displayed in the *RecyclerView*. Layout of the training quite basic but appealing and displays all the information required. Item displaying this week's activity also appealing and useful

Plus, symbol *FloatingActionButton* widget for adding activities.

NavigationDrawerLayout (slide out menu), for navigating from activity to activity. Can also navigate fragments and trigger other events.

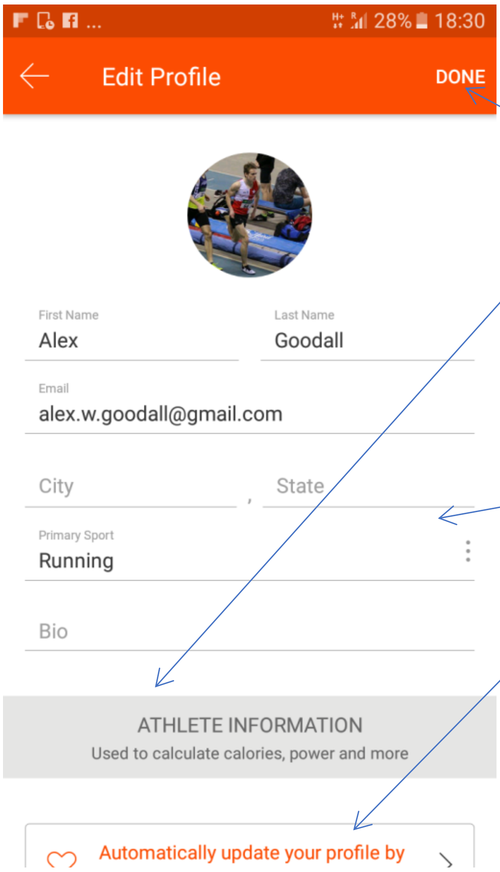


Figure 9 - STRAVA Edit Profile

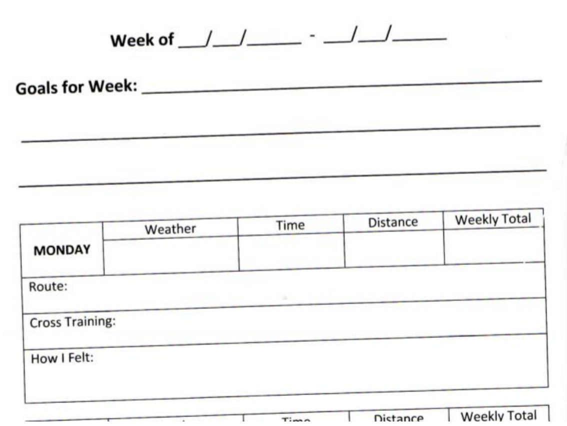
'Done' button lights up white when all the fields have been filled. If data fields are invalid a text dialog pops up notifying user. This shows a consistent theme of the UI throughout the application.

TextView widgets that display information for the user to understand what the details are used for.

When *EditTextView* widget is focused on dialog pops us for selecting race distance, gender and primary sport. The dialog is a list to select from. This can be created with the *AlertDialog.Builder* class and using the *setSingleChoiceItems* method with a string array from resources as the only argument.

For selecting your primary race personal best time, a custom view dialog pops up. This can be done with the *AlertDialog.Builder* class and the *setView* method. However, to select the time we can use a number picker, loading in an integer array from resources using the *setDisplayedValues* method.

Current System



Week of ___/___/_____ - ___/___/_____

Goals for Week: ______________________________

MONDAY	Weather	Time	Distance	Weekly Total
Route:				
Cross Training:				
How I Felt:				

Figure 10 - Training Booklet

Figure 10 is an example format of a training booklet. The table includes details about the training: weather, time, distance, route and feedback. The 'cross training' field is for other forms of exercise apart from running, this will be irrelevant in my system. The weekly total is also added up each day to give the total for that week at the end. There is very little space for time and distance, so this layout would not be ideal for recording all the details of an interval training program. I also find that the weather and route details are irrelevant and will not consider including them in my database. The 'how I felt' section seems rather large and I will consider reducing the feedback to a simple discrete scale on the database, represented by 4 difficulties ranging from easy to hard.

