DEVELOPING AUGMENTED REALITY PLACES OF INTEREST APPLICATION OF UNIVERSITI TEKNOLOGI PETRONAS (UTP)

by

KAMARUN FALIQ BIN AMIR FRIDAUS

DISSERTATION

Submitted to the Business Information System Programme

Universiti Teknologi PETRONAS

In partial fulfillment of the requirement for the

BACHELOR OF COMPUTER INFORMATION SCIENCES (Hons)

(BUSINESS INFORMATION SYSTEM)

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

September 2011

CERIFICATION OF APPROVAL

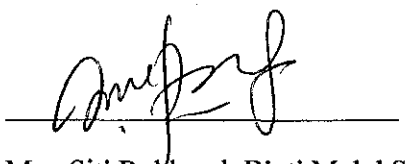
DEVELOPING AUGMENTED REALITY PLACES OF INTEREST APPLICATION OF UNIVERSITI TEKNOLOGI PETRONAS (UTP)

by

Kamarun Faliq Bin Amir Fridaus

A project dissertation submitted to the
Business Information System Programme
Universiti Teknologi PETRONAS
In partial fulfillment of the requirement for the
Bachelor of Computer Information Sciences (Hons)
(Business Information System)

Approved:



Mrs. Siti Rohkmah Binti Mohd Shukri

Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

September 2011

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

Kamarun Faliq Bin Amir Firdaus

ABSTRACT

This report is a preliminary step onto developing Augmented Reality places of interest application of Universiti Teknologi PETRONAS (UTP). As of 2007, UTP holds the recognition with the prestigious Aga Khan Award for Architecture in Malaysia. Having said that, domestic as well as international visitors annually visit UTP to apprehend the unique and award winning design of its campus. But the current problem which persists for visitors is the need for a guide and lack of a general source of information regarding the university's campus during their visits. With such problem in place, it is therefore extremely pivotal to have an Augmented Reality mobile application that will help them find significant places within the campus itself. This allows them to explore the entire university through their mobile phones and have useful information at the touch of their fingertips. The project will involve several phases; firstly the construction of the Augmented Reality application itself, followed by the analysis and design of the rules for the Augmented Reality application, the development of the Augmented Reality application, further testing and finally the implementation of the Augmented Reality application. These implementations, using the Rapid Application Development methodology with the Prototyping approach was further refined through the research and studies as well as feedback obtained from colleagues until it appropriately meets the objectives.

ACKNOWLEDGEMENTS

Alhamdulillah, praise be to Allah S.W.T, the most gracious and most merciful for giving me the strength and wisdom in completing this project. For this golden opportunity, first and foremost, I would like to express my heartiest gratitude to my supervisor, Pn Siti Rohkmah, for her understanding and professional ways in assisting and giving her encouragement, guidance, comment and ideas that are useful towards the development of this project.

In addition I would also like to convey my most sincere thanks to Vincent Yong my former colleague who first introduced me to Augmented Reality which in turn inspired the idea for my final year project and for never failing to help me with the development process of my project.

Last but certainly not least, I extend my deepest gratitude to my parents, Pn Norani binti Hussin and En Amir Firdaus bin Abdullah, and my sisters for their undying support and encouragement. Thank you as well to all my friends who have directly or indirectly contributed to the completion of my project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 PROJECT WORK BACKGROUND

Universiti Teknologi PETRONAS (UTP) was established on January 10, 1997 when PETRONAS was invited by the Malaysian government to set up a university. The campus is built on a 400-hectare site strategically located at Bandar Seri Iskandar, Perak Darul Ridzuan, Malaysia. The University is a wholly-owned subsidiary of PETRONAS, the national oil company of Malaysia. UTP offers a wide range of engineering and technology programmes at undergraduate and postgraduate levels complemented with a strong focus on Research and Development. The programmes are designed with high industry relevance to provide a dynamic learning environment. UTP has not only put Malaysia on the world map as one of the top leading institutions of higher learning in the world; it has also contributed significantly to the socio growth and industrial development for Malaysia. With the said reputation, it is of no surprise that the scope of work for UTP is undeniably vast and when scope is put into consideration, so is the size of the university's campus. Laid out on a 400-hectare site, the sheer size of the campus is impressive.

The scope of this project is narrowed down as such that the developed AR application will only be made possible for usage by using a smartphone and it involves only on the main area of interests of UTP. A few examples of the main areas that would likely be useful for identification in the AR application includes the Chancellor Hall, the Information Resource Centre, the An-Nur Mosque, the Pavilion, the sport complex as well as the villages and the cafes ; to name a few. As mentioned earlier, the size of the university itself is already myriad. Hence this indicates that the possible places of interests in the university are most definitely huge.

Basically an Augmented Reality system generates a composite view for the user. It is "a combination of the real scene viewed by the user and a virtual scene generated by the computer that augments the scene with additional information" (Berlin, Leslie. 2009). The application domains reveal that the augmentation can take on a number of different forms. In all those applications the AR the user enhances that person's performance in and perception of the world. The ultimate goal is to create a system such that the user cannot tell the difference between the real world and the virtual augmentation of it. To the user of this ultimate system it would appear that he is looking at a single real scene.

Having said the above, this project aims to create an Augmented Rreality application that functions as a virtual map of UTP and can be accessed from the smartphones owned by visitors. A further add on to the passive / static indicators of places of interests of the university is the ability to generate viable and useful information to assist the experience of visiting the campus. Because, there are a number of places of interest within UTP's campus, for that reason it is indisputably pivotal to have a reliable and intelligent AR application whereby not only can it be utilized as a virtual map of UTP, the application should also be able to automatically generate the distance of the indicated place of interest from the user holding the smartphone. The application will also contain a radar like feature which would display the places of interest surrounding the user in a 360 degree indicator on the radar to ease navigation.

The above features of the AR application can be determined via designing and personalizing AR technology using Layar AR Browser. Layar provides the mobile application and infrastructure for creating layers which contain information and pictures. Layar can be downloaded for smartphones for free as it is an open platform. Utilizing the technology found in Layar AR Browser such as the method of deploying Points of interest (POI) on to locations, developers can create AR applications that are boundless only to their creativity and will further promote innovative new features for the AR application.

1.2 PROBLEM STATEMENT

Currently in UTP, the standard practice being used is that, visits and tours are carried out by the HR staff of UTP. Whenever there is a need for a tour guide, the availability of the staff is depended on and in most cases they are only reserved for official visits from notable individuals having an official visit. For other visitors such as parents or outside guests such as students and teachers from schools, they are left on their own with not much information made available for them to explore and discover of the university and its campus. This imposes several difficulties especially to the guests and causes degrade in the overall experience:

- **Lack of availability of UTP's staff to carry out tours of the campus for guests.**

  As mentioned earlier, all details regarding the university's campus are being dispersed manually by UTP's personals, usually done by the Human Resource department. Usually these staffs are only available to carry out official visits for important visitors and not for others due to the fact that it's not an official responsibility to be carried out. With the lack of a tour guide explaining the various parts of our campus, it decreases the experience of visits made by people as they are left to wonder on their own without knowing much information of where and what is available in this university.

- **Difficult and tedious for new students and guests to attain information regarding the orientation of the campus.**

  The only available source of navigation for guests within the campus is a signboards and maps which are placed around the campus. These maps which can be found at Chancellor Complex are faded out due and left unmaintained which prove to be impractical to be a source information for guests to use.

1.3 PROJECT OBJECTIVE

To develop a system that meets the following purposes:

- To allow users to attain information on the places of interest of the university's campus for increase in their visit experience and to have it made available and easy for everyone's usage.

- To integrate features that will provide a distance measure in meters of the location of the user and the place of interest as well as radar feature that will display all places interest location surrounding the user.

1.4 PROJECT SCOPE

The scope of this project will be mainly on the pedagogy of the Augmented Reality technology utilizing Layar mobile platform application. This will also cover various areas particularly on:

- Creating a layer containing the information to be displayed in the application
- Creating the Points of Interest (POI) for hotspots in the university
- Creating the layer service for delivering POI content to the Layar Application

This application will also be made as so to meet all the requirements that have been pre – determined earlier and to have made available for public usage once implemented.

CHAPTER 2

LITERATURE REVIEW

One of the objectives of this project is to develop an intelligent AR application which is able to display places of interest and provide information regarding it as well as feature tools such as distance measurements and radar indicator. These added features of the AR application can only be done if it applies rules and concepts of an advanced AR technology found in mobile platforms.

2.1 WHAT IS AUGMENTED REALITY AND HOW IT WORKS.

Before further work can be done, it is best that the actual meaning and concepts of Augmented Rreal in the 70s and by the 90s were already being used by major companies. Now portable computing is finally powerful enough to deliver AR to anyone who has a smart phone or latest generation PC or console.

Mixed Reality combines the content from the real world with virtual imaginary. Augmented Reality is a subset of this where virtual content is overlaid into real objects of the world. Extending the concept of AR, it includes virtual graphics and audio. In 1994, Paul Milgram (Milgram and Kishino 1994; Milgram, Takemura et al. 1994) characterized Mixed Reality interfaces on his Reality-Virtuality Continuum (Figure 1).

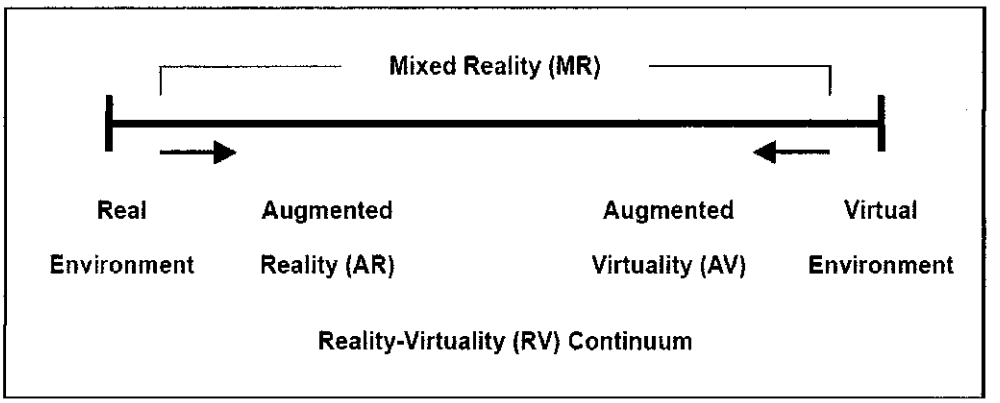


Figure 1: Milgram Reality-Virtuality (RV) Continuum

The real world and a totally virtual environment are at the two ends of this continuum with the middle region called Mixed Reality. Augmented reality lies near the real world end of the line with the predominate perception being the real world augmented by computer generated data. Augmented virtuality is a term created by Milgram to identify systems which are mostly artificial with some real world imagery added such as texture mapping video onto virtual objects. This is a distinction that will fade as the technology improves and the virtual elements in the scene become less distinguishable from the real ones.

