**PROJECT PROPOSAL**

**Prepared for: Marcus Winter / James Burton**

**Module: CI360 - Mobile Application Development**

**Prepared by: Alena Ruprecht**

**Date: 18th January 2012**

**Student #: 09800502**

Jog DJ

- a music app for runners -



Table of Contents

1) Elevator pitch 3

2) Rationale 3

3) Application 3

4) Use case 4

5) Technical overview 5

Positioning and Speed 5

Music 5

Training Plan 6

6) Technical challenges/risks 7

7) UI design draft 8

8) Work plan 12

9) Ethical considerations 14

10) Resources 14

References 16

# 1) Elevator pitch

The application will play a song from the runner’s music library based on the running speed. The runner’s speed can be measured using GPS, which will be output visually on a Google Map or the accelerometer of the phone. It will calculate the bpm (beats per minute) of songs in the music library. The runner’s speed will then be matched against the bpm of a song and this will determine the song to play.

An additional feature will let the user plan his training session by setting various phases (warm-up, main run, cool-down) to a time limit or distance. A voice-over will announce transitions between phases during the run.

# 2) Rationale

The target audience are people that run or walk for exercise. Currently there are no Android apps available that match a song to a runner’s speed. This is surprising considering Android’s market share of 47%.

Runners have to manually choose music that matches their speed. This is tiresome and interrupts the running flow. The runner has to stop and choose a new song if he/she slows down or speeds up. This is important for runners that are training as training usually consists of warm-up (slower speed), the actual run (faster speed) and a cool-down (slower speed). That means the user has to manually set the playlist to contain 3 different sets of songs in the right order for the right length of their training session.

The training plan feature will help by announcing each stage a runner is in. This will save the runner having to look at the time on his phone and thus not paying attention to the environment, which can lead to accidents. This danger could be removed by having an audible indicator such as a voice-over letting the runner know when to slow down/speed up.

# 3) Application

Jog DJ will address the above issues by making an app available that automatically plays a song based on the runner’s speed. The runner will just have to start the app and it will start calculating the runners’ speed by using GPS data or the built-in accelerometer. The resulting speed will be mapped to a song with the associated bpm in the runner’s music library. That means runners just have to start the app and it will play songs until it is stopped. It removes the need to manually choose songs and having to stop during training. The songs will automatically transition to a different song when the runner slows down/speeds up with a fade in/fade out effect like a DJ.

The training plan feature will consist of a voice announcing the stages of the run. There will be a screen that allows the user to make a selection of various phases based on time or distance. This way the runner can for example input that he/she wants to run the warm-up phase for 5 minutes, the actual training phase for 20 minutes and the cool down phase for 5 minutes.

The runner will also be able to save his runs and the associated metadata like speed, distance and time to compare the runs to track progress.

There are currently no Android apps on the market that match songs in a user’s library to the running speed. There are apps like RunKeeper[[1]](#endnote-1) and MapMyRun[[2]](#endnote-2) that allow users to know how far and fast they have run via GPS. This can then be viewed on a map and saved to the runner’s account that has to be set up prior or shared on social networking platforms.

There are also a couple of apps in the app store for iOS that have a similar approach to matching music to speed. The app SynchStep[[3]](#endnote-3) allows users to just push start on the app and get a song instantly matched to their speed using the built-in accelerometer. This app relies on a server that calculates the bpm of a song and plays the song back to the user. The user can also download desktop software to determine the bpm for each song in his/her library and import them to the phone. This seems quite a complicated process.

The second app in the app store is called Cadence[[4]](#endnote-4) and allows the user to set the bpm manually. The user gets a manual that shows how to count how fast he/she is running by measuring the steps against a digital metronome. The user can then set the bpm and songs that match that bpm will play.

The Jog DJ will aim to simplify the music experience while running. The runner will push start and the app will automatically determine which song to play. This will make it easier to use than Cadence and it will exclusively access music on the user’s smartphone, which will remove the need for access to a server (like SynchStep). This is beneficial for users that have limited data plans. Instead the information of bpm for each song will be stored locally with the user.

# 4) Use case

Actor: Runner/Walker

A runner gets ready for his run. He takes his phone with him to be able to listen to music and track his distance/time while running.

