**CS221 Group 15**

**Project Plan**

|  |  |  |  |
| --- | --- | --- | --- |
| **Authors** | Jcm14; jam67; lpd1; mab68; jaj42;  gad16; owd2; jap38 | **Address** | Department of Computer Science |
| **Config Reg** | DOC\_DES\_PrjPlan\_001 |  | Aberystwyth University |
| **Date Last Edit** | 7th November 2013 |  | Ceredigion |
| **Version** | 1.0 |  | SY23 3DB |
| **Status** | Released |  |  |
|  |  | **Copyrights** | © FuzzyNinja (Group 15), Aberystwyh University |

# Introduction

## Purpose of the Document

The purpose of this document is to show that the requirements specification, for Walking Tour Creator, and Walking Tour View, (WTC WTV respectively), have been translated, and understood. The Software is in essence a tour creator, with a viewer element based in the web. It will serve two audiences; Client to show our interpretation of project, Development Team to provide a starting point for building the project.

## Scope

This document will include an over vie w of the proposed system, including platform choice; a description of target users, including expected skill level; use case diagrams and descriptions, indicating the actors of the system, and the core functions they will perform with the application; core UI features planned layout; a Gantt Chart to show the expected timescale this project will be performed, including major tasks; a risk analysis of the project, where we see issues could arise, how we are intending to prevent or reduce the impact of the event should it occur.

## Objectives

The objectives within this document are to:

* Provide a brief over view of the proposed system, what technologies we expect to encount
* Show major components of the proposed system
* Show elements will interact with each other.
* Display a basic model for the UI.
* Identify possibly issues with the development of the system, and how these issues were remedied.

# System Overview

The Proposed system is an Android Application to record and view tours around places of interest, via use of pre recorded video, audio, text, and pictures.

* 1. **Platforms and High Level Architecture**

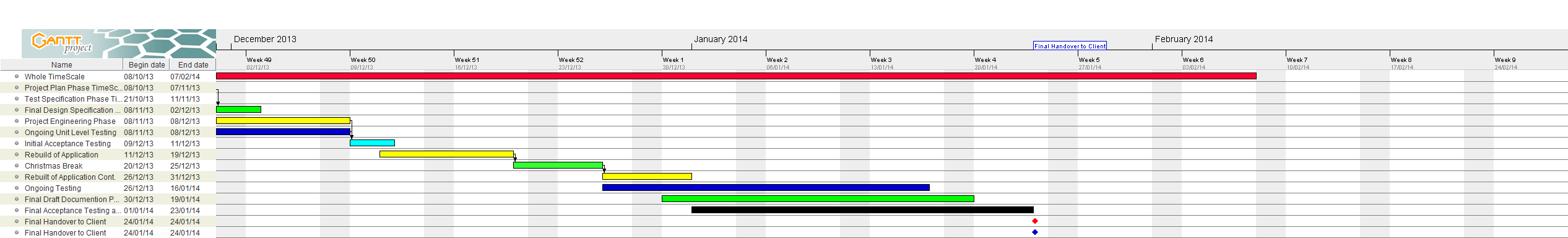
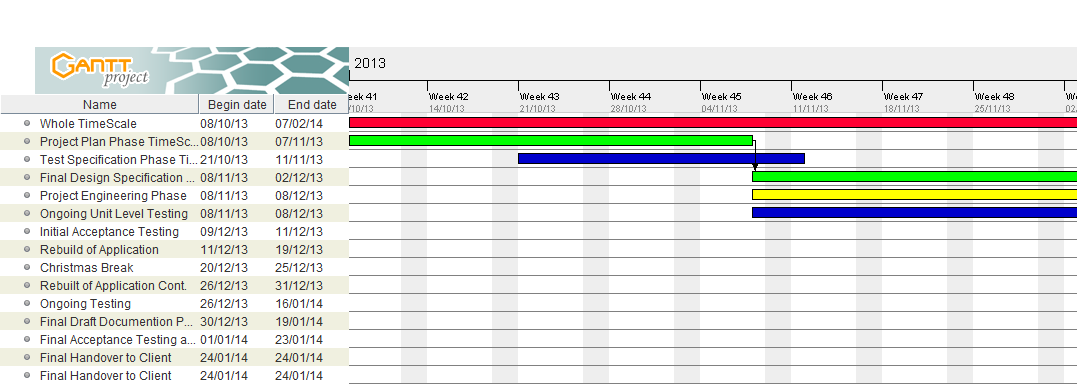
The following is a list of prospective technologies we expect to use throughout the project.

* + 1. **Android Platform** – The client wishes to commission an application, solely, for the android mobile platform
    2. **Android API and Java Language** – These are the most common tools for android systems, although C is an option, the majority of standard android mobiles do not support it out of the box.
    3. **PHP** – This Web Language will be used on the Web Side of the application between viewing uploaded tours, or using HTTP POST request etc. It is available on a vast majority of servers, including the designated system.
    4. **MySQL** – This is one of the most widely used database platforms, which we have chosen to use in Lieu or MS Access of PostGreSQL. I’s simple command interface, and support for PHP MyAdmin will be very useful through the project
    5. **Google Maps API** - Google’s Maps API for Android will be used as the interface for the tour. This was chosen as the well-known controls and features will go well with users, likely increasing the acceptance of the application by users.
    6. **Internet Connectivity**
       1. **Wifi** – Android requires the developer to specify Wifi access for internet connectivity, this prevents excessive mobile data usage at premium charges.
       2. **3G / 4G** – Android requires the developer to specify that Cellular internet is required, this allows it to prevent excessive data usage by accident. The client will have an option for Wifi only or Wifi and 3G internet connections
    7. Login data for the user will be cached locally using an md5 hash for security.
  1. **Target Users**

As defined in the requirements specification, the proposed system will be aimed all ages, to provide an open tour for areas of interest. As such, for those below the age of minority (18 in UK, 21 in some US states) certain tours may be unavailable due to content. It is yet to be decided how this will be managed, whether by the initial user provided a flag for explicit content, or system detection, likely the former however.