A range of technologies can be used for Augmented Rbject or space the user is observing. In the case of a technical course on PC maintenance, for example, Augmented Reality might overlay a schematic diagram onto the inside of a computer, allowing students to identify the various components and access technical specifications about them. Smartphones or other portable devices can use GPS data to provide users with context which includes visual, audio, or text-based data about real objects or places. Augmented reality is not merely a companion text or multimedia file but a technology designed to "see" a real object or place and provide the user with appropriate information at the right time. Augmented reality is designed to blur the line between the reality the user is experiencing and the content provided by technology.

2.2 HOW AUGMENTED REALITY HAS EVOLVED INTO EVERYDAY USAGE.

Augmented reality blurs the line between what's real and what's computer-generated by enhancing what we see, hear, feel and smell. On the "spectrum between virtual reality, which creates immersive, computer-generated environments, and the real world, AugmentedRreality is closer to the natural world" (Ensha, Azadeh, 2009). Augmented reality adds graphics, sounds, haptic feedback and smell to the real world as it exists. Both video games and cell phones today are driving the development of AR from tourists, to soldiers, to someone looking for the closest bus stop can now benefit from the ability to place computer-generated graphics in their field of vision.

As was previously stated, an Augmented Reality system supplements the real world with virtual objects. It means that virtual (computer-generated) content is added to the real world (Ganapati, Priya, 2009). An AR system has the following three main characteristics:

- Combines real and virtual objects in a real environment.
- Runs interactively, and in real time.
- Registers virtual objects onto the real world.

Augmented reality is changing the way we view the world or at least the way its users see the world. Imagine walking or driving down a street and with augmented-reality displays on, which will eventually look much like a normal pair of glasses, informative graphics will appear in your field of view and audio will coincide with whatever you see. These enhancements will be refreshed continually to reflect the movements of your head.

The ultimate goal is to create "a system such that the user cannot tell the difference between the real world and the virtual augmentation of it" (Valino, James R, 1998). Similar devices and applications already exist, particularly on smartphones like the Android phones and iPhones.

What's changed in the past year is that AR has come within reach of all sorts of developers and the technology powerful enough to make use of it is owned by millions of people, often in the palms of their hands. The arrival of powerful smartphones and computers with built-in video capabilities means that we don't have to wait for the AR effects as we normally do with TV. They can simply be overlaid onto real life. This is where Apple's iPhone, and phones using Google's Android operating system such as HTC Desire, both of which are capable of overlaying information on top of a picture or video.

In August 2009, iPhone users were surprised to discover an augmented-reality feature hidden within the Yelp application. Yelp is a famous application known for its user reviews of restaurants and other businesses, but its hidden augmented-reality component, called Monocle, takes things one step further. Just start up the Yelp app, shake the iPhone 3GS three times and Monocle activates. Using the phone's GPS and compass, Monocle will combine the phone's camera view with tiny tags indicating the names, distances and user ratings of nearby bars, restaurants and more information about local restaurants, including ratings and reviews, on the cell phone screen (Parr, Ben, 2009).

2.3 AUGMENTED REALITY BUSINESS MODELS.

Based on a study on augmented reality made by Gary (Hayes, Gary, 2009) there are five technical types of augmented reality, which are basic types of AR from a slightly technical perspective but which can be applied to commercial and marketing applications:

- **Surface**

  The most understandable form of 'reality that is augmented' would be screens, floors, walls etc. that respond to the touch of people in them providing them with virtual real time information or collaboration.

- **Pattern**

  The AR system performs simple pattern recognition on a shape, marker (usually on a framed card in the real world scene) or face and replaces it with a static or moving element e.g.: a 3D model, info, audio, video stream or loop etc.: You view the 'items' in the scene with you.

- **Outline**

  This is where your hand, eye or body outline is picked up and seamlessly 'merged' with the virtual elements. A simple example is where you can pick up a 3D object that doesn't exist because the system is tracking your hand outline.

- **Location**

  Based on detailed GPS or triangulation location & position/view of the camera/device the AR system can overlay information precisely over buildings or people as you move through real space.

- **Hologram**

  Using 'smoke & spinning mirrors' literally in some cases, virtual or real items are 'projected' into the physical space you are in and can be interactive with based on cameras tracking real world impulses e.g.: hand gestures or audio signals.

A Business application/model for augmented reality (see Appendix 1) made by Gary which lists down top 16 marketing or commercial 'intention' vs. a traditional pure business model and broad approach, shows which segment would be successful in applying AR technology. "It is a starting document to aid classification of an emerging commercial sector that is hoped to be useful" (Hayes, Gary, 2009). The fifth top segment in the model is:

- **Location Layers**

  Blended guides to new places, tourism, enhanced travelling or themed space. For travellers just arrived at your city, theme park or other experience you can provide them with pay for tools that will help them take the most 'mutually beneficial' route after they arrive. Free data from wikipedia, or more commercial entities add depth.

It is highly applicable and effective to create a virtual assisted information centre which comes under the location layers of the business model to create augmented reality application.

2.4 FUNDAMENTALS OF LAYAR AND HOW IT WORKS.

Layar is a mobile application firstly designed and developed in Netherlands May 2009, for discovering information about the cities in the country and after several releases and positive feedbacks, in August 2009 Layar 2.0 was released worldwide. Using Augmented Reality technology, Layar display its digital information called "layers" into the smartphone's field of vision. The Layar Reality Browser shows what is around you by displaying real time digital information on top of the real world as seen through the camera of the mobile phone.

Layar is a free application which offers an open platform for developers to publish, discover as well as search for AR layers published by other developers. The basic structure of how Layar functions is that it provides the mobile application and infrastructure. The ready-made AR technology with its various features are made available for developers to use and manipulate based on their creativity and interests. Layar is also available and supported across four different mobile platforms which are Android, iPhone, Symbian and Bada. Since it is available across all these mobile platforms, it is highly advantageous to use Layar.

2.4.1 Terminologies of Layar

- **Layar**

  Layar is the AR browser which offers open platform for developers to use and provides mobile application and infrastructure for AR technology. It is the application for which users can select from a collection of numerous available Layers to be used.

- **Layer**

  Layer is a content layer in the Layar application which is built by developers and contains all the information such as pictures, animations or icons for it to be displayed once selected by the user. It is customized using the developer publishing site on Layar.

- **Point of Interest (POI)**

  Point of Interest is the element of data which when triggered once it is in the view of the device, will prompt element of data to be displayed in the Augmented Reality view such icons, pictures, animations as well as 3D objects. Points of interest are chosen by the developer using coordinates selecting by him / her of the location using Google Maps.

2.4.2 Components of a layer

The layer definition is provisioned on the Layar Publishing site via a web interface. Here all the fixed parameters associated with a layer can be defined. These parameters are stored in the Layar Database and can be accessed by the Layar app.

Once a layer has been defined, it can be listed in the Layer Gallery in the client. When the user selects this layer in the client, the POIs for the layer will be fetched for the current location, using filter parameters that can be set on the device.

The POIs are fetched directly from the Layer Service Provider by the Layar Server using the Layer Developer Application API. The calls are real-time, that is each time the client needs to fetch a new list of POIs, a call will be made by the Layer Server using the API. The developer should therefore expose a REST webservice to the Layar Server according to this API specification.

For each POI, there may be more detailed content to show, or interaction possible with the end-user. This can be achieved by hosting web pages (made for mobile) that will be shown within the Layar client app as web views when a user selects the POI. These pages are accessed directly by the Layar client over the Internet.

2.4.3 Requirements for Layar Work?

Layar works by using a combination of the mobile phones camera, compass and GPS data to identify the user's location and field of view, retrieve data based on those geographical coordinates, and overlay that data over the camera view. For Augmented Reality to work on a smartphone, some requirements of the phone must first be available which includes:

- **Global Position System (GPS)**
  GPS is used to locate the users location to provide accurate information on the possible Points of Interests found near the user.

- **Accelerometer**
  The phone accelerometer is used to provide an orientation of the phone in the world once the AR view is activated on the phone.

- **Compass**
  The compass is used for showing which direction the user is facing and looking at.

- **Camera**
  To show the user what he/ she is looking at in the real world, the camera is what makes it possible for such AR application to exists and function on phones.

- **Internet Connectivity**
  Internet connection is required to connect the phone to the digitial world by accessing the layers which then includes the databases that contains the content of that layer to be displayed for the user.

2.5 RELEVANCE ON THE USAGE OF AN AUGMENTED REALITY APPLICATION FOR THIS PROJECT.

As previously mentioned, among the problems faced by visitors and even newly admitted students of UTP, is that there are no means to introduce or notify them on the orientation of the university's campus and its main facilities and buildings. UTP's personnel will have to constantly provide a tour for visitors which in this case is only restricted to entertain important guests and not for the general mass of other guests. Indeed, the lacking of an information and navigation tool has caused decrease of satisfaction in the campus visits. Because an Augmented Reality Application is capable to provide a means for these targeted group of audience to gain their information and navigation of the university campus, this innovative technology to build the tool for assisting the mentioned group of users.

In determining the functions of the AR application, several features can be written and manipulated to make certain that the tool is working as per intended. The distance measurer and radar feature of the AR application can be applied and further enhance to ensure that the system acts both as an indicator and navigator to the user of the various places of interest found within the campus which would help orient their visit a more better experience.

CHAPTER 3
METHODOLOGY

3.1 RAPID APPLICATION DEVELOPMENT (RAD)

A methodology is a formalized approach to implementing the System Development Life Cycle (SDLC), (Dennis, Wixom, Tegarden, 2005). There are various different methodologies used to build a system depending on the different factors of the system itself. However for this project, it is decided that the most suitable methodology to be implemented is the Rapid Application Development (RAD) - based methodology. Within the RAD methodology as well there are four different approaches; Parallel Development – based, Phased Development – based, Throwaway Prototyping – based and finally the Prototyping – based. The latter, which is the Prototyping – based is by far the most suitable approach onto building the Augmented Reality UTP Application.

