LOCATION BASED EDUCATIONAL WEB SYSTEM DESIGN AND IMPLEMENTATION

A thesis submitted
to Kent State University in partial
fulfillment of the requirements for the
degree of Master of Science

by

Xu Han

May 2017

© Copyright

All rights reserved

Except for previously published materials
Thesis written by

Xu Han

B.S., Tianjin University Ren’ai College, China, 2012

M.S., Kent State University, USA, 2017

Approved by

Dr. Cheng-Chang Lu  ,  Chair, Master Thesis Committee

Dr. Richard E. Ferdig  ,  Members, Master Thesis Committee

Dr. Austin Melton  ,  Members, Master Thesis Committee

Dr. Xiang Lian  ,  Members, Master Thesis Committee

Accepted by

Dr. Javed Khan  ,  Chair, Department of Computer Science

Dr. James L. Blank  ,  Dean, College of Arts and Science
TABLE OF CONTENTS

LIST OF FIGURES ............................................................................................................. V

ACKNOWLEDGEMENTS ................................................................................................. VI

CHAPTER 1 INTRODUCTION ............................................................................................. 1

1.1 Location-Based Web System ...................................................................................... 1

1.2 Requirements of Web System .................................................................................... 1

1.2.1 Programming Language ....................................................................................... 2

1.2.2 MySQL Database .................................................................................................. 2

1.2.3 GeoJSON Database ................................................................................................ 2

1.3 Design System ........................................................................................................... 4

1.4 Implement .................................................................................................................. 5

1.5 Test and Optimization ............................................................................................... 5

CHAPTER 2 DESIGN SYSTEM ........................................................................................... 6

2.1 System Models Visualization .................................................................................... 6

2.1.1 Access control system ....................................................................................... 6

2.1.2 Ajax method to process data .............................................................................. 10

2.1.3 Adventure track ............................................................................................... 12

2.2 Work Prepare ........................................................................................................... 16

2.3 Database Design ...................................................................................................... 17

2.4 Models Design ......................................................................................................... 21

2.4.1 Map Model ........................................................................................................ 21

iii
2.4.2 Data Model ................................................................................................................. 22
2.4.3 Adventure Track Model .......................................................................................... 24
2.4.4 Citizen Science Model .......................................................................................... 25
2.4.5 Badge Model ....................................................................................................... 26
2.4.6 News Model .......................................................................................................... 28
2.4.7 Picture Model ........................................................................................................ 28

CHAPTER 3 IMPLEMENT AND OPTIMIZATION ......................................................... 30
3.1 Using Mapbox Implement Maps ................................................................................. 30
3.2 Collect Multi-Structure Data .................................................................................... 31
3.3 Implement the Adventure Tracks .............................................................................. 32
3.4 Implement the Badge System ................................................................................... 33
3.5 Implement News System ............................................................................................ 34
3.6 Implement upload various of Pictures ........................................................................ 34

CHAPTER 4 TEST AND OPTIMIZATION ................................................................. 36
4.1 Test ............................................................................................................................ 36
4.2 Optimization .............................................................................................................. 36

CHAPTER 5 CONCLUSIONS AND FUTURE WORK ............................................. 39
5.1 Conclusions ............................................................................................................... 39
5.2 Future work ................................................................................................................ 39

REFERENCES ................................................................................................................. 41
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GeoJSON example</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>User role case diagram</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>Logical model for administrator</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>Classic vs. Ajax web application model</td>
<td>11</td>
</tr>
<tr>
<td>5</td>
<td>Adventure Track Map sample</td>
<td>13</td>
</tr>
<tr>
<td>6</td>
<td>Multiple choice question table structure</td>
<td>14</td>
</tr>
<tr>
<td>7</td>
<td>Adventure Track question shows on application</td>
<td>15</td>
</tr>
<tr>
<td>8</td>
<td>Entity Relationship Diagram</td>
<td>18</td>
</tr>
<tr>
<td>9</td>
<td>GeoJSON structure of one point</td>
<td>19</td>
</tr>
<tr>
<td>10</td>
<td>Public information map example</td>
<td>20</td>
</tr>
<tr>
<td>11</td>
<td>Heat Map example</td>
<td>23</td>
</tr>
<tr>
<td>12</td>
<td>update the first login badge sample</td>
<td>27</td>
</tr>
<tr>
<td>13</td>
<td>User answer data structure</td>
<td>32</td>
</tr>
</tbody>
</table>
ACKNOWLEDGEMENTS

I would like to thank all the people who contributed in some way to the work described in this thesis. First and foremost, I thank my academic advisor Dr. Cheng-Chang Lu and co-advisor Dr. Richard E. Ferdig. They give me this precious opportunity to be part of the ParkApps research. Their guidance helped me in all the time of research and writing of thesis. During this two years, they not only gave me very professional academic advising, also encouraged me when I was facing challenges in the research.

