Server side Design Specification

Table of Contents

Database Design 1

Database UML 2

Data Delivery Design 2

POST Request 2

MIME message 3

Sample MIME message 3

JSON code design 3

Sample JSON format 4

Data Delivery Algorithm 5

Website Design 5

Site map 5

User System Design 5

Interface/ Graphical Design 5

Mapping Design 5

# Database Design

## Data Definition Dictionary

Table 1: Database Tables

|  |  |
| --- | --- |
| Table Name | Table Description |
| tours | A list of walks/ tours which the program will display. |
| locations | A list of geographical locations referencing a record in the tour table, describing the route of the tour as a sequence of locations. |
| pointsOfInt | A list of points of interest along tours referencing a location. |
| photos | A list of photographs referencing a point of interest |

Table 2: List of Walks Table

|  |  |  |
| --- | --- | --- |
| Field Name | Field Description | Field Data Format |
| id | Primary Key (auto increment) | integer |
| title | Title of the tour | text |
| shortDesc | A short description of the tour (<100 characters) | text |
| longDesc | A detailed description of the tour. (<1000) | text |
| hours | The number of hours the walk will take | float |
| distance | The total distance of the tour in kilometers | float |

Table 3: Location Table

|  |  |  |
| --- | --- | --- |
| Field Name | Field Description | Field Data Format |
| id | Primary Key (auto increment) | integer |
| walkID | Foreign key, referencing the id field of the tour that the location is associated with | integer |
| latitude | The latitude map reference for the location | float |
| longitude | A detailed description of the tour. | text |
| timestamp | The time in hours from the beginning of the tour | float |

Table 4: Place description table

|  |  |  |
| --- | --- | --- |
| Field Name | Field Description | Field Data Format |
| id | Primary Key (auto increment) | integer |
| locationID | Foreign key, referencing the location that the point of interest is referencing | integer |
| description | The description of this point of interest. (<500 characters) | integer |

Table 5: Photo Usage Table

|  |  |  |
| --- | --- | --- |
| Field Name | Field Description | Field Data Format |
| id | Primary Key (auto increment) | integer |
| placeID | Foreign key, referencing the point of interest that the image is attached to | integer |
| photoName | The name of the jpg file for the photo (without “.jpg” suffix) | text |

## Database UML

TODO: Create UML diagram for database design

# Data Delivery Design

## POST Request

All the data will be packed into a single string, which will be transmitted from the android phone in an HTTP POST request. The value will be paired using the key “message” and will be accessed in PHP using the $\_POST[‘message’] handle.

The request will be made to the file upload.php which will be stored in the root of the website. All requests will be recorded in by upload.php in a file called log.txt in the root of the site as per the testing strategy.

## MIME message

The mime message will contain a “From” field which will store the user’s name and email (From: User’s Name <user@usershost.com>) and the name of the tour in the “Subject” field (Subject: My Tour). Writing the tour name to the subject field will allow the server to record the process in the log, even if there is an error with the JSON code.

It will include a MIME version declaration of version 1.0 (MIME-Version: 1.0) and a multipart content type declaration (Content-Type: multipart/mixed; boundary=”part”). The JSON code will be stored in the only text type part. All of the images will be stored as attachments in jpeg format.

## Sample MIME message

From: John Doe <example@example.com>

Subject: TOUR NAME

MIME-Version: 1.0

Content-Type: multipart/mixed;

boundary="part"

--part

Content-Type: text/plain

JSON CODE GOES HERE

--part

Content-Type: image/jpeg;

Content-Disposition: attachment;

filename="file1.jpg"

jgfc,jbjytf,nmvk-0987y6trfgi9876trdfvbhjytrfdc

--part--

## JSON code design

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 6 Fields for root of JSON data set

|  |  |  |
| --- | --- | --- |
| Variable Name | Description | Format |
| title | The title of the walk | A string of <30 characters |
| shortDesc | A short description of the tour to be displayed in lists of tours on the website. | A string of <100 characters |
| longDesc | 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. (See Table 3) |
| 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 7 Fields for location/ route 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 8 Fields for POI 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…

],

"pointOfInts": [

{

"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

}

## Data Delivery Algorithm

1. Android app produces JSON text representation of tour objective data structure
2. Android app packages JSON text and images (as attachments) in a MIME message format
3. Android app uploads MIME message to a PHP server page over the internet using the HTTP POST method
4. The PHP file unpackages the image attachments and saves them to a temporary directory with a unique name (ie tmp/05012013my\_tour). It records success/ failure to the server log and if successful, will progress to step 5.
5. The PHP file extracts the JSON code and produces an SQL query to insert the data into the database. It records success/ failure to the server log and if successful, will progress to step 6.
6. The files are moved from the tmp directory to a permanent directory with a unique name. (The primary key of the tour in the database)

# Website Design

## Site map

Site structure…

Tree diagram…

## User System Design

Use Cases

Details of implementation…

## Interface/ Graphical Design

Web page layouts…

Wire frame diagram…

## Mapping Design

Mapping API design

Layout design

Algorithm design for processing data from database for API