The main factor to justify why this method was chosen is the fact that it is due to it being designed to increase the speed of development. Since the timeline for building this project is only within 5-7 months, this method encourages flexibility and allows changes to be made while still in the development process. Development of the project will be based on several cycles whereby the Analysis, Design and Implementation phase will be revisited as required to suit the changes and requirements of the project. The cycle repeated continually based on the end users' comments until the prototype successfully meets the requirements and satisfies all conditions. The last prototype will then be called the system. This methodology is chosen as a means to adapt to changes found during the development of the project, such as a new requirement or a time constraint.

The phases of the project will be divided into:

1. Planning & Requirements Gathering
2. System Analysis
3. System Design
4. System Implementation
5. System Testing (Prototype Evaluation)
6. System Deployment

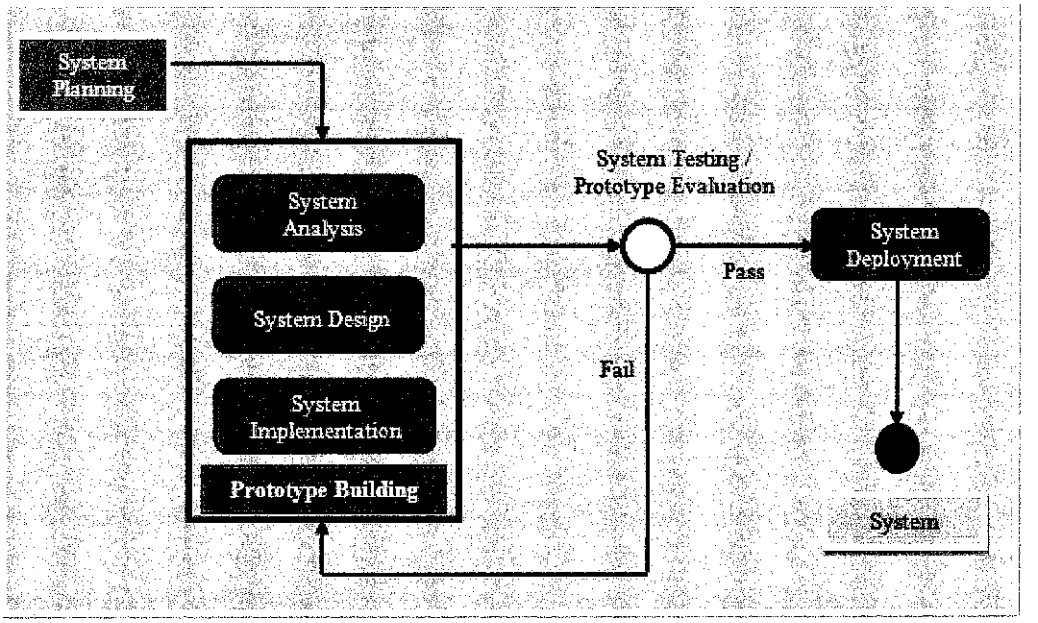


Figure 2: Project Methodology

3.2 PROJECT PHASE

3.2.1 System Planning

For the first stage of the project, the initial requirements are based on information from the requirement gathering methods. Among the initial techniques involve are constant interviews or meetings with the outside guests of UTP and colleagues. Apart from that, it is also imperative to perform a thorough observation of their current situation in place or what their normal experience is. Information is also gathered from journals, books, newspapers, and also case studies that have any relevance to the topics. In the planning phase as well, a Work-Breakdown Structure (WBS) is done to depict all the tasks that as well as its status. Should there be any changes in requirement, or if there is a necessity to make any refinements on the system, the requirement gathering phase is revisit to suit any change. Finally, two feasibility analysis were done; Technical Feasibility Analysis as well as the Organizational Feasibility Analysis.

3.2.2 System Analysis

Once the basic and significant idea of the system have been figured out, the project proceeds to the system analysis phase whereby it started off with the requirements gathering process which includes two main methods of information gathering; interview and documents gathering. The interview process is solely done via phone call with a former colleague from PETRONAS Carigali, En. Shankar Jayaram. Questions asked during the interview mainly revolve around the current feeling of visiting UTP, the awareness of certain places on campus as well as any notable places of interests, the awareness of Layar Application available for mobile phones and the expected functionalities of the AR application.

Based on the requirements gathered, an analysis is done to see what functions and modules are to be developed. Aside from that, the outline system area model and scope of the proposed system were also developed in this task.

Finally, this phase includes a more thorough study on the relationship between the different phases of creating a layer which includes creating the layer contents, setting up the database and web service, setting up the POIs, and testing the layer. Also, in order to create use and create layers using Layar, it is required to sign up for a developer account on Layar Developer Website.

3.2.3 System Design

After thorough analysis were done in the previous phase, the project can then move on to the design phase. The purpose of the analysis phase is to figure out *what* the project needs whereas the purpose of the design phase is to decide *how* to build it. Within this phase as well, all the requirements and information gathered during the planning phase, were organized and presented in the form of an activity diagram and a use case diagram. In addition, this phase also includes taking into account the user interactions with the system and how this will affect the program flow. All the contents of the layer that was needed were generated during this phase. This phase is the starting point of the entire segregation between idea and action of the AR application. Based on all the data, the preliminary system interface as well as a prototype of the system was designed accordingly.

3.2.4 System Implementation

In the system implementation phase, a prototype designed in the design phase is built based on the system analysis and design. This phase involves the building and designing of the UTP layer using the available provisioned layer definitions on the Layar publishing site via a web interface and the development of the database to store the information of the POIs using MySQL Database and the web service which will provide the right response to the Layar platform using PHP. Apart from that, the coding of rules to set up the communication and responses between the layer and the Layar platform will also be done in this phase. Tools such as the MySQL Database software will be utilized in this phase. The implementation phase depicts the actual building phase of the system. Once compiled, the built prototype is the main deliverable.

3.2.5 System Testing

As soon as the system is built successfully, the system testing phase was used to test the implementations. The system testing phase is also known as the prototype evaluation phase whereby during this phase, the prototype was thoroughly checked and evaluated if there are any requirements not met. Because the RAD methodology is being used for this project, if an error or lacking of function is found during this phase, the prototype building phases (system analysis, system design and system implementation) will be continuously repeated only until it has satisfy all requirements. The activities done in this phase also include the Usability Testing from which the principal means involved finding out whether the system meets its intended purpose. This testing phase was also used to identify and fix any bugs that occur within the built system.

3.2.6 System Deployment

The final phase of the system is the system deployment where the system is ready for use. This activity measures the effectiveness of the system when used in the actual learning situation.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 PLANNING PHASE DELIVERABLES

The planning phase is the fundamental process of understanding *why* an information system should be built and determining how to go about building it (Dennis, Wixom, Tegarden, 2005). Before the system can be built, it is important to understand first the purpose or objectives of it. Therefore an initial contact was made with a former supervisor and mentor from PETRONAS Carigali Sdn. Bhd, En. Ghafar Dawam. As a father of a student that has just been enrolled into UTP, he is a good source of information to investigate how much a new visitor knows about the university and the campus itself. From past conversations, it was stated that UTP is a huge place which would easily get people lost and at first was disorientating and a general lack of awareness of the many interesting places that are found on campus. Having said that, the objective of the project is to design an AR application which can store and display places of interests within the university's campus for visitors' awareness as well as become a navigation tool for their orient of the campus.

Prior to building a system, a feasibility analysis has to be made in order to examine key aspects of the proposed project, as shown in Figure 3. The technical feasibility analysis was done to identify the extent to which the system can be successfully designed, developed, and installed. This feasibility analysis was done to answer the question, *"Can we build it"*. Apart from that, the organizational feasibility analysis was also done to estimate how well the system ultimately will be accepted by its users and how the system will be beneficial.

Once the feasibility analysis has been completed, a work plan such as depicted in appendix b was made to record and keeps track of all the tasks that need to be accomplished over the course of the entire project. The work plan lists each task along with important information regarding it. The work plan also quotes the status of the task whether it is open, in progress or completed. The approach for the work plan is to organize a work-breakdown structure (WBS) by the System Development Life Cycle phase.

Tools that will be used in order to create an AR Layer using Layar Platform:

- A web server with PHP (5.2 or above) which supports JSON
- A MySQL database (preferably with phpMyAdmin)
- BuildAR web tool for storing information and layar settings

4.1.1 Feasibility Analysis

The highlights of the feasibility analysis are:

Technical Feasibility Analysis

The UTP AR layer application is feasible technically, although there are some risks:

Risk regarding familiarity with technology is high
The back end database will be developed using MySQL which will then be connected using PHP to the front end user interface designed using the publishing site. It is noted that end users may not have very high familiarity of the tools; however, the system only interacts with users on the interface level and does not require back-end coding activities.

Risk regarding familiarity with application is medium
Since Layar is still very new and is restricted to smartphone users, users may require some getting used-to to the system built. However, certain precautionary measures will be taken whereby guides and introduction of the application will be seriously considered to be implemented.

Risk related to compatibility is low
The layers will be fully developed using MySQL database and PHP. But Layar, the mobile application which houses them is only compatible with four mobile platforms, Android, IPhone, Symbian and Bada. With both the increasing number of smartphones available in the market and the increasing number of users, compatibility isn't an issue.

The project size is considered low risk
This application is built and designed to be used by the general public and is open for them to explore its functionalities. Moreover, the scope of project is also small and the functionalities are not too many.

Organizational Feasibility Analysis

From an organizational perspective, this project has low risk. The objective of the system which is to ease the burden and increase satisfaction from visitors and guests is aligned well with university's goals of encouraging innovative and creative solutions. The move to employ a AR application from current practice (brochures or maps posted around campus) helps achieve the target. Main users of this system are the guests and visitors as well as the new batch of students enrolled in the university.

Figure 3: Feasibility Analysis Summary

4.2 ANALYSIS PHASE DELIVERABLES

The analysis phase involves taking into consideration all the objectives outlined earlier in the planning phase and gather all the information to further proceed with the project. This phase is also where all the requirements are gathered and defined. Most of the requirements are defined from discussions with colleagues and analyzing other similar AR application. Apart from analyzing other AR application and their features, another method used to define requirements is by interview. In many ways, the requirements determination step is the single most critical step to the entire System Development Life Cycle (Dennis, Wixom, Tegarden, 2005).

