Software Engineering Group Project
Project Plan

Author:	Group 14
Config. Ref.:	SE-14-PLAN
Date:	2014-02-14
Version:	2.0
Status:	Final

Department of Computer Science,
Aberystwyth University,
Aberystwyth,
Ceredigion, SY23 3DB,
U.K.

CONTENTS

1   INTRODUCTION

1.1   Purpose of this Document

The purpose of this document is to define the plan of the architecture of the software and the way in which the project will be managed.

1.2   Scope

This document shows a high-level overview of the system without going into detail. The use-cases and user interface for both app and web side, the time schedule for the project and finaly an analysis of the risks of the project are explained.

1.3   Objectives

The main objective of this document is to show the plan of the system being made, and when, how and with what tools it will be made and at what times the different components need to be done. Also making clear what can be done for users and administrators from both phone and web side.

2 OVERVIEW OF PROPOSED SYSTEM

2.1 Project Overview

Our app that we are going to produce will be android based. Our app is going to be a walking tour app that will help people make their own walks and to be able to edit and save them to view again a later day.

Platforms to be used and Architecture for the app: gskip

2.1.1 Android

The assignment stated that we have to make an app for android phones so thats the platform we will be using, as well as using android platform version 2.2

2.1.2 PHP

We will be using PHP for the website and for the phone to communicate to the server (database). I think we will find it easier to use than javascript and it will be a good opportunity to get better at writing in that language. We plan to host the Website on our group member Jake Maguire's personal VPS and the web address for the project will be www.jakemaguire.co.uk/projects/wtc

2.1.3 JAVA

Java is what we will be using to program the app. We just need to decide what we will use to write it in, there is a choice between eclipse with the android plug in or android studio. Our preference will be to use android studio

2.1.4 MySQL

MySQL is what we will be using to make the database to be able to store the walks that have been created.

2.2 High Level Architecture

2.2.1 Map API

The map will be a big part of our app because we will need to display the routes and be able to let people see where there going and make their own route. We will be using the map api on our web server to display a selected route from the database.

2.2.2   App UI

The Menu / UI, first we will have the main page where you will be able to click on the button create new route to be able to start mapping the path while you are walking, right underneath this button their will be 2 more preferences and exit. All together there will be up to 4 menus to this app: main page, naming of the walk, route map editor and the adding location.

2.2.3   Internet Usage

The app will only need to use the wifi or 3G/4G to be able to improve the tracking capability of the app and to be able to upload the finished walks onto the website.

2.2.4   Target Users

Our target users are people that enjoy going on walks and would like to start marking out their walks and to be able to save them so they can look back at them and do that walk a later day.

3 USE-CASES

The use cases for our application will be split into two cases. These are the "App" and the "Server". The App use case describes all interaction with the application itself and, as such, only contains interaction with the mobile user. The server is not listed as interacting with the app here as interaction between the app and the server is functionally one-way (the app pushes walks to the server, but cannot retrieve or view walks).

The Server use case describes all interaction with the server, which includes the web user, app, and administrative user.

