**Web-based Interactive Map of MSU-IIT**

**using OpenStreetMap and OSM Buildings**

A Capstone Project

Presented to the

Faculty of the Information Technology Department

Mindanao State University - Iligan Institute of Technology

In Partial Fulfilment

Of the Requirements for the Degree of

Bachelor of Science in Information Technology

by

Amba, John Vincent

Revelo, Paul Emmanuel

Erik Louwe R. Sala, MSIT

Faculty Adviser

May 2016

**Table of Contents**

1 [INTRODUCTION 1](#_Toc450202138)

[1.1 Background of the Study 1](#_Toc450202139)

[1.1.1 Narrative Listing of Existing System 2](#_Toc450202140)

[1.2 Statement of the Problem 4](#_Toc450202141)

[1.3 Objectives of the Study 5](#_Toc450202142)

[1.3.1 General Objective 5](#_Toc450202143)

[1.3.2 Specific Objectives 5](#_Toc450202144)

[1.4 Scope and Limitations of the Study 6](#_Toc450202145)

[1.5 Significance of the Study 7](#_Toc450202146)

[1.6 Project Design 7](#_Toc450202147)

[1.6.1 User-Centered Design 7](#_Toc450202148)

[2 REVIEW OF RELATED LITERATURE 16](#_Toc450202149)

[2.1 Interactive Map 16](#_Toc450202150)

[2.1.1 Studies on Using Interactive Maps in Universities 16](#_Toc450202151)

[2.2 Local Existing Virtual Maps 18](#_Toc450202152)

[2.2.1 Lakbay IIT: 3D Virtual Tour of MSU-IIT Campus 19](#_Toc450202153)

[2.2.2 Suroy Iligan: A Web-based Virtual Tour of Iligan City 21](#_Toc450202154)

[2.2.3 MSU-IIT Geolocator Information System: An Isometric Virtual Tour of the MSU-IIT Campus 23](#_Toc450202155)

[2.2.4 Summary of Local Existing Systems 25](#_Toc450202156)

[2.3 Usability 27](#_Toc450202157)

[2.3.1 System Usability Scale 27](#_Toc450202158)

[2.3.2 Likert Scale 28](#_Toc450202159)

[2.3.3 SUS Scoring Methodology 28](#_Toc450202160)

[2.3.4 Interpreting SUS Scores 29](#_Toc450202161)

[2.3.5 Analysis of Scaling Methods Used by Existing Systems 29](#_Toc450202162)

[2.4 Summary and Conceptual Framework of the Study 30](#_Toc450202163)

[3 THEORETICAL FRAMEWORK 32](#_Toc450202164)

[3.1 Interactive 3D Maps 32](#_Toc450202165)

[3.2 WebGL 33](#_Toc450202166)

[3.3 Git Workflow 34](#_Toc450202167)

[3.4 SOLID Principles 34](#_Toc450202168)

[3.4.1 Single Responsibility 35](#_Toc450202169)

[3.4.2 Open Close Principle 35](#_Toc450202170)

[3.4.3 Liskov's Substitution 36](#_Toc450202171)

[3.4.4 Interface Segregation 36](#_Toc450202172)

[3.4.5 Dependency Inversion 36](#_Toc450202173)

[3.5 MVC 36](#_Toc450202174)

[3.6 Laravel 5 37](#_Toc450202175)

[3.7 OpenGL 37](#_Toc450202176)

[3.8 OpenStreetMap 38](#_Toc450202177)

[3.9 Leaflet 38](#_Toc450202178)

[3.10 OSM Buildings 38](#_Toc450202179)

[3.11 GeoJSON 39](#_Toc450202180)

[4 WEB-BASED INTERACTIVE MAP OF MSU-IIT 40](#_Toc450202181)

[4.1 Systems Requirements Analysis and Specifications 40](#_Toc450202182)

[4.1.1 System’s Functions 40](#_Toc450202183)

[4.1.2 Physical Environment and Resources 42](#_Toc450202184)

[4.2 Design Models 43](#_Toc450202185)

[4.2.1 Context Diagram 43](#_Toc450202186)

[4.2.2 UML Use Case Model 44](#_Toc450202187)

[4.2.3 Activity Diagrams 53](#_Toc450202188)

[4.2.4 Sequence Diagrams 55](#_Toc450202189)

[4.2.5 Class Diagram 58](#_Toc450202190)

[4.2.6 Component Diagram 59](#_Toc450202191)

[4.2.7 Deployment Diagram 60](#_Toc450202192)

[4.2.8 Database Design Model 60](#_Toc450202193)

[4.3 Structure and Interface 61](#_Toc450202194)

[4.3.1 Program Structure 61](#_Toc450202195)

[4.4 Development Phases 62](#_Toc450202196)

[4.4.1 Pre-development Phase 62](#_Toc450202197)

[4.4.2 Prototype Development 63](#_Toc450202198)

[4.4.3 Summary of the Development Process 64](#_Toc450202199)

[5 RESULTS AND DISCUSSION 66](#_Toc450202200)

[5.1 Feedback Results of the Interactive Map 68](#_Toc450202201)

[5.2 Scoring System Usability Scale 75](#_Toc450202202)

[5.3 SUS Scores of Each Participant of Each User Group 76](#_Toc450202203)

[5.4 System Usability Scale Analysis 77](#_Toc450202204)

[5.4.1 System Usability Analysis for Administrators 77](#_Toc450202205)

[5.4.2 System Usability Analysis for Students 77](#_Toc450202206)

[5.4.3 System Usability Analysis for Visitors 77](#_Toc450202207)

[5.4.4 System Usability Analysis for Prospective Students 77](#_Toc450202208)

[6 SUMMARY, CONCLUSION AND RECOMMENDATIONS 78](#_Toc450202209)

[6.1 Summary of Development 78](#_Toc450202210)

[6.2 Conclusion 78](#_Toc450202211)

[6.3 Recommendations 79](#_Toc450202212)

[7 References 80](#_Toc450202213)

**List of Figures**

[**Figure 1.1** Fishbone Diagram of the System 4](#_Toc450202252)

[**Figure 1.2** User-Centered Design Process of the System 8](#_Toc450202253)

[**Figure 1.3** Dashboard Wireframe 10](#_Toc450202254)

[**Figure 1.4** Interactive Map Prototype using Google Maps 11](#_Toc450202255)

[**Figure 1.5** Interactive Map Prototype using OpenStreetMap and OSM Buildings 11](#_Toc450202256)

[**Figure 1.6** Module Development Process 12](#_Toc450202257)

[**Figure 2.1** SUS Score Equivalent Scale to Acceptability Range and Grade Scale 29](#_Toc450202258)

[**Figure 2.2** Conceptual Framework 30](#_Toc450202259)

[**Figure 3.1** Sample WebGL Representation of a Terrain 33](#_Toc450202260)

[**Figure 3.2** Git Operations 34](#_Toc450202261)

[**Figure 3.3** OSM Buildings Representation 39](#_Toc450202262)

[**Figure 4.1** Context Diagram 43](#_Toc450202263)

[**Figure 4.2** Use Case Diagram for Administrator and User 44](#_Toc450202264)

[**Figure 4.3** Login Activity Diagram 53](#_Toc450202265)

[**Figure 4.4** Map Module Activity Diagram 53](#_Toc450202266)

[**Figure 4.5** Draw Module Activity Diagram 53](#_Toc450202267)

[**Figure 4.6** Content Module Activity Diagram 54](#_Toc450202268)

[**Figure 4.7** Search Module Activity Diagram 54](#_Toc450202269)

[**Figure 4.8** Sequence Diagram for End-User 55](#_Toc450202270)

[**Figure 4.9** Sequence Diagram for Administrator Authentication 56](#_Toc450202271)

[**Figure 4.10** Sequence Diagram for Administrator Create and Edit 57](#_Toc450202272)

[**Figure 4.11** Sequence Diagram for Administrator Delete 57](#_Toc450202273)

[**Figure 4.12** Sequence Diagram for Administrator View 58](#_Toc450202274)

[**Figure 4.13** Class Diagram of the Developed System within Laravel Framework 58](#_Toc450202275)

[**Figure 4.14** Component Diagram 59](#_Toc450202276)

[**Figure 4.15** Deployment Diagram 60](#_Toc450202277)

[**Figure 4.16** Entity Relationship Diagram 60](#_Toc450202278)

[**Figure 4.17** Database Schema 61](#_Toc450202279)

[**Figure 4.18** Program Structure of the System 61](#_Toc450202280)

[**Figure 4.19** Interactive Map Prototype using Google Maps 62](#_Toc450202281)

[**Figure 4.20** Interactive Map Prototype Using OpenStreetMap and OSM Buildings 63](#_Toc450202282)

[**Figure 5.1** Test User Groups Percentage of the Pilot Test 66](#_Toc450202283)

[**Figure 5.2** Gender Percentage of the Pilot Test 67](#_Toc450202284)

[**Figure 5.3** Interactive Map System Usability Feedback of Administrators 69](#_Toc450202285)

[**Figure 5.4** Interactive Map System Usability Feedback of Students 70](#_Toc450202286)

[**Figure 5.5** Interactive Map System Usability Feedback of Visitors 72](#_Toc450202287)

[**Figure 5.6** Interactive Map System Usability Feedback of Prospective Students 73](#_Toc450202288)

[**Figure 5.7** Interactive Map System Usability Feedback of Prospective Students 75](#_Toc450202289)

**List of Tables**

[**Table 1.1** Hard and Soft Constraints of the System 6](#_Toc450202290)

[**Table 1.2** Tools and Scripting Languages 9](#_Toc450202291)

[**Table 2.1** Problem and Recommendations of Lakbay IIT 24](#_Toc450202292)

[**Table 2.2** Feature Comparison of Systems 26](#_Toc450202293)

[**Table 4.1** Login Use Case Specification 44](#_Toc450202294)

[**Table 4.2** Logout Use Case Specification 45](#_Toc450202295)

[**Table 4.3** View Map Use Case Specification 46](#_Toc450202296)

[**Table 4.4** Draw Building Use Case Specification 47](#_Toc450202297)

[**Table 4.5** Edit Building Use Case Specification 48](#_Toc450202298)

[**Table 4.6** Remove Building Use Case Specification 49](#_Toc450202299)

[**Table 4.7** Add Building Information Use Case Specification 50](#_Toc450202300)

[**Table 4.8** Edit Building Information Use Case Specification 50](#_Toc450202301)

[**Table 4.9** Remove Building Information Use Case Specification 51](#_Toc450202302)

[**Table 4.10** Search Building Use Case Specification 52](#_Toc450202303)

[**Table 5.1** Preliminary Results 67](#_Toc450202304)

[**Table 5.2** Administrators’ Feedback on System Usability of the Interactive Map 68](#_Toc450202305)

[**Table 5.3** Students’ Feedback on System Usability of the Interactive Map 69](#_Toc450202306)

[**Table 5.4** Visitors’ Feedback on System Usability of the Interactive Map 71](#_Toc450202307)

[**Table 5.5** Prospective Students’ Feedback on System Usability 72](#_Toc450202308)

[**Table 5.6** Total Feedback on System Usability of the Interactive Map 74](#_Toc450202309)

[**Table 5.7** Administrators’ System Usability Results 76](#_Toc450202310)

[**Table 5.8** Students’ System Usability Results 76](#_Toc450202311)

[**Table 5.9** Visitors’ System Usability Results 76](#_Toc450202312)

[**Table 5.10** Students’ System Usability Results 76](#_Toc450202313)

**CHAPTER 1**

# **INTRODUCTION**

## **Background of the Study**

Universities around the world have incorporated the digital revolution. The 21st century university student understands his or her campus in a profoundly different manner from students in decades past. Today, students are enveloped in wireless Internet access, are never more than a few thousand feet away from a computer or laptop, and manage their various student responsibilities and social activities through digital, web-based applications (Roth, et al., 2009).

University campuses are complex and difficult to navigate for the tens of thousands of people that visit in a given year. The campus map is therefore a vital piece of information, challenging in its design in that it covers a large space, needs to incorporate tremendous detail about building, housing, recreation, and access, and, needs to be displayed on a relatively small computer screen (Roth, et al., 2009). This explains why universities and colleges nowadays often have campus maps to visually represent their campuses. These are mainly found in their school websites.

For high school students who have the time and money to travel, visiting a college campus is the best way to get a sense of the students, the faculty, and the feel of the place where they'll be spending their college years (ICEF Monitor, 2012). However, we must keep in mind that some prospective students simply are unable to visit a campus prior to applying or accepting due to monetary or time issues, but by having a campus map online that students can interact with, colleges and universities gain a competitive edge over schools without one (James, 2011). This allows prospective students to get a feel of the campus without actually visiting. An interactive map allows students to examine the campus size, understand where dorms are in relation to academic buildings, and see how far apart potential classes will be from one another.

Creating campus maps prove to be crucial to competing with universities and colleges. The basic driving force for developing a school’s own campus map is staying competitive with what other schools are doing to increase student recruitment and retention and it proves to be crucial to competing with universities and colleges in order to improve visibility in academic industry (James, 2011).

An interactive map concentrates on the interaction between the map and the user. It provides a means to give input and get results (James, 2011).  Interactive maps can reach a wide variety of people at a very low cost and can be modified to reflect changing business requirements and objectives. With interactive maps, organizations can update centralized databases independently of the map (Fitzpatrick, 2002).

One of the few instances interactive maps are useful is a unique representation of a place particularly of a city (AddisMap, 2014) or in a campus setting (Roth, et al., The 21st Century Campus Map: Mapping the University of Wisconsin-Madison, 2009).

In the local setting, prospective students from neighboring cities and provinces would require themselves to visit the campus of MSU-IIT for initial inspection. ICEF Monitor (2012) suggested that a campus map is a necessary requirement for future endeavors of most academic institutions.

### **Narrative Listing of Existing System**

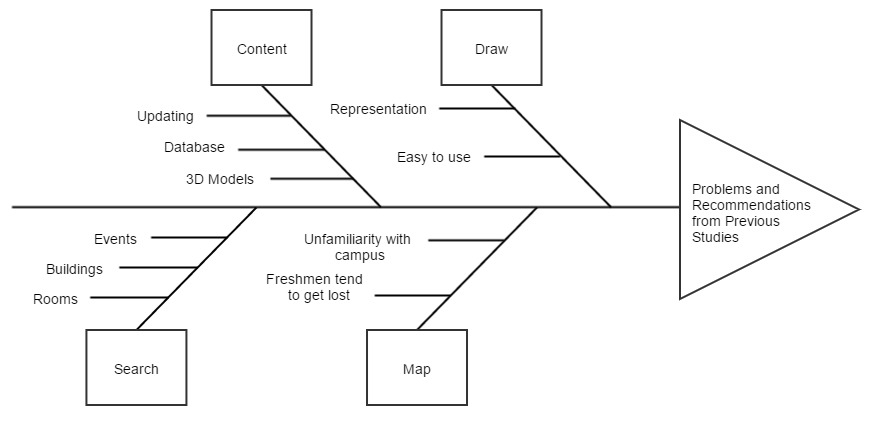
Local studies were pursued in the attempt to fix the problem of visualization of the MSU-IIT campus and Iligan City’s tourist destination spots. Campil, Gomez and Paulin (2010) first addressed the need for a virtual campus map since their preliminary interviews with prospective students and visitors reveal that they experienced several inconveniences in navigating inside the campus. Similarly, Ferrater, Gimeno and Lipa (2013) highlighted the former that the lack of reliable and updated content gave rise to problems for both prospective students and visitors. Further, Bala, Chiu and Yparraguirre (2012) addressed the lack of information regarding potential travel spots in Iligan City's tourism website.