4.2.1 Discussions with Colleagues

A number of discussions were carried out to determine the required places of interest within the university that will be defined as a Point of Interest (POI) which is a vital part in creating an AR layer. It is very important to determine which places are considered interesting from both the perspective of students of this university as well as the perceptive of outsiders. These POIs are places which are both a source of interest as well as proven to be beneficial knowing they're location.

Among some places that are agreed upon and will be certainly be featured in the layer content during the design phase are:

- Chancellor Hall
- Information Resource Centre
- Pocket D and Pocket C
- Sport Complex
- Registry Office
- Student Villages
- The numerous Cafes

4.2.2 Interview Conducted

Interview is the most common and direct method to gather information. Because, the interviewee was chosen as he has a daughter who had just enrolled in UTP and since the interviewee is located in Kuala Lumpur and a time constraint prevents us from meeting, the interview process can only be done vie phone calls. Below is the Interview Report produced consequent to the interview process itself:

Interview Notes Approved By: En. Ghafar bin Dawam

Person Interviewed: En. Ghafar bin Dawam, Technology & Information Management, PETRONAS Carigali Sdn. Bhd. (PCSB)

Interviewer: Kamarun Faliq bin Amir Firdaus

Purpose of Interview: To obtain better understanding on the as-is (current) situation, the awareness of the technology and the requirements for new system to be developed.

Summary of Interviews:

- Two biggest problems with the current situation are:
  - Lack of any introduction or tour programs for visitors to know better about the facilities and infrastructure that is available in UTP.
  - Lack of materials and information available to the public for learning more regarding UTP's campus.
  - New students find it disorientating and troublesome to make their way around the campus and find key places such as the clinic.

- The main features of the to-be application should include the following:
  - Places of interest indicators: To point out key and important facilities such as the clinic and places of high interests such as the IRC.
  - Radar and distance indicators: To monitor location of the user within close by POIs, this can then indicate the distance from it.

Open Items:

- Get user knowledge on the existence of personal AR applications for smartphones and applications that use its technology such as Layar.

Figure 4: Interview Notes

4.3 DESIGN PHASE DELIVERABLES

4.3.1 System Architecture Diagram

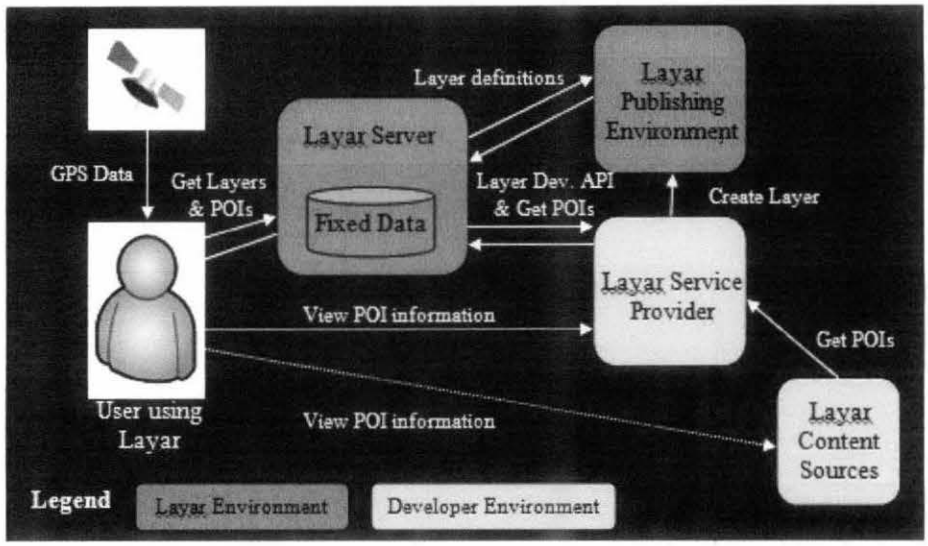


Figure 5: System Architecture Diagram

Figure 5 depicts the system architecture for the UTP layer Augmented Reality Application. As determined during the planning phase, there are three main components which interact with one another to ensure the application functions which are the users, the developer and the Layar Server. The user will be accessing the contents of the layer via the Layar Server as well as the Content Source. The developer on the other hand is responsible for creating the layer definition, the Content Sources (Database) and Service Provider (Web Service). The front-end system interface and application will be based on Layar architecture and the layer and its content will be based on the JSON and PHP scripts to communicate and deliver the POI content to the Layar Application from the databse. All this data are to be stored in the database created by the developer. The database is powered by MySQL.

4.3.2 Business Process Modeling

Business Process Modeling portrays all the activities done with regards to the system that when combined together support a business process.

The first set of activities in the far left of the diagram shows Firstly, The developer is responsible for creating the layer and publishing it's definitions in the Layar Publishing Environment. The developer then creates the Service Provider (Web Service) to interact with the Layar Server for passing the API and POIs, scripts and auto-triggered actions of the layer. The developer also creates a Content Source (Database) which stores the information content of the POIs. The database will be access by the Service Provider to attain information and content of the layer once the layer is chosen to be used.

The second set of activities involves the initial activity of the user initial contact with the system. The user will be accessing the UTP AR layer by using the Layar application that has already been installed on his or her smartphone. During this stage, the actions that are involved include getting the location of the user in the world via GPS and using internet connectivity the user will acquire the layers library. The user will then choose the UTP layer from the library displayed within the Layar Application and once the user accepts to use the layer, the user prompt the attainment of the content for the layer and POIs from the Service Provider.

The third set of activities involves the Layar Environment and depicts the flow of activities for the Layar Server and the Layar publishing environment. Here, the Layar Server is only accessible by using the Layar mobile application. The server will contain fixed data such as the application codings and scripts for the AR technology to function. The Layar Server also serves as a digital browser for the user to browse through existing and published layers created by other developers around the world. Once the user selects the UTP layer to be viewed, the layer definition is then acquired from the Layar publishing environment and then displayed back to the user smartphone. Once the user accepts to use the UTP layer, the Layar server communicates with the Service Provider for acquiring the layer developers API, POIs and scripts/ triggers and displays the contents onto the users view.

4.3.3 Use Case Modeling

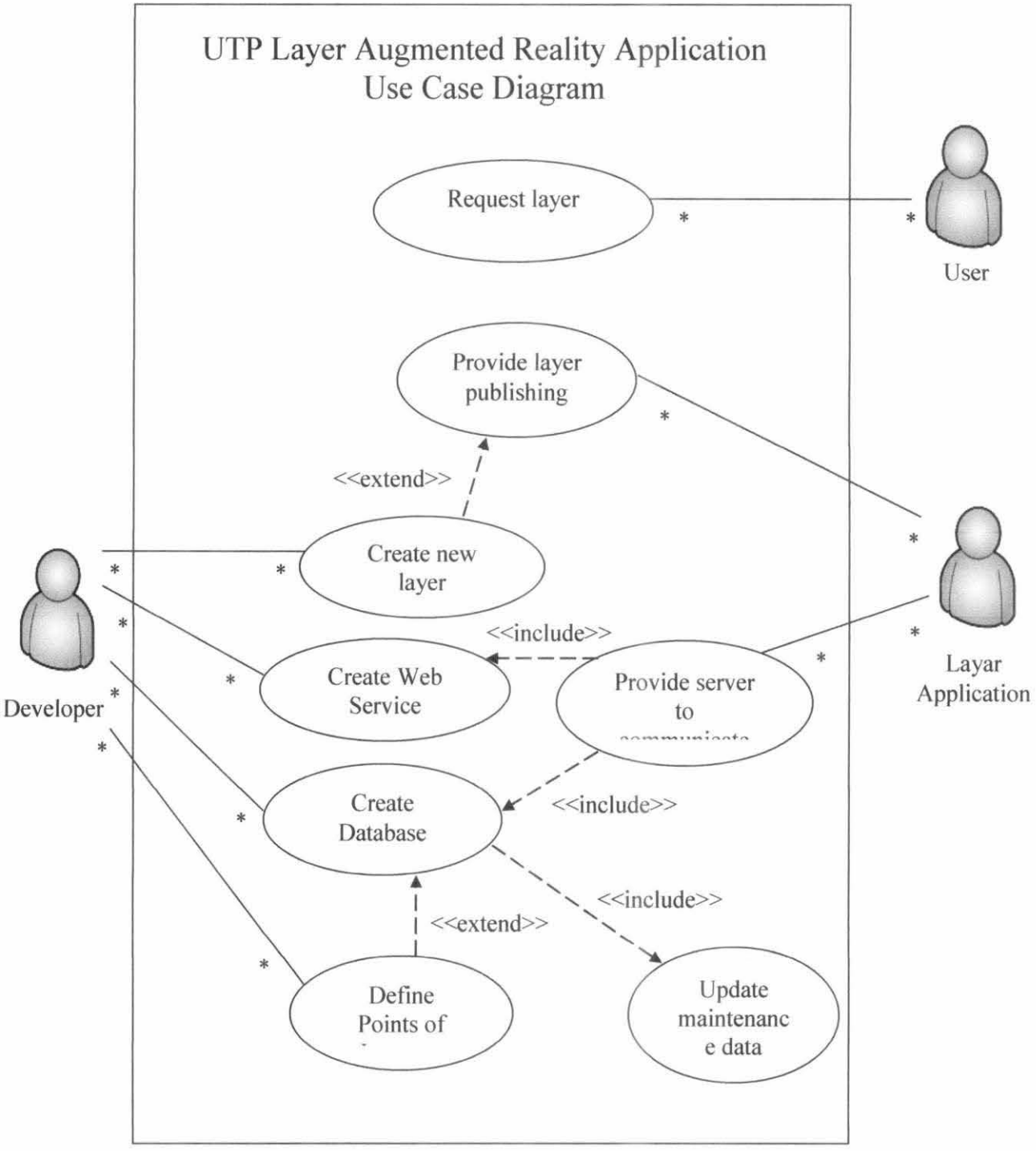


Figure 6: Use Case Diagram

In the design phase, use case modeling is done by constructing a use case diagram. This use case diagram is used to better understand the functionalities of the system at a very high level. Figure 6 depicts the flow of activities for the UTP layer AR application. Typically a use case diagram is drawn to provide a simple, straightforward way of communicating what exactly the system does. It also defines the boundaries for the system.