I would also like to thank my friends who helped me get through two years of research. They are Chang Shu and Xiqian Han. Chang gives me many helps when I was working in the mobile applications’ APIs; Xiqian gave me support about the user interface in the research. Finally, I would like to thank my parents who support me with all their hearts. This accomplishment would not have been possible without them. Thank you.
CHAPTER 1

Introduction

1.1 Location-Based Web System

ParkApps is a location-based web system (LBWS) for teaching and learning in natural settings, like national, state, and local parks. It is a collaborative project between Kent State University, the Cuyahoga Valley National Park (CVNP), and Cleveland Metro Parks (CMP), that gives visitors access to real-time information as they explore various geological or historical landscapes and as they encounter unknown plants and animals. LBWS will use geo-data to integrate location-based information for mobile applications, and provide location and educational information for scientists.

1.2 Requirements of Web System

This thesis focuses on introducing three functional models. One is the LBWS map model, which processes geographic data for to show thousands of points on the geographic information system (GIS). It is important to notice that in many cases updates are performed by remotely connected users and that, moreover, the way spatial data is usually presented to users is through maps [1]. The second model is the Adventure Track Model, which enables users to explore content within the trails. The mobile application will send notifications to users when they move into the geofence for each point. Then the mobile application will list questions for users to answer. LBWS can edit trail information and update data. It also supports the processing of
question data and users answer data by using the Ajax method. The data analytics is the part used to analyze user answer data, helping the research team to get useful information.

1.2.1 Programming Language

PHP is a reliable and efficient programming language with which to build web systems. Its clarity in design, well organized modules, and superior upkeep of various technologies, make it the most popular language in the online industry today [2]. It is used to build highly interactive and appealing dynamic web pages. Its popularity and credibility can be gauged by the fact that reputed organizations like Harvard University and popular social networking websites, like Facebook. LBWS requires high interactive abilities for mobile clients. PHP’s advantages will make the development more efficient and easy to maintain.

1.2.2 MySQL Database

MySQL is an open-source relational database management system (RDBMS). It runs as a server providing a very fast, robust SQL (Structured Query Language), multi-threaded and multi-user access to a number of databases. MySQL is globally renowned for being the most secure and reliable database management system used in popular web applications like WordPress, Drupal, Facebook and Twitter. In this system, we need to save thousands of points and pictures into the database. MySQL’s characteristics make system process data efficiently.

1.2.3 GeoJSON Database

In this research study, we focus on processing and displaying geo-data from mobile applications in real-time. Compared to the new technique with which to process data by using JavaScript Object Notation (JSON), the RDBMS process makes the system very sluggish. In
order to improve the data processing performance, a professional method is needed for geo-data to storage, indexing and updating [3].

GeoJSON format is an open standard format for encoding a variety of geographic data structures, based on JSON. Its structure is like figure 1:

```json
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "type": "Point",
        "coordinates": [102.0, 0.5],
        "properties": {"prop0": "value0"}
      },
      "type": "Feature",
      "geometry": {
        "type": "LineString",
        "coordinates": [
          [102.0, 0.0], [103.0, 1.0], [104.0, 0.0], [105.0, 1.0]
        ]
      },
      "properties": {
        "prop0": "value0",
        "prop1": 0.0
      }
    },
    {
      "type": "Feature",
      "geometry": {
        "type": "Polygon",
        "coordinates": [
          [100.0, 0.0], [101.0, 0.0], [101.0, 1.0], [100.0, 1.0], [100.0, 0.0]
        ]
      },
      "properties": {
        "prop0": "value0",
        "prop1": {"this": "that"}
      }
    }
  ]
}
```

**Figure 1 GeoJSON example**

GeoJSON simplicity and loading speed make the system processing huge geographic data very fast. Although it need a little more storages, in general it is more advance than just uses traditional SQL database to process different conditions’ data [4]. The reasons mainly from the following aspects:
• Using SQL language to query more than thousands points in the database and load them on the map will spend more than four seconds. That will cause the map crash, since geo-data will be saved more and more in the future.

• In this system, we have three kinds of maps. The first one is public points map, which is used to display six hundred public points in the National Park. Public information map uses the static geo-data, it has less communicated with web server. Thus we use loading the static GeoJSON file to process data, not only reduce pressure when data processing with server, it also gives a good user experience when using the map [4].

• The other two kinds of maps are private points map and adventure tracks map, both of them need to communicate with server side frequently. We save each user’s data in one GeoJSON file, and these files’ average size is 50KB. That’s means twenty thousand users’ geo-data just need 1GB storage. Compare with the server’s huge storage, use less storage to improve the map’s fast response is very important for our system.

1.3 Design System

To design whole location based educational system, we need several important steps to achieve its requirements. Entire system frame is the first step to start the project. It includes main function models design, database design, logical process design and extensibility considered. This stage is used to develop system summary, draw clear picture of model frame and determine each model’s relationship. In this project, we use Unified Modeling Language(UML) to visualize the design of system. Then based on models and their functions to design system’s database, we use entity relationship (ER) diagram to illustrate the logical structure of database. We also considered the system’s scalability both database and models. With the new features require, for
example, the public map need new feature to display amenity in the points. We set several empty properties in the points’ table of database and set the pop up content window can show more extra information when we need.

