

T LEVEL

**Technical Qualification in
Digital Production, Design
and Development**

Guide Standard Exemplification Material

Occupational Specialism:

Digital Production, Design and Development

PASS

Version 1.0

© everything possible / Shutterstock
© Independenz / Shutterstock
© 3dreams / Shutterstock



Pearson

23:35:60
Business Str
Management

Digital Production, Design and Development: PASS

Introduction

The exemplar materials within this document relate to the Digital Production Design and Development Occupational Specialism and have been created using the approved Specimen Assessment Material (SAM). These exemplification materials are designed to give Providers and students an indication of the knowledge, skills, and understanding that attest to a threshold Pass grade. Each task within the GSEM shows the students work at that level for that task at the threshold of that grade.

The Guide Standard Exemplification Material is split into two sections: Exemplar material, the exemplars provided are representative of the tasks in the SAM. It is important to note that in live assessments a student's performance is very likely to exhibit a spikey profile and the standard of their performance will vary across the tasks, unlike this exemplar material. The overall grade will be based on an aggregated mark for all tasks.

Examiner commentary includes detailed comments to demonstrate how the exemplar material attests to the minimum standard of Pass for the Occupational Specialism. This may include commentary on improvements in the work to obtain higher marks/grades.

Providers can compare the exemplar material against the mark scheme within the SAM, to give guidance on the standard of knowledge, skills and understanding that need to be met for Pass.

Task 1 Analysing the problem and design a solution

Activity A(i)

Research

To show and emphasise our design, the following section is a comparison to an existing fitness website (in this case: <https://www.mapmyrun.com/>).

Homepage:

From the above, you can see that the website have a minimalist style which refrains from data/information being crushed into a small screen space. The page is split up into sections where additional information to address the advertising and promotion of the website are placed. These still incorporate a minimalist design, in keeping with the theme.



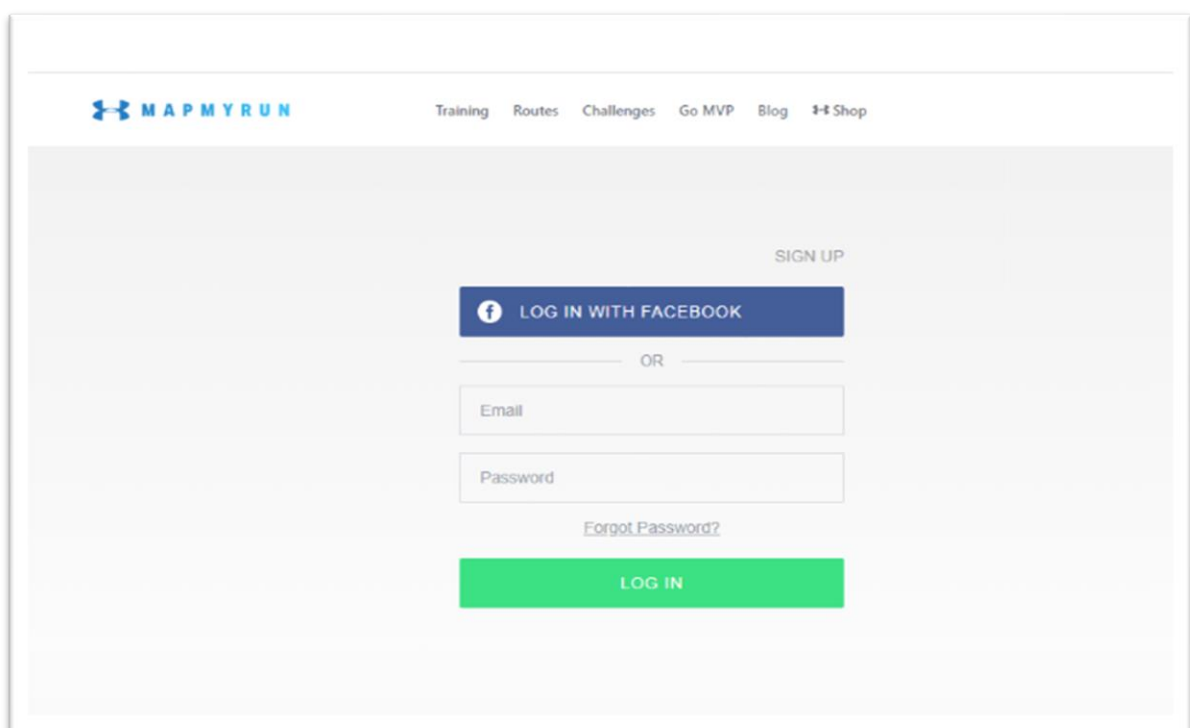
The differences are that the main colours that are used, correspond to the colour of the logo. In our design we have used the same methodology, and have used colours included in the newly designed logo and the original. Furthermore, MapMyRun (Under Armour) have used links to an app as well as logging information on a desktop. Their reasoning will be that it makes it easier for the user to access the data from their smartphone instead of signing into a desktop website, however choosing to include the desktop option allows for more customers who may not have a compatible smartphone to use the service.

Loading Times of Homepage:

1.27s - MapMyRun
970ms - Our Mock Website

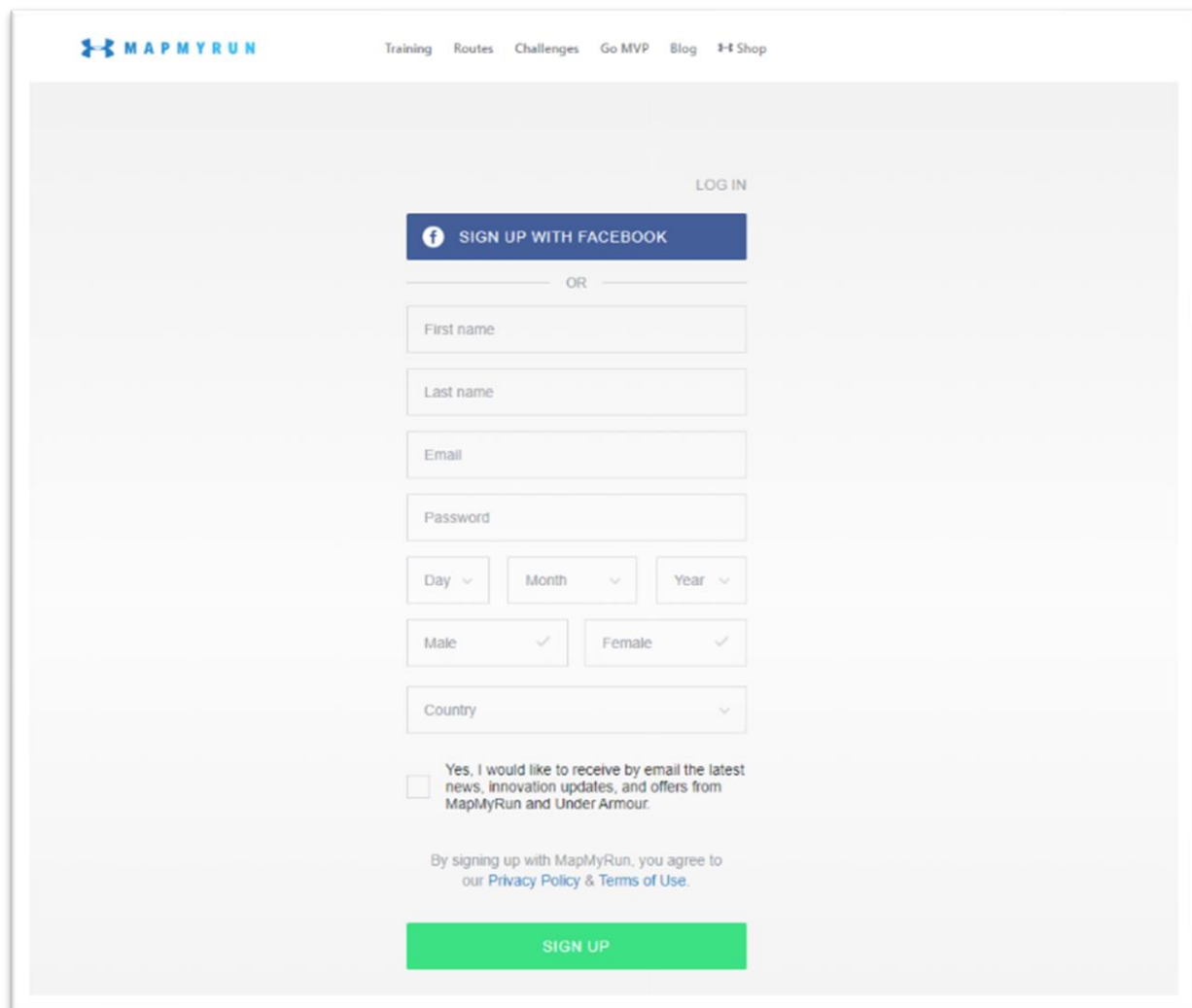
This shows that our website will give people a faster response and will allow for users to access the website quicker than larger corporations such as Under Armour's MapMyRun.