The three actors show the two main groups of people involved in the system which is the user and the developer and the third actor is a system which is the Layar application. The use cases depicted by an oval are major processes that the system will perform that benefit the actors in some way. Referring back to Figure 6, there are 8 use cases including extension and inclusive use cases. The first use case is the Request Layer from the user, where the user requests make request for a layer through Layar application.

The second actor is the developer in which 5 use cases come under. The first use case under the developer is the Create a new layer which is where the developer will create the layer definition. The second use case is the Create Web Service where using PHP, it will contain the scripts for the layer to respond appropriately once selected. The next use case is the Create database which using MySQL will store the contents of the POI's for it to be referred by the application later. The fourth use case is the Define Point of Interests which has an extension relationship with the Create database use case. Create database use case also shares an inclusive relationship with the Update maintenance date use case as later on as newer POIs are revealed, it is necessary to update the collection of POIs.

The third actor is the Layar application itself which has 2 use cases which are Provide Server to communicate and provide layer publishing. These 2 use cases are the middle factor that forms a bridge between the previous 2 actors and their actions and make the system function.

4.3.4 Preliminary Interface Sketch

I have designed a preliminary interface Sketch of how the UTP layer will look like once it is implemented. Figure 6 depicts how the UTP places of interest layer looks like once it is chosen in the Layar application. The Augmented Reality interface blurs the distinction between the real and the digital world to create a seamless image in the view of the user.

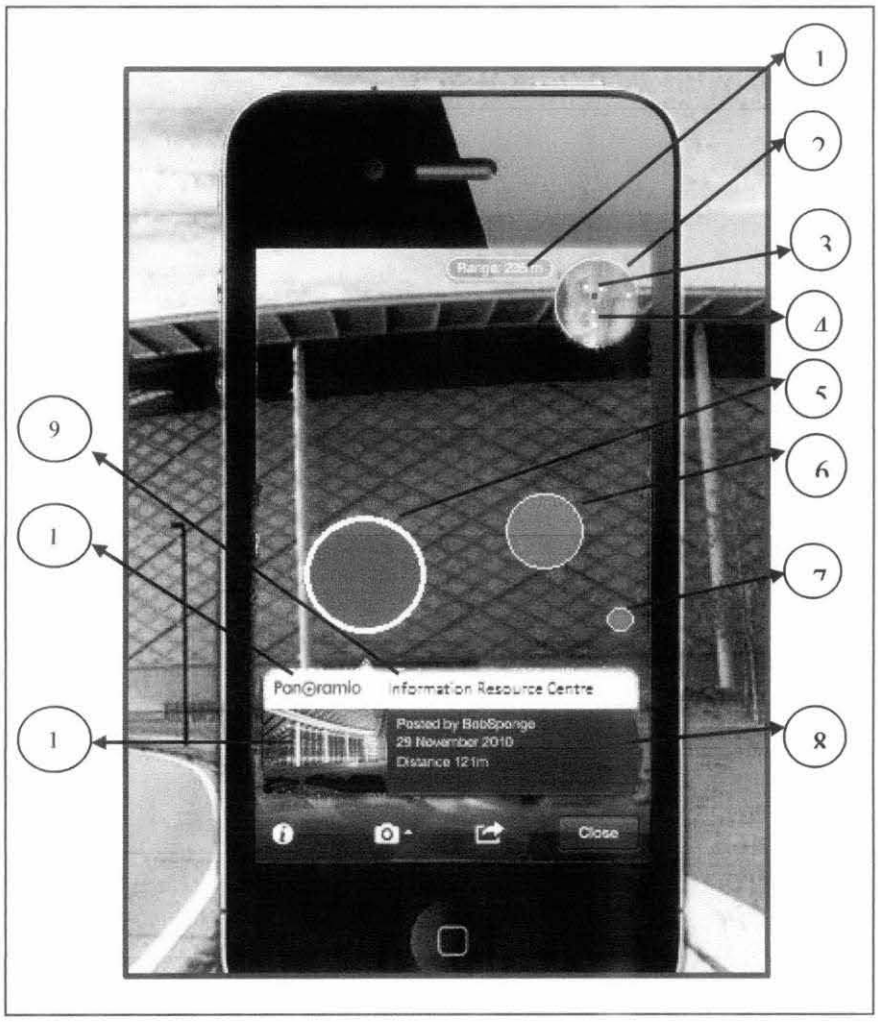


Figure 7: UTP Augmented Reality Layer

4.3.4.1 Definition Of The Attribute/ Features In The Layer

Based on the figure above of the UTP AR Layer, the different attributes that make the entire interface has been labeled from 1 to 11. An understanding of what those attributes and features are is important to create a better understanding of what the AR Layer is capable of doing in view of its main objectives.

Number	Attribute Name	Function
1	Distance Display	Display the distance from the user to the selected POI.
2	Radar Display	Displays all other POIs surrounding the user.
3	Spot colour (Selected POI)	Shows the selected POI on the radar.
4	Grey Spot (all other POIs)	Other available POIs nearby.
5	Focused Circle (Selected POI)	The selected POI aimed in the view of the user.
6	Middle Size Circle (Middle distance POI)	A medium distance POI that isn't selected.
7	Small Size Circle (Furthest distance POI)	A far away POI that isn't selected.
8	Information Banner	Contains information regarding the selected POI.
9	Name of the selected POI	Display name of the selected POI.
10	Banner Icon	Developer's icon on the banner.
11	Image of the Selected POI	Picture of the selected POI

Table 1: Layer Definition

4.4 IMPLEMENTATION PHASE PROGRESS

4.4.1 STEPS TO CREATE A LAYER

In principle to get an Augmented Reality layer to be functioning, a database with the points of interest (POI) is needed to be set up first. Then a PHP script on is required to put on the Layar server to communicate the layer with the actions of the user. Then it can proceed to the testing phase which can be done on the Layar Developer site. Basically, the whole layer creation and publication process can be described using the following 7 steps (refer to flowchart below):

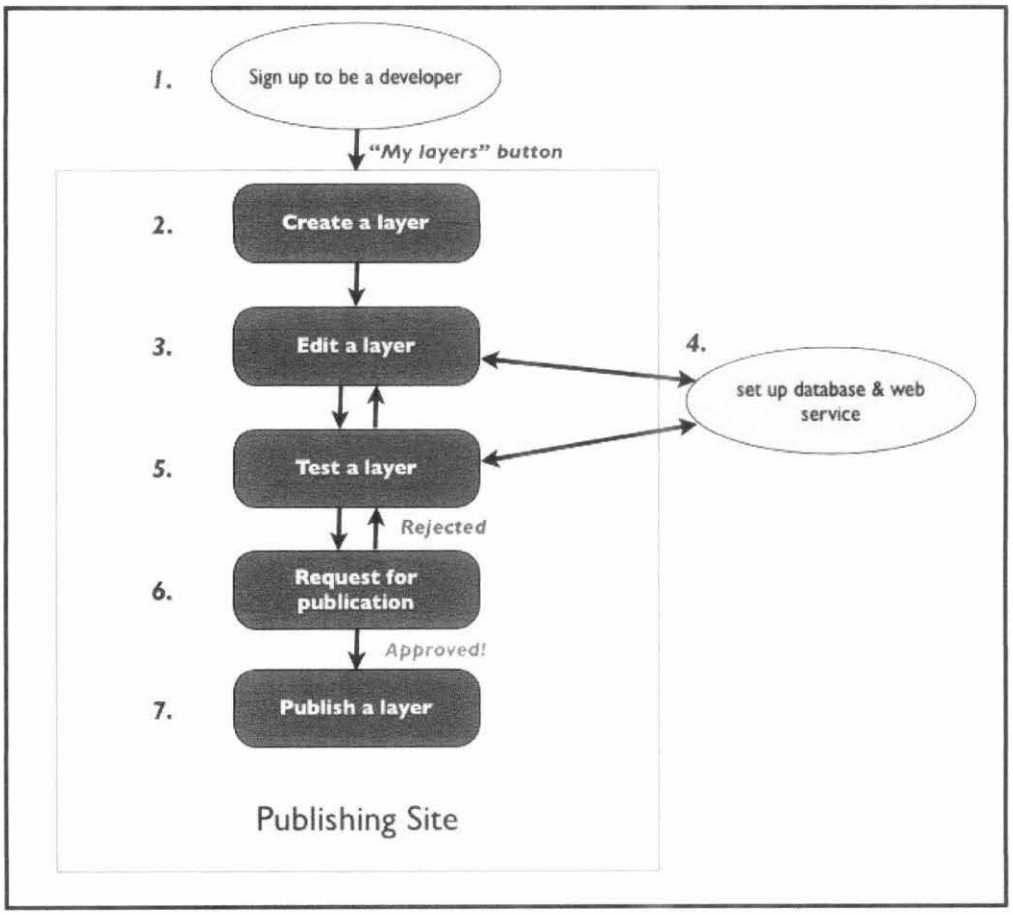


Figure 8: Steps to create and publish a layer

4.4.1.1 Step 1: Signing Up To Be a Developer

Before being able to use the developer publishing site, it is required to register a Layar account at Layar website (www.layar.com). The registration is free and simple as the main business idea of Layar is promote more and more developers to increase their layer catalogue.

Figure 9: Developer Site

Once logged in with the Layar developer account, it should be easy to see a "My layers" button next to the Layar account user name. Clicking on the "My layers" button, it will redirect us to the publishing site environment.

4.4.1.2 Step 2: Creating a Layer

To create and define a layer, it is required to utilize the webpage on the Layar Publishing Site. After clicking the "create a layer" button, it will transfer us to page with form like attributes to customize the layer. As was previously mentioned this stage of defining the layer is made easy by the already provisioned features and options made available on the publisher site.