1.4 Implement

In our research, we use multi technology to implement different models. Using HTML and CSS to build up basic static UI; use JavaScript and jQuery to archive the processing data dynamically; use PHP language to control models logical function and MySQL database. We designed many algorithms to help the development more efficiently. The details implementation will introduce in the chapter 3.

1.5 Test and Optimization

Software testing helps in identifying and fixing bugs before the software becomes operational, the risk of failure can be reduced considerably. In a system, many modules do not work along, they need to integrate and function with other existing modules. Thus, to keep each module can work well is very important for our system. In our research, we use several testing methods to test the system. They not only help us finding the bugs, they detect the mobile battery life draining fast when using the mobile application. In the chapter 4, it will introduce two testing methods which we use in our research project, and explain the method how we optimize the mobile battery life by implement a battery saving algorithm.
2.1 System Models Visualization

In this chapter, we will present the entire process how to design the system. At first, we need to determine the user roles and functional models by the requirements. We use UML (Unified Modeling Language) to visualize the entire system design process. UML is a kind of diagram enabling to specify, visualize, construct and document artifacts of a software system [5].

2.1.1 Access control system

The access control system is a component in charge of ensuring data protection within a data management system. Basically, data access is controlled through a set of authorization rules stating who can access which resource for doing what [7]. In this system, we have three user role:

- Park Administrator: users under this role, they have the most authorities to manage the system. They can edit the public information map, it includes create new public points and update the points information, manage the adventure tracks information and edit the questions in the adventure track, collect the users’ location data and users answer data, and manage the user roles.
• Normal User: all normal user who have registered in the system can use the private map’s features, they can create and update their own points and see them on both web server and mobile application.

• Citizen Science: citizen science is a special role in this system, they can create their private citizen science project and add science information in it.
The figure 2 is the user roles case diagram:

Figure 2 User role case diagram

Functional models will integrate datasets with logical processing to archive their requirements. In this system, they include maps model, data collections model, educational
model adventure tracks and citizen science. Geo-data can be loaded from two side in map. While administrators access system, they can use map to update data which data is saved in database. Public points show on the public information map that everyone can see the points’ information through the website. Normal users use private map to update its own points by using GeoJSON files. Then when users visit park in each time, they not only can see the public points on the map, but also their private markers on the map. Adventure Track is used to help visitors learn more about park when they use mobile application to visit park. Citizen Science is a sub-project for scientists use for their research. Its structure is like adventure tracks. Data collection is developed to help research team to analysis anonymous users’ questions response and find visiting rate in the park area in different seasons.

The example of logical model for administrator shows in figure 3. As admin have the highest permission in this system, they can create, update and delete all points data and determine to prove which point is public point which can display on the public map.

Adventure track model includes many points, admin need to manage points, give points a special key to mark which adventure track they are classified. Administrator also can create and update adventure tracks’ information. It includes adventure track name, introduction, description, difficulty level, track length, track image and track badge, etc. At last, admin can determine which adventure tracks can be approved to publish in the mobile application.
For the education purpose, adventure track model is the main part to archive the functional requirements. Map helps to display points in the current track, different kinds of questions can be created and updated in the points or based on the list.

### 2.1.2 Ajax method to process data

In this part of maps, we use the Google Maps APIs to set their framework. While questions need to be updated, the third JavaScript library- jQuery will use Ajax function to send request to web server.
Ajax is an acronym for asynchronous JavaScript and XML, it is not actually a single technology but a combination of several different technologies. The term was coined by Jesse James Garrett in 2005 [8]. Ajax applications use the XMLHttpRequest object in the web browser to retrieve data from a server or service asynchronously. This means data can be loaded into a web application in the background, the information can be partially updated the screen without having to reload the whole web page. Figure 4 shows the difference between the classic method and the Ajax method to transport the data.

![Figure 4 Classic vs. Ajax web application model](image-url)
Ajax technology makes the questions update data from server more fast and points get the response on the map without refresh whole map. It is different with users’ map which need update whole map’s information, adventure track load static point data and update single point’s data at most of the time.

2.1.3 Adventure track

The figure 5 is an example shows the adventure track map. Map use different colors to define three kinds of points. Red points are current track points; yellow points are normal information points which not belong to this track; blue points are learn as you go points. Admin can edit the questions in the points after these points are saved in this adventure track. All this track’s points are queried from database, and list on the content page. While user click red track point, it can pop up a question list window which shows current point’s amount in each kind of questions. Admin adds and edits questions through the link. Using map based to display points and their questions can make administrator managing questions easily and flexible.
Figure 5 Adventure Track Map sample

These various of questions will be saved in different question type tables in database. Each question table has its own structure. For example, text/image question support user upload photos to answer question, this kind of question table need a picture attribute; multiple choice question support user to choice multiple options in the application, this kind of question table need two special attributes to record the right option and left option, below figure is the multiple choice question data table.
Figure 6 Multiple choice question table structure

All questions can be saved in points, also can saved by track’s list. Questions will show on the mobile application when users click the points or see the track questions list.