Lakbay IIT was developed by Campil et al. as a solution to the limited visual perspective of the campus and the problems that were brought about from it. It aims to supply the user with 3D visualization of the whole campus in a desktop application. Additionally, it provides building data in Portable Document Format (PDF) that contains room identification key and a short description which can be obtained. Furthermore, it provides a hyperlink to a particular college website (Campil, et al., 2010).

Ferrater et al. (2013) developed the Geolocator Information System of MSU-IIT that features isometric projection. Its services include directing paths for users to follow in navigating within the campus and a search functionality for searching destinations, building information, and key personnel of a department or office. It also offers view functionality for college building floor interiors. The system was built for a kiosk machine under Flash-based architecture supporting touchscreen interaction.

In the context of tourism, Bala et al. (2012) introduced a web-based virtual tour integrated with Google Maps services to promote virtual tourism through Iligan City’s tourism website. It provides a set of multimedia elements such as panoramic views and interactivity of top inns, restaurants, and attractions of Iligan City. It also offers extensive navigation functionality on panoramic data for users to navigate freely.

These systems fail to provide an option to update all of their information, thus making them unsuitable for long term use. The systems stated in this section extensively introduced problems that still need to be addressed and functionalities that can be further developed to fulfill the needs of the current audience.



**Figure 0.1** Fishbone Diagram of the System

Figure 1.1 presents a fishbone diagram showing the problems and recommendations stated in the previous studies that are used as guide by the project developers in the design and development of the system.

## **Statement of the Problem**

Campil et al. (2010) developed a 3D virtual map as a desktop application which requires high end hardware capable of rendering 3D models. On the other hand, Ferrater et al. (2013) developed a kiosk-ready Flash-based information system that shows a virtual map of the campus.  Yet, the implementation and maintenance of kiosk machines can be costly in both time and money since they need to be strategically placed all around the campus.

Moreover, these systems were developed as static applications without any form of database and with little to no instructions on how to update relevant data, making the content almost obsolete in the present. Insufficient searching of keywords was also found on the latter where keywords of relevant content are case-sensitive and limited.

This problem is also visible in Suroy Iligan since the information in the system were statically integrated. Appropriately, they recommended adding backend functionalities to make the system dynamic (Bala et. al 2012).

## **Objectives of the Study**

The project developers adhered to the following objectives to guide them in answering the problem stated in the preceding section.

### **General Objective**

As illustrated in Figure 1.1, the project developers aim to develop a web-based interactive map of MSU-IIT by applying the recommendations from the systems of Campil et al., Bala et al. and Ferrater et al. and unifying their functionalities and integrating them into a single dynamic web-based system.

### **Specific Objectives**

To be able to meet the general objective of this project, these specific objectives need to be achieved:

1. To design and develop:
   1. A map module to efficiently render 3D structures of MSU-IIT;
   2. A draw module for drawing buildings as polygons and assigning data on them;
   3. A search module for the application;
   4. A content module with CRUD (Create, Read, Update and Delete) functionalities for all data;
2. To test and evaluate the application.

## **Scope and Limitations of the Study**

This project’s sole focus is on the design and development of an interactive campus map of MSU-IIT. The project developers focused on all the buildings within the campus, excluding external buildings (e.g. College of Medicine and other branches of MSU-IIT Cooperative).

The project also catered the development of a simple tool for drawing buildings as low and lightweight polygons. The system does not allow the user to create and update room perspective as this needs expertise in the field. Users allowed to do so will need to have experience in the field of 3D development.

The project includes a search module that enables the users to search for buildings and colleges. For security purposes, this module does not include searching for specific faculty members, schedules and rooms with unavailable data. The features of the system are divided and classified by the project developers as hard or soft constraints. Hard constraints are the features that are needed to be developed for the system to work significantly. These are considered top priority and should be developed before other features. On the other hand, soft constraints are the features that can be developed after all the hard constraints are integrated. These are usually treated as “add-on” features since they function dependently on the set of hard constraints. These constraints are found in Table 1.2.

**Table 0.1** Hard and Soft Constraints of the System

|  |  |
| --- | --- |
| **Hard Constraints** | **Soft Constraints** |
| Modules:   * Map * Draw * Search * Content | * Geolocation (IP address or GPS); * Room perspective; |

## **Significance of the Study**

The system can help students, faculty members, staff, visitors and prospective students find locations of buildings and/or rooms within campus and also provides them with necessary information about the locations. This also significantly helps in attracting prospective students from distant places to enroll in the university.

The system can also be beneficial to the open source programming libraries used in the design and development of the system especially to OpenStreetMap, OSM Buildings and GLMap. Advancements of the technology used by the developers are contributed to the creators of OSM Buildings and GLMap as these technologies were just recently developed.

## **Project Design**

This section shows the methodology used to develop the web-based interactive map of MSU-IIT.

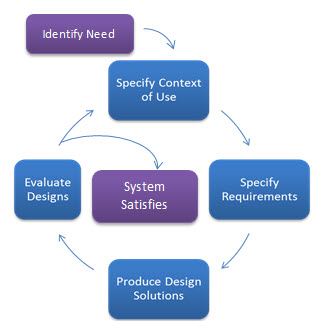
### **User-Centered Design**

User-Centered Design (UCD) is an approach that supports the entire development process with user-centered activities, in order to create applications which are easy to use and are of added value to the intended users. The biggest cost benefit that UCD can provide is by more accurately defining requirements. Ideally, UCD activities should be integrated with other development activities. They should be planned and managed by the development team. Over time, UCD activities will become common practice, and existing members of the team will be able to carry them out. However, usability skills will most probably be needed within the project and if necessary one or more members of the team should possess these skills (Kangas, 2005).

ISO 13407 Human-centered design process for interactive systems states that *“User-centered design is an approach to interactive system development that focuses specifically on making systems usable. It is a multidisciplinary activity.”*

Several studies have used the User-Centered Design, one of which is the study by Góralski (2009) about developing a three-dimensional interactive map to depict geographical information of a place. Another study that used UCD was by Roth et al. (2009) in his study entitled *The 21st Century Campus Map: Mapping the University of Wisconsin-Madison* and added that “design is based upon an explicit understanding of users, tasks, and environments.”

The web-based interactive map developed for MSU-IIT used the User-Centered Design. Here, development starts with planning and goes through Analysis and Requirements, Design and Evaluation.



**Figure 0.2** User-Centered Design Process of the System

#### **Planning**

UCD need not be extensive or costly. A few simple activities initial in development will significantly lessen the overall cost of developing a conventional system.

**Choice of Technologies**

The project developers used HTML5 (Hypertext Markup Language) along with CSS (Cascading Style Sheets), Laravel, a PHP framework intended for the model–view–controller (MVC) architectural pattern and JavaScript in developing the interface of the system. To aid in the design and layout of the user interface, the developers used Bootstrap 3. Geospatial mapping API libraries used include OpenStreetMap, OSM Buildings, Leaflet and Leaflet Draw.

To store the collected data in the data gathering process, the developers needed database storage. The system used MySQL since it’s used by most Web applications nowadays.

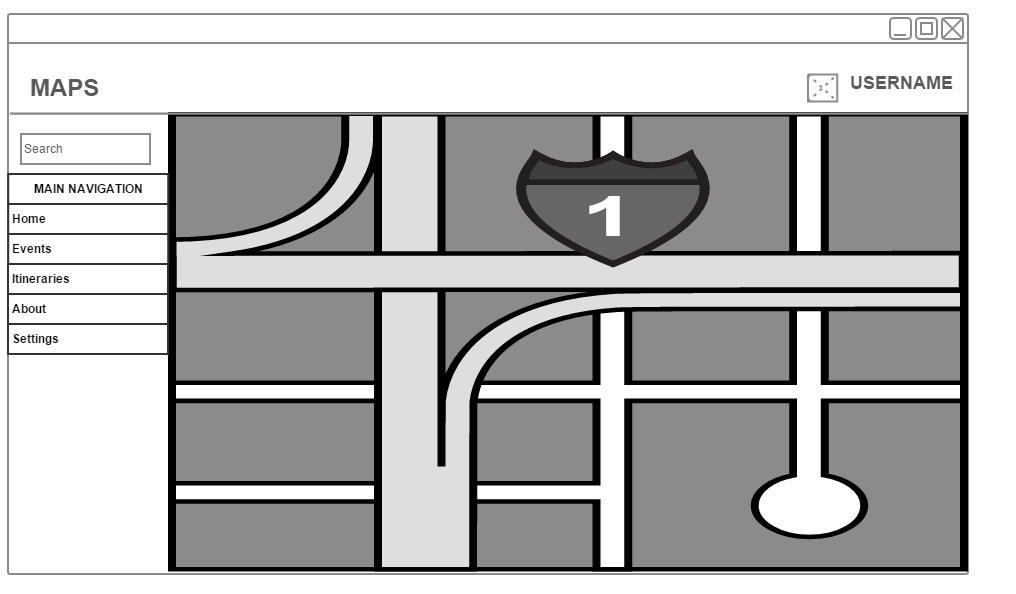
**Table 0.2** Tools and Scripting Languages

|  |  |
| --- | --- |
| **Front-end** | **Back-end** |
| * HTML5 * Bootstrap * CSS * JavaScript * OpenStreetMap * OSM Buildings * Leaflet * Leaflet Draw | * Laravel framework * XAMPP * PHP * MySQL * AJAX |

#### **Design and Development**

**Wireframes**

Wireframes were created by the developers for the initial concept and layout of the system. A wireframe, also known as a page schematic or screen blueprint, is a visual guide that represents the skeletal framework of a website.Wireframes are created for the purpose of arranging elements to best accomplish a particular purpose.



**Figure 0.3** Dashboard Wireframe

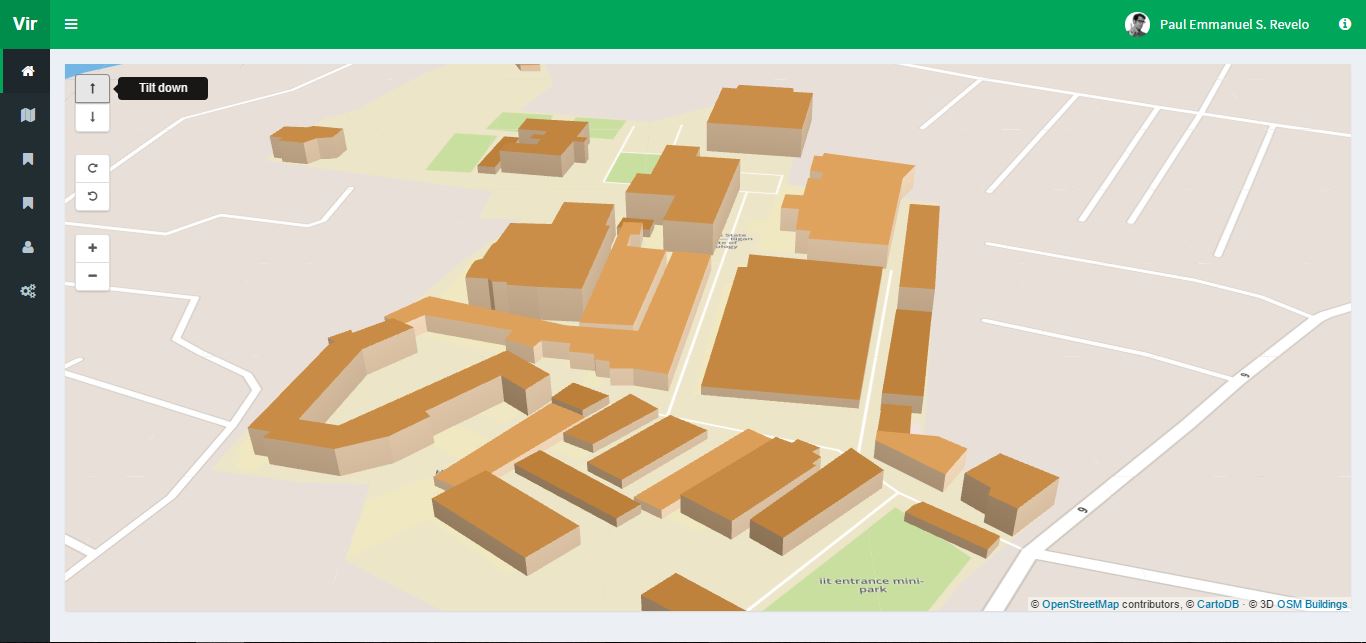
##### Prototype Documentation

Attempts to construct an early prototype for this system has already been made through multiple trials. The prototypes were based on different mapping API such as Google Maps, OpenStreetMap and Leaflet. The prototype using Google Maps did not meet the requirements of the developers. Figure 1.3 shows a screenshot of the prototype using Google Maps.



**Figure 0.4** Interactive Map Prototype using Google Maps

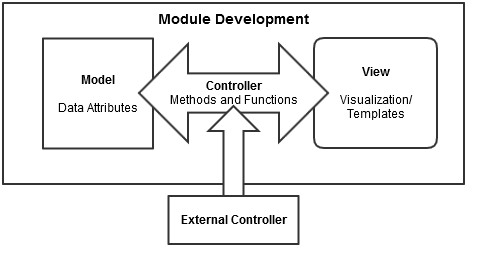
After developing the first prototype, the project developers used another mapping API library named OpenStreetMap. This mapping API library, paired with OSM Buildings, a plugin that allows users to load 3D content in maps, met the requirements set by the developers and the consultant, Prof. Erik Louwe Sala. Figure 1.4 shows a screenshot of the prototype using OpenStreetMap and OSM Buildings.



**Figure 0.5** Interactive Map Prototype using OpenStreetMap and OSM Buildings

##### Module Development

The specific objectives were interpreted as modules for the system. Proposed features based on the conclusion of the system prototype and usability testing were written as recommendations for further development of the system.