Summary

My findings from the interview show that it is very important to allow the user to record interval training programs and all their details to the database. The ideas of a 'training generator' and 'personal best predictor' were also well received, development of them will continue as planned. The interview finally outlined the importance of making information about weekly activity readily available to the user.

The modelling research was very useful for giving me an idea of how to model a training program and quantify the difficulty of a training program for a particular athlete. I will use these ideas for my 'training generator' and for modifying the statistics of an athlete after they have recorded or completed a training.

The UI research allowed me to evaluate a current system and take away UI ideas, everything I highlighted I plan to use in my application.

Finally, the example training booklet allowed me to assess the problems with the current system athletes use to record their trainings, and decide what information is relevant and worth storing on the database. I will store the details of a training program as text on the database in a standard description format. From this text format the training sets, times and distances can be drawn and used as quantitative data. The description format will allow the user to record a variety of distinct types of training programmes, continuous and interval sessions. The details such as, training type, date, name and difficulty will also be stored in different fields. And the weekly total will be calculated from a function that accesses the database directly rather than being stored as a value.

Objectives

1. Creating an account by inputting: username, password, email and date of birth, details saved to database.
2. Forgotten password – sending an email (containing password and username) to a user after they have entered their email correctly (internet required for this).
3. Secure system by logging into the application by entering your username and password correctly as private data will be stored.
4. Display some information regarding the athlete and recent activities on the home page.
5. Navigating from activity to activity using a *Navigation Drawer Layout* (slide-out menu).
6. Viewing active trainings given by training generator or added manually. Functionality to replace, delete or complete an active training.
7. Dialog to complete and select the difficulty of an active training.
8. Activity log displaying all activities (trainings and races) in the database with search functionality to search by name and description. Functionality to edit or delete any of the displayed activities.
9. Statistics activity which displays information about the athlete's training history and average weekly activity calculated from the database.
10. A training generator activity which allows the athlete to name their training, select the type of training and generate an appropriate training to be saved as an active training to the database. Only 5 active trainings maximum allowed.
11. 'training generator' function must adapt to the athlete's habits, needs, abilities. By consistently generating distances and programs the athlete wants to run at the appropriate speed.
12. A 'Next' and 'Previous' button to choose between the training programmes generated for the athlete. Using a stack object to store previously generated trainings.
13. A Personal Bests activity which displays all the records and predictions the athlete has recorded on the database. Functionality to edit and delete any records displayed.
14. Search functionality to search through the personal bests by distance.
15. Prediction functionality, to make predictions for race distances based on records of other similar distances.
16. Profile activity that allows the user to view all the details of their account, along with an option to log out.
17. Edit password activity that allows the user to change their password.
18. Add activity, allows the user to add a training (select: date, difficulty, distance and speeds), race, active training and personal best to the database.
19. Splash screen with a logo that automatically logs in a user who has been using the app on the device.
20. Items from database loaded into a 'Recycler View' with a swipe up to load widget.
21. Attractive dialogs for selecting dates, times, distance, and training sets.

22. Edit profile activity, that allows the user to edit their account details.
23. Appropriate dialogs asking the user to continue when navigating the application and saving data to the database.
24. Appropriate validation and completion checks when entering data into the database.
25. Information buttons displaying a dialog that reads "Track Trainer will assume you take a 5-minute rest in-between each training set, this allows your heart-rate to return to normal".
26. Easy and logical navigation throughout the application
27. Logical entity relationship between Race and Personal Best objects. Whereby a personal best represents the fastest instance of a Race objects of a given distance. A personal best can also be defined uniquely without a Race object associated to it, if it is the faster than all instances of Races of a given distance.