Users also can get notification when they are in the adventure track point’s geofence area, the radius of geofence can input by admin. Then the mobile application sends a request, web server get the request and use APIs to send multi-levels structure data to application. All the data is in real-time, it makes sure users can get the latest question information from server side. The figure 7 is two examples the questions show on the mobile application. Left figure is the single choice question, right figure is the multiple choice question.
Figure 7 Adventure Track question shows on application

In entire system, because of models’ special feature, need system store data in different ways. Traditional relation database MySQL is used process small data in a high frequency or simple relationships in each other. Most of data we use database to save in our system. GeoJSON file is used to save multi-structure data and special large geo-data, which will be used to show map information and collect user’s answer data. For example, location points and users’ information use database to store data. They both have relationships with each other and we use SQL language to get and update them from MySQL database easily.

User table saves all of the users’ information in the system, there are fifteen attributes in this table. The first attribute is the primary key – user_id, to identify users, the username and
password attributes help users can access system and manage their account, the user_role is the users’ level, it includes three different levels users to use system. Administrators have the highest level in the users, they can edit all of the points information and control the role management, change user’s permission to use system. They also can edit adventure tracks and their questions, see all the data collections. Normal users can create and update their points and see the basic adventure track information. Guest just can see the public information map in the web, they have no points and cannot access in the system. Location table saves all the points information, they have coordinates, created by which user and point’s type.

Map cluster points and users’ answer data use JSON file to save. Map cluster points need store geographical information of thousands of points to display them on the map, and users’ answer data have complicated data structure. To save them in JSON file will make sure they can load data by read file single time not like each time to search specific information by using SQL language.

2.2 Work Prepare

In this project, we use multi program skills to archive its requirements. PHP is used to develop back-end server system; JavaScript and the third Maps API library – MapBox to build the public and private information points, and Google Maps API to build the Adventure Tracks maps; HTML/CSS and Bootstrap to set the front-end UI. MySQL and GeoJSON files as the Database to save all of the data in the system [6].

An important characteristic of PHP is the flexible develop and high performance at software bundle – LAMP. LAMP is an archetypal model of web service stacks, named as an acronym of the names of its original four open-source components:

- Linux operating system
• Apache HTTP Server
• MySQL relational database management system (RDBMS)
• PHP programming language

While service stack environment has been set up, PHP and MySQL can work efficiently in each other [9]. In PHP, there are lot of functions can operate MySQL, for example, query() is used to execute SQL language in the MySQL database; fetch_assoc() is used to get the result from SQL execution and format them as a string array data structure; num_rows function is used to return the number of SQL execution, etc.

2.3 Database Design

When the service environment is set up, and all the design work have been finished. Then we need to start the back-end work, the first thing is to create the whole system database. To make the complicated relationships more clearly, we use the ER diagram to build the tables. Figure8 is the ER diagram in this system.
Figure 8 Entity Relationship Diagram

ER diagram includes several attributes we will use in this project:

- The rectangle module is an entity, it is an object in the system.
- The diamond shape represents an action, it shows how two entities share information in the database.
- The oval represents attributes of an entity. A key attribute is the unique, distinguishing characteristic of the entity.

Some tables have relationships in each other, their relationships can be one-to-one, one-to-two-many, and many-many. For example, we set the Location table’s User_ID as a foreign key.
which is the User table’s primary key- user_id. In this way, these two tables set up connected each other, we can base on user id to find all this user’s points and get each point’s creator. In our database, we create many foreigner key in the tables to connected them each other.

To show the different maps, we will mainly use GeoJSON file to loading the geo-data. The multi-levels structure geo-data will show by below figure:

```json
{
    "type": "FeatureCollection",
    "features": [
        {
            "type": "Feature",
            "geometry": {
                "type": "Point",
                "coordinates": [
                    -81.4899204,
                    41.50191905
                ]
            },
            "properties": {
                "title": "Acacia Clubhouse",
                "description": "",
                "marker-size": "medium",
                "marker-symbol": "star",
                "marker-color": "#ff8888",
                "category": 1348,
                "p_id": 1,
                "img_id": 239,
                "images": ".\ uploads\ / f803f1533dbc7df93c0ca825bfe2171e.jpg",
                "img_num": 1,
                "attribute": "",
                "icon": {
                    "iconUrl": ".\Categories\pictures\structures.jpg",
                    "iconSize": [
                        30,
                        41
                    ]
                }
            }
        }
    ]
}
```

**Figure 9 GeoJSON structure of one point**

The attribute coordinate will locate the position on the map; title is the point title when is pop up to display to users; the attribute p_id is the point id saved in the database, it will help to
connect the point with point’s question for adventure track. We can display the points’ shape by using the attribute icon. In this attribute, it includes the iconUrl which is the picture path, iconSize which show the picture’s size on the map. When JavaScript loading the GeoJSON file, the map will show the points like below figure:

![Public information map example](image)