# Project Time Scale

Below is the planned timescale for each phase of the project. This has been developed by the project leader with time estimations from each team within the group. The Chart is Colour coded to help identify tasks.

****

# User Interface Design

## User navigation

Figure 1 is a flow chart showing how users can navigate through the program.

Start app

Starts the Application

Login

Menu start new tour/ View saved tours

If not a member press sign up

If a member login

After creation of the new user automatic to menu screen

Press home to return to menu

Press home to return to menu

Add way points

From the map/tour or old tour screen press the waypoint button to access the waypoint adding screen

Choose a tour from the list to go to the old tour screen, a screen similar to the tour view but with a tour already displayed

View saved tours

Sign-up

Press start tour to go to the main tour view or press view saved tours to go to the saved tours screen

Side swipe to enter the options menu swipe back to exit

Map/tour view

Old tour screen

Options screen

# GUI Design

The diagrams below are not the final designs for the program and are used as a basic guide to what the program will look like they are subject to change. Most screens contain a home button returning the user to figure 4.

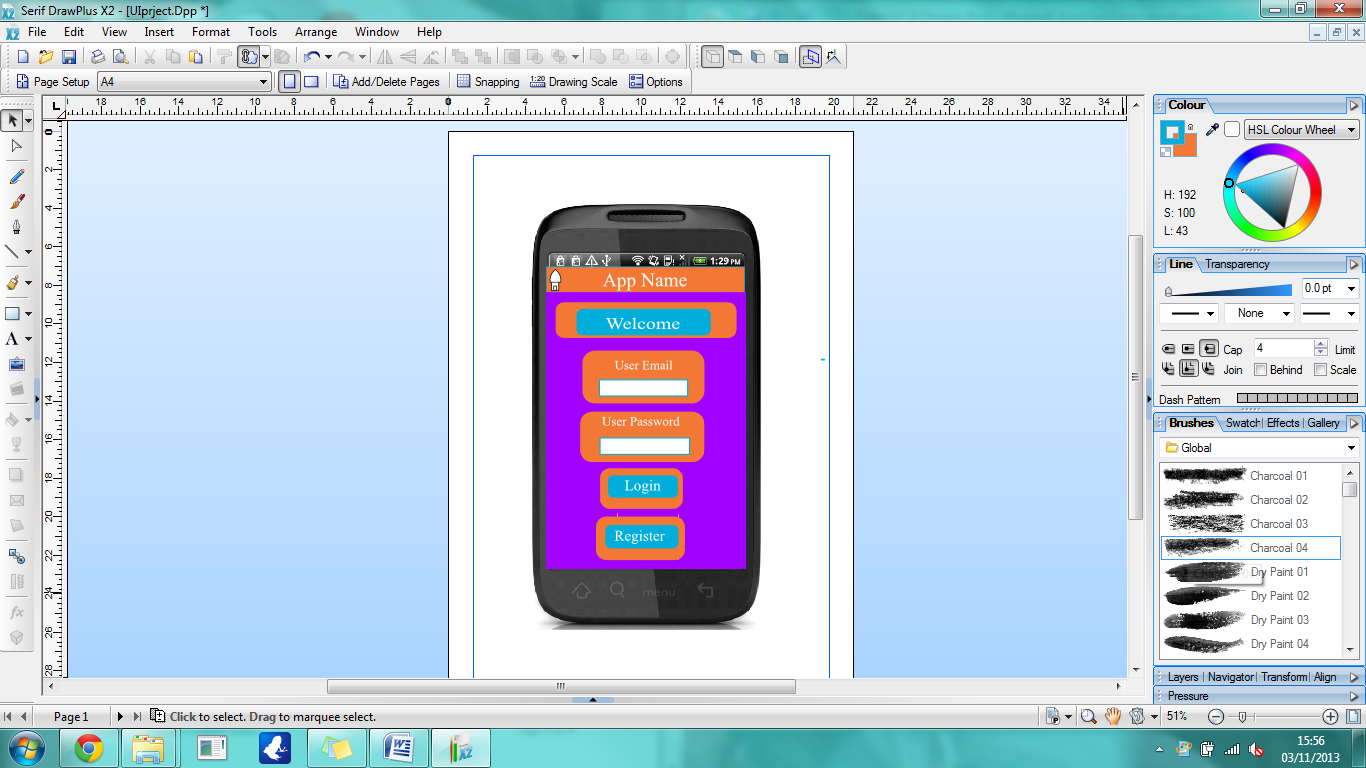
* 1.  **Login screen**

Figure 1 is the login screen this is the first screen the user will see when they start the app. It will contain the name of the app a welcome message and options for the user to either login or register to the app.