3.1 App
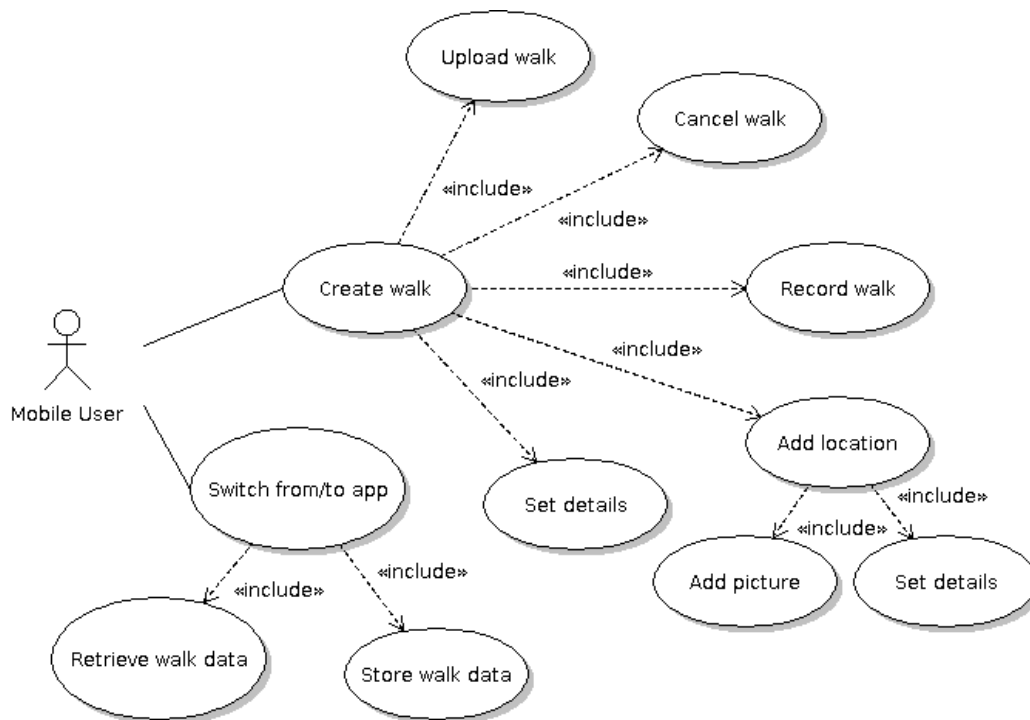
The app's use cases include:

3.1.1   Create Walk (FR1)

Here the user starts to create a walk. Once this interaction is started (by pressing the button in the app's main menu), and a valid walk name is provided, the user's location is tracked and periodically added to the on-going walk's list of GPS points. As a part of this case the user may choose to:

- Add location: This will allow the user to add the current location to the walk as a point of interest. (FR3)

    - Take picture: The user can take a picture to add to this location. (FR4)

    - Set details: The user can set the location's details (name, description).

- Set details: The user can set the walk's details (title, short decription(¡100 characters), long description(¡1000 characters). (FR2)

- Cancel walk: The user may choose at any point to cancel their walk. This will stop recording their position and drop all current data of the on-going walk. (FR5)
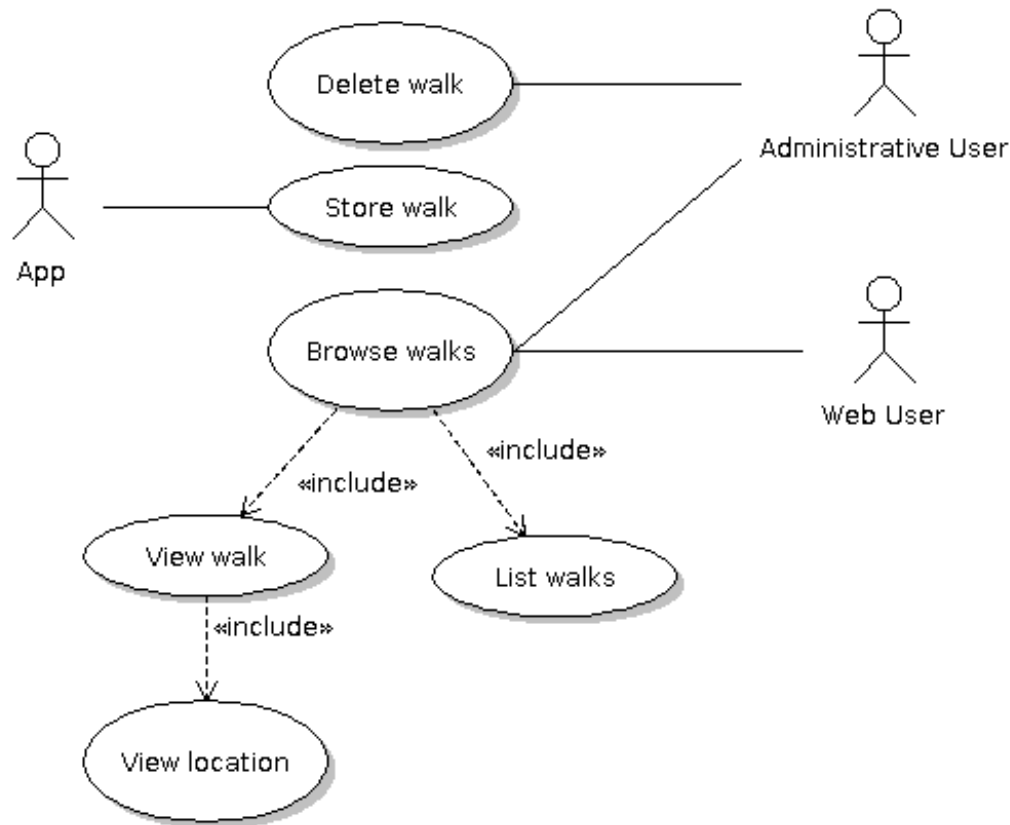
3.1.2   Upload Walk (FR6)