Database Entities

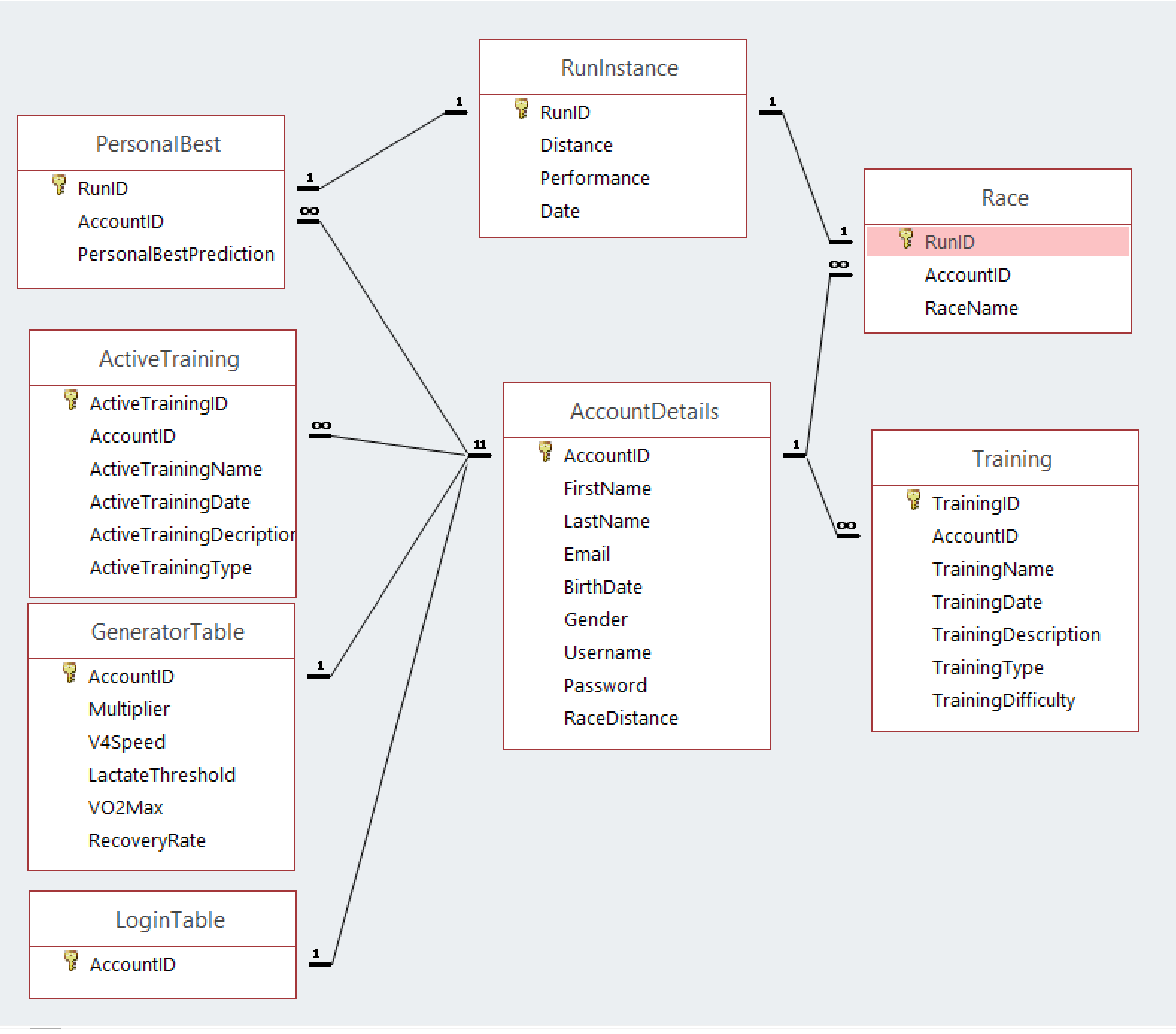


Figure 11 - Dynamic Database E-R Diagram

The E-R Diagram *Figure 11* represents all the data that will be store in the database. I will inherit from the *SQLiteOpenHelper* class to create a local database on the device that stores all the relevant data. I will write methods that execute SQL code to access and store information on the database. Because the database is created locally on the device I need a way of providing external real-life data for prediction models and assigning values to the athlete when an account is created. So, I plan on implementing a static database with all the relevant real-life data I will have extracted, data will be transferred from the static database to the dynamic database when it is first created on the device.