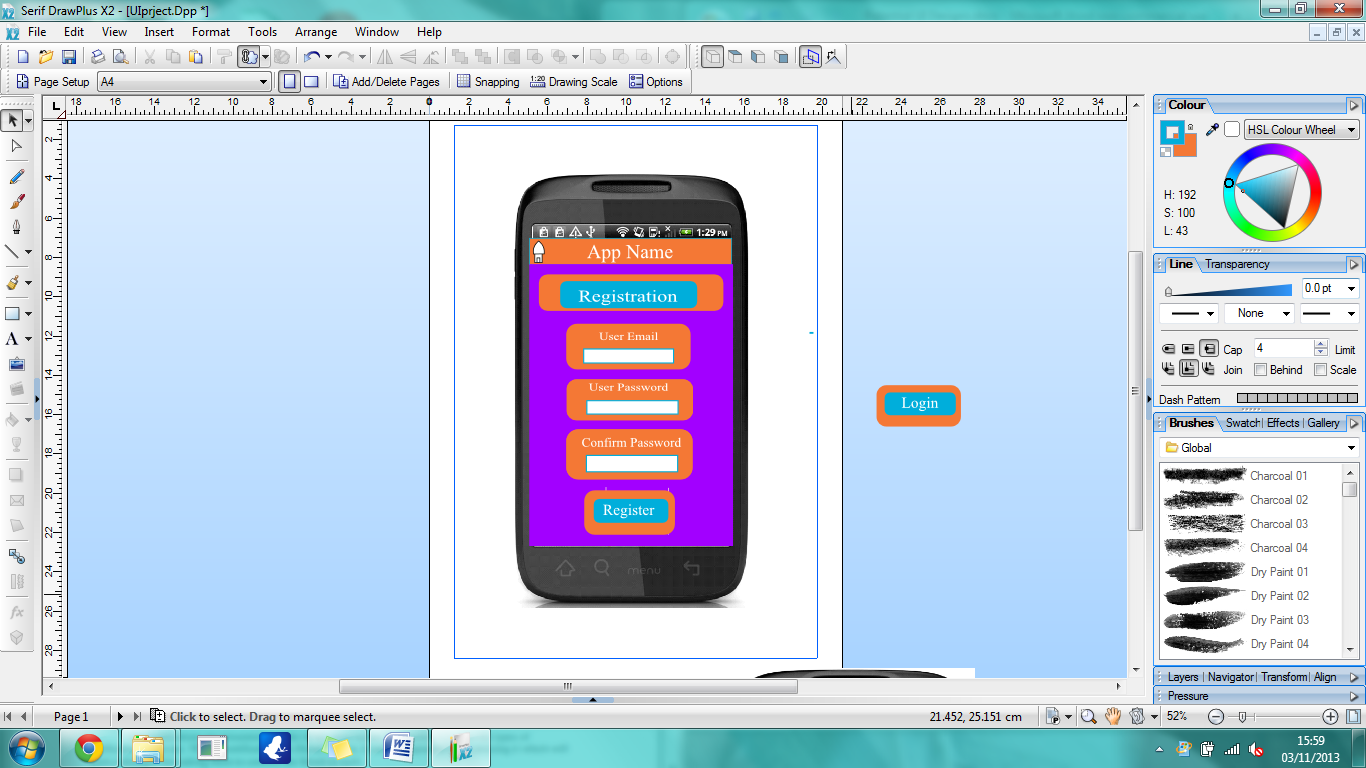
* 1.  **Register screen**

Figure 2 is the registration screen this is where new users sign up to the app. It will contains fields allowing the users enter their email and password as well as confirm their password and finally register to the app.

