

Project Proposal

Project Title

Progressive overload mobile application

Module Number

U08096

Student Number

09092543

Name

Lee Hudson

Supervisor

David Lightfoot

Date

6th October 2013

Contents

Introduction and rationale:.....	3
Background Review:.....	5
Methodology and Resources:	6
Project Plan:	8
References and bibliography:	9

Introduction and rationale:

Technology can be used to solve many problems in day to day life. The aim of this project is to apply mobile technologies to solve the problems inherent in a weight lifting method known as progressive overload.

Progressive overload is a weight lifting technique that is commonly used to improve performance and weight gain. The principle of progressive overload is simple, one must strive to increase the load placed on one's body in order to stimulate muscle growth (Goulet,2004). In practice this means having a set workout and increasing each exercise in some way, be it number of reps, weight or time between sets. The aforementioned increases are quantative and easy to represent numerically but technique is also important, this will be hard to record.

The above technique of progressive overload can be hard to implement effectively in a gym environment, it requires a lot of note taking and can be time consuming in itself. It can sometimes be hard to remember where you left off and it can be hard to determine if you really are improving. This is where the advantages of mobile computing can assist.

The above activity is essentially record keeping and data analysis. A mobile application can drastically improve the practicality of progressive overload.

Aim:

The aim of this project is to create a mobile application to address the issues of progressive overload. The finished application will allow users to enter in their performance for a given work out and to compare their performance as a function of time. It will tell the user what benchmark they reached on their previous workout so they can better it. The exact requirements are detailed in the requirements specification.

With the above aim in mind the following objectives can be set:

1. Complete literature research and planning
2. Decide which platform would be most suitable for this application
3. Compose detailed requirement specification
4. Design the class structure of the application
5. Design and normalise a suitable database
6. Design user interfaces for a both physical orientations
7. Design user interfaces for large and small screens
8. Compose and submit interim report
9. Implement the application
10. Create test schedule for the application
11. Test the application
12. Publish the application on the app market
13. Compose and submit final report

In terms of implementing the above on a mobile device there are the following key issues to be considered:

1. What platform should this be developed on?
 - a. The main choices are Android, IOS and windows mobile.
2. Battery usage
 - a. This is not normally a consideration when programming for a static device but for a mobile device this needs to be considered. Sometimes good programming practices have to be re-considered in favour of efficiency.
3. Portability
 - a. If this application is developed on IOS or android it must be able to be used on a variety of screen sizes and resolutions. Some applications on the market do not consider this and it is evident when you run the application. Android and IOS have the facility to display certain GUIs depending on the size of the screen. In Android this is called a fragment. The application will appear different from device to device.

4. Non-volatile data storage

- a. The data entered into the application must be retained on power off. The data need to either be written to a file or stored in a database. Both android and IOS have SQL interfaces, a database such as SQL would be favoured over a text file as data consistency is important. When implementing a database it is important to ensure the database has been normalised to at least Boyce-Codd normal form.

5. Gui orientation

- a. Most mobile devices allow both portrait and landscape views, this has to be considered.

6. Graphical representation of performance

- a. It would be useful to represent the users performance in the form of a graph, this will most likely need to be programmed from scratch and will have features such as auto scaling.

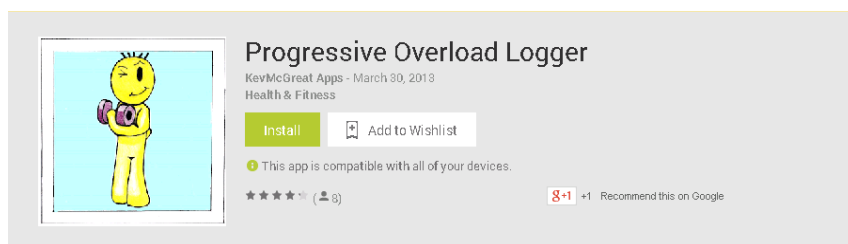
7. Innovative

- a. The application must be easy to use and surprise the user with features such as customisation and nice graphics.

