# YYST

## YESTERDAY YOU SAID TOMORROW

Project Management Plan

VERSION 1.0.0

# Contents



Kevin Hoh, George Oscar Millington, Joon Xian Ng, Tae Hyun Jin
TEAM 16[1]

20/10/2015

# Introduction and Purpose

Yesterday you said tomorrow is a running application designed to enhance the running and exercising experience by providing the ability to map and store runs, analyse statistics and provide motivation to users. The target audience is anyone who is looking to improve their fitness as it is a free web application and due to its simple design and nature, is even suitable for those who are less tech savvy making usability and accessibility a key component in design.

The document aims to provide an overview for the development of the application, detailing the various levels of management involved in the developer process. The document will cover the scope, communications, human resources management and rules and methods implemented to maintain quality and minimise risk.

# Summary of Project

## Assumptions

- Users will have access to the latest version of 'Google Chrome'
- Users will have and maintain an internet connection via 3G/4G or Wi-Fi
- The device that the application is run on will have GPS and accelerometer functionalities
- Users understand English

## Client/Users

The app is designed for people of all fitness levels and smartphone owners who do not necessarily need to have experience with software. Specifically the app targets young teens to adults in an attempt to reduce the number of overweight children in today's society by promoting a healthier lifestyle.

The stakeholders in the project are the clients at 'Yesterday You Said Tomorrow' who have requestsed the product's development.

## Deliverables

The end goal is a web application that contains the following:

- Main Menu Page

  This page allows the user to view the instruction manual, start recording a run and load up a previous run.

- Record Run Page

  The User can begin recording a new run on this page, during the recording process a timer is displayed as well as coordinates from the GPS and the signal strength as well as the distance covered. In the background are calculations for the calories burned and the average speed. Additionally, the beginning gnad end points as well as the run path are also displayed on this screen followed by a stop recording button that prompts the user to save the run and give it a name.

- Archive Page

  This page displays a list of previous runs and their corresponding title and date. Selecting any run will take the user to that runs individual ‘history page’. There is also a drop down menu that can be accessed to sort the archive page by name, distance and date.

- History Page

  The user can review their past run in detail including the path, average speed, total time, total distance, calories burned and can edit or remove the run.

# Scope

## Approach/Methodology

The methodology applied to this project will mainly be the agile method of development as it means that there is a larger room for error and it is possible to get feedback from the client to point the development in the right direction if needed.

An example of the agile method can already be found in this project in the usage of the functional specifications document, as the design has somewhat differed from then to now.

## Tools and Methods

As previously mentioned Asana will be used to gauge progress and delegate tasks through the applications development and to modify the specifications to meet time constraints.

For documentation, Google Docs will be used to create collaborative formal reports.

Github was also used allowing different members to contribute by pulling requests into necessary branches and also making it easier to revert mistakes as a changelog is created.

## Timelines

| Week | Task | Due |
|---|---|---|
| 6 | • Work on User Stories.<br>• Draw activity diagrams. | 3/9(Week 6) |
| 7 | • Draw wireframes<br>• Write storyboards<br>• Complete functional specifications document. | 13/9(Week 7) |
| 8 | • Submit the functional specifications document.<br>• Delegate work for coding of app (from Activity Diagrams) and incorporate into asana | 14/9(Week 8) |
| 9 | • Receive feedback on submission(Client approval)<br>• Take note of any modifications and feedback from client.<br>• Implement the new additions to the app. | 24/9(Week 9) |
| Break | • Group meetings | 2/10 |

| | | |
|---|---|---|
| | • Finish coding the app | |
| 10 | • Plan client presentation<br>• Delegate work<br>• Schedule practice presentations and meetings | 8/10 |
| 11 | • Outlining of user interface considerations from client<br>• Completion of Assignment 2B | 10/10 |
| 12 | • Submission of Assignment 2B<br>• Client presentation | 13/10 |

## Personnel/HR management

**An overview of responsibilities for each member are as follows:**

George Oscar Millington – Acted as the team leader and delegated the tasks on Asana while working on the main functionality of the application and was the leader in meetings

Joon Xian Ng - Familiarised himself with MDL and worked on the UI and the instructions manual and assisted in writing the documentation, also was the scribe in meetings

Kevin Hoh – Assisted with the coding of the main application and worked on the google maps API and assisted in the documentation, was the minute taker

Tae Hyun Jin – Worked on some of the more basic features in the code and documentation required for the project, was the time taker in meetings

## Communications management

Team meetings were held every Thursday in practical classes and in irregular meetings outside of class following an agenda.