**Figure 10 Public information map example**

Adventure Track table is used to store all the adventure tracks and citizen science information. It has relationship with six kinds of questions table. They are Text/Image Question, Fill-in Blank Question, Single Choice Question, Multi Choice Question, Correct Order Question and Match Question. All the questions support displaying multi pictures and text on the mobile side. Each time mobile application use API to request question data, server side use Inner Join to select multi tables to execute the query operations. Using Inner Join select statement can make the query more efficient in the circular statement than use separate query in these tables in each time. Users’ answer data and information map points data will be saved in the GeoJSON files. This is an optimized method to process this kind of data. They all have more complicated structure than other data. Most of them need more than three tables to be connected.
To reduce the pressure of the database and make querying data much faster, let the data save in the JSON file when they are created, make sure all the relationships are stored as the JSON format in it. In this way, PHP use function `file_get_content()` to read the JSON file and use `json_decode()` to transfer JSON format to array structure in PHP. JavaScript can read JSON format very fast, they even do not need changing format, can load the JSON file by using `loadURL()` directly. Thus, these two kinds of database can work divided to process different conditions’ data, in special condition when we build the complicated maps will use both database, get useful data by using better choice algorithm to coordinate them work together. This algorithm will be introduced in the test and optimization section.

2.4 Models Design

System models are the conceptual model as a result of system modeling that describes and represents a system. They help the analyst to understand the functionality of the system and models are used to communicate with users. Moreover, these models show different perspectives, describe the overall behavior of this system. In our system, there are seven models to archive system’s function requirements. They are Map model, adventure track model, citizen science model, badge model, data model, news model and picture model.

2.4.1 Map Model

Map model is designed to build all maps and display useful information on different kinds of maps. Using map model can make setting same architecture map easily. To change some parameters in the functions of maps model while the maps have same structure, this method will reduce process time when we develop it. For example, we have three maps to display different kinds of points, they are public map, public information map and private
information map. They use same structure map based on Mapbox APIs, their distinctive is the points data. To deploy these maps, we only need to change the data on them. Thus, we changed the interface of loading data, let public map to read static GeoJSON file which saved the four hundred public points’ basic information; let public information map to read database and get more information from it, then send all of the public points information to the public information map; let private information to read the GeoJSON file which get the private users’ data from database. In this method, system need to create two different structure maps, and change the loading data if they use same map structure.

2.4.2 Data Model

Data model is responsible for collecting various of data in the system. For research purpose, we collect users entire answer data in adventure track and citizen science, also collect each single point’s data in them. Moreover, we collect users’ location when they use mobile application in the park anonymously.

A standard users and points coordinate is like this:

Latitude: 41.50191905  Longitude: -81.4899204

Location data will be saved in database as string format, and display them on a heat map. Heat map(figure 11) shows each area’s visiting frequencies in the park. We use red color to represent the area is visited more than two hundred times, orange color to represent the area is visited between twenty to two hundred times and green color to represent the area is visited less than twenty times. It also can get the visiting data by choosing different moth. To find when the people like to visit the park and which places are the most popular for people to visit.
The other anonymous users answer data will be saved to the JSON files. These answer data include six types of questions, and each type question has different question structure. SQL language to save users’ answer data need using many times SQL query language. This kind of datasets covers the information from many database tables, they need using user table, location table, adventure tracks table and many different question tables to connected to query. When system use this kind of multi-tables to query in a high frequency, the data will be very slow to collect. As the data more and more in the future, the system will be crashed by using the data collection model. To solve this problem, we design a JSON format with multi-structure, and implement it when the data have been initialized. Most of the data collections will be saved in excel files automatically. In our research, we use PHP’s built-in function readdir() which to search all files name in the directory, it will return the name of files in each time, then use PHP’s built-in function file_get_contents() to get the files’ content. In order to read more than one thousand files much faster, we rename all the files’ by their user id and create date by using
special format, based on user id to get all useful information such as question title, location, question type, user’s response, etc. More data collection details implement method and the will be introduced in the implement section.

2.4.3 Adventure Track Model

Adventure track model is the most important model in the system, it is the part which to archive the main educational purpose in it. Admin use this model to create adventure track’s information which will display on the mobile application. They also can control the time period to show on the mobile application by using timestamp. For example, admin can set show this adventure track from March first to end of May. At these period, visitors can see some special landscape at this season. Badge model connect to adventure track model in this place, a specific badge can be added in this track which will be earned by users who have answered all questions and traveled all points in this adventure track. Each adventure track includes some sub-models. Data visualized model can show the questions type amount by using pie chat. We use the third JavaScript library canvasjs to implement them. Canvasjs can use JavaScript features which display on the web browser dynamically. Map list model use Google Maps APIs to build the basic map. As the previous section we introduce, most the map need update data when user use mobile application. Public points location information and all types questions information need to load on the map.

