Walking Tour Server Side Requirements and Specification Analysis

Table of Contents

[Scope of server side aspects of project 1](#_Toc370843795)

[Transportation of Data 1](#_Toc370843796)

[Storage of Data 2](#_Toc370843797)

[Access Control 2](#_Toc370843798)

[Presentation of Tour (Mapping API) 2](#_Toc370843799)

[Platform Setup / Maintenance 2](#_Toc370843800)

[Quality Assurance 2](#_Toc370843801)

[Testing 2](#_Toc370843802)

[Transportation of Data Analysis 2](#_Toc370843803)

[Requirements 2](#_Toc370843804)

[Outline of Design 3](#_Toc370843805)

[Comparison of data set formats 3](#_Toc370843806)

[XML Pros 3](#_Toc370843807)

[XML Cons 3](#_Toc370843808)

[JSON Pros 3](#_Toc370843809)

[JSON Cons 3](#_Toc370843810)

[JSON data set format specification 4](#_Toc370843811)

[Sample JSON format 5](#_Toc370843812)

[Access Control Layer Analysis 6](#_Toc370843813)

[Explanation of requirement for ACL 6](#_Toc370843814)

[Options for ACL 6](#_Toc370843815)

[CMS comparison 6](#_Toc370843816)

[Similarities 6](#_Toc370843817)

[Differences 6](#_Toc370843818)

[Conclusion 6](#_Toc370843819)

[Mapping API 7](#_Toc370843820)

# Scope of server side aspects of project

The greater project has been split into two key areas of scope: the client side android application and the server side application and website. The server side aspect of the project can be broken down further into the following key tasks.

## Transportation of Data

The team needs to work on a method of abstracting the data that is collected on the phone and transporting it to the server for storage/ presentation. This aspect of the project requires close cooperation between the web team and the android team.

**William Arslett** is responsible for the data handling aspects of the server side project and will take the lead on this task. **Harvey Clark** will be our point of contact in the android team for cooperation.

## Storage of Data

The team will design a system for storing the abstract data that is sent to the server. This document will investigate the use of databases and file systems server side for storing data. **William Arslett** is responsible for this task, which is linked in with the other data handling aspects of the project.

## Access Control

The team will need to implement some form of access control layer to control which users are able to publish tours and to associate tours.

This could be achieved using the development framework of an existing content management system. This document will explore the use of a content management system in the implementation of the website.

**Stephen Clasby** will be responsible for the implementation of the content management system/ access control layer as well as the other web/ administrative aspects of the system.

## Presentation of Tour (Mapping API)

The team will need to design and build a system for presenting the abstract data stored on the server in a way that is coherent and attractive to the user. The system will use some form of mapping API to display the tours. **Neil Vicker** will be taking the lead on this aspect of the project with programming support from **William Arslett** and **Stephen Clasby.**

## Platform Setup / Maintenance

The team will need to setup a platform for this software and this document will identify the requirements of the system and compare languages, databases, web servers and other key platforms available. **Stephen Clasby** will be responsible for the administration and maintenance of these services during the implementation and testing stages of the project.

## Quality Assurance

**Ben O'Donovan** is responsible for the quality assurance for both the web team and the android team.

## Testing

Each member of the web team will be responsible for testing his implementation. **Neil Vicker** will be responsible for the overall testing strategy.

# Transportation of Data Analysis

## Requirements

* The system will need to transport data from the phone to a web server in an abstract format, which can be stored in a persistent database.
* The system will also need to transport any image files required by the tours to be stored in a file system on the server.
* The system will deliver the data in a MIME message format
* The MIME message will be delivered using HTTP POST
* The data provided will include
  + The route of the tour
  + The locations of interest including description and images
  + Information about the tour including a long and short description, the distance and the time.

## Outline of Design

The group has already discussed that data will be stored locally to the phone and the user will choose when to upload the data so that network issues do not interrupt the delivery. The android application will produce a MIME message where the data for the tour is represented as a plain text data set and the images will be represented as attachments to the message using standard MIME attachment encoding. This eliminates the need for packaging multiple files.

The MIME file will be sent as a string in the HTTP POST method to the web server. An application on the server will then convert all of the MIME attachments into files which will be stored on the servers file system and the data for the tour will be stored in a database.

The team has two options for the format of the data set. The system could use an XML data set format or a JSON data set format.

