Group Project 05

Design Specification

Authors: bmo; sr11; hac22; wia2; Department of Computer Science

wjl3; njv1 Aberystwyth University

Config Ref: SE\_05\_DS\_01 Aberystwyth, Ceredigion

Date: 14/11/13 SY23 3DB

Version: 1.0 Copyright © Group 05

Status: Draft Aberystwyth University 2013

Contents

[1. Introduction 3](#_Toc373244409)

[1.1. Purpose of this document 3](#_Toc373244410)

[1.2. Scope 3](#_Toc373244411)

[1.3. Objectives 3](#_Toc373244412)

[2. Architectural Description 4](#_Toc373244413)

[2.1. Android program 4](#_Toc373244414)

[2.2. Web program 5](#_Toc373244415)

[2.3. Table mapping requirements 6](#_Toc373244416)

[3. Dependency Description 6](#_Toc373244417)

[3.1. General architecture 6](#_Toc373244418)

[4. Interface Description 7](#_Toc373244419)

[5. Detail Design 8](#_Toc373244420)

[5.1. Sequence Diagrams 8](#_Toc373244421)

[5.2. Algorithm Description 8](#_Toc373244422)

[5.3. Data Structures 8](#_Toc373244423)

[5.4. ER 8](#_Toc373244424)

[6. References 9](#_Toc373244425)

[7. Document History 9](#_Toc373244426)

# Introduction

## Purpose of this document

The purpose of this document is to describe the outline design for the Walking Tour application. Taking into account the details of the group project assignment and group project quality assurance.

## Scope

This document includes detailed description of: classes used in the Android application; methods used in each class; sequence diagrams; interaction between the application and the database; database design and data handling.

## Objectives

The objective of this document is:

* To describe the main components of the Walking Tour Application
* To describe the main components of the Web site
* To depict the dependencies between the components

# Architectural Description

The Walking Tour System consists of two main components:

* The Android application;
* The Website;

## Android program

The android application will allow the user to create their walking tour and add photos and more detailed information about places of interest they come across during the walk. The walk can then be uploaded to the online database.

**Significant classes and methods:**

* **Tour class:**

1. Description:

The main storage class of the application. Tour will hold a linked list of locations for the walking tour; it will contain variables as follows:

* + a string for the location of the tour e.g. “Aberystwyth”;
  + a string for the long description of the tour that will have no more than 1000 characters;
  + a string for the short description/summary of the tour that is limited to 100 characters;

1. Methods:
   * **Constructors:**
     + Default:   
       The default constructor will generate a blank tour object, with blank values (will be used for testing);
     + With name:  
       This is the most basic constructor that the user will have access to, all tours will need a name to progress, the values for long description and short   
       description will be left blank;
     + With name and short description:  
       Similar to the previous constructors, just without a blank short description;
     + With name, short description and long description:   
       Full constructor, consisting of all text fields filled for the walk;
   * **Add/Remove Location:**   
     Takes a Location and adds/removes it into the linked list of locations.

* **Location class:**

1. Description:

A key location within the tour. It consists of:

* + a longitude and latitude for the location;
  + a name of the location where the user is;
  + a short description of the location limited to 140 characters;
  + a time stamp that can be used to calculate a total time for the walk;
  + an array of photos taken along the walk with a maximum of 5;

1. Methods
   * **Constructors:** 
     + Default

Constructs a blank location with default values for the longitude and latitude, and a photograph(again for testing).

* + - With longitude and latitude

Constructs a location with longitude and latitude taken from the phones current location, will prompt the user if he/she wants to attach a photograph.

* + **Add/Remove Photo:**

Adds/removes a file path of a photograph to the array.

* **Communication class:**

This is the class that handles the communication to the server. It will implement the Link interface.

* **Walk activity class:**

This is the class that links the model and the user interface together. From here the user will add locations to the tour with descriptions and a set of up to 5 photos.

## Web program

The Website allows the user to view walks currently stored in the database using the mapping API.

## Table mapping requirements



How does the overall system fit together? What are the pieces?

Significant classes description for each program.

# Dependency Description

UML component diagrams?

http://www.sparxsystems.com.au/resources/uml2\_tutorial/uml2\_componentdiagram.html

## General architecture

How do the pieces interact with each other?

# Interface Description

public interface Link{

/\*\*

\* Connects to a server with a given IP address and port number.

\*

\*

\*@param ipAddr The IP address of the server you wish to connect to.

\*@param portNum The port number of the application that you are using.

\*/

public void connect(String hostName,int portNum);

/\*\*

\* Sends data to the server.

\*

\*

\*/

public void send();

/\*\*

\* Receives data from the server.

\*

\*/

public void receive();

/\*\*

\* Destroys the connection to the server.

\*

\*

\*/

public void disconnect();

/\*\*

\* Pauses the current connection to the server.

\*

\*

\*/

public void pauseConnection();

/\*\*

\* Re-opens a paused connection the server.

\*

\*

\*/

public void reconnect();

}

What is the way each piece allows others to talk to it?

Should provide everything designers, programmers and testers need to know to use the facilities provided by a module.

Interface specification for each class in the system should include:   
The name of the class also modifiers such as public or abstract.   
Classes which it extends (and why).

Public methods implemented by the class.

Properly formatted Java code with Java doc comments

Method bodies empty

# Detail Design

Significant algorithms / Significant data structures

From requirements, identify and write out main operations of system, including start-up/closedown operations (UML has USE CASES for this)

From operation descriptions, use sequence diagrams to show how each main operation would work by calling a sequence of different classes/ methods.

## Sequence Diagrams



## Algorithm Description

## Data Structures

## ER

Especially where classes represent real world objects, there is often a complex data structure in the program made by objects pointing to each other

Readers of the design needs to get a feel for that structure, and nothing we have documented so far gives that feel   
Class interaction diagrams and object interaction diagrams in UML can do this, but often, just drawing the relationships between a set of objects is enough.

# References

# Document History