****

**Figure 0.6** Module Development Process

Initial development of a module started on constructing its model as a class. It is followed by integration process for its controller and testing through visualization on view process. By following programming principles, external controller(s) were made for complexity and readable code. Additionally, its purpose was to provide interconnectivity between modules to be constructed.

**Map Module**

This module counts as a hard constraint of the system as this is the main feature. Development of this module include doing prototypes using visualization technologies (library or framework) and testing whether the technologies used provided what the researchers need for this module. OpenStreetMap was used for the integration of the map and OSM Buildings for the 3D content in the map.

**Content Module**

This module consists of two parts: database and data management. This module is responsible for storing all of the data gathered by the researchers. These include building, structure, room and event information. Also, this module is accountable for the management of these data, which means users can securely access and manipulate the data. Developing this module includes creating a database and creating CRUD (Create, Read, Update and Delete) functionalities using MySQL and JavaScript libraries.

**Draw Module**

This module is responsible in the drawing of low polygon buildings. Developing this module includes the integration of Leaflet’s Draw tools in the map to let the user draw polygons using coordinates, OSM Buildings’ visualization of these coordinates to make 3D buildings and GeoJSON as the initial data format that references to data of buildings involved in the system. These data are from the information provided by the Physical Plant Division.

**Search Module**

This module is responsible for the searching of all data stored in the system. Developing this module includes integrating JavaScript libraries to create the search function and algorithm, HTML to create the elements and CSS to create a style suitable for the overall design.

##### Module Integration and Testing

In this section, each module follows MVC as software architectural pattern, where it stands for Model, View, Controller respectively. In the integration process, the module created to a class referring it as a model. This is composed of the modules data attributes. It is followed by a controller of a different class containing different methods and functions in manipulating the module data.

These provide the system security in accessing the modules attributes and their interactivity with other modules. In testing, view component is built for visualization whereas it provides the basic user interface for the module. Additionally, it also shows the interactivity between modules thus gives the programmers initial response if problems or bugs are to be found.

#### Prototype Refinement

Refining of the system prototype commences once all of the modules are developed. This phase is monitored by the Product Owner. With this, the researchers follow the SOLID Principles as basis in refining the product for this study.

#### System Testing

Testing the system as a whole was done through a usability test using SUS (System Usability Scale). Thirty (30) respondents were asked to participate in the said test. With this, the developers can evaluate and measure the usability of all modules based on the different objectives identified.

##### Testers

Target testers should be computer literate and are owners of electronic device(s) capable of web interactivity i.e. users who have computer units and/or mobile devices. The testers are defined not on the basis of race, religion, gender, civil or any classification. Any classification to do so are not for this study aims.

Preliminary questionnaires were handed out to thirty testers. These testers were given the opportunity to test the proposed system.

Based on the testers’ general information, they were classified into four main user groups composed of administrators, students, prospective students and visitors.

##### Evaluation Process

Answers supplied by the testers were retrieved in Microsoft Excel format. The data obtained were mainly interpreted to how users interact with the system, including its navigation and design, and dealing quality to other users. Additionally, from each of user group, a mean for the SUS Score was collected and will be evaluated based on the answers of testers.

**CHAPTER 2**

# **REVIEW OF RELATED LITERATURE**

This chapter provides related topics and studies about the concept of interactive mapping. Also, this chapter covers past researches in the context of interactive mapping in the scope of universities, MSU-IIT in particular. Technologies and principles used will be tackled in the succeeding part of this chapter.

## **Interactive Map**

Interactive maps can be a great way of displaying useful information in an engaging and attractive way by inviting the user to take action. This technology has given a lot of websites an ability to embed the map on their web pages and use it according to their purpose. It is an additional tool or feature in providing information especially on different academic institutions and businesses (Eder, Nocete, Rances, Tarrosa, Yanson, 2015). These help in keeping visitors engaged and can demonstrate information relevant to the institution in several ways. Interactive maps are also always up to date (Three Scale, 2014).

### **Studies on Using Interactive Maps in Universities**

Roth et al. (2009) presented two web-based interactive maps of the University of Wisconsin-Madison campus: The University of Wisconsin-Madison Interactive Campus Map and the Lakeshore Nature Preserve Interactive Map. Although the two projects represent the same university campus, the former follows a wayfinding-based model of campus mapping while the latter follows an atlas-based model of campus mapping. The purpose of the University of Wisconsin-Madison Interactive Campus Map is to search for, navigate to, and retrieve information about specific features on campus. The purpose of the Lakeshore Nature Preserve Campus Map is to present the rich history and unique geography of the University’s Lakeshore Nature Preserve. The interactive, online development of these maps follows the broader university transition to digital media for retrieving information and organizing student activities. Although both maps represent the same campus, the two interactive, online maps take two distinct approaches to campus mapping. We describe these two different approaches as a wayfinding-based model and an atlas-based model of campus mapping. As demonstrated through the two examples, both approaches can produce a useful and engaging tool.

Also, Roth (2013) provided a review of the current state of science regarding cartographic interaction, a complement to the traditional focus within cartography on cartographic representation. Cartographic interaction is defined as the dialog between a human and map, mediated through a computing device, and is essential to the research into interactive cartography, geovisualization, and geovisual analytics (Roth, 2013). The review is structured around six fundamental questions facing a science of cartographic interaction: (1) what is cartographic, (2) why provide cartographic interaction, (3) when should cartographic interaction be provided, (4) who should be provided with cartographic interaction, (5) where should cartographic interaction be provided and (6) how should cartographic interaction be provided. The review showed how traditional and interactive maps differ so much from each other. It is really suggested to create an interactive map based on the science of cartographic interaction.

Similarly, Eder et al. (2015) developed a web-based interactive campus map using the Google Maps API. Availability of the overlay function has been taken advantage to create custom map functionalities. Collection of building points were gathered for routing, and to create polygons which serves as a representation of each building. Storing data about the building, room, and staff information, and university events and campus guide are among the primary features that their study has to offer. The campus map is open to constant updates, user-friendly for both trained and untrained users, and capable of responding to all needs of users and carrying out analyses. Based on the data gathered through questionnaires, researchers analyzed the results of the test survey and proved that the system is user friendly, deliver information to users, and the important features that the students expect.

Moreover, Nikoohemat (2013) implemented comprehensive indoor navigation systems that not only needs technology platform but also cartographic and well-designed maps especially for mobile devices. A set of thirty (30) respondents composed of students and staff were asked to answer an exploratory survey. The focus group of the survey are campus map users, staff and students. Some of these students were familiar with the campus and some are not. The survey was also answered by students from other universities who were just enrolled in the campus for one semester and experienced being in different campuses. The survey questionnaire consisted of 20 questions and were distributed in printed form among focus group.

The study concluded several points from its survey. One of which was that finding lecture rooms is the most important functionality in the application followed by routing, finding points-of-interests and staff’s offices in undetermined order. Another point was students preferred 2D maps than 3D maps, preferably without captions. Additionally, using background maps and images has less significance to students to which they prefer better functionality than design for the application. Lastly, reliability and fast loading were the key factors for the importance of the smart campus map.

## **Local Existing Virtual Maps**

Through an extensive collection of related studies, the project developers obtained the following studies that were developed in the local setting. On this part, the developers will comprehensively tackle the methodology introduced by each existing system with a short evaluative summary at the end of this chapter.

### **Lakbay IIT: 3D Virtual Tour of MSU-IIT Campus**

A 3D environment is one that exists in three dimensions and is capable of being moved around in and explored. Campus virtual tours are becoming popular in the websites of many colleges, universities and organizations. It has become a favored medium of communicating to prospective students since it will provide useful information about the campus to the prospective students, parents and any interested party.

In a study conducted by Campil et al. (2010), the problem they faced were (1) exploration of the campus was limited because the campus was presented in 2D graphics, (2) prospective students are not given the opportunity to explore the campus in detail before actually going there, (3) guides around the campus are not enough to help the students with directions from a location to another and (4) students seldom experience several inconveniences during times of enrolment. Preliminary interviews claimed that although there were guides (e.g. maps, signage, posts) found around the campus, those will not still be enough because people, in general, do not give much attention to those kinds of guides. Most of them were not familiar with the buildings and others don't have the resources to visit before busy days (e.g. enrollment, clearance signing).

To address this problem, the researchers developed a desktop application that shows a 3D virtual map of MSU-IIT. This map will enable students to visit different 3D rendered buildings around the campus. However, their application required standard computer unit or device capable for 3D visualization. 3D visualization requires heavy computer specifications. particularly processor speed, graphics processing speed and memory capacity. A prototype was developed but was limited to desktop use only. The prototype only included individual buildings (e.g. Main Library, College of Business Administration and Accountancy, College of Engineering, College of Education, College of Science and Mathematics. School of Computer Studies and the MSU-IIT Gymnasium).

A set of thirty (30) respondents were asked to evaluate the system. These are students of the campus. These users are limited to navigating the map, generating reports and viewing information. The researchers conducted a usability test during and the development phase of the system. Three sets of questionnaires were given to the respondents. These are the pre-test, post-task and post-test questionnaires. The pre-test questionnaire was used during the session introduction to verify the qualifications of the participants and gather additional background information to interpret test data.  The post-task questionnaire was used to capture the participants’ perception of the task’s difficulty and to gather relevant comments where applicable.  The posttest questionnaire should capture the participant’s overall perception of the system’s usability and specific perception.  User sentiment was also considered during the system test.

The researchers also conducted a usability test after the development phase of the system. The observations gathered during the first system test were recognized and changes were made in the system. The users’ sentiments during the first system test were also addressed causing significant changes to the system’s usability.

After the tests were conducted, the researchers were able to find out the participants’ overall and specific perception of the system’s usability.

In conclusion, recommendations were formulated by the researchers, to wit: (1) Additional collision detection on walls, gates, and other objects, (2) Enhancement of the texture of models, (3) Inclusion of minute details like room numbers, bulletin boards, cars, computer, etc., (4) Additional exploration modes like guided tour mode and walk mode, (5) Additional human model for a 3rd person point of view, (6) Additional clickable mini map feature, (7) Optimization of the system for better performance, (8) Additional full screen feature, (9) Improvement on sky limitation, (10) Additional mouse hover labels of rooms and buildings, (11) Inclusion of shortest path feature, (12) Additional pop-up bubble of building or college information and (12) Additional current location indication.

### **Suroy Iligan: A Web-based Virtual Tour of Iligan City**

Many locals and tourists want to discover and explore Iligan City but lack the information of famous places to visit. If they do, it seems to be some common tourist spot. There are tourists who want to check the place first by looking at multimedia elements, such as photographs and videos before visiting the place. They want to have familiarity with what experience these tourist spots can give them. These tourists depend mainly on what the Internet can provide.

Bala et al. (2012) stated that the problem is that the Iligan City tourism website does not have these elements that would give the tourist information about tourist spots and encourage them to visit.

To address this problem, the researchers focused on creating a web-based tour where one can explore the place and give users a first-hand preview on their possible experience if they visit the place. The functionality of the project includes Google Maps integration where locations and information of the places where provided, such as the entrance fees and business hours. The system also aimed to develop a virtual tour of the top inns, restaurants and attractions of Iligan City as recommended by the city tourism office. The system gathered data that are provided by brochures, tourist guides and web pages. The system is easy to use and will aid tourists in finding exact locations of some tourist spots in the city. Every spot has its own location, description, pictures and animations. Panoramic views of suggested places are shown to help tourists have an idea about the spot. The system gives an overview of different destinations that tourists wanted to know about. The aim of the system is to supplement the existing tourism website of the city. The system is an interactive environment where users can explore a panoramic representation of different spots in the city.

A set of thirty-five (35) respondents were asked to evaluate the system. The researchers considered the effectiveness of the system to the intended users (locals and tourists). The researchers first showed a prototype to the client and proceeded to Alpha testing. They did some tests and checked all the features of the system and defined those that did not function properly. Beta testing followed in which the researchers let the respondents use and explore the system. The respondents then were given questionnaires to answer. The questionnaires include evaluation on how reliable the system is in providing information about the city. It then served as a guide for the researchers to determine the needed improvements based on the locals’ and tourists’ feedback and attitude towards the system. The questionnaire was composed of different categories: (1) Ease of Use, (2) Content, (3) Technical Aspect and (4) Usability.

In conclusion, recommendations were formulated by the researchers such as: (1) Make the system kiosk-ready, (2) Make it dynamic by adding backend functionalities, (3) Include places that are not commonly visited, (4) Make panoramic views in hotels, (5) Include big hotels in the city, (6) Additional mouse hover labels, (7) Shortest path feature, (8) Optimization of the system for better performance, (9) Include menus of featured restaurants, (10) Add current location feature, (11) Convert to 360 view panoramic tour and higher quality of panoramas.

### **MSU-IIT Geolocator Information System: An Isometric Virtual Tour of the MSU-IIT Campus**

Ferrater et al. (2013) stated that some graduates and undergraduates stated that they are not familiar with the structure of the campus due to lack of reliable and updated campus map. This gave rise to problems, (1) locating particular areas of the campus was difficult due to lack of information provided and (2) new students and visitors will not have dependable information regarding locations.

They focused on developing a Flash-based system that displays a virtual tour of the MSU-IIT campus using isometric projection. Isometric projection is a method for visually representing three-dimensional objects in two dimensions in technical and engineering drawings. The system guides users in searching for specific locations in the said campus by providing visual and text direction of the most direct route to any college building in the campus. In this study, a kiosk machine was strategically placed in the entrance of the campus, providing initial information about MSU-IIT and short path navigation from the kiosk to any building. Also, the system was limited to one kiosk, disabling the users to freely roam while using the system.

A set of one hundred (100) respondents were asked to evaluate the system. These are students or visitors of the campus. Five instructional statements serve as tools for evaluating and determining the overall value of the system design. These statements are (1) The overall Geolocator is attractive, (2) The Geolocator’s graphics are of good quality, (3) The Geolocator has a good balance of graphics versus text, (4) The colors used throughout the Geolocator are attractive and (5) The typography (lettering, heading, and titles) is attractive.

With the implementation of the system, an alternative approach of organizing, integrating, and presenting information about buildings was developed. The system offers a way users search for buildings, colleges, offices, and significant people.