Login Page:



Again, a minimalist design has been used in the login page which shows continuity within the website and provides a professional look for the user. However, MapMyRun have used a separate page for the user to login whereas, in our design, the login page is a pop-up which allows for faster access speeds as the element has already loaded when the user entered the URL. In addition, a "forgot password" feature has been implemented so that users can access their accounts even if they don't remember their password. In our design we have not included this as many client will be able to have their password reset by users instead of remotely.

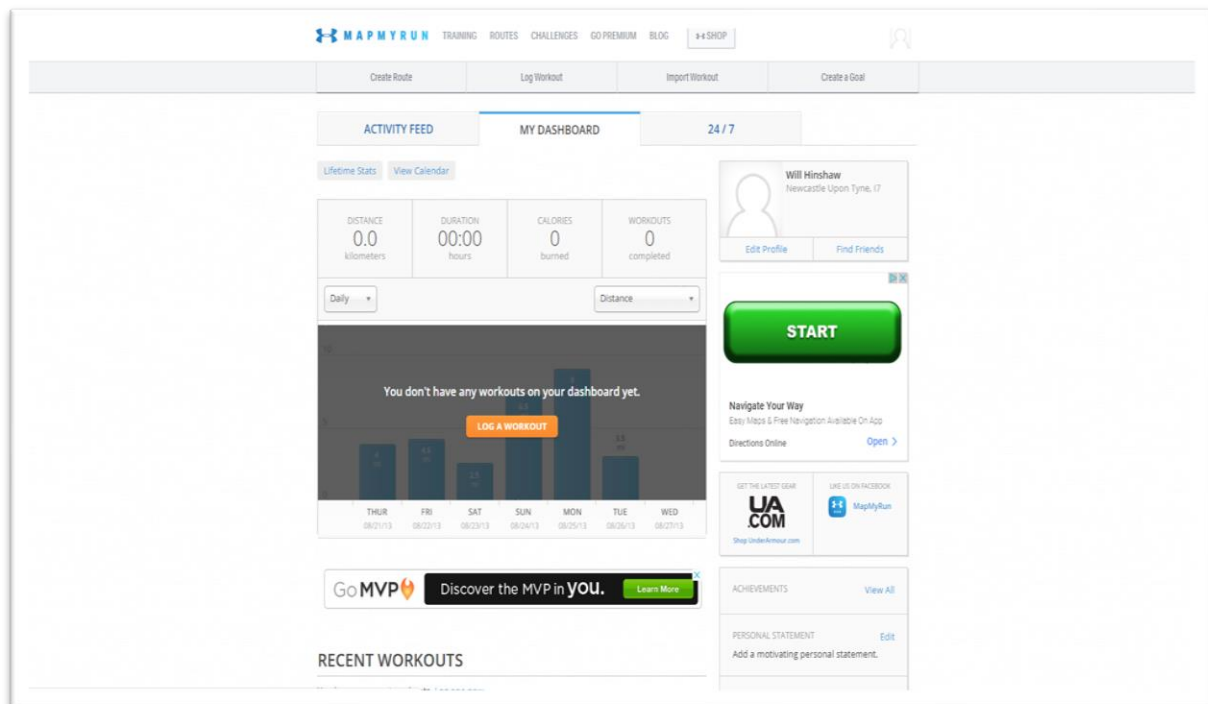
Sign in Page



The image shows a web page for MapMyRun with a navigation bar at the top containing links for Training, Routes, Challenges, Go MVP, Blog, and a Shop icon. The main content area is titled 'LOG IN' and features a 'SIGN UP WITH FACEBOOK' button. Below this is an 'OR' separator, followed by a series of input fields: First name, Last name, Email, Password, and a date selector with Day, Month, and Year dropdowns. There are also radio buttons for Male and Female, and a Country dropdown. A checkbox is present for receiving emails, with a note that it applies to MapMyRun and Under Armour. At the bottom, there is a disclaimer about agreeing to the Privacy Policy & Terms of Use, and a large green 'SIGN UP' button.

There are no differences within overall layout of the MapMyRun design. However, the sign up criteria is different as the system is not based around a client and users platform. With this in mind, we have included a “class code” requirement in the signup process so that the system can validate the user and the school which they attend. This adds another layer of security to the system as the general public will be unable to access the features.

Dashboard:

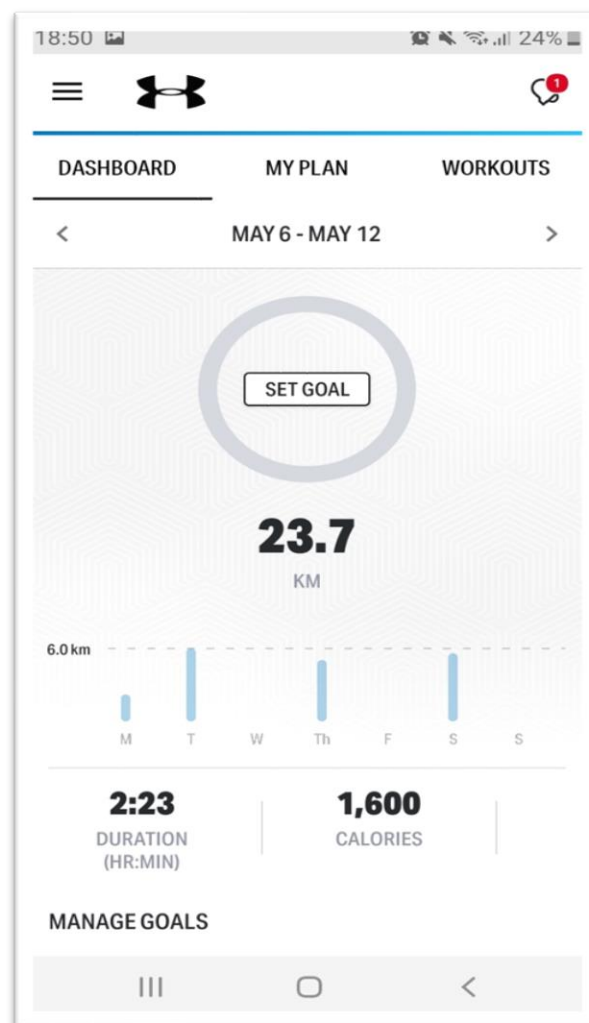


The overall dashboard of MapMyRun is entirely different as the user is able to access their profile settings without going to a dropdown menu, the user can view data directly as they login to the system without navigating to a menu and a second navigation bar has been added so that the user can directly access features such as “Create Route” or “Log Workout” without looking through the page to find said features. In our design the user has the ability to navigate all options through the navigation bar at the top of the page, which allows for the user to remain orientated on the whereabouts in the website they are. This is key design feature of a website as a user may become frustrated if they get lost, and have to navigate using the browsers “back” function. Within the navigation bar, the current page has been highlighted to further help the user locate themselves within the website.