* 1. **Menu Screen**

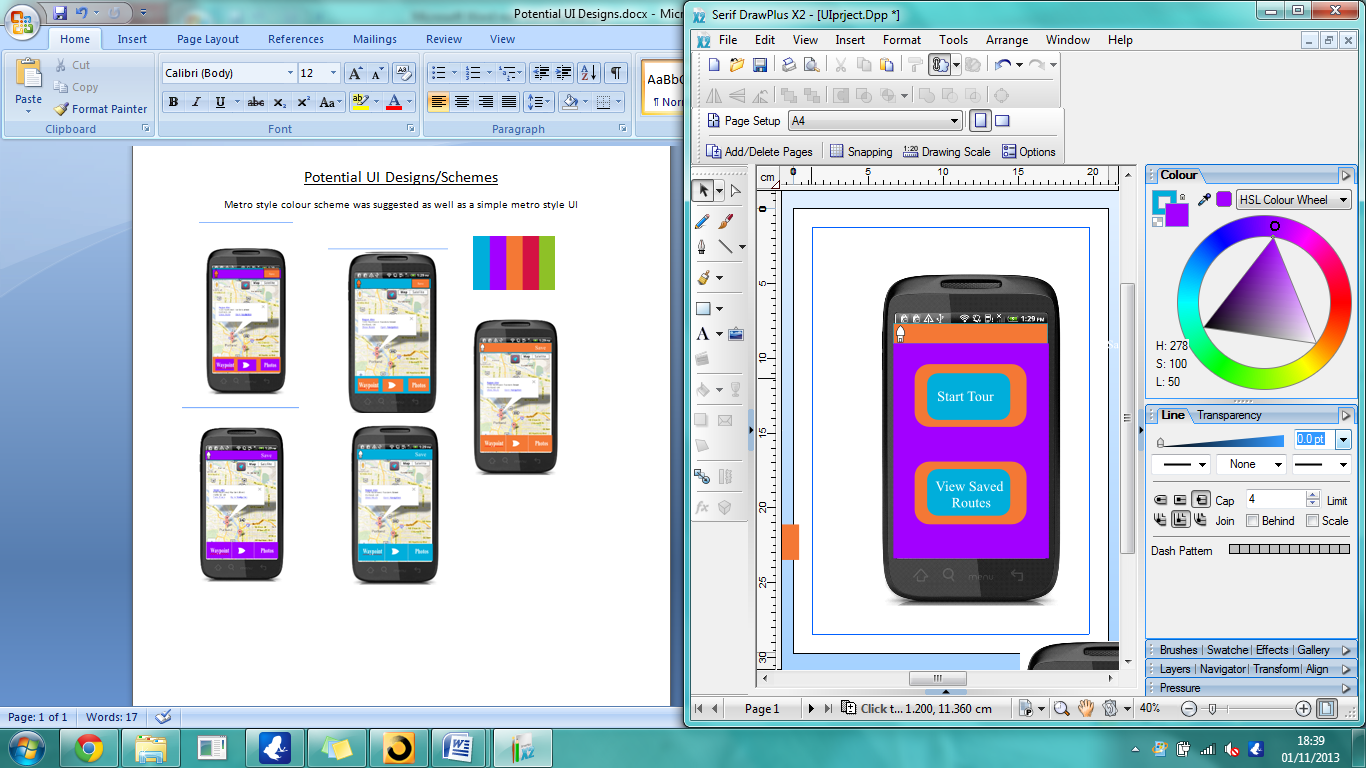
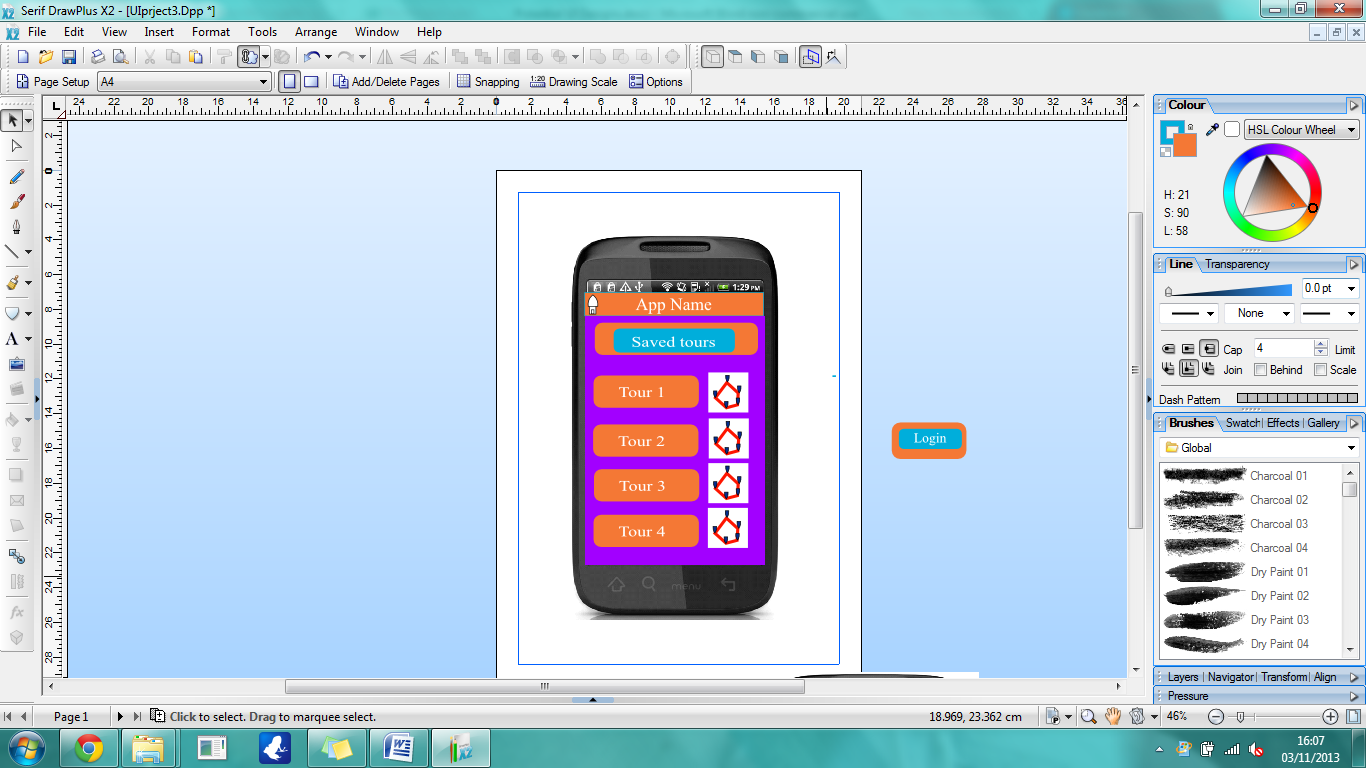
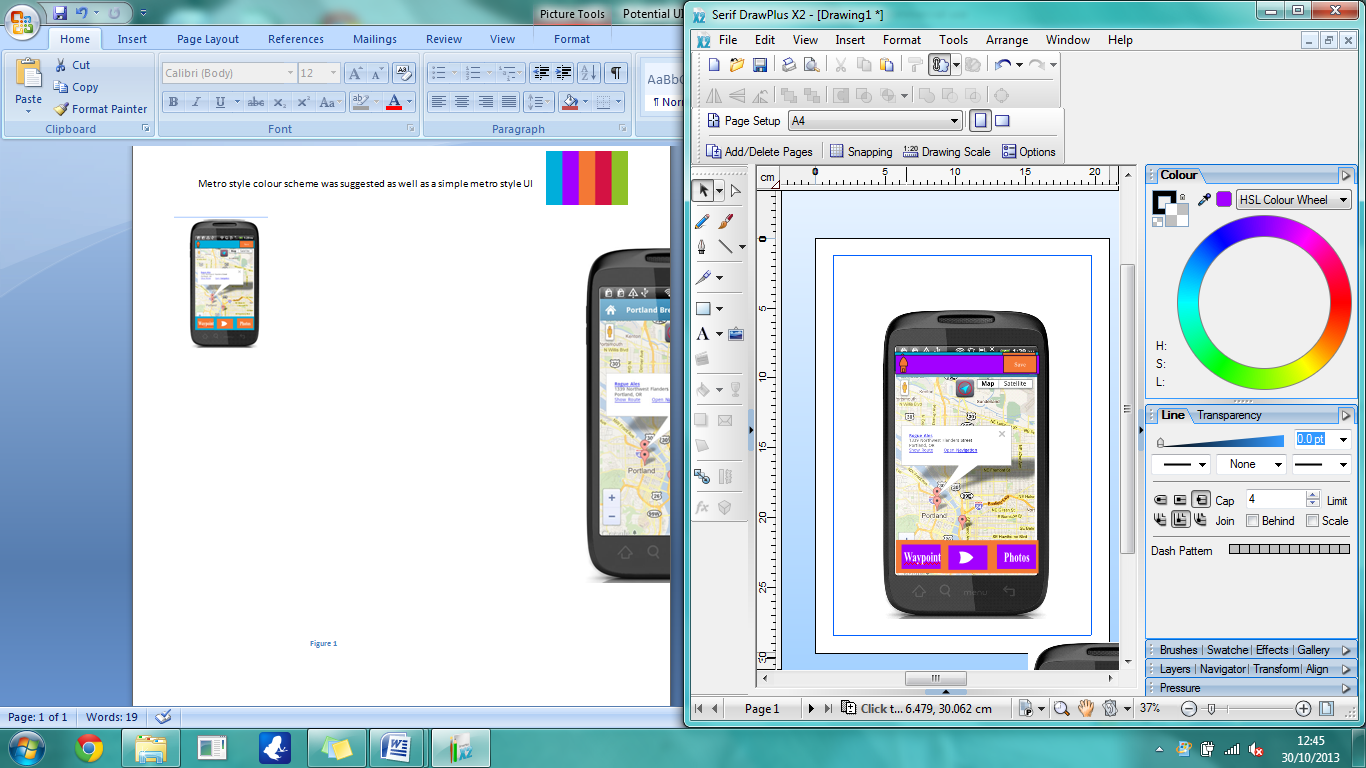