**Table 2.1** Problem and Recommendations of Lakbay IIT

|  |  |  |
| --- | --- | --- |
| **System** | **Problem** | **Recommendations** |
| **Lakbay IIT** | Respondents experience inconveniences during times of enrolment. | 1. Collision detection;  2.Texture enhancement;  3. Minor details;  4. Walk mode;  5. Human model;  6. Clickable map;  7. System optimization;  8. Full screen feature;  9. Sky limitation;  10. Mouse hover on buildings;  11. Shortest path;  12. Info window on buildings;  13. Additional current location indication. |
| **Geolocator** | 1. Locating particular areas of the campus;  2. New students and visitors will not have dependable information regarding locations. | 1.Expansion of geographic scope;  2. Guest book;  3. Search for people;  4. GUI enhancement;  5. Search for courses;  6. Button to show all searchable keywords;  7. First person view;  8. System optimization. |
| **Suroy Iligan** | Iligan City tourism website lacks necessary information on tourist spots and services to encourage prospective tourist to visit. | 1. Kiosk application;  2. Backend functionalities;  3. Include new/rare places;  4. Panoramic view for hotels;  5. Include big hotels;  6. Introduce hover labels;  7. Shortest path feature;  8. Optimization for better performance.  9. Include menus of featured restaurants;  10. Add current location feature;  11. 360 view panoramic tour and higher quality of panoramas. |

In conclusion, the following recommendations were stated by the researchers: (1) Expand the geographic scope of the application, (2) Additional guest book that would monitor the number of users of the application, (3) Extend the search for people, i.e., allow the user to see other faculty and staff of the campus, (4) Enhancement for the graphical components of the system, (5) Extend the search for courses, i.e., allow the user to see graduate programs and the location of each program, (6) Additional button to view all the searchable keywords, (7) Additional walking human model for guiding for the shortest path and (8) Optimization of the system for better performance.

Table 2.1 presents the problems and recommendations of existing systems. These were used as guides for the developers in the implementation of the system that suits the needs of the current audiences.

### **Summary of Local Existing Systems**

It was first addressed through Campil et. al (2010) the need for a virtual campus map since they have learned that prospective students and visitors experienced several inconveniences on navigating the campus site. Collectively, Bala et. al (2012) addressed the lack of information regarding on potential travel spots on the local (Iligan) city's tourist website. Ferrater et. al (2013) highlighted the former that the lack of reliable and updated content still produced problems to visitors. Even with different methodologies exercised by these studies, potential problems would always spring up in them that in the long run these systems would become obsolete.

Campil et, al (2010) produced a 3D virtual map built as a desktop application which requires high end hardware capabilities to render 3D building models. Constructing 3D models alone require excessive time to which the problem was not properly addressed. Furthermore, their system was constructed as a static application and thus data would eventually become obsolete. This problem can also be found on Suroy Iligan not because it was a web application but the contents in general do not promote interactivity since these data are statically integrated. Additionally, it was also addressed in their recommendation to add backend functionalities to make the system dynamic thus the problem (Bala et. al 2012).

**Table 2.2** Feature Comparison of Systems

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Features** | **Lakbay IIT** | **Suroy Iligan** | **Geolocator System** | **Newly Developed System** |
| 3D Graphics | ✔ |  |  | ✔ |
| First person POV | ✔ | ✔ |  | ✔ |
| Interactive |  | ✔ | ✔ | ✔ |
| Informative |  | ✔ | ✔ | ✔ |
| Map |  | ✔ | ✔ | ✔ |
| Desktop-ready | ✔ | ✔ | ✔ | ✔ |
| Mobile-ready/web-based |  | ✔ |  | ✔ |
| Search function |  |  | ✔ | ✔ |
| Event Management |  |  |  | ✔ |
| Updatable Content |  |  |  | ✔ |

Although with good remarks on visual representation of MSU-IIT campus data, the problem was also recurrent in the system of Ferrater et. al (2013) since it was built on a kiosk machine with little to no instructions on how to update relevant data. They had the same problem with Suroy Iligan whereas the data presented was also statically integrated. Additionally, mobility of kiosk machine can be costly on the end of users since they need to find these machines placed on multiple areas around the campus.

Even with these problems present, visualization of different data from the stated systems were potentially viable that their strong suits can still be integrated into one long-term dynamic application.

## **Usability**

Usability is a quality attributethat assesses how easy user interfaces are to use. The word "usability" also refers to methods for improving ease-of-use during the design process. Usability is defined by 5 quality components: *Learnability, Efficiency, Memorability, Errors* and *Satisfaction*. There are different methods in usability testing (Nielsen, 2012). One of which is the System Usability Scale.

The study provided valuable data concerning the usability of each individual tool; the researchers also derived some conclusions relevant to geovisualization techniques in general. They found that users were, in principle, able to understand and adopt the new ideas of map interactivity and manipulability. However, these ideas needed to be appropriately introduced; people could hardly grasp them just from the appearance of the maps and controls.

### **System Usability Scale**

SUS or System Usability Scale is a Likert Scale and a psychometric tool for measuring psychological quality of a product or a system. It provides measurement of people's subjective perceptions of usability in short time available mostly during evaluation. SUS has become an industry standard, referencing to over 1300 articles and publication (Brooke, 2013).

SUS has proved to be a valuable evaluation tool, being robust and reliable. It correlates well with other subjective measures of usability. SUS has been made freely available for use in usability assessment, and has been used for a variety of research projects and industrial evaluations. The SU scale is generally used after the respondent has had an opportunity to use the system being evaluated, but before any debriefing or discussion takes place. Respondents should be asked to record their immediate response to each item, rather than thinking about items for a long time. All items should be checked. If a respondent feel that s/he cannot respond to a particular item, they should mark the center point of the scale.

### **Likert Scale**

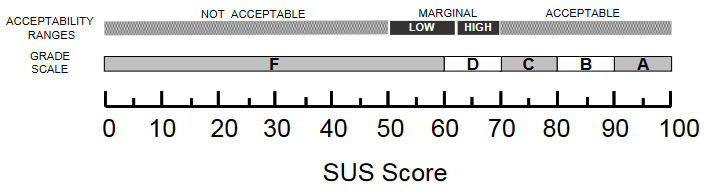
A Likert scale is a common rating format for surveys. Respondents rank quality from high to low or best to worst using five or seven levels. Likert scales were developed in 1932 as the familiar five-point bipolar response that most people are familiar with today. These scales range from a group of categories—least to most—asking people to indicate how much they agree or disagree, approve or disapprove, or believe to be true or false.

### **SUS Scoring Methodology**

For the system, SUS scoring is used. To calculate the SUS score, first add all the score contributions from each item. Each item's score contribution will range from 1 to 5. For items 1,3,5,7 and 9 the score contribution is the scale position minus 1. For items 2,4,6,8 and 10, the contribution is 5 minus the scale position. Multiply the sum of the scores by 2.5 to obtain the overall value of SU.

### **Interpreting SUS Scores**

A study by Bangor, Kortum & Miller (2009), through psychometric theory, suggests that multiple questions are generally superior to a single question to which many studies have found that multiple question surveys tend to yield more reliable results than single question surveys. Additionally, multiple interfaces of a product require multiple item questionnaires and single interfaced product requires single item questionnaire. This study constructed a useful analog, from 964 usability studies, to classify ranges of SUS Scores into traditional school grading scale, whereas it has proven useful in declaring strong face validity and the grading scale matches quite well with the acceptability scores, which is another tool constructed by this study of the same reference.



**Figure 2.1** SUS Score Equivalent Scale to Acceptability Range and Grade Scale

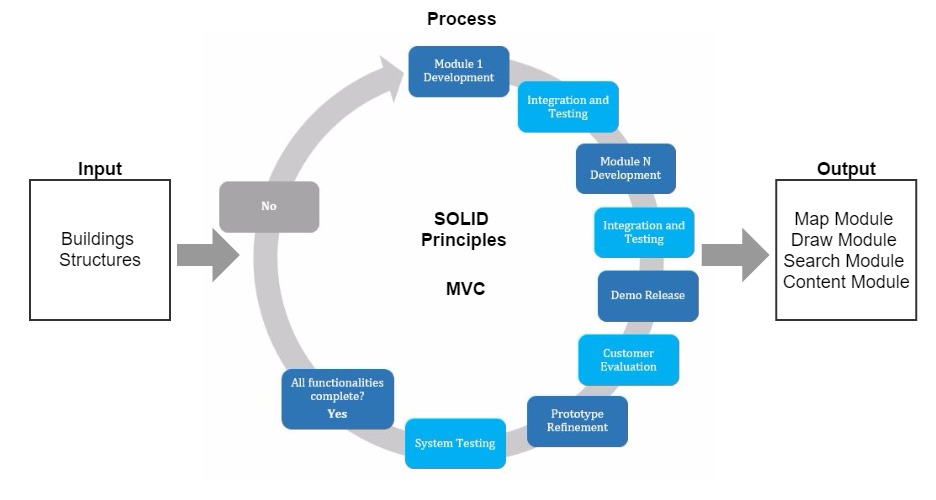
Although this project aimed to present an adjective rating as an additional single questionnaire survey to the collection of SUS scores scale interpretation, it is highly advisable not to include this rating in this research because of incompatibility presented in its discussions, whereas the adjective words used were not specific or were not understood enough for the testers.

### **Analysis of Scaling Methods Used by Existing Systems**

Lakbay IIT used a 5-response Likert scale ranging from one negative response to 4 gradually positive responses (Poor, Good, Average, Satisfactory and Excellent). This makes the product of the study perform with biases in such that the perception to the product itself is always positive. This made the product unreliable enough since it incorrectly conditions the user that the system functionalities are positively usable (Bangor, et. al., 2009).

        MSU-IIT Geolocator Information System used a standard scaling method for their analysis. It also uses 5-response Likert scale but with wordily equal scaling (Strongly Disagree, Disagree, Neutral, Agree and Strongly Agree). However, they presented their question in a positive tone, whereas it could bypass the negative traits of the system that could potentially take the long term usability for the system. With this, the researchers would be able to accurately evaluate the system's usability by using SUS score.

## **Summary and Conceptual Framework of the Study**



**Figure 2.2** Conceptual Framework

The input for this research is the information gathered by the project developers regarding buildings and structures. The developers will use GLMap and Leaflet, both are open source libraries similar to Google Maps. Both libraries use OpenStreetMap (OSM) for the mapping interface because it offers a familiar interface for the users and it allows them to layer custom information on top of the map and provide a platform that will allow them to add new features, like delivering relevant, location-based information. The project developers will also use OSM Buildings so the interactive map will display lightweight 3D-like polygons as buildings.

The development of this system composes of different modules is the terminal outcome of this research. These modules include a map module for displaying the map, a search module so that users can easily search for relevant data, a draw module so designated users can draw buildings in the map themselves and a content module for the database and CRUD functionalities of all events inside the campus. A schematic diagram is presented in Figure 2.1 that shows the relationship between these variables.

**CHAPTER 3**

# **THEORETICAL FRAMEWORK**

In this project, the project developers will develop the system by following a set of coding principles for a more dynamic and robust code structure. SOLID is a mnemonic acronym introduced by Michael Feathers for the five basic principles for object-oriented programming and design. It will be followed by constructing aesthetics as front-end of the system through several open source platforms and frameworks.

## **Interactive 3D Maps**

Three-dimensional maps use 3D computer graphics to present geographical information, using perspective representations that, to a certain degree, correspond to the real world. The view presented on a 3D map is more natural, intuitive and easier to comprehend than its 2D equivalent. 3D maps may, but do not have to, use real-3D data and volumetric objects. Earth surface representation, that incorporates height information, called 2.5D, complemented with 3D symbols, is decent enough for many applications (Raper, 1989).

3D maps are interactive by definition. Their usability is very restricted without the likelihood of interactive manipulation of the presented view, and unobstructed setting of the wanted perspective. The higher the level of interactivity which is provided, the more useful a 3D map becomes (Góralski, 2009).

However, it is not easy to define what is and what is not a 3D map. As with traditional maps there are diverse types and sorts of geographical presentations that use 3D visualization. A majority of research focuses on the broader subject of application of 3D in geographical visualization, or geovisualization, (MacEachren and Kraak, 2001). Representations used in this discipline range in levels of realism and presented data types. In the geovisualization’s understanding, a 3D map may be a realistic reconstruction of a city, or a planned landscape, as in geospatial virtual environments, (MacEachren et al., 2003).

This system will focus on a 3D campus map of MSU-IIT that present topography of buildings and structures and use a combination of different types of textures and symbols (3D objects and labels, 2D symbols, lines and polygons, text, numbers, points; selectable or not; multimedia and hyperlinks).

## **WebGL**

WebGL (Marrin, 2011) is an extension of HTML5 canvas element, which is now widely used for developing web applications requiring 3D visualization. It is a 3D graphics API, written in low level language and is based on OpenGL ES 2.0. To avoid complex low level programming, several WebGL-based frameworks have been developed, providing ease of development.

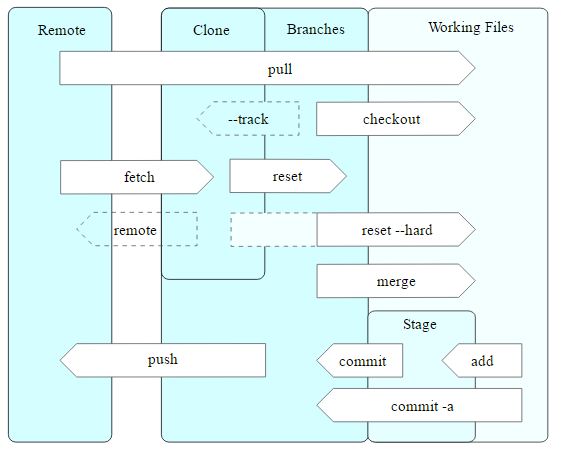
The system greatly uses WebGL in displaying the interactive campus map. It is with this technology that the map can be accessed in any browser anywhere.



**Figure 3.1** Sample WebGL Representation of a Terrain

## **Git Workflow**

By far, the most widely used modern version control system in the world today is Git. Git is a mature, actively maintained open source project originally developed in 2005 by Linus Torvalds, the famous creator of the Linux operating system kernel. A staggering number of software projects rely on Git for version control, including commercial projects as well as open source. Developers who have worked with Git are well represented in the pool of available software development talent and it works well on a wide range of operating systems and IDEs (Integrated Development Environments) (Atlassian, 2005).



**Figure 3.2** Git Operations

## **SOLID Principles**

Since the application requires an extensive but robust number of APIs, extra functionalities might get entangled with the other functions or mismanagement in writing the code would result to an unstable product. With this, the researchers will use the 5 basic principles of object-oriented programming in creating a flexible and ready to use and updatable product.

### **Single Responsibility**

In implementing functions of the system, classes of these functions should be considerably written to one reason to chance. This represents a good way of identifying classes (as to their use) during the design phase of the application.

*Example (might change due to the relevance of code)*

// single responsibility principle - good example

interface IEmail {

public void setSender(String sender);

public void setReceiver(String receiver);

public void setContent(IContent content);

}

interface IContent {

public String getAsString(); // used for serialization

}

class Email implements IEmail {

public void setSender(String sender) {// set sender; }

public void setReceiver(String receiver) {// set receiver; }

public void setContent(IContent content) {// set content; }

}

In the project, single responsibility will be mostly observed in constructing functionalities of the system to minimize confusion but maximizes effectiveness of different tools from different frameworks.