In this map, we designed a point sorted algorithm to help admin to control the points order to show on the mobile side. Points’ default order is sorted by the time they were created, use this algorithm, admin can set each point’s display number. They will be saved in a temporary array, and we will use quick sort method to sort this array, send it to mobile side. When user
click the point on this adventure track map, it will pop up this point’s question information. Questions’ information can be edited and delete by admin on this place. Admin can edit question’s tile and description which will show on the mobile side when visitors walk in the is adventure track area. Right or wrong response will show to users after they answered the question, this information also can be edited in question model. In this part, question can show multi pictures to help user to answer the special question. We use single database table to store one question type, each question type requires different structure, different tables can keep each type’s special characteristic and make the query questions much easier. All the interactive data use Ajax method to communicate with server-side script, it makes the process map information and question information data without reloading browser.

We also design and implement point order algorithm to help sort adventure track points, admin can determine the points order in the track which can help visitors to learn the track in a right sequence.

2.4.4 Citizen Science Model

Citizen science model has some aspects are similar with adventure track. They use the same basic model to develop. Citizen science role is different with park admin and normal user. They have special permissions to create their citizen science project. Citizen science has totally different purpose with adventure track which is used to provide a method for park visitors to help scientists create new knowledge. In this research, they can create private adventure tracks and set questions in them. In their own project, visitors use mobile application to find the published citizen science project, and answer the questions which help scientists can useful information and analysis them by using data collection. All these adventure tracks will not show to normal users before park admin to approve them. While admin think this is a legal project, it will be published
to the users. In the future, we will link some adventure track with a specific citizen science project, that will have a logical connection, help users can learn some knowledge from easy level. When users finish an adventure track and get its badge, they can find the next citizen science automatically to continue their exploration.

2.4.5 Badge Model

As its name, badge is a distinctive emblem worn as a mark when users archive some specific requirements can earn it. In badge model, admin can create, issue and verify digital badges. It is an incentive mechanism to encourage user to explore the park to find more adventure tracks and answer their questions. In badge system, we have for kinds of badges. They are adventure track badge, park visits badges and distance hiked badge. While user first use mobile application to login the system, it will send a notice that you have earned your first badge -first login badge in this park. If user archive some specific requirements, we will set a badge criteria verify automatically. The figure 12 is the user interface how the admin to update a badge.
Figure 12 update the first login badge sample

All the badge’s information and their criteria are saved in a single badge table. Admin uses badge system can set badge criteria in it. For example, admin can create an adventure badge which users must to answer all questions in it and explored all the points, then they can earn this badge automatically. There are two ways to set the adventure track badge, one is admin set the badge criteria to be an adventure track badge and link the badge to the adventure track name. The other one is that adventure track model can create its default badge after admin create the adventure track, this badge use default badge icon before admin to edit it. Admin also can set a badge multi criteria let users finish it. We have distance hiked badge which is to encourage users walk or run in the park, when you start to explore park and reach some requirements like finishing hike 5 miles, it will reach the 5 miles’ criteria then get this badge. Badge system will
continue to record user’s distance when the finish the previous badge. Users can get a better badge after system calculate their whole hike distance.

2.4.6 News Model

News model is designed to help admin ca push news to the website. This kind of news can be watched by all the visitors who use the website. Only admin have the permission to edit and update the news. Picture model can provide API to let admin to upload news’ picture. News order is sorted by the news create date. when admin approve it, then it can be published on the news page.

2.4.7 Picture Model

In the system, many models need upload pictures function. Public information map need that each one point can upload multiple photos; adventure track need to upload some questions’ picture and user can upload picture to answer some specific questions; badge model need to upload badge’s icon and news model need a function to upload news’ picture. In order to make the development more efficient, we develop the picture model to help user and admin to upload different purpose photos. Most of the upload code will be in the picture model which can be re-used in many times. We use HTML and PHP multi technology to develop this model.

Form is a HTML basic tag which is used to collect user input data, it has a enctype attribute to specifies how the form-data should be encoded when submitting it to the server. When the file is uploaded to server side, it will be saved in a temporary place as a cache in server. PHP use the built-in function $_FILES["file"]['tmp_name'] to get the uploaded picture’s temporary name, then use move_uploaded_file() function to put the photo in specifies place. At last, the picture’s path will be saved in the picture table in the database. Different model use
picture model’s API to get the uploaded photo and just need to rename it to a specifies place. Models show the picture can query the picture table by using picture id and other attributes.
CHAPTER 3
Implement and Optimization

This chapter gives more details how the models are developed and implement. Moreover, we will introduce some optimization methods to improve the performance of system.

3.1 Using Mapbox Implement Maps

Mapbox is a Map APIs based on JavaScript, it is built on vector maps, an advanced approach to mapping where data is delivered to the device and precisely rendered in real-time. In our research, we need to set the launch coordinate which is the first locate place in the map, and use addLayer(L.mapbox.tileLayer('mapbox.streets')) to build the basic static map. Then we need load the points data on the static map. In Mapbox, it has a built-in function L.mapbox.featureLayer().loadURL() to get the GeoJSON file by using the path. At the start of the project, we try to use SQL language to get the points information, and use PHP to manage MySQL database to put the point data on the map one by one. Using loop to get the data and send data on the map is a very inefficient way. In this method, the map need more than five seconds to display all points on the map, and each time to update the points data will make the map reload for some seconds. It is a bad user experience for a map system.