Originality:

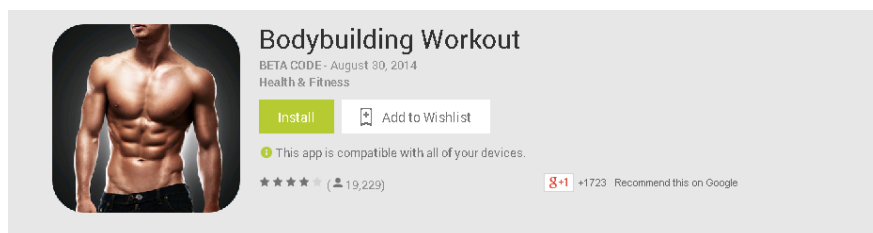
Currently there are a few applications available for workouts but none that address the issues of progressive overload effectively. The following applications are available for android:

Progressive overload logger - KevMcGreat Apps



This application addresses the needs of progressive overload to a certain extent by allowing the user to log their performance, however, it is hard to use. It requires too many user interactions for a single operation which is unsuitable for use in a gym.

Bodybuilding workout - Beta Code



This application does nearly everything needed for progressive overload but doesn't do it in a robust way. This app is full of bugs and is not intuitive. It also doesn't allow you to view your progress in a graphical way.

The above applications are the only ones to be found that directly target progressive overload. Neither of these applications fully address the needs of a progressive overload so the market would benefit from an application that does specifically target progressive overload.

The project outlined above is a suitable project for a software engineering degree as it contains many aspects of software engineering. It will be very code intensive, make use of a database and graphical user interfaces. Good software engineering practices will be observed where possible.

Background Review:

There are two main subjects to research, the technical aspect and the topic area. From a technical point of view some research is needed into mobile applications and programming. From a topical point of view research is needed into the method of progressive overload. For this literature search the IEEE library was used along with a generic internet search engine.

Article 1: **How to choose the best platform for your app**

Sherman,E.(2012) *How to choose the best platform for your app*. [Online] 08.10.12. Available from - <http://www.inc.com/erik-sherman/avoiding-the-single-mobile-platform-trap.html> [Accessed: 03.10.14]

The above article is a personal blog posted at the referenced URL. The article details things that must be considered when choosing a platform such as the possible risks associated with developing with apple, specifically the app store and the control apple asserts over it. It also looks at the risks with android, specifically the challenge of developing for many devices at once. It doesn't appear to have a bias to any particular platform and does highlight things that must be considered. The article is very brief but it does make the reader aware of some interesting considerations. It touches on the technical limitations of platforms and limitations from an infrastructure point of view, specifically the apple market place.

The author Erik Sherman is a journalist and also has experience as a communications consultant. He is also the author or co-author of 10 books, some of which are technical.

This article is still relevant as it's only 2 years old. Technology moves fast but these key issues are still present so this article is still applicable. This article will be useful when it comes to deciding which platform to develop on.

Article 2: **The ten rules of progressive overload**

Bret(2013) *The ten rules of progressive overload*. [Online] 26.02.13. Available from - <http://bretcontreras.com/progressive-overload> [Accessed: 03.10.14]

This article is also a blog but it is focused on the subject material rather than the technical aspect of an application. It does a good job of explaining the principle of progressive overload but as with a lot of weight training articles it is subjective. It uses examples of how well the technique works and is useful in that it underlines all the basic principles of progressive overload. This article will be very useful when designing this application.

The Author is just a personal blogger who has a passionate interest into the subject. There are so many opinions on this subject it is hard to verify the credibility of this article or the author but it must be said that this article is methodical and logical.

This article is a year old so it is definitely relevant. The subject of weight training has been around for a very long time with very few developments in technique so this will probably remain relevant for some time to come. This article will be useful to ensure the mobile application meets the requirements of progressive overload and to make sure the concept of progressive overload is fully understood.

In addition to the above articles many other articles have been seen to be relevant and so have been used. These have been detailed in the references.