### **Open Close Principle**

Software entities like classes, modules and functions should be *open for extension* but *closed for modifications***.** This creates flexibility on the child classes by introducing abstraction to its parent class methods. So if a functionality of a system can be reused to another, it is applicable through this principle.

### **Liskov's Substitution**

Derived types must be completely substitutable for their base types. This ensures the system that any new derived models or classes to be implemented in the system would not change the behavior or the previous functionalities.

### **Interface Segregation**

The Interface Segregation Principle states that clients should not be forced to implement interfaces they don't use. Instead of one fat interface many small interfaces are preferred based on groups of methods, each one serving one submodule. In the system, the principle will be useful especially when applying tools of different APIs on a single method or functionality of the application.

### **Dependency Inversion**

High-level modules should not depend on low-level modules. Both should depend on abstractions. Abstractions should not depend on details. Details should depend on abstractions. With this, the system's modules would be flexible for future modification and/or updates.

## **MVC**

In object-oriented programming development, model-view-controller (MVC) is the name of a methodology or design pattern for successfully and efficiently linking the user interface to underlying data models. The MVC pattern has been prefigured by many developers as a useful pattern for the reuse of object code and a pattern that permits them to significantly lessen the time it takes to develop applications with user interfaces.

The model-view-controller pattern proposes three main components or objects to be used in software development:

* A *Model*, which represents the underlying, logical structure of data in a software application and the high-level class associated with it. This object model does not contain any information about the user interface.
* A *View*, which is a collection of classes representing the elements in the user interface (all of the things the user can see and respond to on the screen, such as buttons, display boxes, and so forth)
* A *Controller*, which represents the classes connecting the model and the view, and is used to communicate between classes in the model and view.

## **Laravel 5**

Laravel 5, created by Taylor Otwell and intended for the development of web applications following the model–view–controller (MVC) architectural pattern, is a web application framework with expressive, elegant syntax. Laravel attempts to take the pain out of development by easing common tasks used in the majority of web projects, such as authentication, routing, sessions, queueing, and caching. It aims to make the development process a pleasing one for the developer without sacrificing application functionality. It is accessible, yet powerful, providing powerful tools needed for large, robust applications. A superb inversion of control container, expressive migration system, and tightly integrated unit testing support give you the tools you need to build any application with which you are tasked.

## **OpenGL**

Open Graphics Library is a cross-language, cross-platform application programming interface (API) for rendering 2D and 3D vector graphics. The API is typically used to interact with a graphics processing unit (GPU), to achieve hardware-accelerated rendering. OpenGL is designed as a streamlined, hardware-independent interface to be implemented on many different hardware platforms.

## **OpenStreetMap**

Google Maps is a commercialized data map library whereas its services were exponentially expensive. Meanwhile, OpenStreetMap is an open data project. They release raw data "vector" maps that together with OSM Elements make up the virtual map. With this short discrepancy, OpenStreetMap has the greater potential for larger contribution since data presented are raw and open to all. However, Google Maps has more active services and larger data diversity which means that potentially the developed system architecture has already been established (which also costs more). With this considered, the researchers would use OpenStreetMap as one of the main frameworks to be used in the system.

## **Leaflet**

Leaflet is the leading open-source JavaScript library for mobile-friendly interactive maps. It has all the mapping features most developers ever need. It is designed with simplicity, performance and usability in mind. It works efficiently across all major desktop and mobile platforms, can be extended with lots of plugins, has a beautiful, easy to use and well-documented API and a simple, readable source code that is a joy to contribute to.

## **OSM Buildings**

OSM Buildings is a 3D renderer that uses geometry data from OpenStreetMap available under OpenDatabase License. Three.js is also a 3D rendering Library written in Javascript that uses WebGL technology. Although they act on the same purpose, Three.js has more dynamic functionalities than OSM Buildings. However, OSM Buildings works well with OpenStreetMap since every visible element within the OpenStreetMap acts as a rendered 3D and 2D resources.

In the initial development of the system, the researchers started with Three.js with Google Maps as framework. But the extensive library of Three.js and the limited free resources from Google Maps hinders the researchers to control the aesthetic design of the developed system. Thus then the researchers decided to use OSM Building and its related library for the system instead.



**Figure 3.3** OSM Buildings Representation

## **GeoJSON**

GeoJSON is a format for encoding a variety of geographic data structures. A GeoJSON object may represent a geometry, a feature, or a collection of features. This format will be used to most functionalities required by this system since the selected frameworks follows the same format as well as it has human readable syntax.

**CHAPTER 4**

# **WEB-BASED INTERACTIVE MAP OF MSU-IIT**

This chapter presents the overall specifications, functional requirements and nonfunctional requirements of the software being developed.

## **Systems Requirements Analysis and Specifications**

### **System’s Functions**

The system was developed in a modular approach. Each module has a set of functions. These functions are listed and summarized at the succeeding section.

#### ***Functional Requirements***

The main functional requirements of the system are the following:

* Displays a campus map of MSU-IIT using WebGL.
* Allows the users to navigate around the campus map.
* Presents information of a specific building in the map.
* Stores building and event information in the database.
* Allows management of data through CRUD functionalities.
* Allows the administrator to search, add, edit and delete building polygons in the map.

#### ***Non-functional Requirements***

The most important non-functional requirements of the system are the following:

* The system requires an internet browser.
* The system supports interaction with mouse, keyboard and touch screen.
* The system will run in both 64-bit and 32-bit.

#### *Map Module*

The map module is responsible for the visualization of the campus map. Visualization will be done using OSM Buildings’ integration of the coordinates of the buildings. Functions of this module are displaying of the campus map in 3D perspective, navigating around the map and displaying building information on the map.

#### *Content Module*

This module consists of two parts: the database and data management through CRUD (Create, Read, Update and Delete) functions. This module is mainly developed using MySQL for the database. This module is responsible for storing all building and event information and management of these data through the use of jQuery.

#### *Draw Module*

This module is responsible for the drawing of low polygon buildings. This lets the administrator draw polygons using coordinates. These coordinates will be saved in the database through the content module. This module is implemented through the use of Leaflet and Leaflet Draw library. This module’s functions include adding a new building polygon, editing the coordinates of a building polygon and deleting a building polygon.

#### *Search Module*

This module is responsible for the searching of all building and event information stored in the system. This module is developed using Datatables JavaScript library.

### **Physical Environment and Resources**

The interactive map is a web-based application that can be run in any internet browser. WebGL is used in visualizing lightweight 3D elements in any platform, thus enabling the system to be accessed almost anywhere as long as the user is connected to the internet.

#### Technical Requirements

System Requirements

Desktop

* CPU: 2.33Ghz or faster x86-compatible processor, or Intel® Atom™ 1.6GHz or faster processor for netbooks
* RAM: (WebGL adjusts to the provided memory)
* Video Card: 1GB of graphics memory
* Operating System**:** Windows XP (or higher), Linux or Mac
* Browser**:** Internet Explorer 8.0 or later, latest versions of Microsoft Edge, Mozilla Firefox, Google Chrome, Opera or Safari

Mobile

* RAM: WebGL adjusts to the provided memory
* Browser**:** Internet Explorer 8.0 or later, latest versions of Microsoft Edge, Mozilla Firefox, Google Chrome, Opera or Safari

#### User Specification

*End Users*

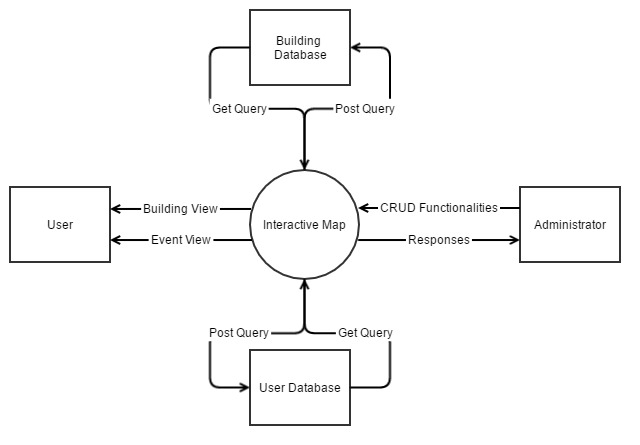
End users can be divided into different user groups: prospective students, students and visitors. These are anyone who are interested to explore the MSU-IIT campus. Educational level and experience is not a constraint but the user should be computer literate enough to know the elements and functionalities of the system.

*Administrator*

        The administrator is capable of adding, editing and deleting buildings and information in the system. S/he is also responsible for the overall maintenance of the system. S/he should be computer literate, has knowledge of how the system works, should identify the different elements and functionalities of the system. S/he should be a member of the Physical Plant Division.

## Design Models

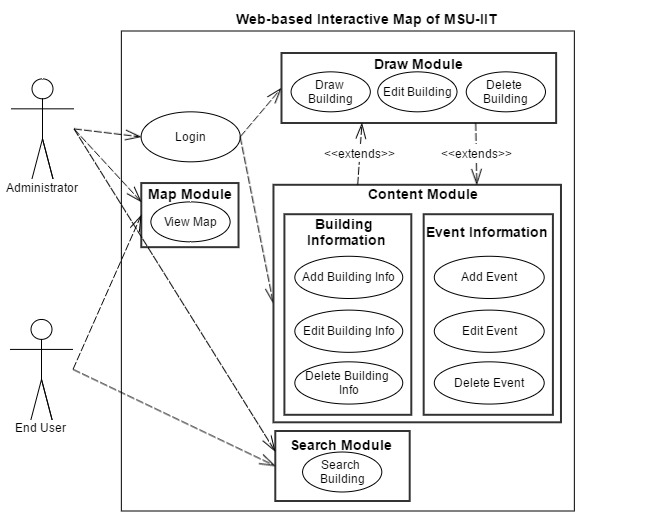
### Context Diagram



**Figure 4.1** Context Diagram

### UML Use Case Model

#### Use Case Diagram



**Figure 4.2** Use Case Diagram for Administrator and User

#### Use Case Specification

##### Login/Logout

**Table 4.1** Login Use Case Specification

|  |  |
| --- | --- |
| **Use Case Name** | **Login** |
| **Use Case ID** | **L1.1** |
| **Description** | This use case describes the actions the administrator will do to log into the system. |
| **Actor/s** | Administrator |
| **Trigger** | The administrator will click the “Login” button. |
| **Preconditions** | All elements of the page are loaded successfully. |
| **Normal Flow** | 1. The application requests for user’s login credentials. 2. The administrator enters his/her username and password. 3. The administrator will click the “Sign in” button. 4. The application validates the username and password provided by the user. 5. The administrator will be directed to the main page. 6. Use case ends. |
| **Alternative Flow** | 1A. The user forgot his/her credentials, s/he will click on the “Forgot Password?” link.   1. The administrator will be redirected to the “Retrieve Password” page. 2. The administrator will input his/her email address. 3. The administrator will click the “Send to Email” button. 4. The application will make the API call for password retrieval. 5. The application will send an email to the email address provided. 6. Use case resumes in step 1. |
| **Exceptional Flow** | If the administrator fails to enter the right username and password, the system will prompt an error message: “The username/password is incorrect.” |
| **Post condition** | The administrator is now logged into the system. |

**Table 4.2** Logout Use Case Specification

|  |  |
| --- | --- |
| **Use Case Name** | **Logout** |
| **Use Case ID** | **L1.2** |
| **Description** | This use case describes the actions the administrator will do to logout of the system. |
| **Actor/s** | Administrator |
| **Trigger** | The administrator will click the “Logout” button. |
| **Preconditions** | The administrator is in the main page. |
| **Normal Flow** | 1. The administrator will click the “Logout” button. 2. The application will prompt the user if s/he is sure. 3. The use case ends. |
| **Alternative Flow** | 2A. The administrator clicks the “Yes” button.   1. The administrator will be back to the login page. 2. Use case resumes in step 2.   2B. The administrator clicks the “No” button.   1. The administrator remains in the main page. 2. Use case resumes in step 2. |
| **Post condition** | The administrator is now logged out of the system. |

##### Map Module

**Table 4.3** View Map Use Case Specification

|  |  |
| --- | --- |
| **Use Case Name** | **View Map** |
| **Use Case ID** | **M2.1** |
| **Description** | This use case describes the actions the administrator or end user will do to view the map. |
| **Actor/s** | * Administrator * End user |
| **Trigger** | * The administrator will visit the website. * The end user will visit the website. |
| **Preconditions** | * The administrator is in the main page of the website. * The end user is in the main page of the website. |
| **Normal Flow** | 1. Administrator    1. The administrator will go to the website.    2. Use case ends. 2. End user    1. The end user will go to the website.    2. Use case ends. |
| **Alternative Flows** | * The administrator can cancel by clicking the “Cancel” button. * The end user can cancel by closing the page. |
| **Exceptional Flow** | If login credentials are incorrect, the website will prompt an error message saying “Incorrect username/password.” |
| **Post condition** | The administrator/end user can view the map. |

##### Draw Module

**Table 4.4** Draw Building Use Case Specification

|  |  |
| --- | --- |
| **Use Case Name** | **Draw Building** |
| **Use Case ID** | **D2.1** |
| **Description** | This use case describes the actions the administrator will do to draw buildings in the map.  This extends with the content module. |
| **Actor/s** | Administrator |
| **Trigger** | The administrator is in the Map Editor by clicking on the “Map Editor” menu item. |
| **Preconditions** | The administrator is logged into the application. |
| **Normal Flow** | 1. The application will display a map with Leaflet’s map drawing tools. 2. The application will also provide a form for the administrator to fill out with information about the building. 3. The administrator will draw a polygon using points. 4. The administrator will fill out the form provided. 5. The application will prompt if the process is successful. 6. Use case ends. |
| **Alternative Flow** | The administrator can cancel by clicking the “Cancel” button. |
| **Exceptional Flow** | If the administrator doesn’t fill out all the required fields, the application will prompt an error message saying “You need to fill out all fields.” |
| **Post condition** | A new building is drawn. |

**Table 4.5** Edit Building Use Case Specification

|  |  |
| --- | --- |
| **Use Case Name** | **Draw Building** |
| **Use Case ID** | **D2.2** |
| **Description** | This use case describes the actions the administrator will do to edit buildings in the map. |
| **Actor/s** | Administrator |
| **Trigger** | The administrator is in the Map Editor by clicking on the “Map Editor” menu item. |
| **Preconditions** | The administrator is logged into the application. |
| **Normal Flow** | 1. The application will display a map with Leaflet’s map drawing tools. 2. The administrator will click on a building in the map. 3. The application will give an option to edit the building.   C1. If the administrator chooses to edit the building:   1. The application will display the points of the building; these points can then be manipulated in the map. 2. The administrator will also replace the information about the building. 3. The administrator clicks the “Update” button. 4. Use case ends.   C2. If the administrator cancels:   1. The administrator clicks the “Cancel” button. 2. Use case ends. |
| **Post condition** | The chosen building is updated. |