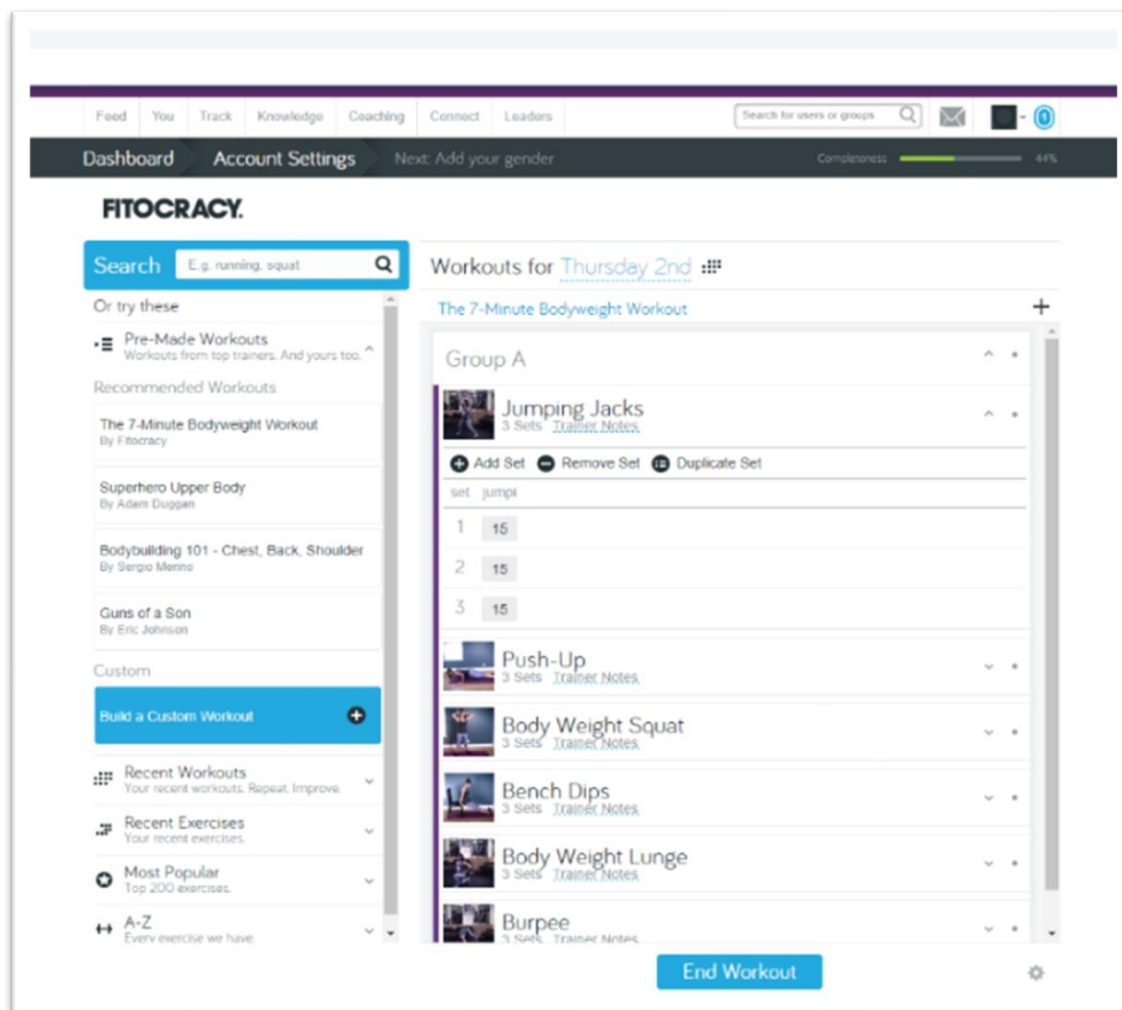
In addition, the page includes adverts which can clutter the page and make it feel unprofessional. On our design, the page doesn’t include any “bad” adverts which have no remote link to the website’s purpose. Furthermore, in the above diagram, there were no pop-up adverts that restricted the functionality of the website.

Other pre-existing fitness application include which are dedicated to fitness in general – UA's MyFitnessPal for example allows users to log their food intake as well as a range of predetermined sports, and calculates their overall calorie needs, intake and expenditure, as well as providing guidance on macros and showing complex statistics on your eating habits. However, it has no facility for users to create workouts based on strength exercises or even just track the exercise itself, as the application is focused on calorie usage. However, it is a very popular app with a very easy to use UI, depicting each day on one screen and allowing the user to flick between, the guidance from its interface found it to be the easiest to use. Most of these general fitness application also have premium features only accessible to those who pay, usually the features which allow users to customise their experience more.

MyFitnessPal also interfaces with many applications such as the preinstalled Samsung Health, other UA apps such as MapMyRun or MapMyRide, and wearable technology such as heart-rate monitors and shoes with GPS or accelerometers inside.



FITOCRACY COACHING



This website was particularly interesting, you can sign up for free, and provides you with work out programs and personal trainer you can select depending on what your goals are. Found it easy to navigate, you can also view other member's progress and what problems they have encounter on a live feed. Once have selected you work out session you can then log your progress as shown below. Found this to be very valuable.

Activity A (ii)

I have been asked by the owner of ToKa Fitness to develop a digital system that will:

- provide information on fitness training
- provides help on healthy living
- provide access to digital content
- provides to support customers with their training and healthy lifestyle
- encourage existing customers to use more of the services provided by ToKa Fitness
- Allow customers individual accounts
- Provide eating plans

The Purpose of the App

In the context of our App, the purpose of the site is to store records of physical activities that are collected by wrist worn devices that can be accessed by a user or an instructor remotely or in class to view activity data at different levels (i.e. individual level or class level). In addition, the project team wanted features such as setting challenges and making the data available in different forms for use especially by user for teaching of other subjects (E.g. Math's, English etc.) Furthermore, the interface should show different level of details when used in class by the instructor/user than when used by them remotely.

Target Audience

Designing an App to make it “aesthetically pleasing / state of the art” is not the first port of call, if it doesn't appeal to the targeted audience then there is no point in having a App at all. In the context of this design brief, the App is targeted at client and user who wish to store and keep track of physical activities. Developing an easy-to-understand and user-friendly App is a crucial part of any business as it defines the appropriate target audience – its needs, wants and expectations of the App.

Aims and Objectives (ToKafit Brief)

The App is designed primarily to store and show physical activity data in an easy-to-read format both to instructors/user and client who use the ToKafit wrist worn device. It will allow for user to keep track of client' progress and challenges with the ability to showcase class data to a class without revealing personal data or naming underworking client.

A further possibility of the use of the App is with smaller personal training groups

as the instructor can monitor progress and possibly motivate a certain group member to do more activities and set realistic challenges.

A further goal of the App is to educate client of the importance of personal motivation and fitness in their life.

Functional Design Choices

The following are the main functionality design choices that should be considered when designing an App:

- Should be easy to navigate – Without a navigation system the user can become very lost within a complex App, in addition, the navigation should allow the user to easily access part of the app without having to search through the page for the link.
- Have a clear indication of whereabouts the user is on the App – An indication of the location of the user can help to decide the next move of the user, if, for instance, the user was in a shopping basket and no indication was present, then the user may not realise that clicking on “Buy Now” will process the transaction without their knowledge.
- App should be easy to find/search for – The App should be found at one of the first search, research shows that the first search receives 95% of the traffic, leaving 5% for the other Apps, this means that the App in 1st rank will gain more customers/users attracting a larger audience.
- App & Login Forms should be secure – Protocol **Secure** which is used for secure communication over a computer network and protect against “man-in-the-middle” attacks, where the attacker secretly relays and possibly alters the communications between two parties who believe that they are directly communicating with each other.
- No adverts

Legal Considerations

- Domain Names: Usually they are registered in the name of the designer or developer and not in the name of the client. It is the client who has paid for the domain name and therefore the client should be the owner or registrant.
- Ownership: The creator/designer of the App is by default the holder of the copyright and therefore the designer is the owner of the App, unless decided by the client.

- Content: Protection of the designer from the risks associated with clients giving them information or content that they want to include with the design when they are not entitled to do so. This may be illegal content or something that infringes the rights of the 3rd parties.
- Data Protection: It is under UK law that the App notifies the user of what you intend to use the personal data that you collect for.
- Cookies: Every mobile browser, just like desktop browsers, has different cookie settings and handle first party and third party cookies differently. Cookies also exist within apps when a browser is needed to view certain content or display an ad within an app. However, the cookies are completely "sandboxed" in apps. There is a requirement under the amended E-Privacy Directive and the UK.

Regulations to:

- Inform users about cookies and what you are going to use their information for
- Obtain their consent to the use of the cookies

Currently, gym users who want to track specific details at the ToKa gym are unable to for their workouts and exercises, often have to do this on paper so they can track what they have completed during that workout session. However, this approach means that it's very difficult to these people to quantifiably track their results – the client, ToKa Fitness therefore wants an application allowing users to track and log their activity at the gym. The application would allow users to enter data such as the exercise, weight, reps and sets the user completes about each workout, so that the user can track their progress long-term.

This project lends itself to a computational approach, partly because of the convenience of tracking details by your phone or smartwatch (removing the need to carry pen and paper, and allowing users to track the details immediately, reducing the risk of errors), and partly because a computational solution will allow automatic progress tracking as it can automatically calculate and display your progress from the previous input data. This will make tracking users' gym progress much easier and provide them with details of their progress which would be very difficult and time-consuming to recalculate themselves every week.

The key performance indicators (KPIs) and user acceptance criteria for the proposed ToKa Fitness offers personal training sessions and advice on fitness training and healthy living to its customers and would like an App for their

specific requirements. ToKa Fitness specific requirements are that the proposed solution:

- has free access with some accessibility to services
- customers section for paid content to full services
- accessibility features for users with sight loss
- link to 'social media' features
- ability to add new workouts and eating plans

Other areas to consider are:

- the advice given to reduce health issues or injuries
- level of detail of instructions in a fitness training plan
- user experience to promote company image - visual assets and content
- complying with relevant legal regulations and guidelines
- compatibility across different devices Android and iOS
- privacy and security of user data

Accessibility issues to be considered:

- clear and readable interface and branding
- copyright
- access for disability people
- error messages to also have an error icon

My proposed solution should provide ToKa Fitness with professional and easy to use system that promotes their business and encourage customers to use it. All customers would have their own account. 'Free member' will have a basic level of accessibility and 'Full member' with a monthly fee for more complex features and functions.

