Alex Luc, Tom Gu, Nikolaevsky Nir

TEAM 68

21/10/2015

just do it!

Project Management Plan

Version 1.0.0

Contents

TOC \o 2-2 \t "Heading, 3"

Introduction and Purpose PAGEREF \_Toc \h 2

Summary of Project PAGEREF \_Toc1 \h 2

Assumptions PAGEREF \_Toc2 \h 2

Client/Users PAGEREF \_Toc3 \h 2

Deliverables PAGEREF \_Toc4 \h 2

Scope PAGEREF \_Toc5 \h 3

Approach/Methodology PAGEREF \_Toc6 \h 3

Timelines PAGEREF \_Toc7 \h 3

Personnel/HR management PAGEREF \_Toc8 \h 4

Communications management PAGEREF \_Toc9 \h 5

Quality management PAGEREF \_Toc10 \h 5

Risk management PAGEREF \_Toc11 \h 6

# Introduction and Purpose

Since the early days of humanity fitness was and is an important part of our lives. While the basics of fitness stayed the same, recent advancements in technology allowing us to make staying fit easier then ever. The goal is of the project is exactly that, helping people to stay fit by providing them with a tool which can record their achievements and help them track the progress they made.

This document contains a brief summery of the project and it’s scope together with providing the reader with information about the management aspects of the given project.

# Summary of Project

## Assumptions

In order to successfully handle the project, some assumptions and changes to the functional spec had to be done.   
Assumption made:

* App should be somewhat visually appealing
* Buttons should represent their actions and be easily visible
* Clients got basic understanding of touchscreen devices and familiar with common app design choices (for the actual functionalities of the page an instruction page was added)

Not many changes were made from the functional spec submitted. Most visual changes included layout differences of buttons and text.

## Client/Users

The target audience of this app is anyone who cares about fitness and takes parts in activities like walking, running and cycling.

## Deliverables

The project will include the functional app with all the specified functions (recording and saving runs) together with three documents, this Project Management Plan, Requirements Creep Document and Test Plan & Results Document.

# Scope

## Approach/Methodology

To decide on an approach to this project all aspects of it had to be considered. We decided to use agile approach. This approach was chosen to allow easier implementation of new features and the ability to deliver a product even if not all the planned features were implemented. To achieve the best results, the visual side of the project (HTML/CSS) was focused on first. The design and layout of the app were planned in detailed and polished before diving into the functional coding (JavaScript). This allowed us to insert the functions into an already ready and finalised template and allowed us to visualise to finalised product. The workload was decided between all members to increase efficiency. Weekly meeting and online conversation insured all members know what they are doing and check on each other’s progress.   
To maximise the efficiency and quality of the code all members were working on the same page at the same time (devision would not be ideal as some pages required significantly more complicated code then others) and using services like GitHub and Asana to stay on top of each others work.

## Timelines

HTML layout (non dependent) - 1st - 2nd week of assignment

* Index Page (working period 4/10 - 7/10) :  
  \*Buttons   
  \*Text boxes   
  \*Background
* Instruction Page (working period 6/10- 9/10)  
  \*Buttons   
  \*Text boxes   
  \*Background
* Recording Page (working period 6/10 - 12/10)  
  \*Buttons   
  \*Text boxes   
  \*Background  
  \*Map
* View (History) page (working period 6/10 - 9/10)  
  \*Buttons   
  \*Text boxes   
  \*Background  
  \*Table

JavaScript code (dependent on HTML template) - 3rd -4th week of assignment

* Adding Links to all buttons (back buttons, start action buttons) (working period 13/10 - 14/10)
* Adding timer to Recording page (working period 12/10 - 14/10)
* Adding options for user input and activity choice (working period 13/10 - 16/10)
* JSON and local storage (working period 17/10 - 18/10)
* Tracking and map (including markers and polylines) ( working period 17/10 - 19/10)
* Creating a table and display of stored data (working period 17/10 - 19/10)
* Calorie counter (working period 20/10 - 21/10)

Staying on schedule: to stay on schedule we used the platform offered by Asana. It allowed the team to manage each task and follow due dates and deadlines.   
In addition, frequent online conversation (usually over Skype or Facebook) allowed members to inform each other on the progress made and any delays.

# Personnel/HR management

At the beginning the workload was divided for the first part of the project. Later, when new code had to be implemented or old code had to be tested and updated GitHub was used. Github helped manage the different version of code and make sure everyone is working on the latest version of the code. Asana was used to keep a steady schedule and helped teammates assign tasks to each other. Those resources together with Skype conversation, phone calls and face to face meetings helped insure all tasks are being fulfilled and all members doing what is required of them.