**Table 4.6** Remove Building Use Case Specification

|  |  |
| --- | --- |
| **Use Case Name** | **Remove Building** |
| **Use Case ID** | **D2.3** |
| **Description** | This use case describes the actions the administrator will do to remove buildings in the map. |
| **Actor/s** | Administrator |
| **Trigger** | The administrator is in the Map Editor by clicking on the “Map Editor” menu item. |
| **Preconditions** | The administrator is logged into the application. |
| **Normal Flow** | 1. The application will display a map with Leaflet’s map drawing tools. 2. The administrator will click on a building in the map. 3. The application will give an option to remove the building. |
| **Alternative Flow** | C1. If the administrator chooses to remove the building:   1. The administrator clicks the “Remove” button. 2. Use case ends.   C2. If the administrator cancels:   1. The administrator clicks the “Cancel” button. 2. Use case ends. |
| **Post condition** | The chosen building is removed. |

##### Content Module

**Table 4.7** Add Building Information Use Case Specification

|  |  |
| --- | --- |
| **Use Case Name** | **Add Building Information** |
| **Use Case ID** | **C5.1** |
| **Description** | This use case describes the actions the administrator will do to add information about a building. |
| **Actor/s** | Administrator |
| **Trigger** | The administrator is in the Map Editor by clicking on the “Map Editor” menu item. |
| **Preconditions** | The administrator is logged into the application. |
| **Normal Flow** | 1. The application will display a map of the campus. 2. The administrator will click on a building in the map. 3. The application will provide a form for the administrator to fill out with information about the building. 4. The administrator will fill out the form provided. 5. The application will prompt if the process is successful. 6. Use case ends. |
| **Alternative Flow** | The user can cancel by clicking the “Cancel” button. |
| **Post condition** | Information about a building is added. |

**Table 4.8** Edit Building Information Use Case Specification

|  |  |
| --- | --- |
| **Use Case Name** | **Edit Building Information** |
| **Use Case ID** | **C5.2** |
| **Description** | This use case describes the actions the administrator will do to edit information about a building. |
| **Actor/s** | Administrator |
| **Trigger** | The administrator is in the Map Editor by clicking on the “Map Editor” menu item. |
| **Preconditions** | The administrator is logged into the application. |
| **Normal Flow** | 1. The application will display a map of the campus. 2. The administrator will click on a building in the map. 3. The application will provide a form for the administrator to fill out with information about the building. 4. The administrator will fill out the form provided. 5. The application will prompt if the process is successful. 6. Use case ends. |
| **Alternative Flow** | The user can cancel by clicking the “Cancel” button. |
| **Post condition** | Information about chosen building is updated. |

**Table 4.9** Remove Building Information Use Case Specification

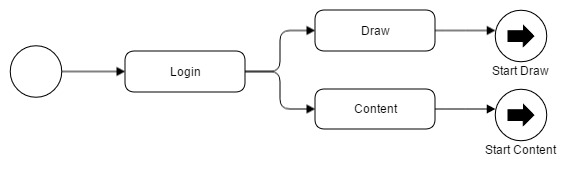
|  |  |
| --- | --- |
| **Use Case Name** | **Remove Building Information** |
| **Use Case ID** | **C5.3** |
| **Description** | This use case describes the actions the administrator will do to add information about a building. |
| **Actor/s** | Administrator |
| **Trigger** | The administrator is in the Map Editor by clicking on the “Map Editor” menu item. |
| **Preconditions** | The administrator is logged into the application. |
| **Normal Flow** | 1. The application will display a map of the campus. 2. The administrator will click on a building in the map. 3. The application will provide an option for the administrator to remove all information about the building. 4. The administrator will click the “Remove Information” button. 5. The application will prompt the administrator if he/she is sure. 6. The application will prompt if the process is successful. 7. Use case ends. |
| **Alternative Flow** | The user can cancel by clicking the “Cancel” button. |
| **Post condition** | Information about chosen building is removed. |

##### Search Module

**Table 4.10** Search Building Use Case Specification

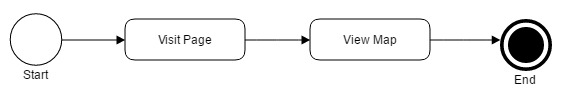
|  |  |
| --- | --- |
| **Use Case Name** | **Search Building** |
| **Use Case ID** | **S6.1** |
| **Description** | This use case describes the actions the administrator will do to search for buildings. |
| **Actor/s** | * Administrator |
| **Trigger** | * The administrator will click on the search bar in the Buildings page. |
| **Preconditions** | * The administrator has logged in and is in the main page of the website. |
| **Normal Flow** | 1. The administrator will input the name of the building inside the campus. 2. The application will present results of the search. 3. The administrator will choose on the list of results. 4. The administrator will be directed to that building. 5. The application will show details of that building. 6. Use case ends. |
| **Alternative Flow** | 1. The administrator inputs an erroneous search. 2. The application prompts with an error message. 3. Use case ends. |
| **Post condition** | The search will be successful. |

### Activity Diagrams



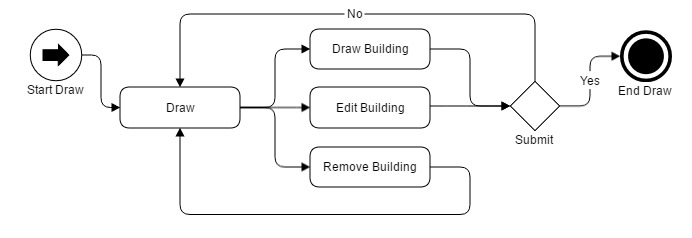
**Figure 4.3** Login Activity Diagram

To control the content within the system, administrators must perform login activity through a special route. Then, s/he can perform draw map activity or add content activity, etc.



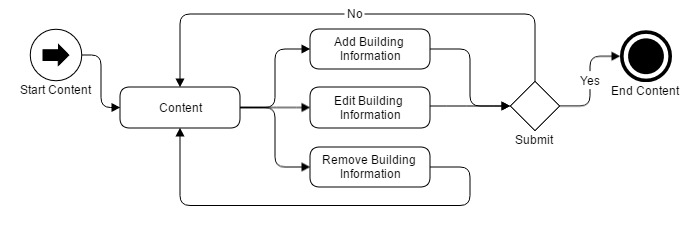
**Figure 4.4** Map Module Activity Diagram

Based on Figure 4.4, web visitors can navigate the main service by just going to its front page. The system then performs a set of task such as requesting data information from the database to render the map module.



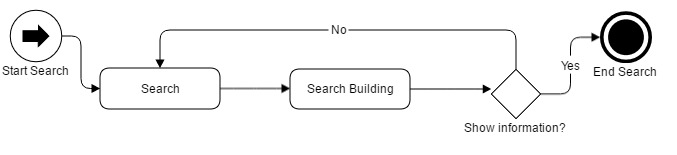
**Figure 4.5** Draw Module Activity Diagram

With authenticated user privileges, draw module enables the user to have draw, edit and remove activity in modifying the data within the system. In Figure 4.5, draw and edit building prompts the user to verify any changes or added information. However, remove building activity would redirect the authenticated user to the main section of the module. Otherwise the activity sufficed then terminated.



**Figure 4.6** Content Module Activity Diagram

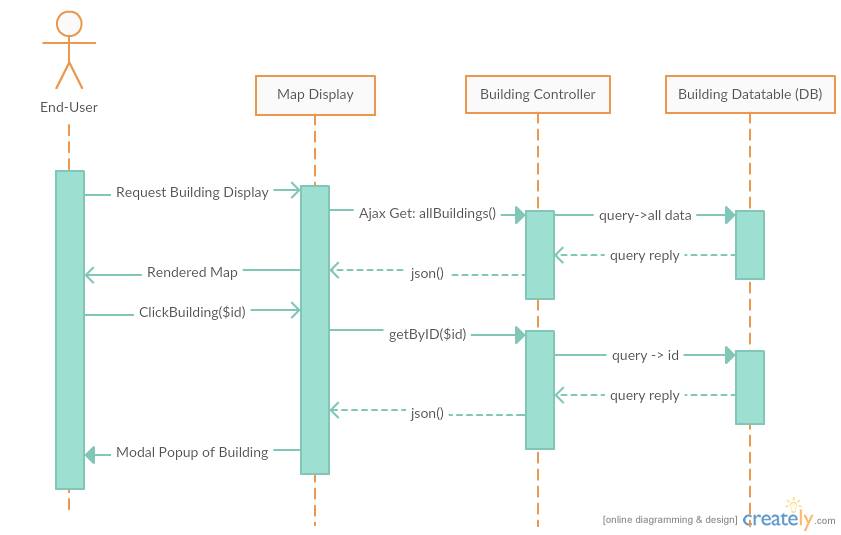
The same with Figure 4.5, the authenticated user can add, edit or delete building information. Add and edit activity also prompts verification message to complete the activity.



**Figure 4.7** Search Module Activity Diagram

Lastly, search functionality was only integrated within the content module thus limited to authenticated user. Users with administrative rights can search and modify the data by clicking it to show its information. Until then the search activity terminates.

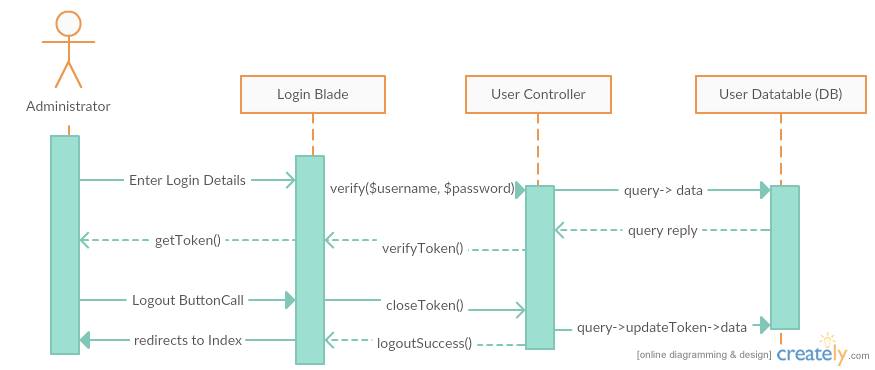
### Sequence Diagrams



**Figure 4.8** Sequence Diagram for End-User

Through Laravel framework, requesting data is easily achievable. As shown in Figure 4.8, the End-user, when visiting the front page, issues a request in his/her device to the browser, the browser which handles the Map Display as view component then sends Ajax Get request on the building controller. The building controller then sends query to the database of structural data and so on. The database service then queries the reply. The controller converts the data in JSON format which then the view component arranges and converts into relevant and readable information for the user.

Same goes when the end user clicked on a selected building, the module interprets a get request, the controller interprets the request and the database issue a query.



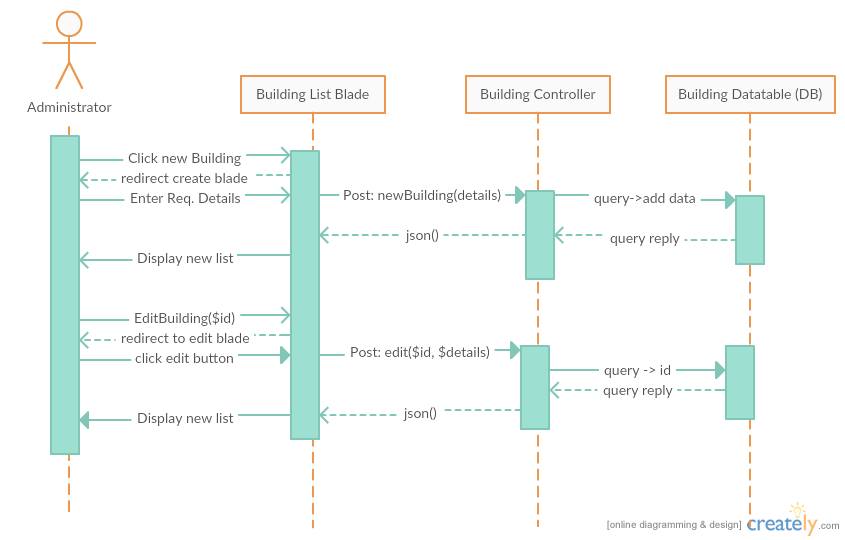
**Figure 4.9** Sequence Diagram for Administrator Authentication

To authenticate a user, s/he must enter a designated login details. The view component which is the login blade will issue a verification to the user controller, then the controller queries the request to the database. The database will issue a reply the controller repacks the reply through verifyToken function. This will send a set of string as token for the user activity.

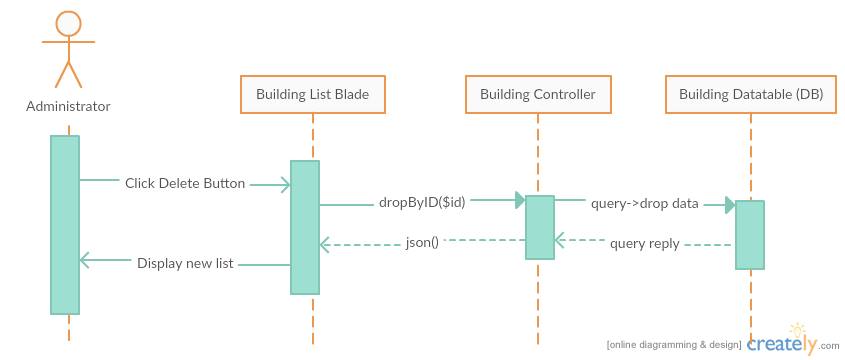
The logout button will be present in the content blades however its functionality runs with the controller. The controller sends a query to the database at the same time closes the verified token of the user and redirects the user to the front page.

In creating building data requires the user to navigate to the create blade view component. Upon redirection to the right blade, the admin may now enter the required details. The data made will be posted to the controller then will be added to the database. The database will then send a reply which will be packed into JSON format for the view component to display the newly made data.

Additionally, in Figure 4.10, edit buildings follow the same steps of creating buildings but requires to choose on what building to edit.