Figure 3 is the menu screen this is where the user decides whether to start a new tour or to look at/go on, a previously saved tour.

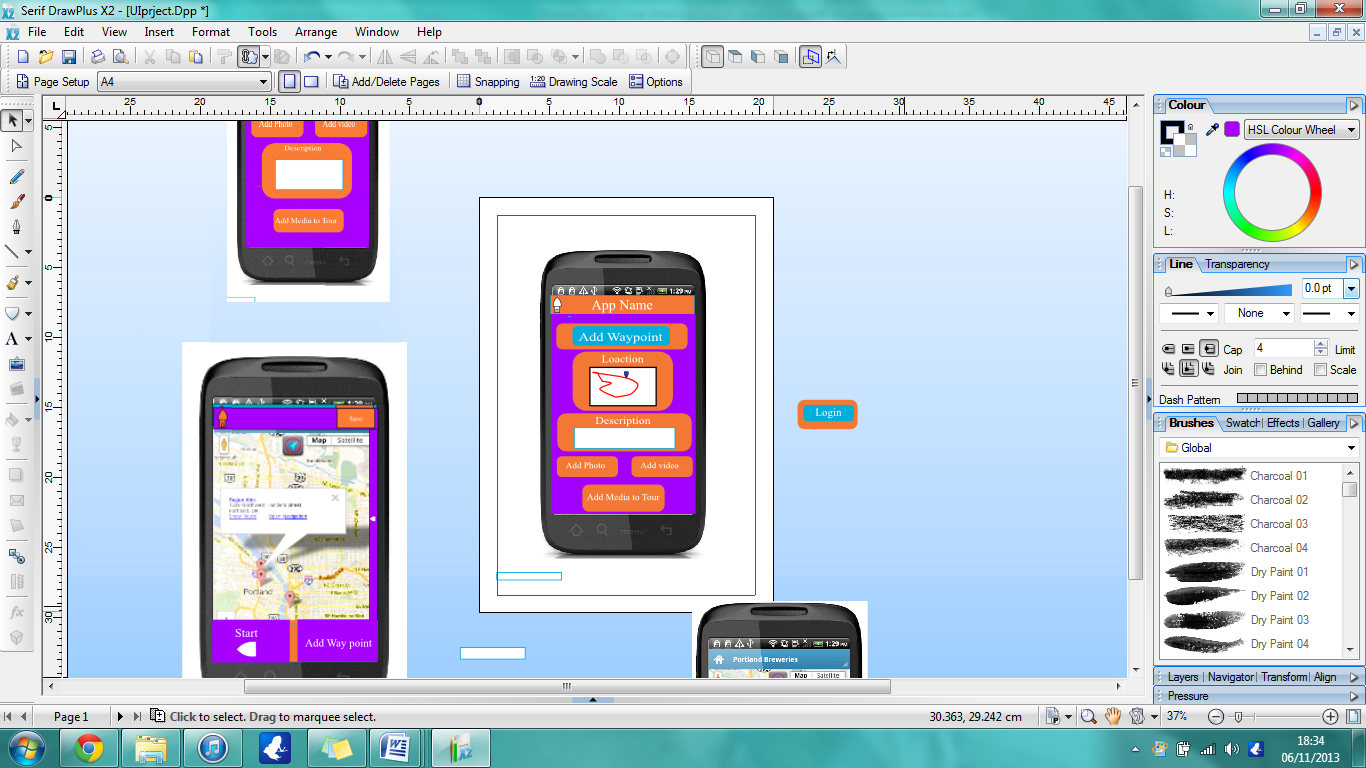
* 1. **View Saved Tours Screen**

This screen allows the user to select a previously created tour, and either run through it, or edit it. It lists the tours in alphabetical order, but considerations have been made for reordering the list