Static database

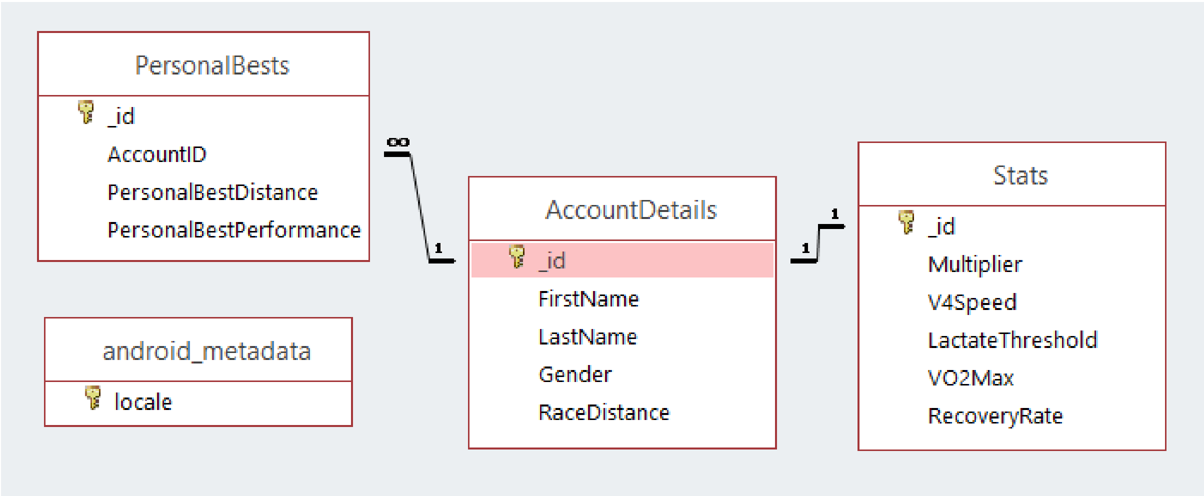


Figure 12 - Static Database E-R Diagram

Figure 12 represents the static database which is far simpler and fully normalised, storing only the required data for making prediction models. The personal bests table will allow the user to make predictions on performance of certain distances, using real life data. The stats table will allow the system to generate some initial statistics for a new account when it is created based on their primary race distance and personal best for that distance. The account details table connects these 2 tables. Real-life data will be manually inputted into the static database and accessed from within the application files (assets directory). A similar class inherited from *SQLiteOpenHelper* will be written for accessing the static database values.

Mathematical Models

Regression Line

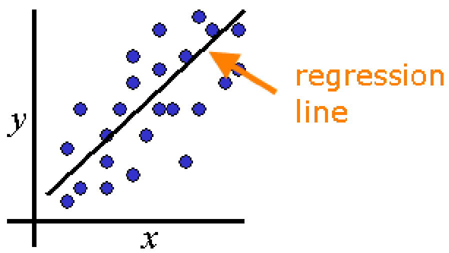


Figure 13 - Regression Line

$$a = \bar{y} - b\bar{x} \qquad \text{and} \qquad b = \frac{S_{xy}}{S_{xx}}$$

$$\text{Where, } S_{xy} = \Sigma xy - \frac{\Sigma x \Sigma y}{n}$$

$$\text{And } S_{xx} = \Sigma x^2 - \frac{(\Sigma x)^2}{}$$

Figure 14 - Regression Line Formula

The idea of regression lines will be used for making predictions using personal bests. For example, an athlete wants a prediction for their performance of a distance 'y'. An array of personal best values (doubles) of 'y' and corresponding values of 'x' (another similar distance to 'y') will be drawn from the database. A regression line will then be made using these 2 arrays and a prediction for 'y' will then be accessed by inputting the value 'x' into the regression line formula. To make the prediction more accurate 2 distances similar to 'y' will be assigned. So, in effect 2 regression lines are created where a value (performance) from the user for these distances exist. The 2 predictions will then be subtracted from the actual value to find the deviation of the actual value from the regression line (in terms of standard deviations). The two predictions will then be considered, however the most accurate one takes priority when formulating the final prediction.