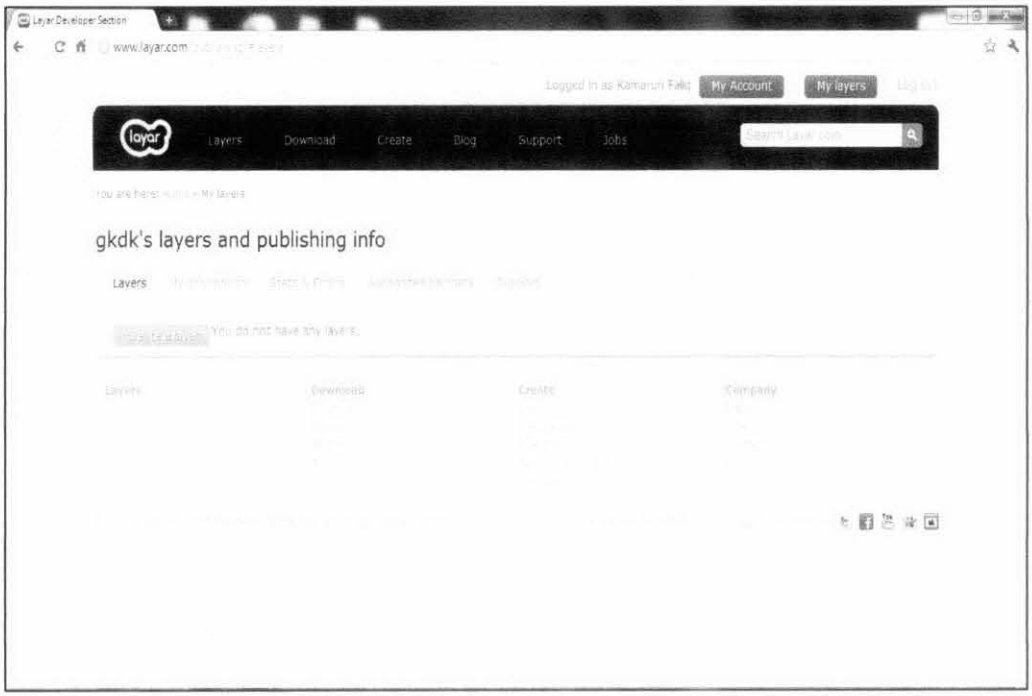


Figure 10: Layar Publishing Site

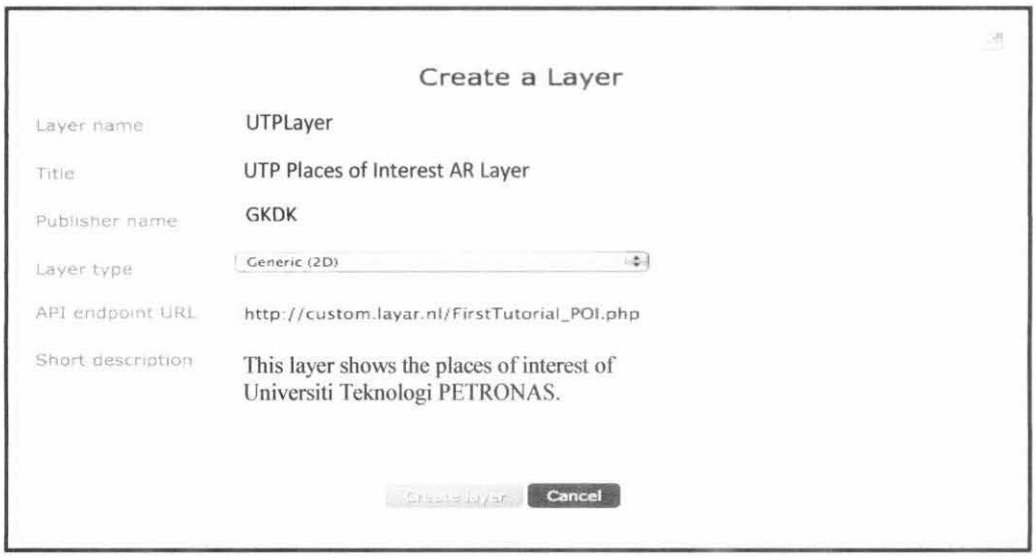


Figure 11: Defining a layer

Fields	**Description**	**Value**
Layer name	A unique name for your layer	UTPLayar
Title	The title that will be visible once it is published	UTP Places of Interest AR Layer
Publisher name	The name of the publisher	GKDK
Layer type	The type of the layer, can also be 3D and 2D objects in 3D space (will be introduced in later tutorials)	Generic (2D)
API endpoint URL	The URL for the web service that provides information about points of interest (POIs) in the database.	http://custom.layar.nl/first.php
Short Description	A short description of the layer and where the POIs are located	This layer shows the places of interest of Universiti Teknologi PETRONAS.

Table 2: Layer definitions Attributes

4.4.1.3 Step 3: Customizing the layer contents

Figure 12.1: Publisher Project Page

Customizing the layer is the next step that is taken in order to fully integrate how the layer will function. Further options of customizing the layer in detail can be found by clicking on the edit button on under the other actions category.

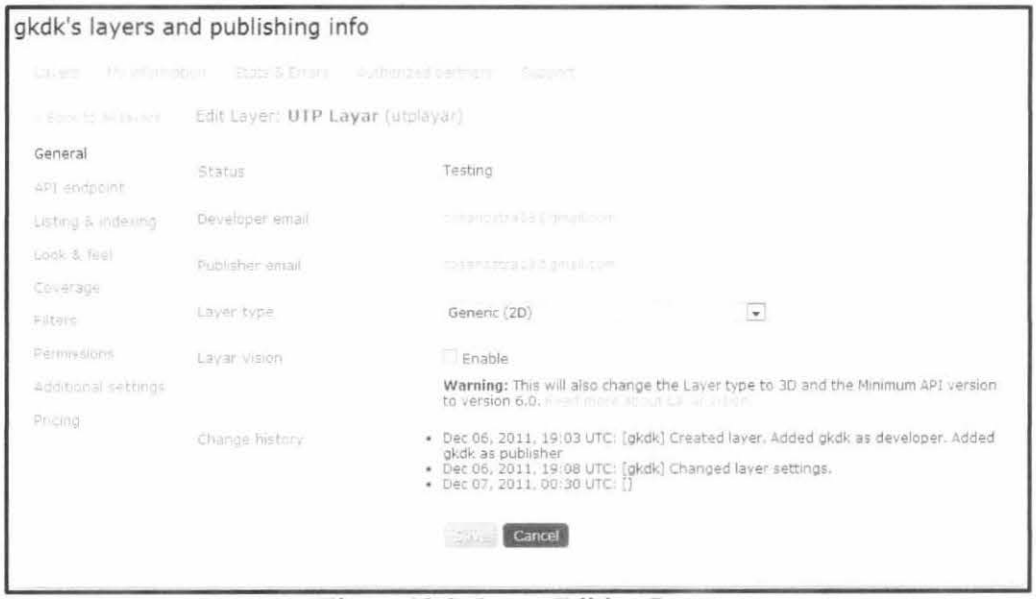


Figure 12.2: Layer Editing Page

The next step is to utilize the main functions found on this page which includes in total nine other further functions. These include:

Tab	Function
General	The basic information about this layer, such as layer status.
API endpoint	Mainly the POI URL for developer's web service that provides hotspots response.
Listing & Indexing	Mainly Layer listing on the phone and indexing in Layar Stream.
Look & feel	Manage the appearance of the Camera view on the phone.
Coverage	Define layer country region and set up bounding box for local layers.
Filters	Set up filter settings for a layer.
Permissions	Assign different publisher and add viewers to a layer.
Pricing	Set up the price tier for your premium layer.

Table 3: Layer Edit Tabs

Each tab will be further explained in detail below on what is needed to set.

1. General Tab

In this tab, basic information about this layer is provided, such as layer status and layer developer or publisher emails. Again, layer type can be changed on this tab for further improvement (refer to Figure 12.2).

2. API Endpoint Tab

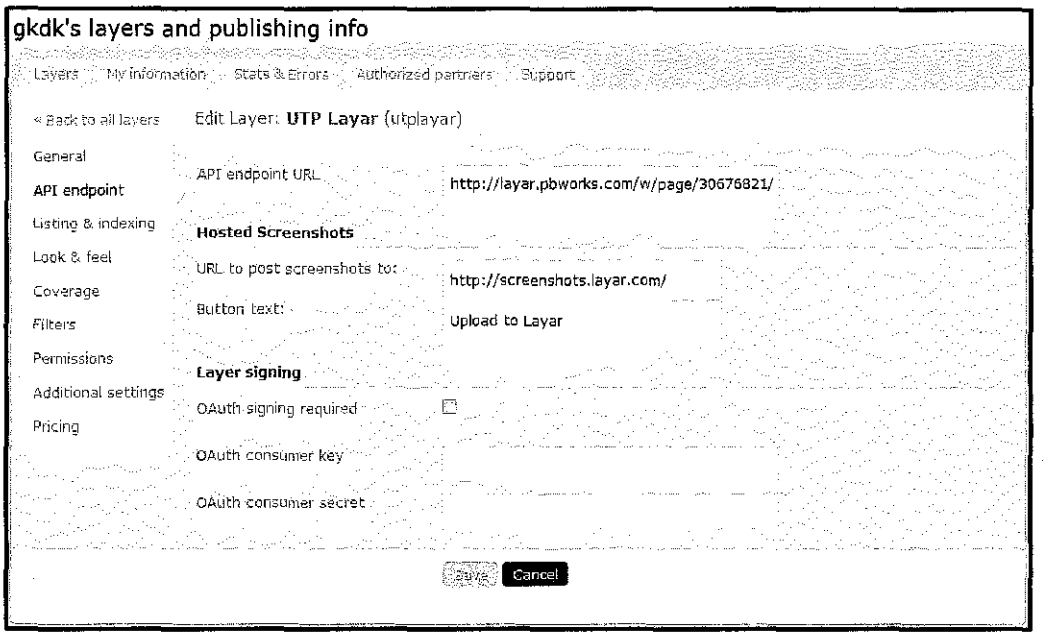


Figure 13: API endpoint tab

As we can see that API endpoint URL (which was defined when this layer was created) can be modified here. Since Layar client 5.0 release, it is possible for users to upload screenshots taken from the camera view directly to layer developer's web server. This can be defined under Hosted screenshots section. The detailed description of each field can be found below:

Fields	Required/Default	Definition
URL to post screenshots to	Optional (mandatory when this function is used)	The URL of the website where end users can post screenshots to. If this is filled out, and a user selects to share a screenshot with this option, an HTTP post will be sent to this URL containing parameters: - screenshot: the image - layer_name: the name of the layer - message: a message from the user - lat: latitude of the user - lon: longitude of the user - location_name: friendly location name
Button text	Optional (mandatory when this function is used)	The text showed when the user chooses to share the screenshot.

Table 4: API endpoint tab definition

3. Listing & Indexing Tab

The next tab now includes the listing, fields related to layer listing in the Layar application that will be displayed for users to see on the phone. This will contribute to user friendliness as it provides a clear and brief explanation of what this layer's purpose is.