For display the dynamic map efficiently, we use another method to build the dynamic map more efficient. That is using JSON format to initial the points data and let JavaScript to read the file directly. JSON is a format for storing and transporting data. It is best for use of data in
web applications from web servers because of JavaScript which can read JSON more quickly than other transfer methods.

Not like query point in the database and send them to the JavaScript one by one, this method uses PHP function file_put_contents() to write the useful data in a JSON file in JSON syntax. JSON syntax is derived from JavaScript object notation syntax, it has a compact architecture and easy to read. In this way, Mapbox APIs to read the JSON files and load points data on the map will use less than one second.

3.2 Collect Multi-Structure Data

In this system, most of the complicated data is stored in the JSON file. It avoids that to get single piece of data by query several tables in the database in many times. When the data is initialized, use PHP function to write it in JSON file can reduce the interaction with database. Make sure the JSON file can get the entire useful information and PHP reads it easily, that is means we can collect more information in same time.

It is different with using the SQL query to get data, we need to set the structure of data clearly, PHP use json_decode() to change the entire data to PHP’s array. Based on the structure when the JSON file is created to get the useful data. Figure 13 is the answer data structure, the attribute “pid” is the point id which can get the adventure track name, question id and question title; the attribute “q_type” is question type which can be get by using $val['questions'][0]['q_type']; the attribute “u_response” is the user’s answer data which will be saved in last column in the excel. The structure of user answer data is based on question’s type, multiple choice question records several options in an array in one column. Users also can answer question repeatedly, each column save one answer data, the attribute “response” will be created by the answer time automatically.
3.3 Implement the Adventure Tracks

We use role permission to control different user use the adventure track function. Normal user and citizen sciences role just can see the basic information of a published adventure track; admin have a full permission to control adventure track.

When admin use adventure track create function to add a new track, it just has the basic track’s information. Maps data and questions data are both empty, need admin to input in it. We have set some necessary restricts when tracks are created, track name, track description, track difficulty levels and track length are disable to write empty. It makes sure the tracks have the basic information and let the points and questions use tracks’ some information when it displays on the mobile application.
For the questions in the adventure tracks, we design the point order algorithm to help admin give the track’s points a correct logical order. First of all, points data is queried from database by using the parameters point’s track_type=adventure id and put them in the order page. Next admin input the correct order to the points and submit them to the back-end server. Finally, the point id with its order are saved in the database and sort them to the application.

All these questions have different structure in the database, they are saved in different tables. This means we cannot load the questions information on the map by use PHP and MySQL. PHP uses function to query data to the map will reload the page in each time. To solve the high frequency to fresh the web page, we use ajax method to transport the points data between server side, database and the user interface. In each processing data, for example, to show the different questions’ amount on the map will set a click listener by using JavaScript function onclick(), when this event is triggered by user, we use post() function send the request to the server without reload the page [10]. And using PHP to query the specific data from database. When PHP gets the result of the data from database, our server sends the JSON data to the map. In the adventure tracks, most of the transporting data are used this method. Thus user to click map or update some questions will be very efficient.

3.4 Implement the Badge System

When we start to design the badge model, the requirement is simple. It just has the adventure track badge in the system. We design badge model do not need considering update the badges and set the badges’ criteria. Then more kinds of badges are required in the system. Badge system helps admin to manage badges easily, it can add new badges and update the badges by using this system. Admin need to upload the badge’s icon firstly, then input badge’s name, badge’s description and badge criteria. Each badge must have at least one criteria. Badge also
can have multiple criteria. PHP sets each criteria verification in the back-end, when badge’s criteria is triggered, user’s information and the badge’s information will be saved in the earn badge table of the database. Users can see their own badges in their badge page.

3.5 Implement News System

News system is a basic management system based on control the database to insert, delete, update and query news. Admin need to input the title, author, link and content in one news. PHP uses built-in function date('Y-m-d H:i:s',time()) to get the currently time and set this time to the news’ create date. The format of the news’ content is controlled by the third JavaScript widget-nicedit.js, it helps admin to edit the content easily, and transfer web content to the html code. When admin click the ispublised checkbox, this news will display on news page.