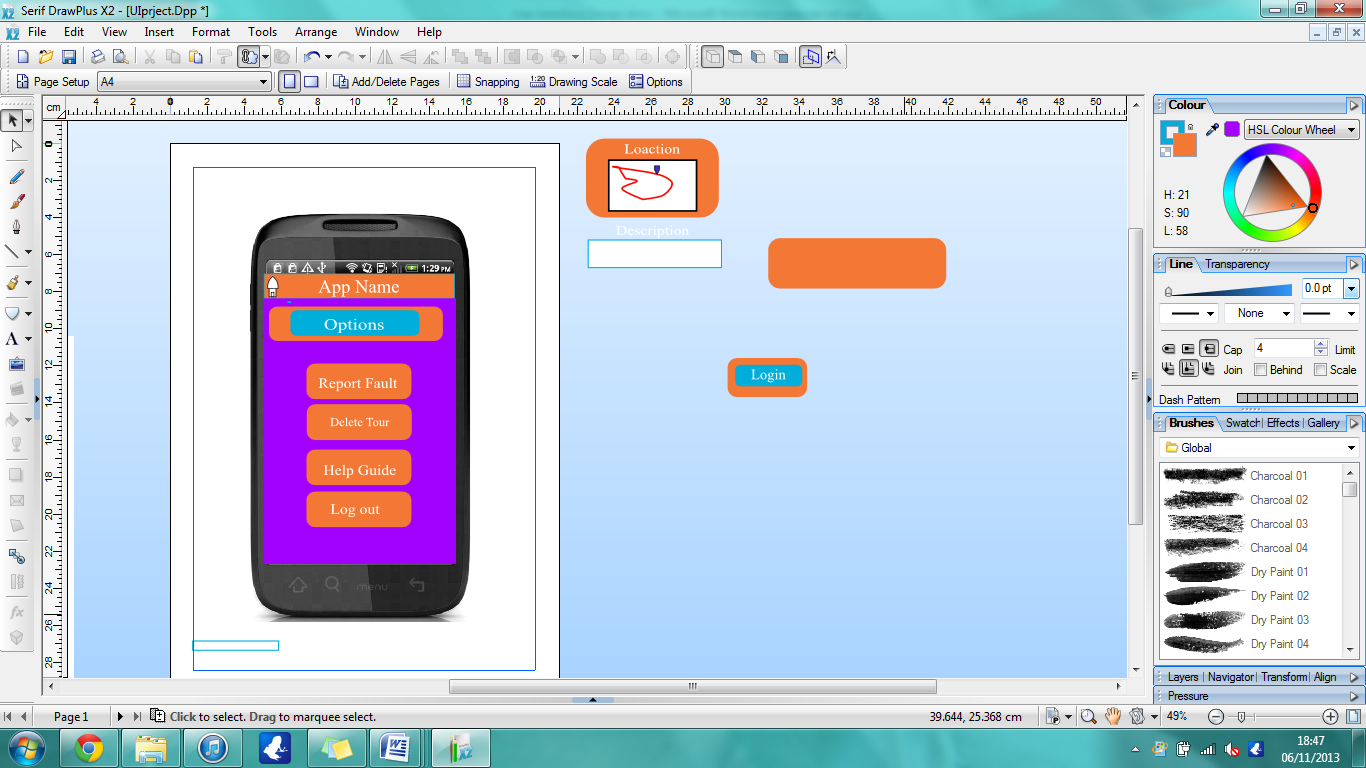
* 1. **Main Tour Screen**

 This screen is the tour screen, and is where the app uses GPS to track a user’s location allowing them to create tours of routes they have walked or follow a saved tour. On this screen the user can press the start/stop button to start or stop the GPS tracking of the tour with the option to save the tour or delete it one stop has been pressed. The user can press the add waypoints button to which takes the user to screen 4.6. On the top bar there’s a home button taking the user back to figure 4 and a save button to send the route to the server. There will also be a side menu activated by a side swipe that will bring up an options menu taking the user to figure 4.8.