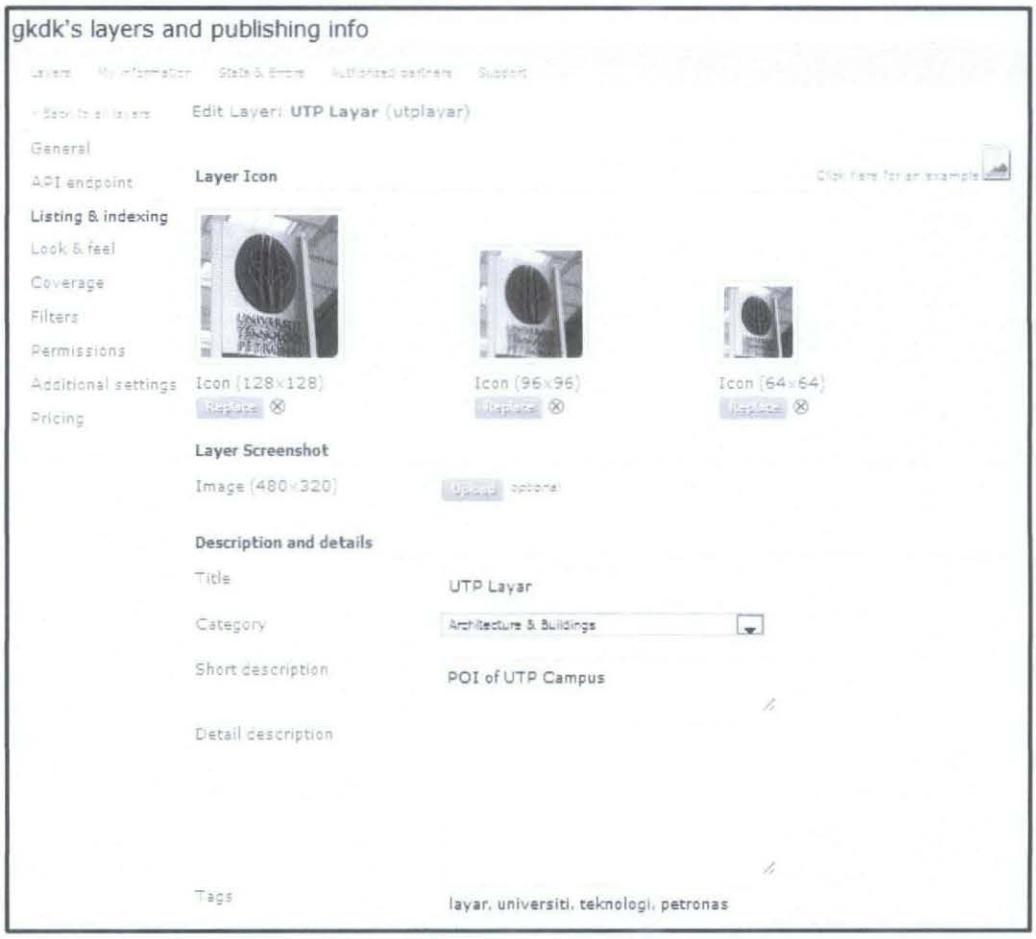


Figure 14: Listing & Indexing Tab

The detailed explanation of what each field that is required to be filled up will be discussed here.

Field: Layer icon	
Required/Default Value	High-res icon is mandatory(while requesting for publication)
Definition	The icon to be used in the list of layers that will be viewed in the Layer Gallery. It is possible to have this icon (in PNG format) in different resolutions, high-res (128X128), medium-res (96X96) and low-res (64X64).
Tooltips	• The medium-res and low-res icons can be generated automatically based on the high-res icon. • Should refrain from using corner effects and/or rounded corners. A mask with 4px radius rounded corners will be applied automatically.
Field: layer Screenshot	
Required/Default Value	Optional
Definition	A screenshot which will be shown below the detail description text on layer listing page.
Tooltips	size 480x320px
Field: Category	
Required/Default Value	Mandatory(while requesting for publication)
Definition	A list of categories that a layer belongs to. The category list and description can be found here.
Tooltips	The layer must be assigned to the right category. This will make sure that users can find the layer using Categories section in Layar application on the phone.

Field: Detail description	
Required/Default Value	Mandatory(while requesting for publication)
Definition	Some text that will further describe what this layer is about and how to use it and where hotspots are located, etc.
Tooltips	• Goal of the layer, e.g. find the places of interest in UTP • Location of the Spots, e.g. UTP campus
Field: Tags	
Required/Default Value	Mandatory/at least 1 tag
Definition	A list of words describing the layer. A maximum of 15 words is supported. Also that the words in the layer title and publisher name are automatically added to the tag list.
Tooltips	• Come up with the right keywords that can cover the content of the layer. Therefore the target audience should be able to find this layer easily.

Table 5: Indexing & Listing fields

To get a better idea on how these fields fit in Layar application on the phone, the screenshot below will provide a clear view of what is displayed by each field:

Figure 15: Indexing & Listing Completed Result

Results

For the layer icon I've chosen to use this picture of Universiti Teknologi PETRONAS as the most appropriate and easily recognizable for users.

Figure 16: Layer Icon

For the required fields of information to be filled up for users to understand this layer and its purpose better, the layer is introduced with the below information.

Field	**Content**
Title	UTP Layar
Category	Architecture & Building
Short description	Points of Interest in UTP Campus
Detail description	This layer provides a clear view of places of interest that can be found in Universiti Teknologi PETRONAS campus. It is intended for those who are new or guests to our university to easily navigate and locate places that are desired.
Tags	universiti, teknologi, petronas, UTP

Table 6: Result of Indexing & Listing

4. Look & Feel Tab

This is to further customize the general look of the interface. It comes with many options that the developer can choose to change. The options that are available in this tab will be further explained below.

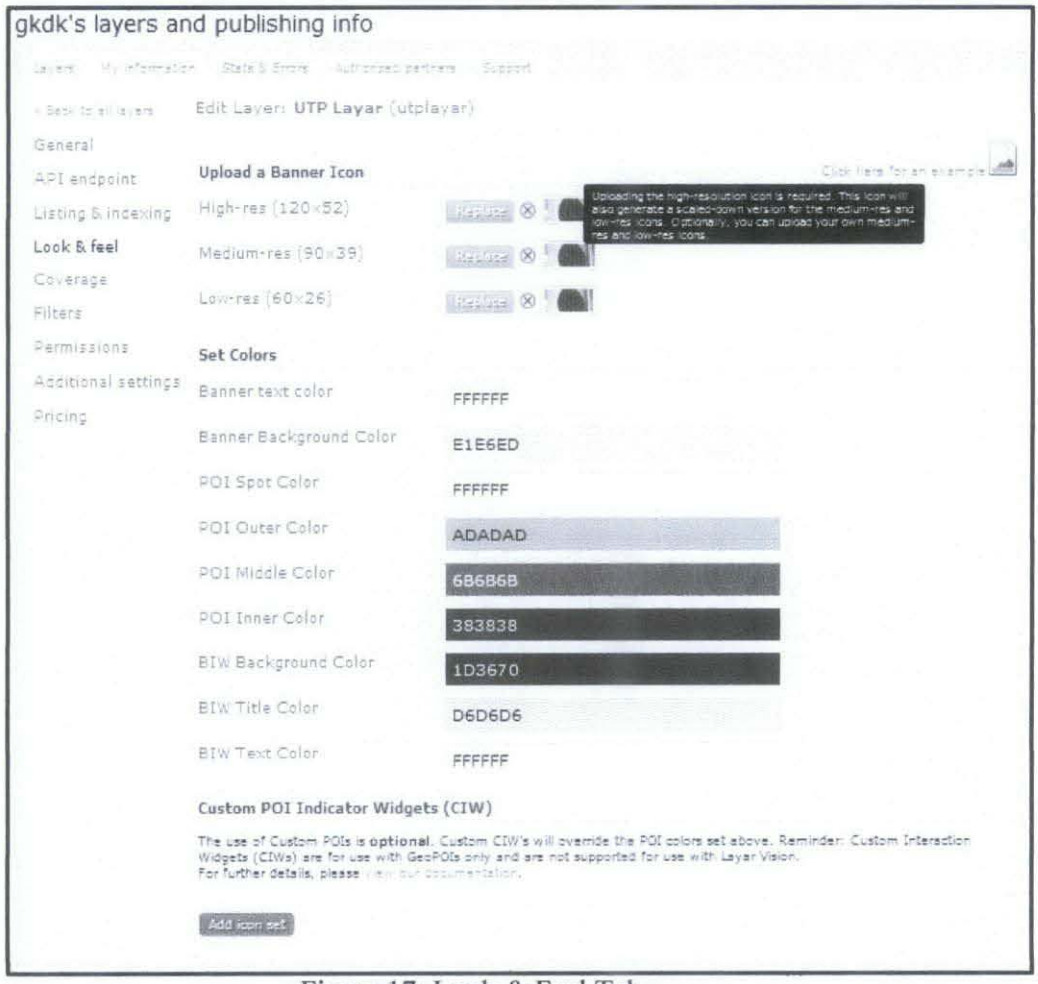


Figure 17: Look & Feel Tab

The banner icon is a required value and it is used for the determining the icon (in PNG format) to be used in the banner for the branding of the layer. The high-res icon is a better choice for this. The high-res icon will also generate a scaled down version of medium-res and low-res. For the banner icon, the chosen icon is used to represent this layer:

Figure 18.1: Banner Icon

The color fields are optional but it is better to field them up as it creates a sort of identity and feel to the layer. All the color fields will affect the layer appearance when it is in camera view and it better to test them out until the right and nice combination of color is chosen.

Figure 18.2: Banner color fields

Custom POI indicator widgets (CIW) is an option that enables the developer to choose to have the standard icon, a default disk shape, to be displayed in the camera view of the POI's or to change that standard icon with a picture of that location instead. This option is used as it creates an easier form for navigation that the users can experience when searching for their desired POIs.