Usability Features

The design of the above system features a lot that makes it accessible to many people. It has a simple layout with minimal options, meaning that the boxes/areas and text for everything in it can be large enough for those who have sight problems to read. The text displayed would also have to be actual text rather than images containing text so that it works with text to speech software. The fonts used should look professional yet easy to read. There is no audio that must be heard in order for the system to work as intended, as although the alert will make a sound when triggered, it will also pop up on screen and persist until dismissed.

There will be use of colours (uncertain about which colours at the current time) to break apart the sections and make it easier to navigate. Although the colours would be something to consult my client about, nothing too bright should be used, so greys, blacks or maybe muted blues and purples would be a good choice to make sure that everything is broken apart quite nicely but it isn't too much that it becomes hard to look at.

The programs basics will function the same as any other so the user has a sense of familiarity. The cross to close the window will be in the top right corner (although due to the nature of the program it will still run in the background) and familiar icons will be used to indicate features such as the dropdown boxes or an input text box. This will prevent the user from becoming confused.

Member
Private and secure customers area – password protected
Ability to customise some workouts and eating plans
Use of daily food planner with limited food analysis and reports
Use of daily fitness training planner with reports
Access to some fitness training/recipe videos
Links to social media/ blog / forums
Use of daily food planner with food analysis and reports/graph analysis
Use of daily fitness training planner with reports/graph analysis
Access to all fitness training/recipe videos
Link to Social media / Fitness training Videos/ Recipes/ blog / forums
Ability to customise a workout and eating plans
Details analysis of daily foot plan and health plan

Customers would have access to a daily fitness training planner and videos that have been verified by sports professionals and doctors. It is important that the videos and planners are safe to use and provide the correct advice and guidance.

The setup of the customised eating plans will also have to be verified by a nutritionist as they must meet the correct and current nutrition guidance. If the nutrition guidance is incorrect, the member could get very ill as the result of following the customised eating plan, and ToKa Fitness could be sued and get a bad reputation and go out of business.

Members should be able to customise a workout and eating plans, depending on the customers' membership level. If the customers' experience of using the App is good, they will continue to use it and promote its use to their friends and on social media.

The App will have to be secure because of the member's privacy and security of user data to comply with legal requirements such as Data Protection legislation and food and nutrition regulatory guidelines. Customer details must be secured by password protection and clear terms and conditions. The interface should be easy to use and navigate, be accessible across different devices and have accessibility features for users with sight loss.

User requirements

ToKa Fitness has commissioned my software development company to develop a digital system. ToKa Fitness offers personal training sessions and advice on fitness training and healthy living to its customers and would like an App for their specific requirements.

ToKa Fitness specific requirements are that the proposed solution.

- has free access with some accessibility to services
- customers section for paid content to full services
- accessibility features for users with sight loss
- link to 'social media' features
- ability to customizable workout and eating plans

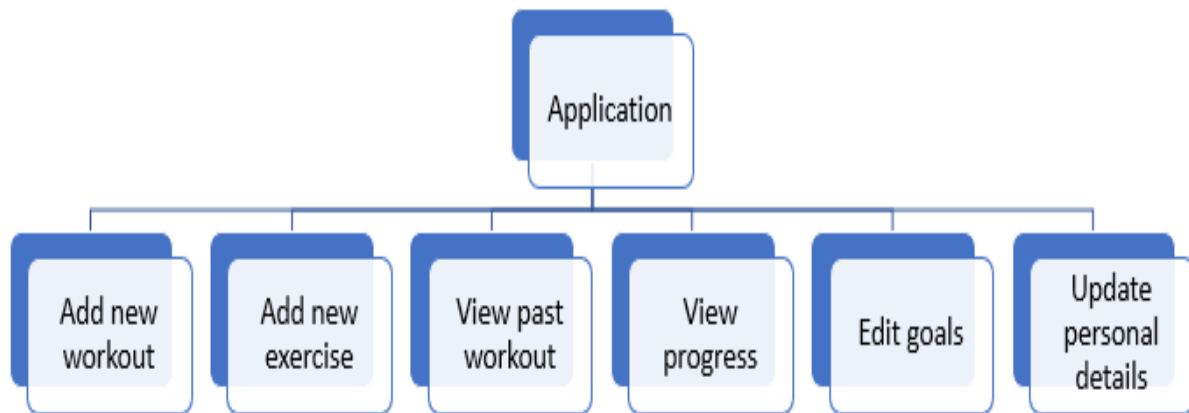
The target user group are adults, male and females. Due to legal and ethical nature of Apps, it might be difficult for under-18s to follow the exercise and enter the correct details and follow the food and recipes correctly. To comply with age restrictions and guidelines, clear advice will be given to reduce health issues or injuries.

The customer area will be secure accessed via a password and payment made in a secure platform. This will comply with legal requirements. The privacy and security of user data is important for each member to have access to their area via a secure login.

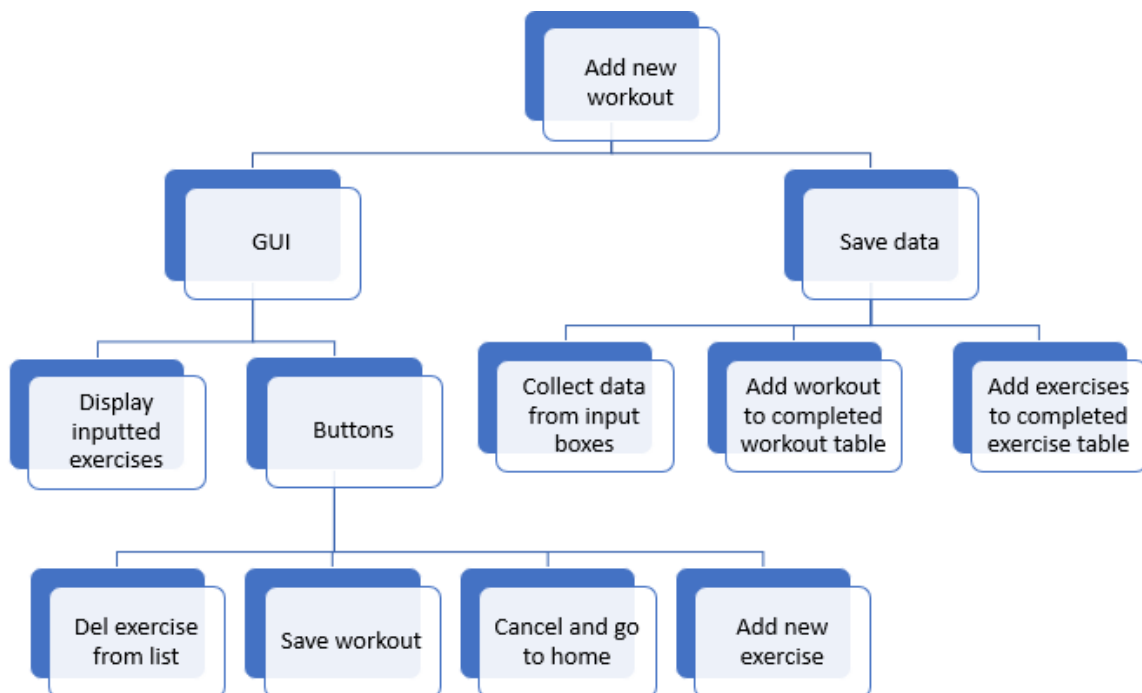
Each customer will be able to customisable workout and eating plans and generate a report on their progress for specific periods, and the level of detail of instructions in a fitness training plan will vary to allow the customer to meet their identified goal.

All links to external sources will take consideration of Copyright and intellectual property and Licensing requirements.

Decomposition



The project consists of 6 main features - the ability to add a new workout and exercises to it, view previously entered data, view progress, edit goals and update personal details. These 6 comprise the main body of the application, as identified by my research as the most important features to include. Each will consist of its own activity and functions, being self-contained. I have not included optional features I will probably not have time to implement, like custom exercises or Facebook integration. Each of these 6 main features can be broken down into their component parts



This is a breakdown of the component parts involved in adding a new workout. It's split between the GUI- containing display elements including a list of inputted exercises and buttons- and saving the data, which contains 3 main parts. The input data has to be collected from the input boxes and stored in local variables. Then I will use SQL to add a new record to the workout table, and any exercises to the exercise table which are associated with this workout. The appropriate relational links will also have to be made. A series of buttons and associated tasks will be necessary, including the ability to cancel the input, delete an exercise and navigate to add new exercise to workout page.

Validation

In terms of user input, there will be a handful of drop-down menus, and lots of buttons, which will require no validation as it will be restricted to correct input.