3.6 Implement upload various of Pictures

Picture model is the most widely used in the whole system. User’s profile page need uploading photo in their photo box; maps need uploading single or multiple photos in their points; adventure track need to display the image questions by uploading the questions’ photo and users can upload picture to answer some special questions; our identify function need uploading users’ photos to iNaturelist server to get the photo’s information. In order to make the reuse uploading picture efficient, we design two function to help uploading the pictures. One is an uploading function class for the server side. Most of the pictures are uploaded by the HTML \texttt{<form>}, when user submit the form, the picture from \texttt{<input type=file>} send to server side and will be a temporary file on the server side. In our system, the picture format JPEG, JPG, GIF and PNG are allowed to upload, and the file max size is 5MB. User need to send a right picture format and a correct size to server. When user send a wrong file, we use PHP to verify the file’s
format and size. PHP function $_FILES["file"]['type'] can get the file’s suffix name, if it is “image/jpeg”, “image/gif” or “image/png”, it is correct picture format. PHP function $_FILES["file"]['size'] to control the file’s max size. In above, all the steps are same in uploading pictures function. We just need to set the path where the temporary photos are saved in the server. PHP function move_uploaded_file() can move the temporary file to a specific place. When uploaded photo moves to the path successfully, it will return a true value.
CHAPTER 4

Test and Optimization

4.1 Test

During the development, several test methods will be used in the system. One is Corner case testing which will use some corner cases to test its stability. Corner case occurs outside of normal operating parameters, specifically when multiple environmental variables or conditions are simultaneously at extreme levels. To implement corner case can make sure function not only can work, it also can work well in the various strict conditions.

We also use Black Box Testing to let many people who do not know the internal structures of this system, providing inputs (clicks, keystrokes) and verifying the outputs against the expected outcome.

4.2 Optimization

During the development, we found a big problem that the battery life of device draining fast when visitors are using mobile application in the park. When mobile devices are recording users’ location with GPS, the battery life can reduce 12% in half hour. The main reason for this problem is that we use mobile application to calculate if the user’s location is in the points geofence area. Even though with the development of mobile CPU technique, the performance of mobile CPU is much better than before, it is still much lower than the server’s. Thus when using
mobile application to calculate the distance between user and point will increase mobile’s CPU pressure. Moreover, the GPS will use some battery power.

To improve the mobile battery life, we design and implement battery saving algorithm on mobile application to improve the battery life. Battery saving algorithm make points distance calculation on the server side, use high performance of server sides’ CPU to calculate points distance based on the coordinates. It uses PHP function distance($lat1,$lng1,$lat2,$lng2) to calculate the distance of two points. The parameter ($lat1, $lng1) is the users’ GPS coordinate, ($lat2, $lng2) is the adventure track point’s GPS coordinate. Mobile application post user’s id and user’s coordinate to server. Server query the Location table from the database, find the points which belong to this adventure track and save them in a points array. Then we use MySQL query array function to search the point in the array one by one. Using array_push() function to set point id and point’s coordinate to an object and save them in array as an element.

```php
$sql_gps = "SELECT * FROM Location WHERE ID IN('" . implode('', $point_arr) . ")";
$result_gps = $conn->query($sql_gps);
$gps_data = array();
while($r = $result_gps->fetch_assoc()) {
    $datas = array(
        'id' => $r['ID'],
        'lat' => $r['Latitude'],
        'lon' => $r['Longitude']
    );
    array_push($gps_data, $datas);
}
```

When PHP puts points data in an array, the distance() function calculate the distance between user and point and put the result in another objective array [11]. At last, usort() function sort the specific attribute in the array. If the first parameter smaller than the second parameter, it will return a negative value otherwise return a positive value. The ordered array will be
ascending. When the distance is smaller than the point’s radius, PHP send the result by JSON format to mobile application that user is in the point’s geofence.

```php
for($i=0;$i<count($gps_data);$i++)
{
    $pid = $gps_data[$i]['id'];
    $pdis = distance($now_lat, $now_lon, $gps_data[$i]['lat'], $gps_data[$i]['lon']);
    $re = array(
        'pid'  => $pid,
        'distance' => $pdis
    );
    array_push($res, $re);
}
usort($res, function($item1,$item2){
    if($item1['distance'] == $item2['distance']) return 0;
    return $item1['distance'] < $item2['distance'] ? -1:1;
});
```

After we implement this algorithm in the server and run this function on the mobile application for half hour in five times, the average of reduce battery life is 7% in half hour. Compare with the distance calculate in the application, it saves battery life 5% in half hour.
CHAPTER 5
Conclusions and future work

5.1 Conclusions

In this paper, we present the whole process of how to develop a location-based educational web system. In order to build a high-performance map with geo-data in real-time, we design and implemented an effective algorithm to determine how to upload data to the dynamic map in different conditions. By applying the GeoJSON method to save multilevel geo-data, this kind of dataset not only loads data in Mapbox API and Google Map API efficiently but also helps to collect complicated user answer data easily and clearly. By developing adventure tracks and citizen science functional model, we use the Ajax method to manage various questions in a specific area on the map. All questions can edit in the list or on the map, and post JSON format questions to the mobile application in real-time. Moreover, we designed several functional classes like uploading photos, getting user geo-data and mobile APIs, which need to be re-used at a high frequency.

5.2 Future work

An increasing numbers of adventure tracks and citizen science projects are created. Many of them are not alone in the system. Some of them have relationships with other. When users want to explore an adventure track, they may need some previous knowledge before they start. Thus, they need to finish an adventure track and earn a badge. Then they can continue to explore
another track and answer the questions. In the future, we will design an algorithm to connect the adventure tracks and citizen science projects. To give them a logical connection which can finish one track and jump to another track automatically.
REFERENCES