## Comparison of data set formats

### XML Pros

* Familiar to development team
* Data is structured as a tree of elements with parameters and content allows for greater detail in data set
* Can be easily formulated in Java Android API
* Can be easily processed by PHP

### XML Cons

* Rigid Structure
* Duplicated tag and parameter labels require more characters and therefor more bytes
* More complicated structure than JSON

### JSON Pros

* Very simple to understand (if well formatted)
* Very minimalist structure uses operators instead of labeled tags therefor considerably less bytes
* Can also be formulated very easily in Java Android API
* Can also be processed easily by PHP and converted directly into an associative array

### JSON Cons

* Elements do not have parameters
* Can be difficult to understand if not well formatted
* Development team less familiar than with XML

The group has discussed the pros and cons of both data sets and has decided that JSON would be the best data set to use for the tours. This is largely due to it being significantly lighter in weight than XML and how easy it is to process in PHP.

## JSON data set format specification

The data set will need to contain a title for the tour, a short description of the tour and a long description, a collection of GPS coordinates for the route, a collection of locations associated with GPS coordinates on the route including images and description, the total time of the route and the total distance of the route.

Table 1 Fields for root of JSON data set

|  |  |  |
| --- | --- | --- |
| Variable Name | Description | Format |
| title | The title of the walk | A string of <30 characters |
| shortDesk | A short description of the tour to be displayed in lists of tours on the website. | A string of <100 characters |
| longDesk | A long description of the tour to be displayed alongside the map on the website. | A string of <1000 characters |
| route | A sequence of GPS locations that describe the route of the tour | A collection of objects representing GPS coordinates. (See Table 2) |
| locations | A set of locations of interest along the tour. | A collection of objects representing locations of interest. |
| time | The number of seconds that elapsed during the recording of the tour. (Not including when paused) | integer |
| distance | The distance of the route of the tour in meters. | integer |

Table 2 Fields for GPS objects

|  |  |  |
| --- | --- | --- |
| Variable Name | Description | Format |
| id | A unique ID indicating the index of the location in the sequence | integer |
| longitude | The longitude of the current GPS location on the route | integer |
| latitude | The latitude of the current GPS location on the route | integer |
| time | The number of seconds that elapsed from the beginning of the tour to this recorded location. (Not including when paused) | integer |

Table 3 Fields for location object

|  |  |  |
| --- | --- | --- |
| Variable Name | Description | Format |
| coord | The ID of the GPS coordinate object that the location is attached to | integer |
| description | A short description of the current location. | A string of <500 characters |
| media | A set of URLs pointing to the images to be associated with the location | A string collection of variable length. |

## Sample JSON format

{

"Title": "My Walk",

"shortDesc": "A walk from grans house to my house",

"longDesc": "This is a walk that I take from my house to my nans. I hope you enjoy it…",

"route": [

{

"id": 0,

"longitude": 345674,

"latitude": 583848,

"time": 0

},

{

"id": 1,

"longitude": 345684,

"latitude": 583848,

"time": 5

},

//LOTS MORE HERE…

],

"locations": [

{

"coord": 5,

"description": "This is where I live",

"media": [

"file1.jpg",

"file2.jpg"

]

},

{

"coord": 17,

"description": "This is about half way",

"media": []

},

{

"coord": 25,

"description": "This is where my gran lives",

"media": [

"file3.jpg"

]

}

],

"time": 45676,

"distance": 23454

}

# Access Control Layer Analysis

## Explanation of requirement for ACL

We have discussed that it would not be appropriate to allow users to use the system anonymously and that we would like to integrate an access control layer which would:

* Require users to register before they are able to publish tours
* Associate every published tour with a registered user

Use of the ACL will make users accountable for the content which they publish, this is particularly necessary in terms of the applications ability to upload photographs.

## Options for ACL

There are two options for controlling user access. The team could develop our own user access control layer from scratch or using a server side framework or the team could make use of the existing access control layer from a content management system. Use of a content management system for Access control will make building the website considerably easier and will also provide us with a system for content management. The three most popular content management systems are Joomla, Drupal and Wordpress.

## CMS comparison

### Similarities

* All three systems use MySQL DBMS
* All three systems are written in PHP
* All three systems are web server independent
* All three systems are platform (OS) independent
* All three systems are XHTML complient

### Differences

* Wordpress does not allow you to extend user profiles with custom properties
* Wordpress does not allow you to create a custom permissions/ role structure
* Joomla does not provide an audit trail
* Wordpress does not provide a login history
* Joomla does not provide database replication

### Conclusion

From this comparison I can conclude the wordpress would be unsuitable for use as an Access Control System because it does not allow developers to extend the user profile.

Joomla or Drupal would both be valid choices, however, Drupal provides some additional features such as an audit trail and database replication which might prove useful. It is also the cms most familiar to Stephen Clasby who will be doing the most work on this aspect of the project.

# Mapping API

The group has studied a number of mapping APIs and has decided to use an API called leaflet. A justification and comparison for this choice has been written up by Neil Vicker.