He unlocks his phone and enables the GPS function of the smartphone if he is using GPS. Then he opens Jog DJ. The runner can opt in to use a training plan. He has to select if he wants to have a warm-up, a main run and a cool-down as well as how long or far each phase should be.

He also has to input if he wants to use the GPS or the Pedometer option. That’s all from the user’s perspective and he can push “Start”. The phone will be put into a running armband (works best with the pedometer) or any other pocket on the runner’s body. As soon as the runner starts jogging, the app will determine his speed based on the distance covered or the accelerometer. The pace will then be matched to a song from the runner’s music library.

The song will fade in and play while the runner keeps a consistent speed. If the song ends, another song with the same bpm will be played. If the runner speeds up/slows down, the current song will fade out and another song with the new bpm will fade in.

When the runner finishes his run, he can stop the app and save his running data of time, speed and distance to his account to track his progress.

# 5) Technical overview

The Jog DJ application will consist of various complex parts. There are three main parts that can be separated and looked at.

## Positioning and Speed

The application will be able to determine the position and/or speed of the runner using one of the following two methods.

**GPS**

Nowadays all smartphones are equipped with a GPS. This positioning information can be used to plot the position of the runner on a map. For this application Google Maps will be used. The GPS will also let the application calculate the running speed. In terms of implementation this feature will rely on the LocationManager class provided by the Android SDK.

**Accelerometer & Directional Compass**

As an alternative it is possible to use the built-in accelerometer to calculate the speed of the runner. The accelerometer itself is actually not enough to calculate the speed and it needs to be used in conjunction with the directional compass. The accelerometer can only tell us if the phone is moving while the directional compass can tell us whether the phone is speeding-up or slowing-down. The Sensor and SensorManager classes will be used to implement this feature.

## Music

**Music Library & Playback**

The application will have to access the music library of the phone. The MediaPlayer class will help to access this functionality. It will make it easy to play/pause/stop songs.

**BPM Calculation**

This feature is not supported natively by Android, however EchoNest might help to address this issue. It is not yet clear to what extent this API will be used but will be solidified during the requirements phase completing on March 15th. The calculation is done for each song in a user’s library. Performing this calculation for each song in the library will probably take some time and therefore we can’t afford to do it when the user opens the application. There are various solutions to mitigate this problem. First this calculation will be done only once per song and the result will be stored in a local database using the Data Storage class. The first time the runner uses the application it might take a while to find a song matching his speed. This is because at that point there are no songs processed in the database. The application will be constantly processing songs in the background and the next time the user opens the app songs will be available immediately. This process will run until there are no more songs to process in the music library and start up again if new songs are added.

## Training Plan

**Options**

This feature will allow the user to set a training plan for the run. The user can choose from a selection of options. The implementation is quite simple. If the user selects a checkbox for each running phase, a dropdown box will become available that allows the user to choose the duration or distance for each phase.

**Voice-over**

The voice-over feature will announce each part of the training stage a runner is in. This will keep the user updated without having to check the time or distance on the screen. The Text-to-Speech Engine will address this part of development.

**Music**

* Music Library &

Playback

* BPM calculation

Jog DJ communicates with all 3 components and processes the information in the app itself.

**Jog DJ**

**Training Plan**

* Options
* Voice-Over

**Speed & Positioning**

* GPS
* Accelerometer &

Directional Compass

GPS

The training plan feature will have to communicate with the Speed & Positioning feature to calculate how far/long a user has been running.

# 

# 6) Technical challenges/risks

Various challenges have to be faced due to the complexity of the application. The inexperience of the developer in Android development will add to the difficulty. However the developer is going to use the extensive information of the Android documentation as well as various other sites including stackoverflow.com designed to help aspiring and professional programmers.

It is to be expected that the matching functionality between the bpm of a song and the speed of a runner will bear significant difficulty. As of yet, it is not established how to ensure the runner’s speed will play the correct song with the right bpm. The developer will establish this in due course but thinks that the prevalent approach might be the manual match of a certain bpm to a certain speed. This doesn’t seem the best approach, but this will be decided during the requirements & analysis phase of this sub task.