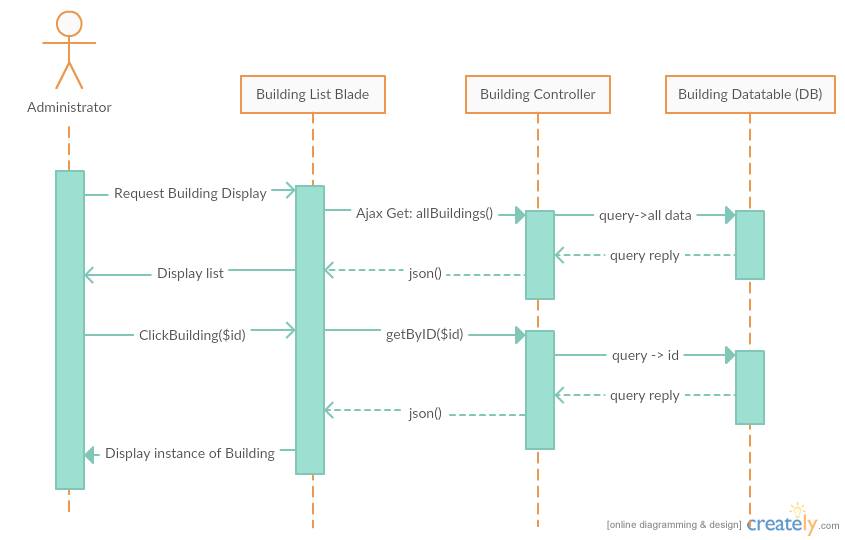


**Figure 4.10** Sequence Diagram for Administrator Create and Edit



**Figure 4.11** Sequence Diagram for Administrator Delete

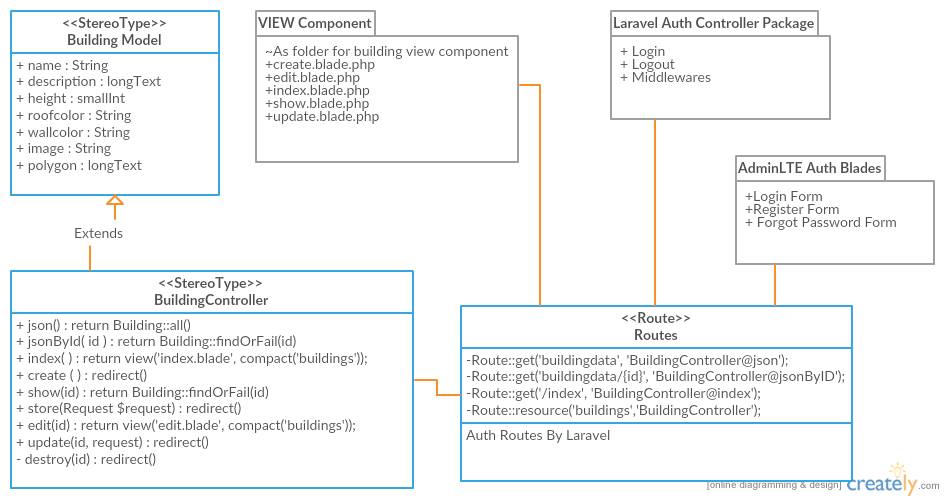
Finally, the administrator can delete selected data by a push of a button. The blade will instantly interpret the data to be dropped and will be sent to the building controller. The controller then queries to drop the data from the table and after which the database queries a reply. The controller issues another JSON format data to refresh the list.



**Figure 4.12** Sequence Diagram for Administrator View

The administrator follows the same sequence with End-user in viewing data but the difference is the blade component used. It can be explained with the descriptions in figure 4.8.

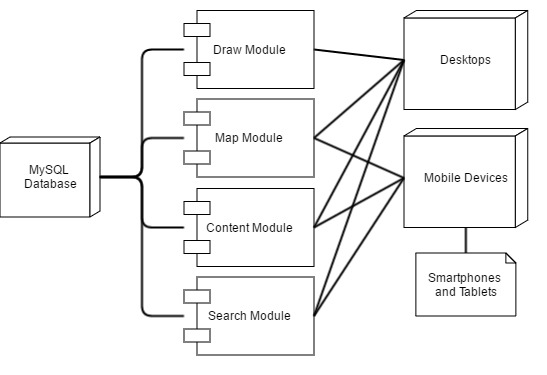
### Class Diagram



**Figure 4.13** Class Diagram of the Developed System within Laravel Framework

Laravel framework follows an eloquent class hierarchy. Models are constructed through Laravel own model class interface. Additionally, it is already connected through a database management system but this project uses MySQL services. Controllers are also extendedly created from an interface of the Laravel services. And routes are premade classes that act as an intermediary to address route, view components and controllers. With the same practices, the project will make use of pre made packages within AdminLTE framework and Laravel for easier build.

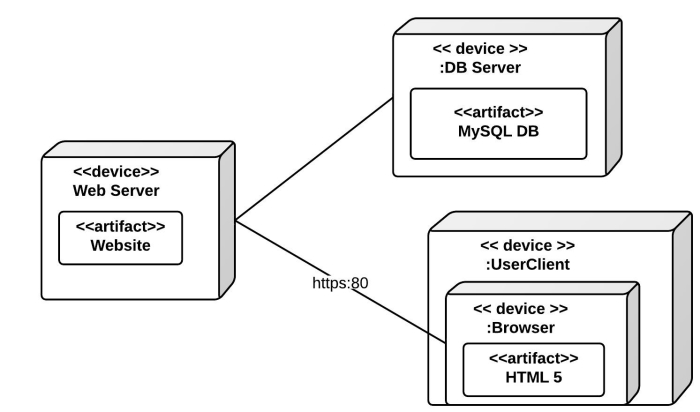
### Component Diagram

****

**Figure 4.14** Component Diagram

Through Figure 4.14, selective modules only work on specific devices, whereas the map, content and search modules are the only modules available for mobile devices. However, all of the modules can be used for desktop and larger screen device users.

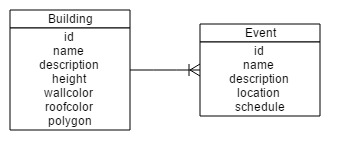
### Deployment Diagram

****

**Figure 4.15** Deployment Diagram

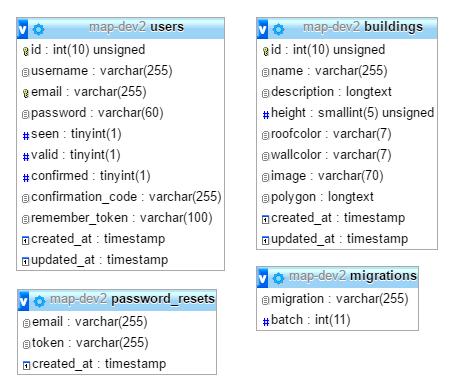
### Database Design Model

#### Entity Relationship Diagram



**Figure 4.16** Entity Relationship Diagram

#### Schema

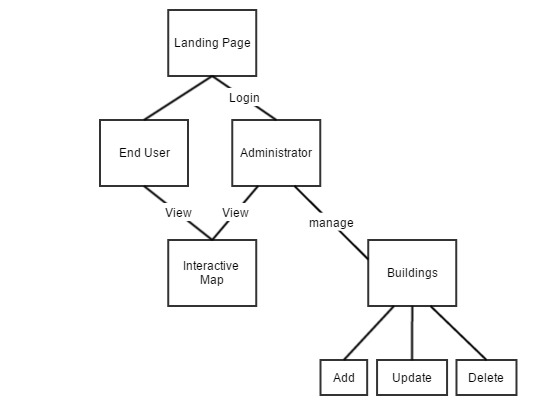


**Figure 4.17** Database Schema

## **Structure and Interface**

This section discusses the system functions through screen shots. Walkthrough to the system’s interface will be reflected here.

### Program Structure



**Figure 4.18** Program Structure of the System

## **Development Phases**

### Pre-development Phase

On a heuristic approach, the project developers established the system that introduces different visualization of buildings and structures of MSU-IIT.



**Figure 4.19** Interactive Map Prototype using Google Maps

Initially, the system used polygon information integrated on Google Maps whereas it highlighted different sections in representation of 2-dimensional shapes within Google Maps. In line with this, the developers aimed to represent buildings with another layer of representation thus the problem for isometric visualization from GIS MSU-IIT. The project then introduced isometric visualization by elevating a duplicate copy of the shape of an area added with additional polygon shape as side or visual perimeter for the represented buildings. However, isometric visualization requires the underlying Google map to be in a 45-degree angle view in order to best represent the buildings and structures.

Isometric representation was limited on Google Maps because some areas were not covered with 45-degree map visuals especially in the Philippines. Google Map requires additional financial cost to developers for its different services particularly in 45-degree visual data. This lags the development process thus different programming and geographical map library was searched.

### Prototype Development



**Figure 4.20** Interactive Map Prototype Using OpenStreetMap and OSM Buildings

In contrast to Google Maps and its API library, OpenStreetMap was chosen to be the geospatial data provider for this project. OpenStreetMap is a free open data system which has limited data regarding the Philippines.

Using Leaflet.js library, the OSM canvas was efficiently controlled and few features of the library was integrated to the system for the functionalities needed in the system. OpenStreetMap API can read GeoJSON format hence illustrating 3D low polygons of buildings and structures of the campus is easier to integrate. It was verified by seeding data of the buildings in the prototype. With this, a database can be expected in the system to contain these 3D data and other content.

### Summary of the Development Process

The prototype was integrated with Laravel 5 website development framework. These include GUI designs by using the AdminLTE template to the environment and migrating sample data to the database was done through Laravel database seeding services. Additionally, models in a form of classes to represent each data entity and controllers for routing and information control were also made through the same service.

With this done, the requirements of making the modules was first supplying the view component through GUI designs. Additionally, models of buildings and user data were created for the view component. These models act as an intermediary component to the database whereas the entity variables were declared in the model classes. These variables then were used by the controller component to be routed to the different parts of the view components.

In the map module, integration of multiple libraries was done mostly within the view components since these were JavaScript libraries and require less in the model component management. First, OpenSteetMap was added to render the map tiles and other services. OSM Buildings was then added to render 3D models of the building structures, however in the initial prototype the system used an external file with GeoJSON data of the structures. Lastly, Leaflet library was added for functionalities present in the map. To make this component dynamic since it was using a static information in displaying 3D maps, an additional code using AJAX to get building data through a selected route then converting the retrieved data into GeoJSON format. With this, a selected address route for building structures was required for this module to work.

The content module was created by following the MVC framework. First, models and its data variables were made using the Laravel Eloquent programming technique which followed SOLID principles. Initially, building information was migrated to MySQL database with a model class within the system directory folder. Inside the model class, data variables were declared. With this, building controller was made to use this model for data management and other end-user functionalities as services. The building controller contains different functions or methods that channels different data based on the requirements. In instance, a function that returns building structural data to complete the map module services was added within the building controller. Additionally, an external class called route.php routes these data by providing a registered address to where they are addressed. Lastly, multiple view components or Laravel blade pages were made to show the controlled data. In an instance, a blade for creating building data was accomplished by registering a routed data that is connected to a controller method within the building controller.

To complete the component module, CRUD functionalities were followed where as a create page was made for creating building data, a list page was made for displaying the list of building data, update page was made for updating the selected data and a delete function to delete a selected data. These pages have their own methods within the building controller and can be replicated for other entities as content.

The draw module is a specialized module of content and map modules since it follows the same mechanics however offers more complex services since this module relies on the creating and updating of a map and its contents. Integration of the same services was done. Additionally, Leaflet draw functions was used to retrieve the data created and updated on the page to be posted through the route class then to the content controller.

Lastly, a search module was integrated within the content page for the buildings. Yajra Datatables services was integrated for the search functionality for the content and can be integrated to other services or content. **CHAPTER 5**

# **RESULTS AND DISCUSSION**

The web-based interactive map of MSU-IIT was evaluated through a pilot testing of thirty (30) respondents. The project developers were present during the hands-on testing of the system.

**Figure 5.1** Test User Groups Percentage of the Pilot Test

The test users were composed of five (5) administrators from the Physical Plant Division, nine (9) college students, eight (8) visitors and eight (8) Grade 10 students from La Salle Academy. The project developers chose these user groups because they represent those who can directly use the system.

**Figure 5.2** Gender Percentage of the Pilot Test

Based on the preliminary questionnaires given to the test users, 16 were female and 14 were male. The test user’s age range was between 16 to 36 years old. 27 out of 30 said that they have used maps or directions as guidance in visiting places. 14 out of 16 visitors and prospective students said that they have visited MSU-IIT before. 17 out of 25 students, visitors and prospective students said that they got lost inside the campus while looking for their destination. Lastly, 22 out of 25 students, visitors and prospective students said that they would prefer a virtual 3D campus map than a traditional campus map. (See Table 5.1 for summary).

**Table 5.1** Preliminary Results

|  |  |  |
| --- | --- | --- |
| Question | Yes | No |
| Q1: Have you used maps or directions as guidance in visiting places? | 27 | 3 |
| Q2: Have you visited MSU-IIT before? (if visitor or prospective student) | 14 | 2 |
| Q3: When inside the campus, have you tried looking for a location but got lost instead? | 17 | 8 |
| Q4: Would you prefer a virtual 3D campus map than a traditional campus map? | 22 | 3 |

## **Feedback Results of the Interactive Map**

This section presents the results that were gathered during the hands-on testing of the interactive map. The pilot testing for the interactive map was done on the 3rd of May, 2016 from 11AM to 5PM.

After the testing, System Usability Scale questionnaires were given to the testers and were later gathered and analyzed by the researchers.

**Feedback on System Usability of the Interactive Map**

**Table 5.2** Administrators’ Feedback on System Usability of the Interactive Map

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Strongly Disagree | Disagree | Neutral | Agree | Strongly Agree |
| Q1: I think that I would like to use this system frequently |  |  |  | 2 | 3 |
| Q2: I found the system unnecessarily complex | 5 |  |  |  |  |
| Q3: I thought the system was easy to use |  |  |  | 1 | 4 |
| Q4: I think that I would need the support of a technical person to be able to use this system | 1 | 2 | 2 |  |  |
| Q5: I found the various functions in this system were well integrated |  |  |  | 4 | 1 |
| Q6: I thought there was too much inconsistency in this system | 1 | 4 |  |  |  |
| Q7: I would imagine that most people would learn to use this system very quickly |  |  |  | 2 | 3 |
| Q8: I found the system very awkward to use | 5 |  |  |  |  |
| Q9: I felt very confident using the system |  |  |  | 3 | 2 |
| Q10: I needed to learn a lot of things before I could get going with this system | 3 | 2 |  |  |  |

Table 5.2 shows the summary of the administrators’ feedback on the System Usability test of the Interactive Map.

**Figure 5.3** Interactive Map System Usability Feedback of Administrators

Figure 5.3 shows the percentage of the students’ individual rating in the aspect of usability for each statement of the SUS.