The input exercises will be limited to the exercises included in the database initially (although may expand on that and add custom exercises if it is feasible) and will have stored what type of data each exercise requires, e.g. sets, distance, weight etc. so that the only types of error possible is the users entering the wrong values in their data. Add some validation in these inputs to ensure users don't enter unreasonable values like lifting 400kg. However, by restricting the user to pre-entered exercises can at least ensure consistency between workouts.

Most numerical input can be checked to see if the range is correct, but other than that it will be very difficult to validate- entering the wrong input will result in errors in the progress page but allowing users to edit past workouts should allow them to change any wrong input themselves when they notice wrong results in the progress page.

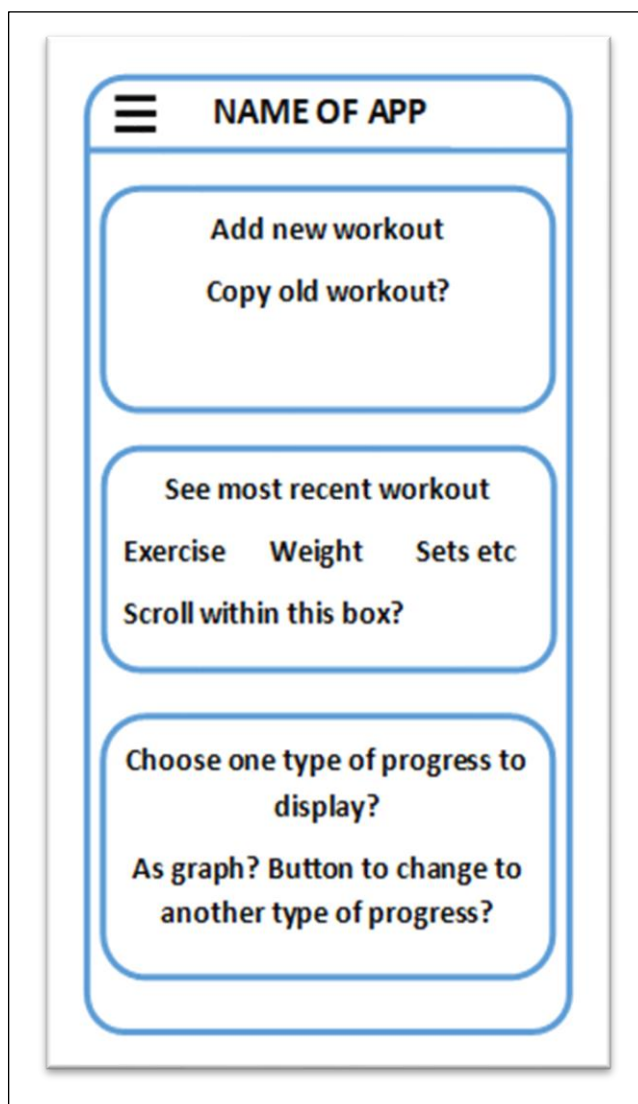
The best enforcements of validation that I can make to ensure incorrect data will not crash the application, is to include a presence check on every input box and a type check on every database insert.

Activity B

Visual/Interface Designs

User Interface

HOME PAGE



The home page is not required to do any major tasks, and is designed to have a simple display and allow users to easily access other parts of the application.

When the 3 black lines in the top left corner are pressed, the options menu will expand- this is a standard design used by most android applications and so will be familiar and intuitive to the user. On this menu, the user will be able to switch between activities, such as to track new workouts, view past workouts, see progress, edit goals and any

GOALS PAGE

A wireframe of a mobile app screen titled "NAME OF APP". On the left side, there is a vertical menu with five options: "Track new workout", "View past workouts", "See progress", "Edit goals", and "Personal details". The right side of the screen is a large empty rectangular area, likely for displaying content related to the selected menu item.

Moving to the edit goals activity in the menu should bring up this page. In my market research, the most common goals potential users were working towards were orientated around how often they work out, how much they can lift, and body measurements such as size, weight and fat. Should also be enable feature of these on the goals page, with varying input options? Next to each goal type will be an input box, allowing the user to enter numbers as appropriate, and save or cancel their changes to the relevant database

A wireframe of a mobile app screen titled "NAME OF APP". The screen contains several input fields for user data: "Workouts per week" with an "INPUT" button, "Bodyweight/fat?" with an "INPUT" button, and a section for "Muscle" with a table. The table has two columns: "Muscle" and "Kgs/lbs". Under "Muscle", there are three rows: "Arms", "Legs", and "etc". Each row has an "INPUT" button in the "Kgs/lbs" column. At the bottom of the screen, there are two buttons: "SAVE" and "CANCEL".

This is where users will log new workouts into their exercise log. This page is accessible both from the home page and from the menu.

Under the page title, an option for users to load previous workouts should be available. The day text next to it should allow the user to select a day from the past, and the workout associated with that day should be loaded automatically into the data below to reduce the difficulty in entering lots of repeat data into the workout.

Underneath, I have displayed some sample data of a workout. It displays the exercise name, and relevant data associated with that exercise, for example sets, reps, weight, time or distance. The add button underneath allows users to add a new exercise to that workout (navigates to an entire new activity for this) and there is the option to save or cancel the users input.

A mobile app mockup for the 'ADD EXERCISE TO WORKOUT' screen. At the top is a header bar with a hamburger menu icon and the text 'NAME OF APP'. Below the header is the title 'ADD EXERCISE TO WORKOUT'. The main area contains a 'Search bar' and a 'Filter by body part' button. Below these are three exercise cards: 'Situps', 'Bicep curls', and 'Squats'. The 'Bicep curls' card has input fields for 'Sets: input' and 'Reps: input', and an 'ADD' button. At the bottom is a 'DONE' button.

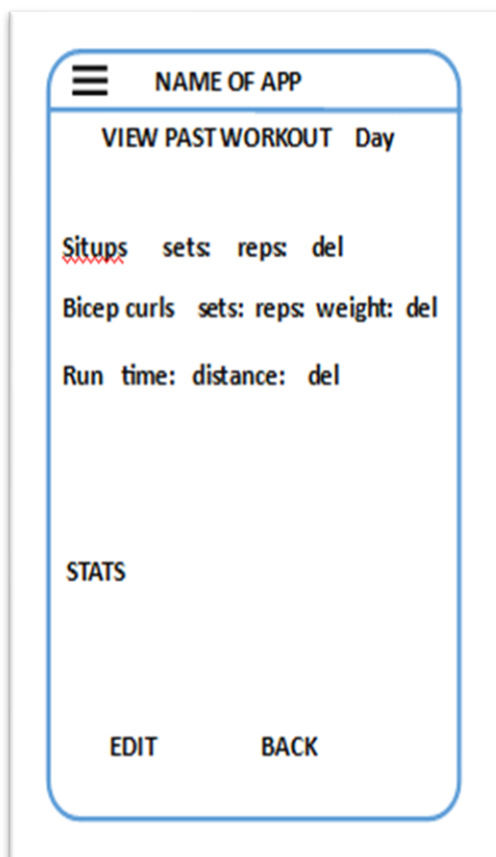
Here, the user can search by name, and filter by body part to select the exercise they want to add to a workout. This is scrollable to include dozens of exercises.

Once the user has selected an exercise, it expands to allow them to enter the relevant details. The add button will add this exercise to the workout but remain on this activity so the user can easily multi-add lots of exercises. The done button will then return the user to view the entire workout.

To allow users to access all their past data, it will display workouts in mini, potentially scrollable boxes within the activity. The activity can either be entirely scrollable or allow the user to move to the next set of 3 activities. If the users want to access a specific workout, they can select the date and it will come up below.

A preview of each workout should come up in each box, and if the user clicks on it should expand into a new activity with all the details about that workout.

A mobile app mockup for the 'VIEW PAST WORKOUT' screen. At the top is a header bar with a hamburger menu icon and the text 'NAME OF APP'. Below the header is the title 'VIEW PAST WORKOUT'. The main area contains a 'CHOOSE DATE' button and the date '09/11/19'. Below this are three workout preview cards. Each card has a title (e.g., 'MOST RECENT WORKOUT'), a 'DATE' field, and a 'CLICK TO EXPAND' button. At the bottom is a 'SCROLLABLE' label.



Expanding each workout from the above page results in this page giving details of each past workout. I could potentially include some stats on the workout here if this was something my client wanted.

This page is potentially very similar to the creating a workout page and therefore I could reuse some of the code.

SEE PROGRESS and PERSONAL DETAILS

NAME OF APP

DROP DOWN

CHANGE IN BODY FAT

CHANGE IN WEIGHT LIFTED

% PROGRESS TOWARDS GOAL

Progress can be viewed in several formats, determined by the drop down menu. The drop down menu will include things like days worked out (in calendar form), % increase of weight lifted over time (by body part) and change in body weight. Underneath I can include quantitative progress like change in body fat since initial value and progress made towards personal goals.