Methodology and Resources:

It has been decided that the platform this application will be developed on is android. The reason for this is due to the extra costs associated with developing for apple devices. Although the one of the most popular app categories on the apple market is fitness (Rahul Varshneya,2013) the ease of developing and publishing on the android market makes android the best choice for this application. Also the diverse hardware associated with android adds an interesting challenge. The variety of screen sizes and resolutions will pose some technical challenges.

With the platform set the development can begin. This application will be developed using Eclipse as an android plugin has been developed for this IDE. I am familiar with both Eclipse and the Android plugin. For testing purposes both a large screen and a small screen device will be needed. These are available and will be a Samsung 10.1 tablet and a Nexus 5 phone. The software development methodology for this project will be the waterfall method. The exact steps that will be taken are detailed below but follow the standard specification, design, implementation and verification process:

1. Refine and finalise requirement specification
2. Decide on classes, interaction and rough functionality
3. Compose UML diagram of system
4. Compose and normalise database
5. Design user interface for landscape, vertical, high res and low res screens.
6. Implementation
 - a. Create android activities
 - b. Implement GUIs for each activity
 - c. Implement communication between activities
 - d. Implement database
 - e. Implement custom graph view
 - f. Implement timer option
 - g. Implement any audio
7. Testing
 - a. Create a test schedule
 - b. Test application

The system can be tested in a software functionality sense by choosing specific inputs that challenge the application with a prior knowledge of the workings of the system (white box testing). Both a beta release and the final release will be given to the public (selected gym users) to ensure the application does address the demands of progressive overload in the real world.

With the above project there are inherent risks. Some of the technical aspects of this application are unknown and while should be possible may be more challenging than anticipated. The most unknown aspect of this application is the use of different GUIs for screen sizes and displaying the application differently on a tablet compared to a phone. If this does prove too challenging the GUIs can be adapted to be good on both screen sizes, this is not ideal and is highly unlikely. The other risk is the project taking too long. In order to mitigate this there has been some spare time added to the project plan in the design, implementation and test stages of the project. A first pass over the possible requirements yields the below requirement specification:

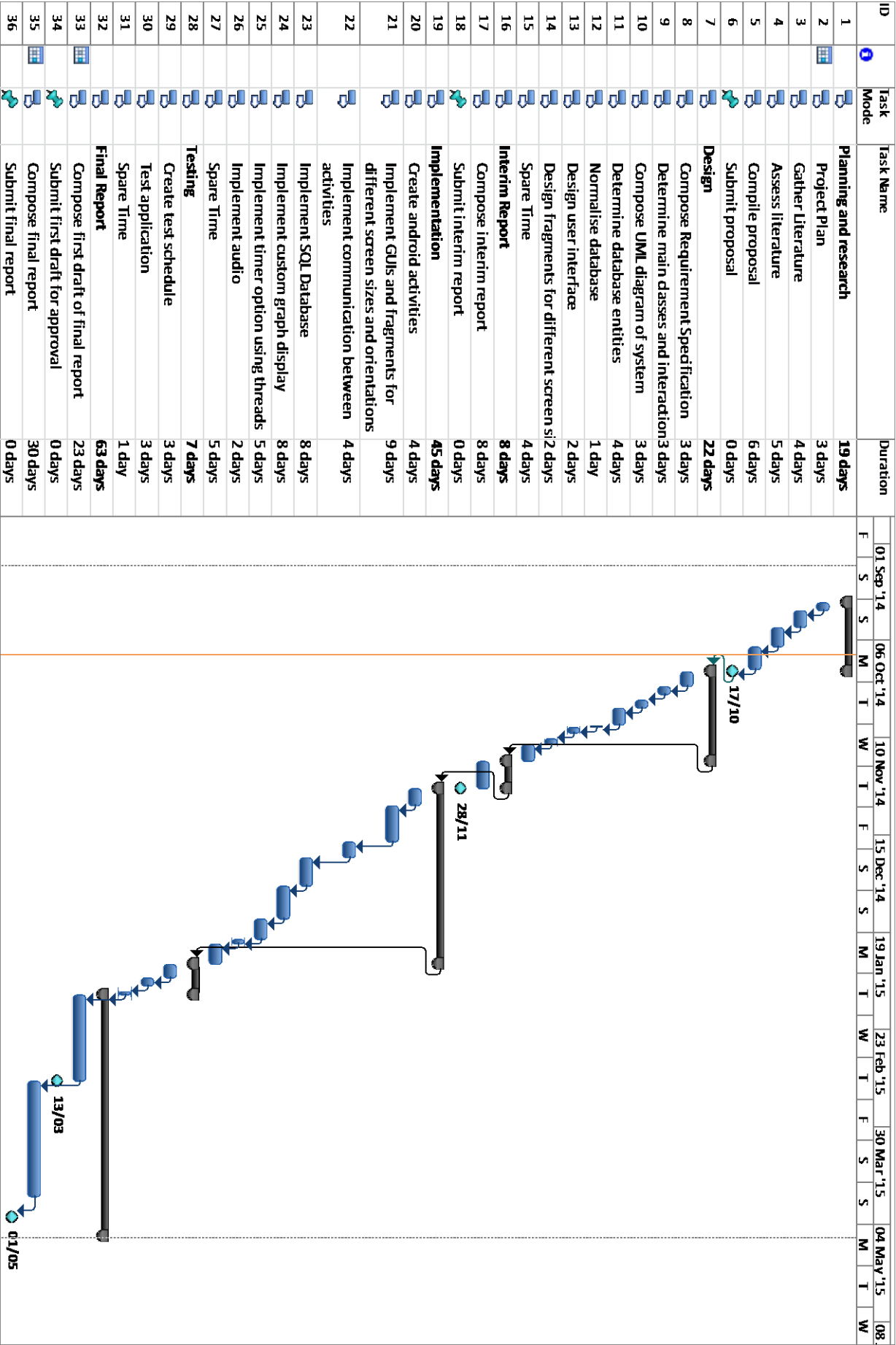
Functional requirements:

1. Allow the user to create a workout .
2. Allow the user to create an exercise.
3. Allow the user to add exercises to a work out.
4. Allow the user to remove a workout.
5. Allow the user to remove an exercise.
6. Allow the user to record the performance each time an exercise is performed.
7. Allow the user to see the performance from their last workout to allow them to clearly see what they have to better.
8. Give the user an audible alarm to signify the end of a rest period.
9. Allow the user to see how their performance has increased in a graphical way.

Non-functional requirements:

1. The application should respond to an input within 500ms.
2. The application must retain data on power off.
3. The data stored by the application must be consistent.
4. The application must work in both landscape and portrait orientations.
5. The applications must work on screen sizes from 640x480 to 1920x1080.

Project Plan:



References and bibliography:

- I. Sherman,E.(2012) *How to choose the best platform for your app*. [Online] 08.10.12. Available from - <http://www.inc.com/erik-sherman/avoiding-the-single-mobile-platform-trap.html> [Accessed: 03.10.14]
- II. Bret(2013) *The ten rules of progressive overload*. [Online] 26.02.13. Available from - <http://bretcontreras.com/progressive-overload> [Accessed: 03.10.14]
- III. Vogel,L.(2014) *Multi-pane development in android with fragments - Tutorial* [Online] 29.04.14. Available from - <http://www.vogella.com/tutorials/AndroidFragments/article.html> [Accessed: 03.10.14]
- IV. Varshneya,R.(2013) *iOS or Android? Choosing the best platform for your application*. [Online] 17.07.13. Available from - <http://www.entrepreneur.com/article/227426> [Accessed: 03.10.14]
- V. Goulet,C.(2004) *Progressive overload: The concept you must know to grow!* [Online] 19.10.04. Available from - <http://www.bodybuilding.com/fun/goulet11.htm> [Accessed: 03.10.14]
- VI. Vogel,L.(2014) *Android SQLite database and content provider - Tutorial* . [Online] 19.08.14. Available from - <http://www.vogella.com/tutorials/AndroidSQLite/article.html> [Accessed: 03.10.14]