Discrete Probability Distribution

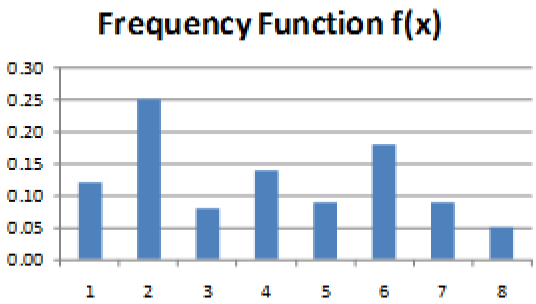


Figure 15 - Discrete Probability Distribution

The idea of discrete probability distributions will be used for determining distances in the training generator. A discrete probability distribution will generate a random value for X given the probability. In the case of the 'training generator' a discrete probability distribution will generate a random distance the user has to run based on a probability. Initial probabilities will be hard coded for each distance. However, as the user logs more information to the database a new probability distribution will be created based on distances the user has run frequently during their trainings. This idea will also be applied to the amount of repetitions generated for a certain distance for interval training and the amount of sets in a training program. Allowing the 'training generator' to adapt to the athlete's habits.

```
public int getX(){
    int randomNumber = MathFunc.getRandomNumber(0,100);
    double probability = 0;
    for (int i = 0; i < PXisx.length; i++) {
        probability = probability + PXisx[i];
        if ((probability * 100) >= randomNumber){
            return valueX[i];
        }
    }
    return valueX[valueX.length-1];
}
```

Figure 16 - Code to return random value of X

Exponential Model

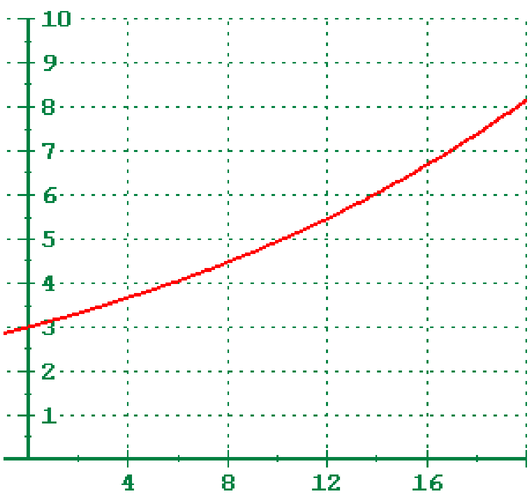


Figure 17 - Exponential Model

The idea of exponential modelling will be used to quantify the difficulty of a training for a particular athlete. Each athlete will have 4 values associated with them (V4Speed, Lactate Threshold, V02Max, Recovery Rate) all of which will be stored and updated in the database. The 2 values VO2Max and V4Speed will be used to model the exponential model of speed (m/s) against blood lactate (mol/l). The lactate threshold value will indicate at what blood lactate level the rate of lactate accumulation equals the rate of blood lactate dissipation. So, if the value of blood lactate calculated from the speed greater than the lactate threshold value then the blood lactate value is increasing with a rate directly proportional to (blood lactate – lactate threshold). The recovery rate is a value that indicates the rate of blood lactate breakdown when resting. Using these 4 values we can model the training and calculate the average blood lactate level throughout the training of an athlete. The average blood lactate value will indicate how difficult the training is for a particular athlete. So, when generating a training the 'x' value (speed) will be varied with a for loop and the average blood lactate value calculated to generate an appropriate training.

```
double b = k / VO2MAX;
double a = V4Speed - (1/b)*Math.log(1/b);
double d = bL - Math.pow(Math.E, -(b*a));
this.lactateCurve = new ExponentialModel(a,b,1,d);
//where f(x) = c.e^(b(x-a)) + d
//where k and bL are a constants
```

Figure 18 - Code to build exponential model