This is used at the end of the "create walk" case in order to upload the walk to the server.

3.1.3   Switch to/from app

Whenever the user switches from the WTC app to another app, it will store the current walk's data. When they switch back it will retrieve the previously stored data. (FR7)

3.2 Server

The server's use cases include several actors.

The "App" actor is an instance of the WTC app on a client's phone and can upload walks.

The " User" is a client accessing the WTC website in a web browser. They may view a list of walks, select one, and view its details.

The "Administrative User" is a user with the power to delete walks as well as view them.

The server's use cases include:

3.2.1 Store walk

A walk recieved from a client using the WTC app can be stored in the database of walks. . (FR9)

3.2.2   Delete walk

This use case allows for administrative users to delete any walks which they feel do not belong in the database. This would include any corrupted/broken walks and walks which contain obscene or offensive content.

3.2.3   Browse walks

This use case provides html pages to users which contains a list of all the walks in the database. The cases in this case are:

- List walks: This provides a list of all the stored walks.

- View walk: This will allow the user to view a specific walk, including its name, short and long descriptions, and its points of interest. (FR8)

    – View location: This will allow a user to look at a specific point of interest. This includes its name, description, location, and any picture it has.

4   USER INTERFACE DESIGN

4.1   Android Application UI Design
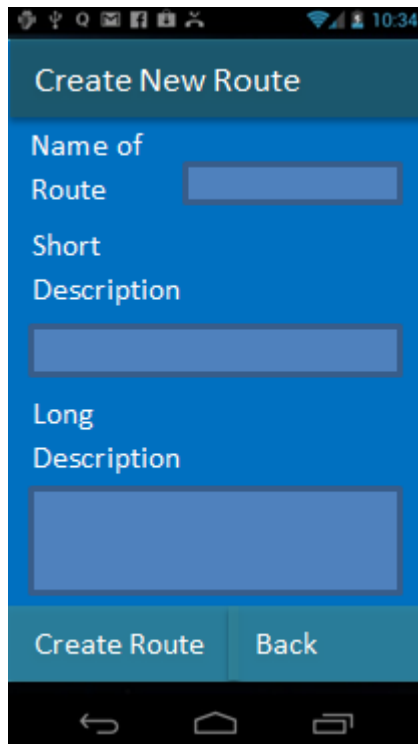
This section of the documentation will give an idea for how the UI of both the Android application and the web application will look like and also the flow of the program from screen to screen. These designs are not final due to technical restraints we may face later on in development but they will give a feel for the sort of layout and feel we are working towards.

4.1.1 Main Menu (FR1)
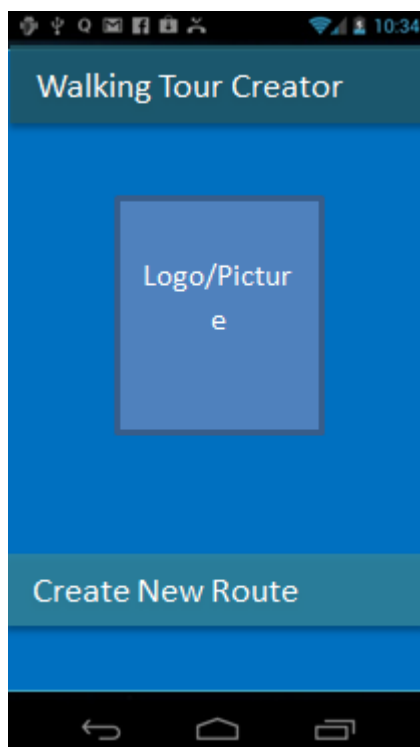
The first screen that the user is presented with is the main menu. This screen has one main option for the user to select. The user can create a new route which allows them to progress into the application and start recording their routes. There will also be some sort of logo or picture above these options.