NAME OF APP

UPDATE PERSONAL DETAILS

Weight Date INPUT

Body fat Date INPUT

etc

SAVE

This page allows users to update data which is not part of a workout, e.g. body measurements like waist size, weight, body fat or any other appropriate measures. This data will be stored and use on the progress page.

Requirements	Requirements accessible in UI?
Allow users to input exercise, weight, reps, sets	Yes- add new exercise activity
Allow users to input personal details	Yes- personal details activity
Allow users to change personal details	Yes- personal details activity
Display past workouts	Yes- view past workouts and work out details activities
Display progress	Yes- see progress activity
Allow the users to modify previous workouts	Yes- view work out details activity
Intuitive UI	Yes
Calculate some statistical analysis on input data, eg change in weight lifted over time, change in 1RM.	Yes- see progress activity
Allow users to see cardio and strength workouts in conjunction	Yes- view past workout activity
Provide features completely free of charge for all android users.	Yes

Usability Features

My main criteria for usability of design is that each feature is clearly signposted and can be completed with a limited amount of clicks to ensure the system is efficient.

Using a standard design for the menu page means users will instinctively know where to go to find features they would like to use, and consistency of the design of each page (i.e. same style of inputs) will allow users to very quickly understand how each activity functions. Each activity has a clear heading and access to the menu page so users know they can return to any point in the application quickly and easily.

The design of the application has been done with the intention of having a designated activity for almost every task, so that each page can be kept as simply as possible and it is easy to signpost the user to each feature. This means that users can access each feature through the menu system and as a result reduces the clicks needed to access each feature and improves time efficiency for them to input their data after each workout – this satisfies my success criteria. The consistency and limited objects on each page means the user will always know

how to navigate the application and should the application should have excellent visual clarity, if you limit the application to a handful of colours and use consistent fonts and font sizes. Having links to essential pages also on the home page is good practice in terms of usability. Keeping a menu button on the top bar of every activity will always allow users to get back to where they were if so needed.

Additionally, as the colours used in XML elements in Android Studio are defined in a values file, it's easy to define the few colours used in a colour scheme there and have the application use them automatically. This makes the application easier to use as each activity is consistent.

Algorithms and pseudocode

As android applications are event-driven, i.e. driven by the users actions rather than the results of previous code, the algorithms given below are used when an application is triggered to create, or when a button is pressed, for example. Not detailed below are the parts of the application which handle this, such as setting listeners for the users actions- these trigger the algorithms given below.

As each activity is designed to solve a particular solution requirement(s), it is easy to develop the application iteratively- each iteration will involve the solution of one or more solution requirements, in the full development of one or more activities.

Add new workout

These are the algorithms associated with add new workout activity.

```
PROCEDURE build()
    CREATE textObject (text="ADD NEW WORKOUT")
    CREATE textObject (text="Load previous workout")
    CREATE buttonObject (text="Day", onclick=expand())
    CREATE buttonObject (text="SAVE", onclick=save())
    CREATE buttonObject (text="CANCEL", onclick=cancel())
    displayInput()
ENDPROCEDURE
```

```
PROCEDURE displayInput()
    IF activity launched from start THEN
        DO NOTHING
    ELSEIF day clicked in calendar THEN
        //DO STUFF
        //verify workout on that day exists
        //get all exercises associated with that days workout
        //input data to displayInput()
    ELSE
        ARRAY data = GET data from "add new exercise activity"
        data.append(previously saved data from this activity)
        FOR i between 0 and length(data)
            CREATE textObject (text = data[i][exerciseName])
            CREATE textObject (text = data[i][exercise stats])
            CREATE buttonObject (text="DEL", onclick
                                =del(exerciseName))
        ENDFOR
    ENDIF
ENDPROCEDURE
```

This procedure is ran on launch of the activity- it builds the GUI of the page and defines its behaviours. It will create each GUI object such as text or buttons and defines which procedures buttons call on click. It also calls the **displayInput** procedure which displays any exercises and corresponding input which has been sent by add new exercise activity. This is a separate procedure as it may be called several times over the life of the activity.

```
PROCEDURE add()  
    STORE input until activity relaunch  
    CLOSE activity  
    LAUNCH "add new exercise" activity  
ENDPROCEDURE
```

The **displayInput** procedure displays any exercises which have been selected as part of the exercise. If the activity is launched from the menu, then none is displayed. If the activity is launched from the add new exercise activity, it will display data passed by that activity and any saved while the user is selecting this input by looping through the data and creating text objects and button objects for each exercise input.

When the user wants to add more exercises to their workout, they can press the add button, which calls this procedure on click. It stores any previously input data until the user navigates back to this page, closes the activity and launches add new exercise activity so the user can input more data.

```
PROCEDURE cancel()  
    DELETE all input data  
    CLOSE activity  
    LAUNCH home page or last open activity  
ENDPROCEDURE
```

If the user decides not to input this data, they can press the cancel button, calling this procedure. It deletes all saved input data, closes the activity and navigates to the home page.

```
PROCEDURE expand()  
  
    DISPLAY calendar  
  
    IF user clicks on date THEN  
        displayInput(date)  
    ENDIF  
  
ENDPROCEDURE
```

This procedure is also optional- if the user wants to auto-input data from their last workout, they can press on the “Day” text and it will call this procedure, which displays a calendar and calls the displayInput procedure if the user does select a day.

```
PROCEDURE save()  
  
    CONNECT to db  
  
    SQL1 = insert into completedWorkouts (Date, TotalTime) values  
    (CurrentDate, TotalTime IF INPUT)  
  
    DO SQL1  
  
    SQL2 = SELECT ID from completedWorkouts where Date =  
    CurrentDate  
  
    WorkoutID = DO SQL2  
  
    FOR i between 0 and length(data)  
        SQL3 = SELECT * from ExerciseBank where ExerciseName =  
        data[i][exerciseName]  
        ARRAY ExerciseData = DO SQL3  
        ExerciseID = ExerciseData[1]  
        ARRAY stats = array of all statistics  
        FOR i between 2 and length(number of possible  
        statistics)+2  
            IF ExerciseData[i] = False THEN  
                stats[i].delete  
            ELSE  
                DO NOTHING  
            ENDIF  
        ENDFOR  
  
        SQL5 = insert into CompletedExercises (WorkoutID,  
        ExerciseID, Relevant attributes) values (WorkoutID,  
        ExerciseID, data[i][statistics])  
  
        DO SQL5  
    ENDFOR  
  
    DELETE data  
    CLOSE activity  
    LAUNCH home  
  
ENDPROCEDURE
```



```
PROCEDURE del(ID)

    DELETE all objects of exerciseName

    DELETE from data internal array of exerciseName

ENDPROCEDURE
```

Each displayed activity will have a delete button next to it to aid usability of the application. The delete button calls this function with its own ID, which then deletes all objects associated with it (i.e. the exerciseName text object, statistics) and deletes the data from the internal array of inputs. This allows users to delete one input exercise without having to delete and re-input everything else.

The save procedure comprises of the main bulk of the code for this activity. When the user has finished inputting their data, they press the save button which calls this procedure. The procedure must then create a new workout in the workouts table, then get the ID of the new record to be used later. Then it loops, so that it can check in the exercise bank which statistics are applicable to that exercise, stores the relevant attribute names, and gets the exercise ID, so that for every set of data in the array, the procedure will add a new record to the exercises table entering data into the relevant statistics and null values into the others. Then the procedure deletes any stored data, closes the activity and reopens the home page.

Add new exercise

These are the activities associated with add new exercise activity.

```
PROCEDURE build()

    CREATE textObject (text="Add exercise to workout")

    CREATE inputObject (text="Search bar",
        onsearch=displayExercises("search",input)

    CREATE dropDownObject (text="Filter by body part",
        onclick=displayExercises("filter",input)

    displayExercises(all, all)

    CREATE buttonObject (text="Done", onclick=done())

ENDPROCEDURE
```

Text here

The build procedure is called on launch of the activity, to build the UI. It creates the objects needed, then calls the displayExercises procedure to display all exercises. The displayExercises procedure is also called from the search bar or drop down menu if the user inputs any preferences on the data they want displayed.

```
PROCEDURE done()  
    CLOSE activity  
    LAUNCH add new workout activity (data)  
ENDPROCEDURE
```

This procedure is called on click of the done button, and sends the user back to add new workout activity, passing all the input data to it.

```
PROCEDURE displayExercises(type, data)  
    //type corresponds to where input comes from  
    //if exercises are all displayed, or filtered by something  
    //what they're filtered by is in parameter data  
  
    CONNECT to db  
  
    IF type == "all" THEN  
        SQL = "Select * from ExerciseBank"  
    ELSEIF type == "search" THEN  
        SQL = "Select * from ExerciseBank where ExerciseName =  
        data"  
    ELSEIF type == "filter" THEN  
        SQL = "Select * from ExerciseBank where BodyPart = data"  
    ENDIF  
  
    exercises = DO SQL  
  
    FOR i between 0 and length(exercises)  
        CREATE textObject (text=exercises[i][1], onclick =  
        expand(exercises[i][1])  
    ENDFOR  
  
ENDPROCEDURE
```