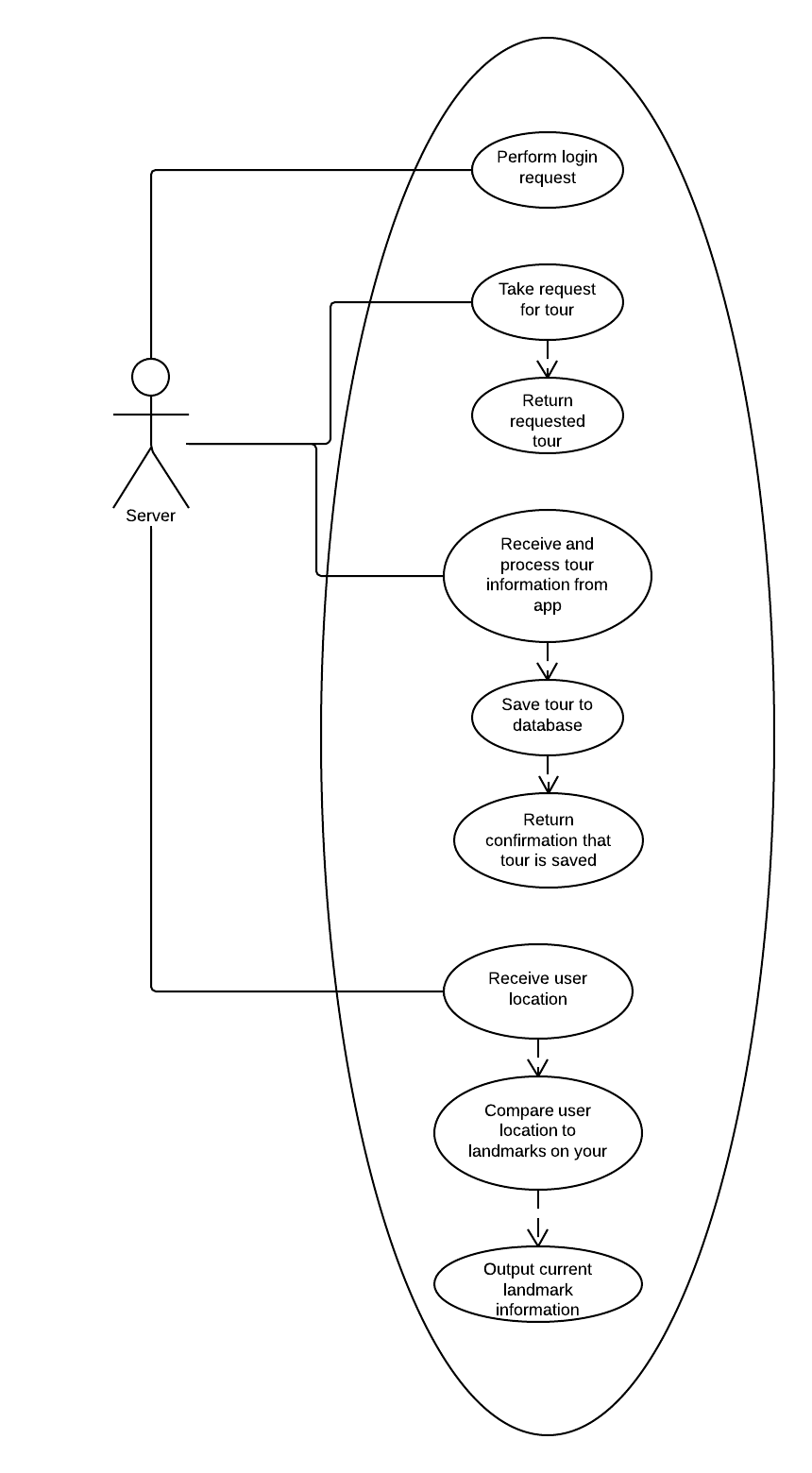
* 1. **Add Waypoint Screen**

Clicking the add way point button brings the user to the add waypoint screen were they can mark a specific location on the route and add a description of the location into the text field (limited to a certain number of words to be decided later) and a photo or video of the way point. The user can get back to the main tour screen by pressing the back button in the top left corner or will be taken automatically after adding the waypoint.