4.1.2 Create New Route (FR2)

On the Create New Route screen the user is able to set up a new route for them
to record. They will be presented with a field to enter the name of the route,
a short description of the route and a longer description. The short description
will be restricted to up to 100 characters while the long description is up to 1000
characters. The user will be required to fill in all of the fields, as per the functional
requirements.

There will be two buttons at the bottom which allow the route to be created or to
go back to the previous page.

4.1.3 Create Route (FR1, FR5, FR6)
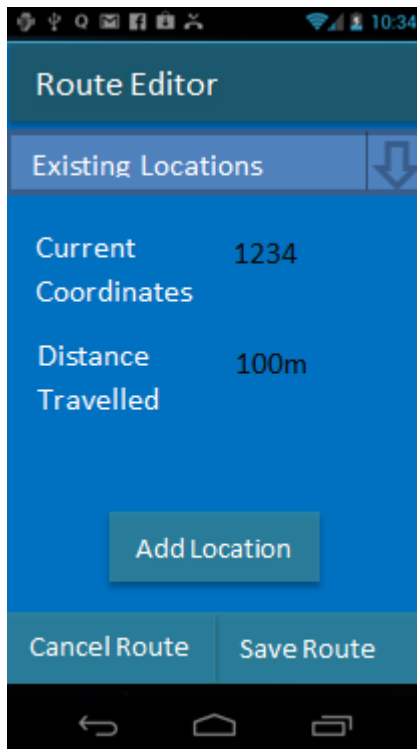
When the route has been created the user will be presented with their current coordinates which will update in real time as the user moves around, their current distance travelled from the starting location, along with options to add a location and save the current route. The existing locations drop down will allow the editing

of locations that have already been added, letting the user change the name, coordinates and description of a previous location. Adding a location will save the current

position and move to the next screen where they can specify information about that location. Once the user has finished recording and has added their locations, the user can press the save location button at the bottom of the screen to send this data to the server where it can be read by the web application. They can also cancel the walk if they do not require it after recording.