The **displayExercises** procedure is responsible for displaying the exercises the

users may want to add to their workout, depending on any previous input. If it's called on launch of the activity, all exercises are displayed. If it's called from the search bar or drop down menu then the SQL is modified to query the exercises related to the users search and display the relevant exercises. The exercises are stored in the exercise variable then iterated through to create all the relevant text objects.

```
PROCEDURE add()  
    exerciseData = GET input from all fields  
    data.append(exerciseData)  
ENDPROCEDURE
```

The add button gets all the input data from the fields of that exercise, and adds it as an array to the data variable, which will store all input data while this activity is running, and be passed back to the add new workout activity when the done button is pressed.

```
PROCEDURE expand(exerciseName)  
    CONNECT to db  
    SQL = SELECT * from ExerciseBank where ExerciseName =  
    exerciseName  
    ARRAY ExerciseData = DO SQL  
    ARRAY stats = array of all statistics  
    FOR i between 2 and length(number of possible statistics)+2  
        IF ExerciseData[i] = False  
            stats[i].delete  
        ELSE  
            DO NOTHING  
        ENDIF  
    FOR i between 0 and length of stats  
        CREATE textObject (text = stats[i])  
    ENDFOR  
    CREATE buttonObject (text="ADD", onclick=add())  
ENDPROCEDURE
```

Once all the exercises have been displayed, they will need to expand to show

input boxes for the relevant statistics so that the user can input their workout stats. The procedure gets the data for that exercise from the exercise bank and displays an input box for each relevant stat. It also create a button ADD for when the user wants to save this data.

Edit Goal

```
PROCEDURE build()

    CREATE textObject (text="Muscle")

    CREATE textObject (text = "Kgs/lbs")

    ARRAY goals = ["Workouts per week", "Bodyweight", "Bodyfat",
    "Arms", "Legs", "Torso"]

    CONNECT to db

    SQL = select goals from personal details

    ARRAY currentValues = DO SQL

    FOR i between 0 and length of goals
        CREATE textObject (text=goals[i])
        CREATE inputObject (text=currentValues[i])
    ENDFOR

    CREATE buttonObject (text="SAVE", onclick=save())

    CREATE buttonObject (text="CANCEL", onclick=cancel())

ENDPROCEDURE
```

This procedure is called on launch of the activity- it creates text objects for the titles, and creates an array of the names of the goals. Then it connects to the DBMS, gets the current goals set, and iterates through the goals to create the text and input objects. It also creates the buttons at the bottom of the page to save or cancel the input.

```
PROCEDURE save()  
  
    ARRAY newValues = GET data from input boxes  
  
    CONNECT to db  
  
    SQL = insert into Personal Details (goals columns) values  
        (newValues)  
  
    DO SQL  
  
    CLOSE activity  
  
    LAUNCH home activity  
  
ENDPROCEDURE
```

The save procedure is called on click of the save button and gets all the data from the input boxes, updates the personal details table to include this information and exits the activity.

```
PROCEDURE cancel()  
  
    DELETE all input  
  
    CLOSE activity  
  
    LAUNCH home activity  
  
ENDPROCEDURE
```

This procedure just exits the application- it exists mostly to aid usability.

Validation

In terms of user input, there will be a handful of drop-down menus, and lots of buttons, which will require no validation as it will be restricted to correct input.

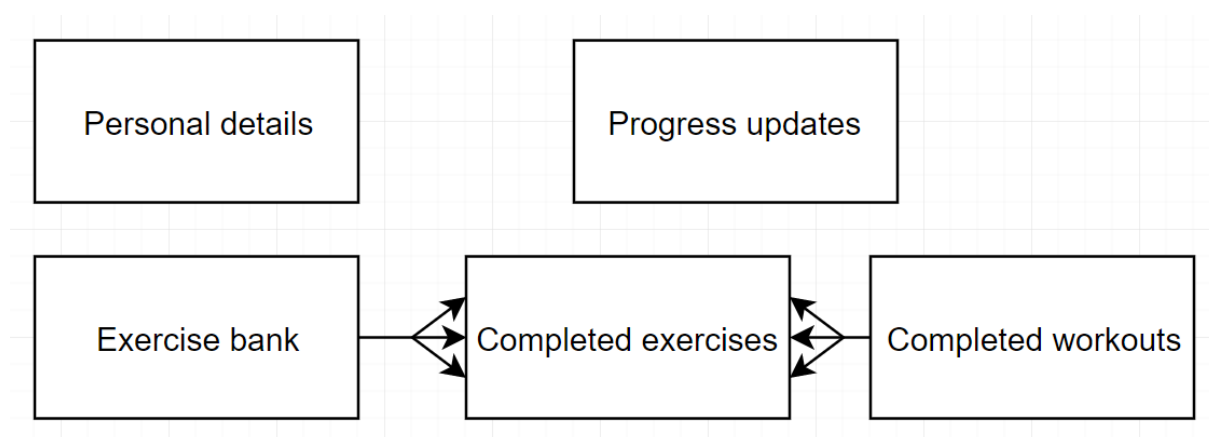
The input exercises will be limited to the exercises to include database initially and stored what type of data each exercise requires, e.g. sets, distance, weight etc. so that the only types of error possible is the users entering the wrong values in their data. Add some validation to inputs to ensure users don't enter unreasonable values like lifting 400kg. However, by restricting the user to pre-entered can ensure consistency between workouts.

Most numerical input can be checked to see if the range is correct, but other than that it will be very difficult to validate- entering the wrong input will result in errors in the progress page but allowing users to edit past workouts should allow them to change any wrong input themselves when they notice wrong results in the progress page.

The best enforcements of validation that I can make to ensure incorrect data will not crash the application, is to include a presence check on every input box and a type check on every database insert

Databases

Android studio allows for the implementation of data storage using SQLite so this will be my primary method of storing all the users' data. As I have restricted the scope of the project to make it manageable, I will store all the users' data on their device and therefore do not need to store and query many users' worth of data. The attributes I need to store for each user are as follows:



These are the entities I will be including in my database. 2 are simple stand-alone tables, while exercise bank and completed workouts both have a one-to-many relationship with completed exercises. This is because exercise bank and completed workouts would have a many-to-many relationship if this entity were omitted.

Personal Details

These are the basic details the user enters as part of their setup. They can be used to calculate stats such as calories burned and the user can update weight, bodyfat etc. as part of their progress to be measured. Weight and bodyfat are not stored in this table- in order to keep a record of previous measurements to

be used for calculating progress, I will need to keep many values so the repeat data will be stored in a separate database exclusively for these values.

Attribute	Data type and restrictions
Name	String (32 char max)
Height (optional)	Integer/float. 0<height<250 cm
Gender (optional)	Char (M/F/N) or null
Date of birth (optional)	Date or null. Must be younger than 100, older than 10.
Goal weight (optional)	Float or null 30<weight<250.
Goal bodyfat (optional)	Float or null. Restrict to safe values,
Goal weight lifted (optional)	2d array of float and body part or null
Goal workouts per week	Integer

Exercise Bank

This will be set up by me so there will be no need for any validation of this database. Upload a sample set of exercises into it as part of my development process but can be expanded to include hundreds and potentially allow the users to put their own custom exercises in it too (which would need validation).

Attribute	Data type and restrictions
ExerciseID	Integer (primary key)
Exercise name	String (32 char max)
Strength	Boolean (false = cardio)
Body area	String e.g. arms, legs, abs
Sets	Boolean (if exercise is measured in sets) or null
Reps	Boolean (if exercise is measured in reps) or null
Weight	Boolean (if exercise is measured in weight) or null
Distance	Boolean (if exercise is measured in distance) or null
Time	Boolean (if exercise is measured in time) or null
Speed	Boolean (if exercise is measured in speed) or null

This is all the statistics that will be tracking, although the design of the database should easily allow me to add more if my clients request any others,

Each statistic is stored as a Boolean- whether than value is relevant to this

exercise or not. On implementation, users will be allowed to enter values into the columns of the completed exercises table where there is a corresponding True in this table only.

Completed Workouts

There is a many-to-many relationship between workouts and exercises, normalised the database to include an extra table for completed exercises (which now restricts the relationships to one-to-many). The completed workouts table contains only the basic information about each workout. Exercises completed are stored in another table.

Attribute	Data type and restrictions
WorkoutID	Integer (primary key)
Day	Date
Total time	Time (could be same as exercise if workout consists of one cardio exercise) or null

Completed Exercises

There will be a one-to-many relationship between the completed workout table and completed exercise table and between the exercise bank and completed exercise table. Each workout is made up of several exercises of this table. The presence of data in each attribute of this table is determined by the Boolean values in the relevant columns of the exercise bank table.