**Table 5.3** Students’ Feedback on System Usability of the Interactive Map

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Strongly Disagree | Disagree | Neutral | Agree | Strongly Agree |
| Q1: I think that I would like to use this system frequently |  |  | 1 | 4 | 4 |
| Q2: I found the system unnecessarily complex | 2 | 2 | 5 |  |  |
| Q3: I thought the system was easy to use |  |  | 2 | 3 | 4 |
| Q4: I think that I would need the support of a technical person to be able to use this system | 5 | 2 | 2 |  |  |
| Q5: I found the various functions in this system were well integrated |  |  |  | 6 | 3 |
| Q6: I thought there was too much inconsistency in this system | 2 | 4 | 2 | 1 |  |
| Q7: I would imagine that most people would learn to use this system very quickly |  |  | 1 | 3 | 5 |
| Q8: I found the system very cumbersome to use | 4 | 4 | 1 |  |  |
| Q9: I felt very confident using the system |  |  | 2 | 2 | 5 |
| Q10: I needed to learn a lot of things before I could get going with this system | 6 | 2 | 1 |  |  |

Table 5.3 shows the summary of the students’ feedback on the System Usability test of the Interactive Map.

**Figure 5.4** Interactive Map System Usability Feedback of Students

Figure 5.4 shows the percentage of the students’ individual rating in the aspect of usability for each statement of the SUS.

**Table 5.4** Visitors’ Feedback on System Usability of the Interactive Map

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Strongly Disagree | Disagree | Neutral | Agree | Strongly Agree |
| Q1: I think that I would like to use this system frequently |  | 1 |  | 5 | 2 |
| Q2: I found the system unnecessarily complex | 3 | 3 | 2 |  |  |
| Q3: I thought the system was easy to use |  |  |  | 3 | 5 |
| Q4: I think that I would need the support of a technical person to be able to use this system | 4 | 3 | 1 |  |  |
| Q5: I found the various functions in this system were well integrated |  |  | 2 | 2 | 4 |
| Q6: I thought there was too much inconsistency in this system | 4 | 2 | 2 |  |  |
| Q7: I would imagine that most people would learn to use this system very quickly |  |  |  | 1 | 7 |
| Q8: I found the system very awkward to use | 3 | 3 | 1 | 1 |  |
| Q9: I felt very confident using the system |  |  | 1 | 3 | 4 |
| Q10: I needed to learn a lot of things before I could get going with this system | 6 | 2 |  |  |  |

Table 5.3 shows the summary of the visitors’ feedback on the System Usability test of the Interactive Map.

**Figure 5.5** Interactive Map System Usability Feedback of Visitors

Figure 5.5 shows the percentage of the visitors’ individual rating in the aspect of usability for each statement of the SUS.

**Table 5.5** Prospective Students’ Feedback on System Usability

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Strongly Disagree | Disagree | Neutral | Agree | Strongly Agree |
| Q1: I think that I would like to use this system frequently |  |  |  | 3 | 5 |
| Q2: I found the system unnecessarily complex | 5 | 3 |  |  |  |
| Q3: I thought the system was easy to use |  |  |  | 1 | 7 |
| Q4: I think that I would need the support of a technical person to be able to use this system | 6 |  | 2 |  |  |
| Q5: I found the various functions in this system were well integrated |  |  | 1 | 4 | 3 |
| Q6: I thought there was too much inconsistency in this system | 3 | 5 |  |  |  |
| Q7: I would imagine that most people would learn to use this system very quickly |  |  | 1 | 3 | 4 |
| Q8: I found the system very cumbersome to use | 6 | 2 |  |  |  |
| Q9: I felt very confident using the system |  |  |  | 4 | 4 |
| Q10: I needed to learn a lot of things before I could get going with this system | 5 | 2 | 1 |  |  |

Table 5.3 shows the summary of the prospective students’ feedback on the System Usability test of the Interactive Map.

**Figure 5.6** Interactive Map System Usability Feedback of Prospective Students

Figure 5.6 shows the percentage of the prospective students’ individual rating in the aspect of usability for each statement of the SUS.

**Table 5.6** Total Feedback on System Usability of the Interactive Map

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Strongly Disagree | Disagree | Neutral | Agree | Strongly Agree |
| Q1: I think that I would like to use this system frequently |  | 1 | 1 | 14 | 14 |
| Q2: I found the system unnecessarily complex | 15 | 8 | 7 |  |  |
| Q3: I thought the system was easy to use |  |  | 2 | 8 | 20 |
| Q4: I think that I would need the support of a technical person to be able to use this system | 16 | 7 | 7 |  |  |
| Q5: I found the various functions in this system were well integrated |  |  | 3 | 16 | 11 |
| Q6: I thought there was too much inconsistency in this system | 10 | 15 | 4 | 1 |  |
| Q7: I would imagine that most people would learn to use this system very quickly |  |  | 2 | 9 | 19 |
| Q8: I found the system very awkward to use | 18 | 9 | 2 | 1 |  |
| Q9: I felt very confident using the system |  |  | 3 | 12 | 15 |
| Q10: I needed to learn a lot of things before I could get going with this system | 19 | 8 | 2 |  | 1 |

Table 5.6 shows the summary of the total feedback of all participants on the System Usability test of the Interactive Map.

**Figure 5.7** Interactive Map System Usability Feedback of Prospective Students

Figure 5.7 shows the percentage of the prospective students’ individual rating in the aspect of usability for each statement of the SUS.

## **Scoring System Usability Scale**

SUS has a way of calculating the scores of each tester. To calculate the SUS score, first sum the score contributions from each item. Each item's score contribution will range from 0 to 4. For items 1,3,5,7, and 9 the score contribution is the scale position minus 1. For items 2,4,6,8 and 10, the contribution is 5 minus the scale position. Multiply the sum of the scores by 2.5 to obtain the overall value of SU. SUS scores have a range of 0 to 100. The standard passing score for SUS is 68, a SUS score above 68 would be considered above average and any score below 68 is deliberated as below average.

## **SUS Scores of Each Participant of Each User Group**

**Table 5.7** Administrators’ System Usability Results

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Participant | 1 | 2 | 3 | 4 | 5 | Total | **Overall** |
| SUS Score | 87.5 | 80 | 90 | 92.5 | 80 | 430 | **86** |

Table 5.7 shows that all of the SUS scores of the administrators passed the 68 score-mark.

**Table 5.8** Students’ System Usability Results

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Participant | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | Total | **Overall** |
| SUS Score | 82.5 | 90 | 67.5 | 87.5 | 75 | 67.5 | 80 | 87.5 | 90 | 727.5 | **81** |

Table 5.8 shows that 7 of the SUS scores of the students passed the 68 score-mark while 2 were below average.

**Table 5.9** Visitors’ System Usability Results

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Participant | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | Total | **Overall** |
| SUS Score | 70 | 80 | 90 | 77.5 | 90 | 90 | 100 | 77.5 | 675 | **84** |

Table 5.9 shows that all of the SUS scores of the visitors passed the 68 score-mark.

**Table 5.10** Prospective Students’ System Usability Results

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Participant | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | Total | **Overall** |
| SUS Score | 80 | 97.5 | 95 | 95 | 85 | 87.5 | 87.5 | 80 | 707.5 | **88** |

Table 5.10 shows that all of the SUS scores of the prospective students passed the 68 score-mark.

## **System Usability Scale Analysis**

### **System Usability Analysis for Administrators**

From the results shown in Table 5.7, the project developers obtained an average score of 86 for the administration group with all of them having scores above 80. This concludes that the administration group which mainly consists of the Physical Plant Division committee of the campus gives this product a passing grade.

### **System Usability Analysis for Students**

From the results shown in Table 5.8, the project developers obtained an average score of 81 for the students group with 2 of the respondents giving 67.5 score which is lower than the passing grade of 68. This may suggest that the student group would likely use the product only in their freshmen year. However, the average score reached the passing grade thus concludes the product can still be beneficial to resident students.

### **System Usability Analysis for Visitors**

From the results shown in Table 5.9, the project developers obtained an average score of 84 for the visitor group with only 3 of the participants have scores lower than 80. The average score passed the average grade thus concludes the system is usable for them.

### **System Usability Analysis for Prospective Students**

From the results shown in Table 5.10, the project developers obtained an average score of 88 for the prospective students with all of the scores above 80. This suggest that the system is useful to them since they will need this type of technology if they sought to enroll in MSU-IIT.

**CHAPTER 6**

# **SUMMARY, CONCLUSION AND RECOMMENDATIONS**

## **Summary of Development**

In this project, the project developers laid out the foundation of a dynamic 3D interactive map by constructing the modules: map, draw, content and search that were based from the recommendations of previous studies: Campil et al. (2010), Bala et al. (2012) and Ferrater et al. (2013).

The map module was integrated with OpenStreetMap for the map tiles, OSM Buildings for the 3D map models and Leaflet library for the integration of a map with CRUD functions. The draw module was made similarly with the map module but integrated with the content module for the add and update functionalities. The content module, which was constucted in Laravel 5 website development framework, allows dynamic data connection to the map module. Additionally, the search functionality was integrated within the content module using Yajra Datatables which allows dynamic searching for relevant data.

With this, the project achieved the first objective of developing and designing the dynamic interactive map for MSU-IIT.

## **Conclusion**

While most of the participants admitted using maps and direction as guidance, only few of them arrived at their destined places correctly. Since, the requirements of having a virtual map has already been established from previous studies. The project developers collected results from the usability evaluation of this system.

It showed that most of the participants were affected by the system positively with its design and potential use. Specifically, administrators were willing to learn and suggest more functionalities of the system. Additionally, visitors and students also gave the product with above average score, only a few of them showed minimal appreciation. Lastly, prospective students showed maximum appreciation of the system suggesting that it is useful for them since they will need this type of technology for future use.

## **Recommendations**

The following are some recommendations gathered from the participants and project developers for others who wish to further improve the system:

* 1. Add on-screen labels for buildings;
  2. Add or use true-to-scale legend;
  3. Additional visuals for buildings in the future;
  4. Show traffic route;
  5. Add room details and description;
  6. Add option to show proposed buildings in more complex 3D model.

# **References**

Allen, I. E., & Seaman, C. A. (2007, July). *Likert Scales and Data Analyses*. Retrieved from Quality Progress Web site: http://asq.org/quality-progress/2007/07/statistics/likert-scales-and-data-analyses.html

Andrienko, G. L., & Andrienko, N. V. (1999). Interactive maps for visual data exploration. *Journal of Geographical Information Science, Vol. 13, No. 4*, 355-374.

Andrienko, N., Andrienko, G., Voss, H., Bernardo, F., Hipolito, J., & Kretchmer, U. (2002). Testing the Usability of Interactive Maps. *Journal pf Cartography and Geographic Information Science, Vol. 29*, 325-342.

Atlassian. (2005). Git Workflow.

Bangor, A., Kortum, P., & Miller, J. (2009). Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *Journal of Usability Studies Vol. 4, Issue 3*, 114-123.

Brooke, J. (2013). SUS: A Retrospective. *Journal of Usability Studies Vol. 8, Issue 2*, 29-40.

Campil, L. F., Gomez, J. M., & Paulin, K. D. (2010, March). LAKBAY IIT: 3D VIRTUAL TOUR OF MSU-IIT CAMPUS.

Chaturvedi, K. (2014). *Web-based 3D Analysis and Visualization Using HTML5 and WebGL.* Enschede, the Netherlands: Faculty of Geo-Information Science and Earth Observation, University of Twente.

Faughnder, R. (2012, July 07). *With virtual campus tours, applicants explore colleges*. Retrieved from Los Angeles Times Web site: http://articles.latimes.com/2012/jul/07/business/la-fi-tn-virtual-tours-20120706

Ferrater, R. R., Gimeno, E. A., & Lipa, A. M. (2013, March). MSU-IIT Geolocator Information System: An Isometric Virtual Tour of the MSU-IIT Campus.

Haklay, M., & Tobón, C. (2002). Usability Evaluation and PPGIS: towards a usercentred. *first URISA PPGIS conference.* London.

Harrower, M., & Brewer, C. A. (2003). ColorBrewer.org: An Online Tool for Selecting Colour. *The Cartographic Journal, Vol. 40, No. 1*, 27–37.

James, K. (2011, August 18). *Top 5 Reasons You Need An Interactive Campus Map.* Retrieved from nuCloud Web site: https://nucloud.com/blog/top-5-reasons-you-need-an-interactive-campus-map/

Kraak, M. J. (2001). Web Cartography: Developments and Prospects. *Cartographic principles* (pp. 53-72). CRC Press.

MacEachren, A. M., & Brewer, I. (2003). Developing a conceptual framework for visually-enabled. *International Journal of Geographical Information Science*.

Marrin, C. (2011). WebGL Specification. *Khronos WebGL Working Group*.

Monitor, I. (2012, September 27). *The power of virtual tours in student recruitment*. Retrieved from ICEF Monitor Web site: http://monitor.icef.com/2012/09/the-power-of-virtual-tours-in-student-recruitment/

Nikoohemat, S. (2013). *Smart Campus Map.* Munich: Faculty of Civil, Geo and Environmental Engineering, Department of Cartography.

Paulk, M. C. (2002, October). Agile Methodologies and Process Discipline. *Research Showcase @ CMU*, pp. 15-18.

Raper, J. F. (1989). Key 3D modelling concepts for geoscientific. *NATO Advanced Research Workshop*, (pp. 215-232). Santa Barbara, California, USA.

Reichenbacher, T. (2001 ). Mobile Cartography – Adaptive Visualisation of Geographic Information on Mobile Devices. *Journal of Geographical Sciences, Vol. 11*.

Reichenbacher, T. (2001). ADAPTIVE CONCEPTS FOR A MOBILE CARTORAPHY. *Journal of Geographical Sciences, Vol. 11*, 43-53.

Roth, R. E. (2012). *Cartographic Interaction Primitives: Framework and Synthesis.* Wisconsin: University of Wisconsin‒Madison.

Roth, R. E. (2013). Interactive maps: What we know. *JOURNAL OF SPATIAL INFORMATION SCIENCE*, 59–115.

Roth, R. E., & Harrower, M. (2008). Addressing Map Interface Usability: Learning from the Lakeshore Nature Preserve Interactive Map. *Journal of Cartographic Perspectives, No. 60*, 46-66.

Roth, R. E., Van Den Hoek, J., Woodruff, A., Erkenswick, A., McGlynn, E., & Przybylowski, J. (2009). The 21st Century Campus Map: Mapping the University of Wisconsin-Madison. *Journal of Maps*, 1-8.

Weber, A., Jenny, B., Wanner, M., Cron, J., Marty, P., & Hurni, L. (2010). Cartography Meets Gaming: Navigating Globes, Block Diagrams and 2D Maps with Gamepads and Joysticks. *The Cartographic Journal, Vol. 47, No. 1*, 92–100.