The main communication tool used was Facebook Messenger for its easy usage.

Ensuring that code was accompanied with comments where necessary meant that there was minimal confusion in between transferring of code to other members

Asana reminded members when their deadline for the task was coming up and the comments sections for each task were used to assist group members in completing their tasks.

# Quality management

The expected functionality of the application was compared to the current outcomes to determine whether the results were satisfactory to fulfil client demands and user expectations.

| **Feature** | **Expected Quality of Outcomes** | **Quality Test** |
|---|---|---|
| Recording runs | ● Initial position and final position is displayed clearly in the map.<br>● When it is recording, the route which the user is running should display on the map as a line.<br>● Speed, distance, cadence should be detected and shown.<br>● 'Stop' button should be provided for user to stop recording.<br>● User can continue running after pressing 'stop' button.<br>● People can clear their record and restart it during their run. | - Test application on multiple devices<br>- Debug the app after running<br>- Random tests with feedback, perhaps family members or members of the public |
| Saving recorded runs | ● The record can be saved when the user stops running<br>● When people are saving the route, the user can enter a name for the route to be saved. | - Make sure that data is being saved and stored in to the local storage<br>- Debugging after running<br>- Feedback based improvements where possible |
| Reviewing recorded runs | ● All of the records can be | - See if the recorded run |

| | | |
|---|---|---|
| | reviewed.<br>● Only one route can be selected and shown at any time.<br>● Initial point, endpoint,run path, average speed, total distance and total time should be displayed on the map.<br>● Record is deletable and user can rename it at any time. | is accurate<br>- Debugging<br>- Add features to increase intuitiveness such as programming the back button on Android devices<br>- Make sure buttons perform tasks as intended |
| Home page | ● It is the initial page for the user.<br>● Users can navigate to other pages/features from the home screen | - Make sure that links/buttons work to transition to necessary features<br>- Make sure it is possible to return to the home page |
| User Instruction Page | ● User instructions can be understood easily.<br>● User instructions are easily accessible | - Test the application on people with no prior knowledge and ask for feedback on the quality of instructions and change accordingly |

## Risk management

| **Risk** | **description** | **Severity** | **likelihood** | **plan/response/mitigation** |
|---|---|---|---|---|
| App updated resulting in lost data | Regular updates to the application result in lost data | Medium | Medium | Keep storage methods consistent throughout updates to ensure data is maintained |

| | | | | |
|---|---|---|---|---|
| Device operating system is updated resulting in failure of the web application | Operating system update on the device leads to a failure in the running of the web application | High | Low | Be aware of updates and release updates accordingly |
| App is hacked by others. | Application is hacked in an attempt to retrieve user data | High | Low | Improve security |
| Data usage is exceeded | Each recorded run can take up space on the local storage to the point where it exceeds the allocated usage amount | Medium | Low | ● Limit the number of records for each user<br>● Store data using JSON which amounts to very little usage |
| Loss of data due to crashes | Computers and devices can crash occasionally leading to a loss of data | High | Medium | ● Adopt a cloud system to maintain data<br>● Asana, Github and Google Docs constantly auto saves work to ensure minimal losses upon crash |
| Google updates | Google chrome or maps APIs could receive updates, making the web application become outdated | Medium | Low | ● Pre-emptively update application to fit new versions of google chrome and maps<br>● Be alert of future updates from google |
| The uploading server may face | The application can be uploaded | Low | Medium | Ensure that work is completed in a timely manner as to not |

| | | | | |
|---|---|---|---|---|
| issues | only to the server, hence if there are issues changes will not be able to be made | | | rely on server usage nearing the end of the project as there may be heavier loads |