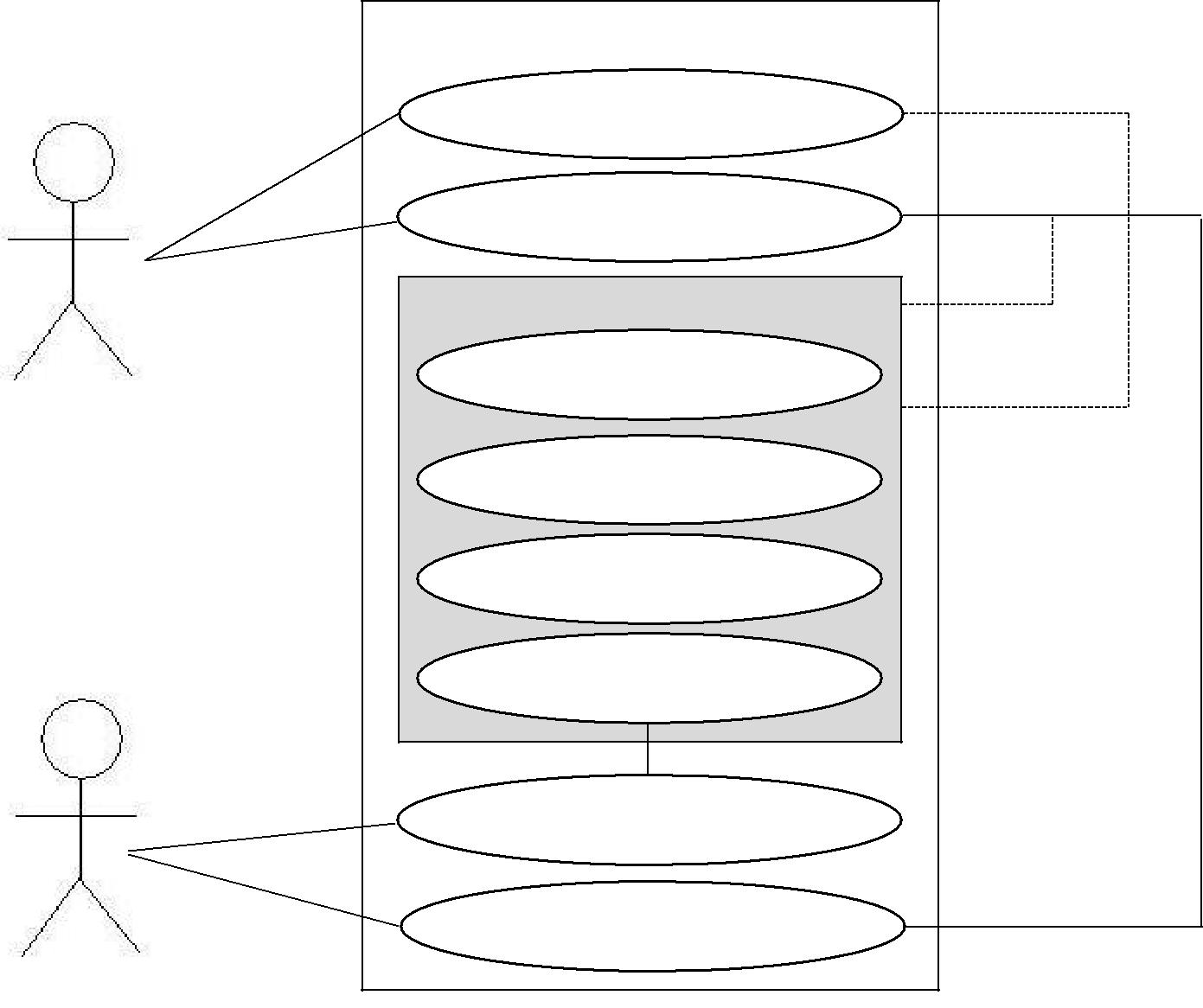
* 1. **Options Screen**

Accessed by swiping from the right on the old tour screen/main tour screen and exited via the back button in the top left or by scrolling from the left. Features buttons to log out, report a fault by bringing up an email for the app. There is a button to delete the current tour, removing it from the device and the server. Also links to a help guide/ tutorial page were the user can view FAQs and learn about the app.

# Use-Case: Server



|  |  |
| --- | --- |
| **Use Case** | **Description** |
| Perform login request | The server will take the user’s login details sent from the app, and search the database for that user. If a user is found and the details match, it will return a success message. Otherwise, it will return an unsuccessful message to the user. |
| Take request for tour | The server will receive a tour ID through a POST request from the app, and will then execute a query to find that tour in the database, and return the details to the app |
| Receive and process app information | The server will receive some data via POST from the app, which will contain the details of a new tour, as well as the details of the user who submitted it. The app will then execute a query to save this tour to the database, and return a confirmation message when this is complete - or a failure notification if this fails. |
| Receive user location | Periodically, the server will receive the users’ location from the app. It will then compare this location with locations saved for the tour they are viewing. If the user is within a predefined range of one of the locations in the tour, then the server will return information for that location to the user. |



**Android App – Walking Tour Creator**

New walking tour

Edit walking tour

**New/Edit Walk**

Add location

**User**

Add photo to location

Cancel walk

Save walk

Send walk to server

Get walk from server

**Server**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case** | | **Description** | |
| New walking tour | | Creates a new walking tour. Opens the walk editor screen. | |
| Edit walking tour | | Edit an existing walking tour. First the tour must be obtained | |
|  | | from the server first. Opens the walk editor screen. | |
| Add location | | Adds a location to the current walk. Includes latitude, | |
|  | | longitude and a timestamp. | |
| Add photo | | Opens the Android camera application. Gets a photo image | |
|  | | taken in the application and adds it to the given location. | |
| Cancel Walk | | Cancels the walk and resets any instance variables. | |
| Save walk | | Saves the current walk. | |
| Send walk to server | | Opens a connection to the server and sends the walk data to | |
|  | | the database. | |
| Get walk from server | | Opens a connection to the server and requests walk data from | |
|  | | the database. | |

# Risk Analysis

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk Event** | **Risk Level** | **Measures** | **Risk Chance** |
| Missing group members | Low | If a member is missing from a meeting the work should proceed as normal. The missing team member should read the minutes to get informed. Team members that do not show on meetings and do their job will get a warning. | Low |
| **Software Development** | | | |
| Communication between application and server | Medium / High | Problems may occur with application-server communication. This issue must be tested before the project submitted. | Low |
| Missing features | Low / Medium | Some features may take more time to implement than expected or a group member may not be able to complete his work. If a group member sees such an issue arising he/she should inform the rest of the group and ask for help. | Low |
| SQL Injection | Medium | There is a risk of SQL injection which may cause problems with the database. PHP has sanitizing functions that we will use to sanitize the input. | Low |
| Server downtime | Medium | Problems may occur with application-server communication. This issue must be tested before the project submitted. | Medium |
| Running out of time | Medium | In-group deadlines before the actual deadlines to make sure everything will be completed in time. | High |
| Poor interface quality | Low / Medium | Information should be displayed clearly and the user should be able to navigate easily. | Low/Medium |
| Git Desynchronisation | Low / Medium | All the group members should make sure that they correctly upload all of their work in order to avoid desychronisation. | Low |
| **Documentation** | | | |
| Late submission of the documentation | Low | In-group deadlines before the actual deadlines to make sure that all the documentation is appropriate and with good quality. If any problems occur and the group member cannot finish his work by the deadline he should inform the group in order to get help with his task. | Low |
| Low quality of the documentation | Low | The documentation should be checked by the group and in case of low quality should be returned to the group member for improvements. The group member should ask for help from the group if needed. | Low |