* Index Page - Alex
* Instruction Page - Tom
* Recording Page - Alex/Tom/Nir
* View (History) page - Alex/Nir
* Adding Links to all buttons (back buttons, start action buttons) - Nir
* Adding timer to Recording page - Alex
* Adding options for user input and activity choice - Tom
* JSON and local storage - Nir
* Tracking and map (including markers and polylines) - Alex/Tom
* Creating a table and display of stored data - Tom/Nir
* Calorie counter - Nir

It is important to mention that while each member took care of specific tasks allocated to him by the whole team, almost every page was eventually edited by all 3 members.

# Communications management

The team used a wide variety of tools to communicate, all of which were mentioned in the report.   
GitHub was used to share code and communicate changes to code.   
Asana was used to assign and follow tasks   
For more general or less focused communication phone calls,Skype, Facebook were used.   
Weekly face to face meetings also helped in organising the project and made working together easier which resulted in synergy.

# Quality management

| Feature | Expected Quality | Testing of Quality |
| --- | --- | --- |
| Buttons | * Look attractive (represent their function) * Be big enough * Preform their intended function (clear, go back, select mod, etc..) | * Look at the buttons and ask for a review from peers and others * Same * Running the app and checking each button preforms the required task |
| Text Fields | * Have clear and visible text * Placed (merged) properly in the page | Ask for a review from peers and others |
| Background and design | * Displayed throughout the page * Not distracting and visually appealing and relevant | - Ask for a review from peers and others |
| Map (design) | * Be big enough for practical use * Placed (merged) properly in the page | - Ask for a review from peers and others |
| Timer | * Be clear with information displayed (hours, minutes,seconds) * Display time accurately | - Ask for a review from peers and others  - Timer will be compared with other pre-existing timers to insure accuracy |
| Tracking | * Show current location with marker * Centre on current location every period of time * Follow user’s movement * Draw polylines between visited coordinates | * Run the app and check if location displayed by marker is accurate relative to actual testers location * Run the app and check if track recorded is similar enough to track taken by tester |
| Input comments | * App prompts user to enter text * Prompt box gives clear instructions of text to be entered * Text entered is saved to local storage | * Ask for a review from peers and others * Same * Check local storage through Developer tools |
| Choice of activity | - Change mode when relevant button is pressed (allows later to extract data for each separate activity) | - Record a run with all different activities and try to extract data. See if data can be extracted in relation to activity |
| Saving runs in local storage and extracting in history page | * Saves all relevant data from run to local storage * Is extracted to a table in history page | * Use developer tools after couple of runs to check if data was saved to local storage * Check if all the correct data is displayed in history page when compared with local storage. |
| Calorie counter | - Save and display calorie count which is calculated by information inputed by the user and saved from the run | * Running the app and checking if the formula produces reasonable numbers * Checking if the results are consistent |

# Risk management

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Risk* | *description* | *Severity* | *likelihood* | *plan/response/mitigation* |
| Teammate might retire | While all teammates have approximately the same skills, they can not be replaced. One less teammate means a lot more work decided between the remaining teammates. | High | Medium | control -> Decide as a team that if someone wants to quit they will notify the rest of the team as soon as possible. The project will be kept as organised as possible and records of work that need to be done will be made and maintained so the work could be divided easily later on and continued with as least effort as possible. |
| All progress is erased | As a result of a mistake all new data and code can end up being deleted | High | Low | control -> Backup all the most updated data on each user’s computer. This can be easily done using GitHub. As another safety measurement the project can also be kept on a USB or hard drive. |
| Teammate can get sick | During the semester teammates might get sick, slowing the overall progress and productivity of the team | Medium | Medium | control -> Decide on action plan in case a member is sick. Decide on ways to communicate and expected workload is member stays at home (considering a light illness and not a serious condition)  OR accept -> in case the condition is serious and prevents the teammate from participating at all, the remaining teammates will take care of his share of the work. |
| Phone breaks | During testing the phone might fall and break, its a problem as the app must work specifically on the device provided | Medium | Medium | control -> Team will decide on appropriate ways to handle the phone (ex. only over soft surfaces, always using both hands, etc…)  If the phone does break a new one can be purchased. |
| Addition of tasks while app is in development | While the app is in development new tasks and requirements can suddenly be added. Those tasks where not accounted for in the planning and no time was allocated to them | Medium/Low  (depending on the required task) | Medium | control -> allocate the time in the planning stage to allow some free time which will account for possible additions or changes.  OR accept -> as it is hard to anticipate if new features will be required during the development and thus time allocation is challenging. In case of a new feature some time from all other tasks will be taken to allow its application in the app or teammates will require to work additional hours on the project. |
| Monash servers might fail/overload | The servers and VPN allocated for the testing of the app might crash due to heavy traffic created by multiple teams testing their apps.  This may cause a delay in developing the app as real time testing will be made more challenging and less accessible. | Low | High | Accept -> as we do not have control over the servers we will have to accept the risk. While heavy traffic on the servers is likely and might delay some testing, many functions can be tested on a computer so frequent testing using the VPN is not necessary. |