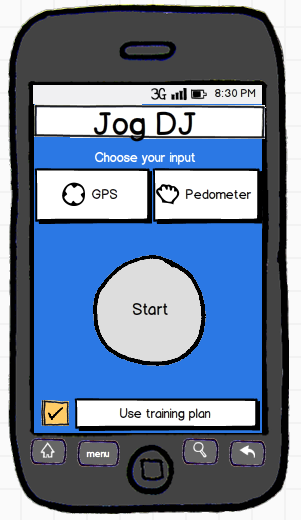
Another issue could arise through the use of the accelerometer and the directional compass. These will have to be programmed to act as a pedometer that can register change of speed. However during the current research it seemed that while it was an easy task to register speeding up, it seems quite a difficult task to master the accelerometer to register slowing down as well.

Risks that can be identified are the potential small timeframe in which this project has to be carried out.

A further risk that might arise is that the inexperience of the developer will add to the weight of the code not being “mobile” enough. This will have to be addressed at a later stage, however the developer will make an effort on keeping the code light and the app responsive. Reason for this concern is the calculations that have to be made to determine a songs bpm as well as having this information readily available for Jog DJ to access.

# 7) UI design draft

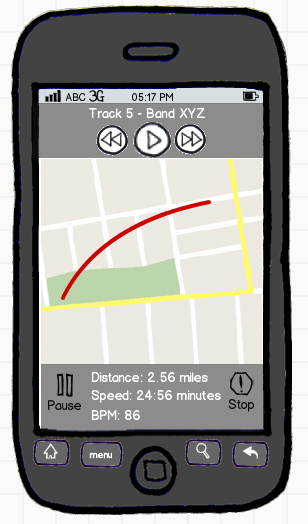
Start Screen



Start button becomes available when the user has picked an input.

User has to tick if they want to use a training plan. If ticked, a button will lead to another screen allowing the user to choose how long the training session should be.

User has to pick input method between GPS (needs to be enabled prior) or the pedometer function, which uses the accelerometer to determine speed.

****

Metadata is showing the runners distance, speed and BPM for the song playing.

There is a pause and stop button for the user to pause/stop the app.

If runner is using GPS, a map will show the runners path.

Track is playing at the top with manual controls.

If the user has picked a training plan, the main app will show the phase the runner is in.

Main App Screen – GPS input

Run path

# Macintosh HD:Users:alena:Desktop:Screen Shot 2012-01-16 at 17.29.32.png

Main App Screen – Pedometer input

# Macintosh HD:Users:alena:Desktop:Screen Shot 2012-01-16 at 17.56.20.png

Main difference to GPS is that bpm is shown as main feature as well as the time the app has been running.

Training Plan Screen



Allows the user to select if and how much time he wants to spend on each part.

# 8) Work plan

The developer is proposing an iterative approach to this project due to the nature of the project having to be tested in a real-life environment.

Deployment

Initial Planning

The initial planning phase has been done with this proposal. Extensive research into the separate components of this application has also been undertaken.

The project has been divided into several sub tasks and will be developed accordingly. The iterative development plan will approach each of the following phases:

* Access to the GPS system
* Calculating distance by time with the GPS
* Displaying GPS output on Google Maps
* Access to the accelerometer and directional compass to output motion into footsteps
* Access to the built-in music library
* Calculating bpm for a song
* Matching bpm to running speed
* Accessing a song and playing it back
* Registering new input on speed and change of song
* Voice-over features for the training plan

Once a planning phase has been concluded, requirements will be set for each sub task. The design and analysis will only approach the sub task and this sub task only. Once this has been done satisfactorily, the task will be implemented and tested. As the GPS and accelerometer are the first tasks to be implemented, they will have to be tested outdoors to determine if they correctly in- and output the information.

The test results have to be evaluated and if satisfactory, the next sub task can be started.

The first sub task is the implementation of the GPS positioning and it’s output to a Google Map. The task can take up to two weeks, however due to the difficulty of this task, it is anticipated that it might take longer to complete thus taking us into the timeframe of the next sub task. Should this happen, the implementation of the accelerometer feature could be omitted. The application can function solely with GPS and the accelerometer is just an extra feature.

Each evaluation will add to the write-up of the project. This is to ensure consistency across all stages rather than having to write up from memory at the end of the project. There will be time dedicated to the task of polishing the report and adding a conclusion.

The proposed timeline is shown in the below diagram:



Each of the above stages includes all phases of

* Planning
* Requirements gathering
* Analysis
* Design
* Implementation
* Testing
* Evaluation
* Additional iterations if needed

at the end of each task. This is to ensure consistency and help adding features only when the current stage has successfully been completed.