4.1.4   Adding Location (FR3, FR4, FR9)
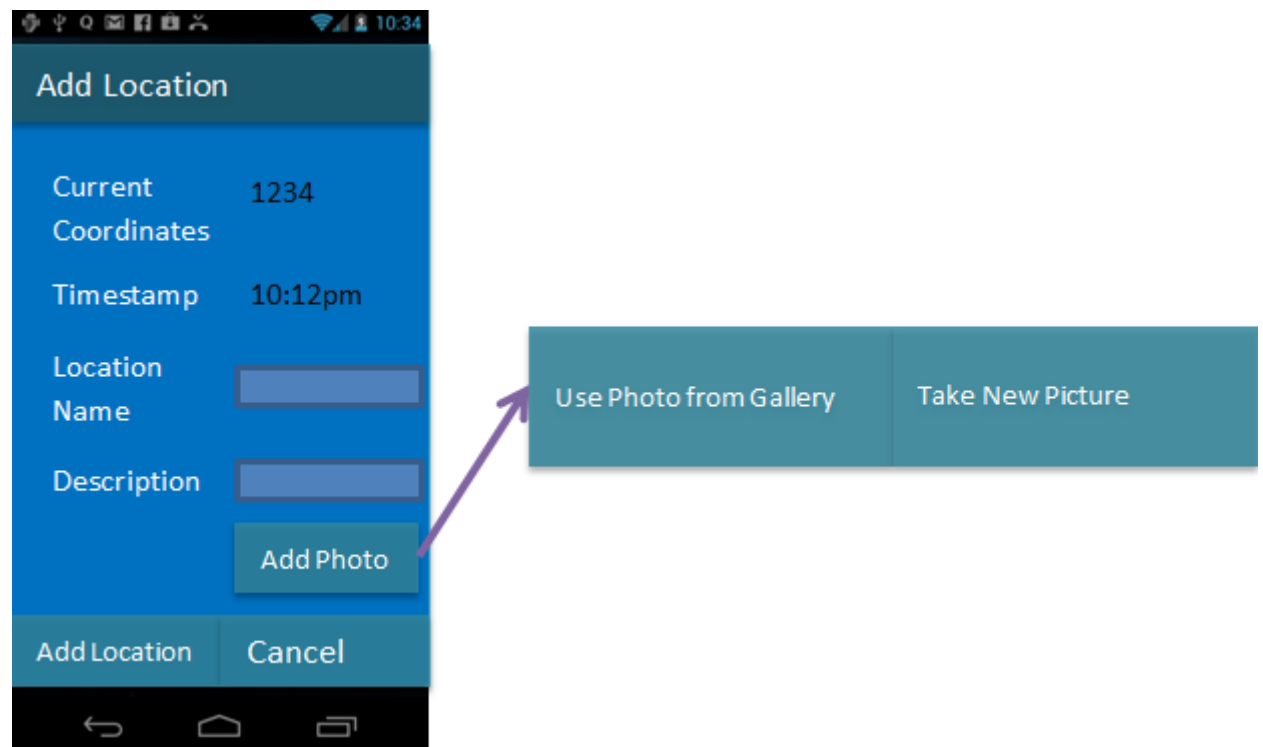
When adding the location the user is required to enter some information along with some already given to you. The user?s current coordinates are displayed along with the current time the location was added. The user then enters the name of the location and a short description. There is also an option to add a photo where a photo can be taken using the camera option or use a photo already in the gallery.

4.2   Web Application Design

4.2.1   HomePage
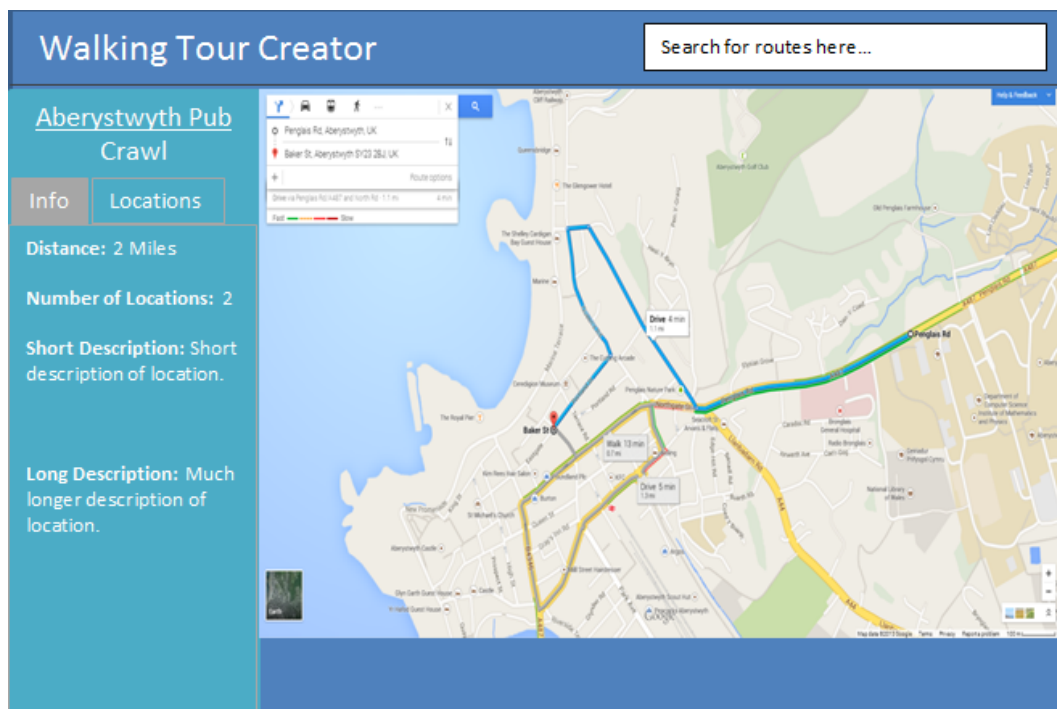
There will be images on the home page of the walks. The user can then select a walk from the homepage. The user can then select this route which will then populate the sidebar and Google Map. The sidebar will allow the user to toggle between information about the walk and information about the locations, as there is not enough space to contain both.