If the clients decide that they want other data tracked than what has been designed, it will be easy to implement in the database. For example, if they want to track another type of stat for an exercise, it can be entered as an attribute in the exercise bank table with a Boolean value dictating when it is used and a corresponding attribute added to this table. This should not affect the normalisation of the database.

Attribute	Data type and restrictions
Primary key	Integer (primary key)
WorkoutID	Integer (foreign key)
ExerciseID	Integer (foreign key)
Sets	Integer
Reps	Array of length size(sets) of integers or null
Weight	Float. Restrict to reasonable weight.

Distance	Float. Restrict to reasonable distance.
Time	Time
Speed	Float. Do not restrict- different exercises will have different reasonable inputs.

The data included with each exercise is dependent on what is relevant- therefore it is possible for these attributes (except for primary and foreign keys) to have a null value. Whether they contain a value is dependent on the Boolean data in the relevant attribute in the exercise bank (i.e. if the exercise bank signals a certain exercise is measurable in sets, the application will allow the user to enter the amount of sets completed. None of this data is mandatory.)

Updates Of Personal Details/Progress Updates

Attributes	Data type and restrictions
ProgressID	Integer (primary key)
Weight	Integer/float or null
Bodyfat	Percentage or null
Date changed	Date

Expansion Of Project

If need to expand the project and have all the users' data stored on an external server, may have to modify these databases to allow for keeping each users' data separate. As this is not within the scope of this project, also designed the database with one user per device in mind.

Querying Database

Intend to query the database for several purposes- will need to get current details in order to display to the user their previously entered personal details and goals. Also need to get exercises from the exercise table to display to the user when adding their exercises to a workout and check if each attribute is relevant to that exercise.

Need to query the database to get data to calculate the users' progress. If, for example, by calculating their body fat change, will need to get the measurement and date of each measurement from the progress updates table, and either put each value into a graph or other visualisation, or calculate the total change and percentage change from the first and last pieces of data

Testing

Approach to testing:

To test my solution white box testing (where I test the internal structure of the App) and black box testing (where I only test the inputs and outputs).

Algorithm	How to test it
Signing in	Use values not in the database Use some values in the database and others not in the database (e.g. correct username but wrong password) Use values in the database that are not related together (using a username but the password is for a different account) Use correct values in the database Use code to try and re-write the checking program (e.g. use JavaScript code)
Signing up	Use correct values and datatypes Use correct datatypes but type different passwords for confirmation Use incorrect datatypes (e.g. use letters in telephone variable) Don't type anything in compulsory inputs Try to re-write code using JavaScript or another programming language
Any validation	Use erroneous data (wrong datatype) Use correct data (right data type) Use code to try and change the validation algorithm (e.g. JavaScript)

Example of test to be carried out

Component to be tested:	Accessibility settings			
Pre-requisites and dependencies	Have access to the App Mobile device Computer			
Description of test	Test data to be used (if required)	Expected outcome	Actual outcome	Comments and intended actions
App accessed on the mobile device		All pages of the App can be accessed on a mobile device		
App accessed on a computer		All pages of the App can be accessed on a computer		
Adjust settings for visually impaired users on the mobile device		Able to adjust settings to make the icons/links and text bigger and easier to use for visually impaired users		
Adjust settings for visually impaired users on the computer		Able to adjust settings to make the icons/links and text bigger and easier to use for visually impaired users		

Test Number	Test description	Expected result	Actual result	Successful?
1	Open navigation menu from button	Navigation menu opens	Navigation menu opens	Yes
2	Close navigation menu with back button	Navigation menu closes	Navigation menu closes	Yes
3	Open and close navigation menu by swiping	Navigation menu opens and closes	Navigation menu opens and closes	Yes
4	Navigate to home activity from home activity	Home activity launches	Home activity launches	Yes
5	As above, from track workout activity	Home activity launches	Home activity launches	Yes
6	As above, from view past workouts activity	Home activity launches	Home activity launches	Yes
7	As above, from see progress activity	Home activity launches	Home activity launches	Yes
8	As above, from edit goals activity	Home activity launches	Home activity launches	Yes
9	As above, from personal details activity	Home activity launches	Home activity launches	Yes
10	Navigation menu closes when back button pressed	Navigation menu closes and activity remains the same	Navigation menu closes and activity remains the same	Yes
11	Navigate to track workout activity	Track workout activity launches	Track workout activity launches	Yes
12	Navigate to view past workouts activity	View past workouts activity launches	View past workouts activity launches	Yes
13	Navigate to see progress activity	See progress activity launches	See progress activity launches	Yes
14	Navigate to edit goals activity	Edit goals activity launches	Edit goals activity launches	Yes
15	Navigate to personal details activity	Personal details activity launches	Personal details activity launches	Yes

Testing – Progress Updates

Test Number	Test description	Expected result	Actual result	Successful?
1	Add a more recent update	Most recent update is displayed on creation	Most recent update is displayed on creation	Yes
2	Add updates from several months and years	Most recent update is displayed on creation	Most recent update is displayed on creation	Yes
3	Set date 35th March	Display error message	Error message displayed	Yes
4	Set date 30th February	Display error message	Error message displayed	Yes
5	Set date 2010	Display error message	Error message displayed	Yes
6	Set weight =5	Display error message	Error message displayed	Yes
7	Set weight =500	Display error message	Error message displayed	Yes
8	Set bodyfat = 0	Display error message	Error message displayed	Yes
9	Set bodyfat = 95	Display error message	Error message displayed	Yes
10	Press cancel	All values are removed from input boxes	Input boxes are cleared	Yes
11	Press save	Data is added to entity	Data is added to entity	Yes
12-16	Leave each input box blank	Display error message	Error message displayed	Yes
17	Enter a letter	Not possible- keyboard restricts to numbers	N/A	Yes

As this activity was rigorously tested during development, no issues in the post-iteration testing.

Testing- Initial Entry

Test	Test description	Expected result	Actual result	Successful?
1-11	Leave each input box blank	Displays error message	Displays error message then saves	No
12	Enter a number in name box	Displays error message	Displays error message then saves	No
13	Enter G in gender	Display error message	Displays error message then saves	No
14	Set weight = 5	Display error message	Displays error message then saves	No
15	Set weight = 500	Display error message	Displays error message then saves	No
16	Set bodyfat = 0	Display error message	Displays error message then saves	No
17	Set bodyfat = 95	Display error message	Displays error message then saves	No
18	Press save with valid values	Data is added to entity	Data is added to entity	Yes
19	Enter goal workouts = 12	Displays error message	Displays error message then saves	No
20	Set date 42th March	Display error message	Displays error message then saves	No
21	Set date 30th February	Display error message	Displays error message then saves	No
22	Set date 200	Display error message	Displays error message then saves	No

As shown in the testing, the logic of my validation but the logic of the code is allowing the incorrect data to be saved. The same error has resulted in many failed tests and will need to be fixed before moving on to iteration 3. In addition, the error message on the phone is displaying on top of one of the input boxes due to the difference in screen size between that and the VM, so will need to readjust it.

Testing- Edit Goals

Test Number	Test description	Expected result	Actual result	Successful?
1	Enter 55 workouts per week	Displays error message	Displays error message	Yes
2	Enter weight = 12	Displays error message	Displays error message	Yes
3	Enter weight = 500	Displays error message	Displays error message	Yes
4	Enter bodyfat = 0	Displays error message	Displays error message	Yes
5	Enter bodyfat = 98	Displays error message	Displays error message	Yes
6	Press cancel button	Displays error message	Displays error message	Yes
7-9	Leave each box blank	Displays error message	Displays error message	Yes
10	Press save on correct data	Data is entered into entity	Data is entered into entity	Yes

Again, as this activity was rigorously tested during development, there are no more issues with the program.

Stakeholder Feedback

I asked a potential user a few questions after they had some time to use the application.

Q1) Do you feel the application is easy to navigate?

Yes- the swipe menu is very similar to most applications so you don't have to learn something different to use the app. And everything is the same sort of colour scheme so it's obvious where buttons and input boxes are.

Q2) Have you experienced any problems using the application?

The first version you gave me couldn't navigate away from the first screen, but after you fixed that, it's all worked fine. But lots of parts of the application are missing- entering workout data is the most important part and you haven't built that yet so there's loads of things I can't do.

Q3) Are there any other comments you have about the application at this stage?

I'm sure I've used apps before which can display a calendar to let you choose a date- that makes it much easier. Would it be possible to change that?

This survey is encouraging- all the completed parts of the application are working as intended and no unexpected errors have escaped testing. The issue of missing parts of the application isn't a problem as these will be developed in the next iteration.