The third last milestone on the timeline is dedicated to ensuring that the delivered product is at the highest possible standard by eradicating bugs through extensive testing in the field.

The second last milestone is to ensure that the project is delivered with a report that rivals equally in accuracy and detail. The last milestone is considered to be the actual hand-in on May 24th, 2012.

# 9) Ethical considerations

Jog DJ will be build as a stand-alone application. For the scope of this project, there will be no retrieval of data from servers. Regarding privacy, the data will be held locally and holds no significant importance to the runner apart from being able to track progress. Data protection as well as health and safety don’t have any part in this application due to the nature of the system.

Moral implications couldn’t be found either.

Involvement of the user on the other hand will have to be quite significant in the requirements analysis and testing stages. The requirements analysis will have to be quite thorough to determine the exact use of a smartphone when running. The runner will have to demonstrate the current use, the strains of having to manually adjust music during a run and what an app should be able to do “hands-free”. This will be thoroughly investigated and the app infrastructure will be based on this information.

At a later stage, the app will have to be tested in the same environment. It will have to be determined if the GPS as well as the accelerometer are working. As the developer is an avid runner, the developer will most likely carry out testing. However a few other test subjects will be asked to involve themselves to receive some objective feedback. The developer will ensure that the testers will not put themselves in any danger testing the app.

# 10) Resources

The project is going to need access to a smartphone running at least Android 2.2. The development environment will be testing the app in 2.2 and 2.3, as currently the largest percentage of Android phones are running Android 2.2 (30.4%) and Android 2.3 (55.5%). It will be ensured that there is compatibility with devices of even higher running Android releases (3.0 and 4.0).

The development environment is the Eclipse platform with Android SDK installed. Access to various API’s will be needed. The GPS tracking software will have to have access to Google Maps to display the map in the app should the user wish to use GPS for the run.

The music matching software will have to be further investigated but at the current stand might be EchoNest.

For the scope of this project there will be no connection to a server, however this could be implemented in a future commercial release allowing the user to upload his running stats to a website or sharing the stats on social networking platforms.

# References

The following books and resources will and have been used to conduct research for this proposal and the on-going project:

Books

Ed Burnette (2009). *Hello, Android: Introducing Google's Mobile Development Platform*. Pragmatic Bookshelf.

Websites

Jay Yarow. 2012. *Android's Market Share Collapses As Apple Surges Thanks To The iPhone 4S - Business Insider*. [ONLINE] Available at:<http://articles.businessinsider.com/2012-01-09/tech/30606530_1_new-iphone-android-sales-verizon-customers>. [Accessed 17 January 2012].

Jim Scarlett. 2003. *Enhancing the Performance of Pedometers Using a Single Accelerometer*. [ONLINE] Available at: <http://www.analog.com/library/analogDialogue/archives/41-03/pedometer.html>. [Accessed 10 January 12].

Jonathan Foote. 2001. *The Beat Spectrum: a New Approach To Rhythm Analysis*. [ONLINE] Available at: <http://www.rotorbrain.com/foote/papers/icme2001/icmehtml.htm>. [Accessed 13 January 12].

Unknown author. 2011. *The Echo Nest Remix API*. [ONLINE] Available at: <http://code.google.com/p/echo-nest-remix/>. [Accessed 15 January 12].

Various discussions on stackoverflow.com have been used to research material regarding GPS, the accelerometer, the directional compass and access to the music library.

API usage

http://developer.echonest.com/

http://code.google.com/apis/maps/index.html

Android Documentation

Accelerometer: http://developer.android.com/reference/android/hardware/Sensor.html

GPS: http://developer.android.com/reference/android/location/LocationManager.html

MediaStore: <http://developer.android.com/reference/android/provider/MediaStore.html>

SensorManager (Compass): http://developer.android.com/reference/android/hardware/SensorManager.html

Text-to-Speech: http://developer.android.com/resources/articles/tts.html

Competitor Applications

1. http://runkeeper.com/ [↑](#endnote-ref-1)
2. http://www.mapmyrun.com/ [↑](#endnote-ref-2)
3. http://synchstep.com/ [↑](#endnote-ref-3)
4. http://cadenceapp.com/ [↑](#endnote-ref-4)