5   GANTT CHART

Weeks

Task	11th / 1	14th / 2	21st / 3	28th / 4	4th / 5	11th / 6	18th / 7	25th / 8	2nd / 9	9th / 10	16th / 11	23rd / 12	30th / 13	6th / 14	13th / 15	20th / 16	27th / 17	3th / 18
Project plan																		
Test Spec																		
Design Spec																		
Prototypes																		
Website																		
Database																		
UI implementation																		
APIs for Android app																		
Android app																		
UI implementation																		
GPS backend																		
Server communication																		
Delivery of software																		
Final report																		
Notes					Work week							Christmas		Exams				

Legend:
- Everyone
- Testing team
- Design specification team
- Website team
- Android team

6 RISK ANALYSIS

6.1 Gold plating

As programmers we sometimes like to show off our skills by adding unnecessary features to our program. The schema clearly states that there are no extra marks for extra features and so doing this would just be a waste of programming hours. This can be avoided by having a clear plan of what is needed for the program before the actual programming starts.

6.2 Illnesses

This is an unavoidable risk. It is inevitable that members of the group will fall ill at some point during the year. To stop this becoming a problem, any members that fall ill will need to let the group know as soon a possible, so that other members can help out with any work that the ill member had been assigned. It is then the responsibility of the ill member of the group to then put in some extra work time when he recovers from illness. Also more than one programmer will work on and understand each part of the code.

6.3 Time management

One of the biggest risks for the group is that we will run out of time and that the project will be unfinished in the due date, or that all members of the group will have to start putting in crazy hours, in a desperate attempt to get the program done. Again this can be avoided by having a clear plan, where each specific part of the website and program have a clear completion deadline.

6.4 Compromising on design

In order to get stuck into the harder tasks of a program earlier, programmers will compromise on design on the easier parts of the program. This is also a waste of programming hours as design is arguably the most important part of a program.

6.5 Backups

The worst situation that could happen is that all the data that the group has been working on gets lost. This is avoided by having multiple backups of our program, and not just on the same hard drive. These backups need to be made in a separate physical location, and also need to be made every time work has been done on the

project. As well as several physical copies, all the work will be constantly updated and stored on github.

6.6   Released project has low quality

There is always the risk that the project, when completed is of low quality. This can be avoided by assuring that everyone is happy with suggested Interface design and everyone is clear on what needs to be done. We will also need to ensure a disciplined development is used.

6.7   Conflicting Ideas

As programmers, we are all likely to have our own ideas on how we think the app should be developed. To avoid any conflict, any issues that an individual may have with the design of the app, will have to bring them forward sooner rather than later. This will overall make for a happy group who will be able to work efficiently as a team.

REFERENCES

[1] *Software Engineering Group Projects* General Documentation Standards. C. J. Price, N. W. Hardy, SE.QA.03. 1.5 Release.

[2] *Software Engineering Group Projects* Requirements Specification C. J. Price and B.P.Tiddeman, SE.QA.03 .1.6 Release

[3] *Software Engineering Group Projects* Project Plan Specification Standards B. P. Tiddeman, SE.QA.05b, 1.2

DOCUMENT HISTORY

Version	CCF No.	Date	Changes made to Document	Changed by
1.0	N/A	2013-11-08	Made first version	lhl
1.1	N/A	2013-11-08	Changed author sections and the author section, header being wrong and use the newest version of the Gantt Chart	lhl
1.2	N/A	2014-01-29	Making fixes to the document	jam66, meo9
2.0	N/A	2014-02-12	Further fixes to the document	jab73