5. Coverage Tab

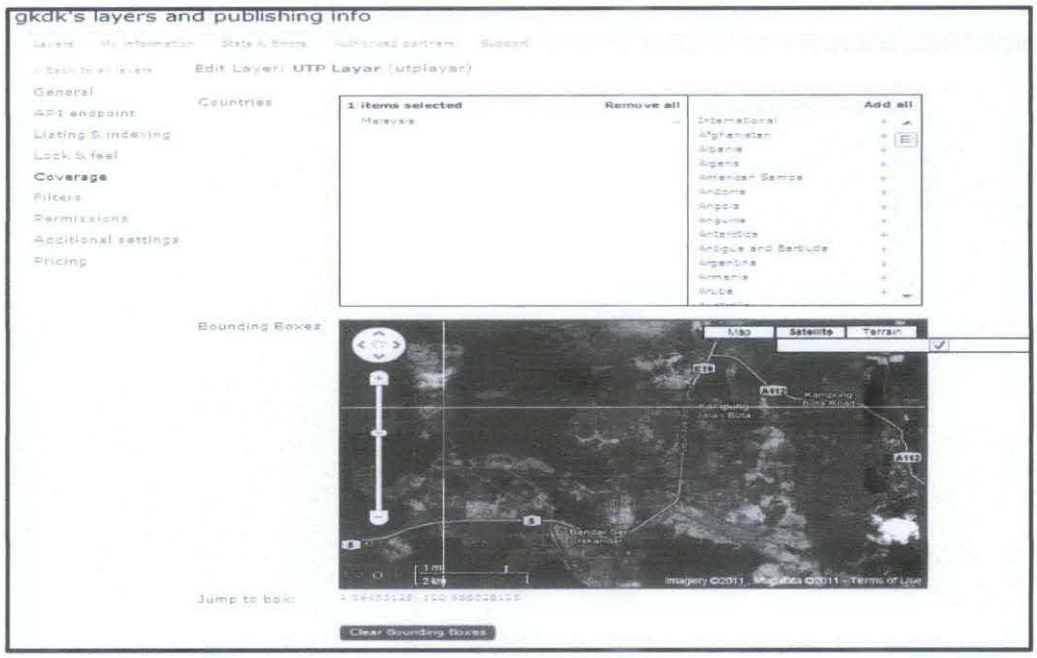


Figure 19: Coverage Tab

The coverage tab is to define in which specific country and region the layer is applicable to use. The countries field is mandatory to be filled as it will be easier for users to search the layer by minimizing the search to a specific country which for this layer will be Malaysia as the applicable country.

Bounding boxes are optional to be field but this enables the layer to be made "local" if all the POIs are clustered in one city or region. With that being said, this option will make the layer more visible to users because it shows in the local tab of the client if a user is in this particular area. The coverage area and the bounding boxes are selected (Refer to Figure 19).

6. Filters Tab

The next tab is called Filters and in this tab there are five different filters that can be associated with each layer. The values of each filter will be sent in the request for hotspots. They are all optional and it is not required for the layer to function as it should. Figure 20.1 display a screenshot of a filter option, Range Slider.

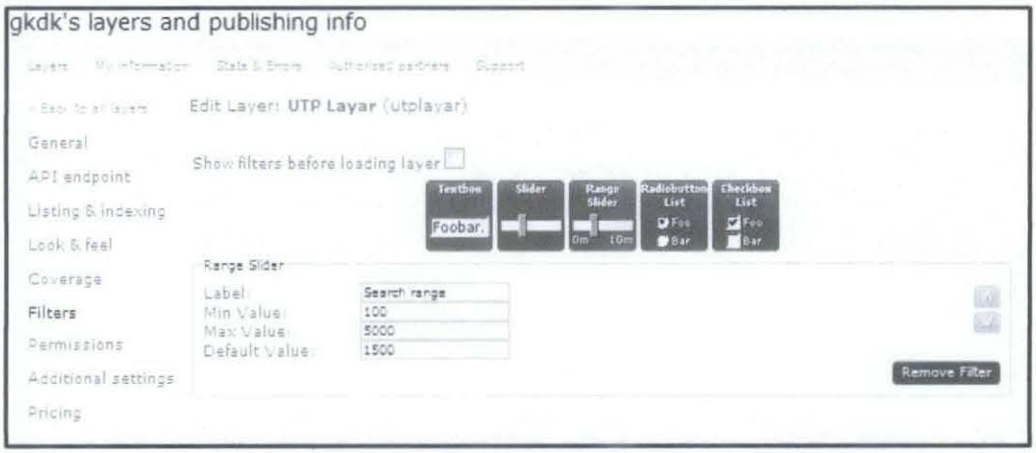


Figure 20.1: Filter Tab

But since it is not required for optimal usage of the layer by the user, this feature won't be included in the UTP Layar as users will be having more trouble as it provides the users to customize the overall view of how they want to see the layer (Refer to Figure 20.2). But perhaps this feature will be included in future versions of this layer to provide better different user experience once this application has been familiarized by the users.

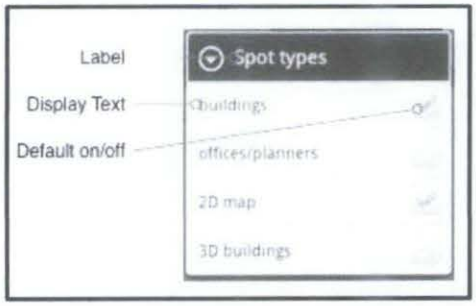


Figure 20.2: Filter Options

7. Permissions Tab

Under the "Permissions" tab a publisher name which was defined while layer creation can be edited as well as at the same time, the ownership of a layer can be transferred from current publisher to another publisher. Developer can also add a viewer for a layer so that this viewer can see this layer on Layar client even if it is still in development for testing and feedback purposes for the developer to make further improvements.

Figure 21: Permissions Tab

By adding viewers to a layer, other people can see this layer on Layar client even it is not in public status. This is especially useful when you want to show it to other people, such as project clients. To add a viewer:

1. Click on "Add viewer" button.
2. Fill in the email address of this viewer.
3. An email is sent to this viewer to notify him/her on this assignment.
4. If this viewer already has a Layar account (with the same email address), he/she can directly start viewing your layer on Layar client after logging in.
5. If not, he will then need to create a Layar account on layar website (www.layar.com) first in order to see this layer on layar client.

4.4.1.4 Step 4: Prepare the Database

After defining a layer on the publishing site, it is time then to prepare the database that will store the POI information. A MySQL database with phpMyAdmin will be used. There many prepared tables already in existence that serves the same function so it is applicable to be used for this application. A table called POI is used. More tables will be created in the coming time to enable more advanced features, such as actions and picture storage.

Based on the POI object description at GetPOIs-JSON Response that displays the required attributes of a table, the POI table is defined displayed in Figure 22.1.

Figure 22.1: POI table

Attribute	Definition
id	Id for the POI to be easily fetched
footnote	Footnote of description for the POI
title	Title of the POI to be displayed in the GUI
lat	Latitude location of the POI
lon	Longitude location of the POI
imageURL	The URl where the image of the POI is hosted
description	The description of the POI to be displayed in the GUI
blwStyle	The font style to be used

Table 7: Database attributes definition

After the initial database is created the other databases are created to store the other necessary attributes related to the POI. This includes the Object, POiAction, Icon and Transform tables which form a relational database with the primary key being ID attribute that forms the relationship among these tables.

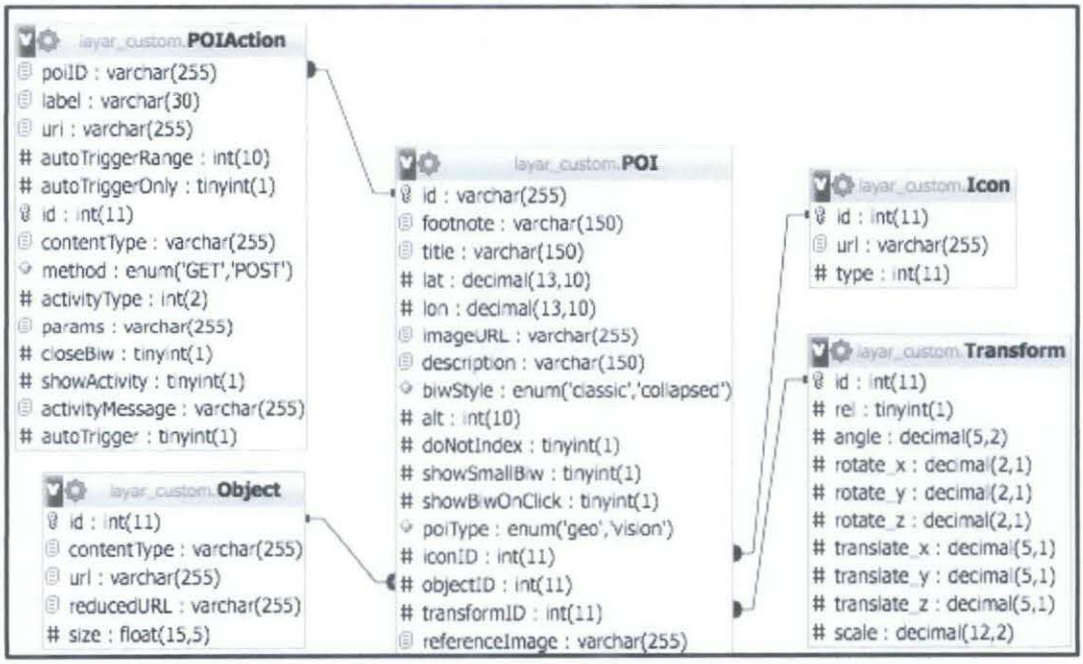


Figure 22.2: Tables for a POI to function

4.4.1.5 Step 5: Gather POIs information

The next phase is to determine the POIs physical location so that it will be possible for the application to trace where the exact location of a POI in real world. This can be achieved in many ways. One of them which is being applied to this particular layer to use the locations exact latitude and longitude coordinates.

For this project, the number of POIs that will be created and tested will be nine locations which are:

- UTP Front Entrance
- Information Resource Centre
- Masjid An-Nur
- Sport Complex
- CIMB USM
- Clinic Pocket C
- Village 2
- Village 4
- Village 5

When the "POI" Table is ready, then it is time to insert POIs into it. For the first POI that will be created, the Information Resource Centre will be targeted. Google Maps is used to get the GPS coordinates for selected locations. Other tools can also be used such as GPS Utility but after testing each method it is way simpler to use Google Maps as it provides accurate coordinates. This is accomplished by doing three steps which are:

Step 1

Search for "Universiti Teknologi PETRONAS" at Google Maps.

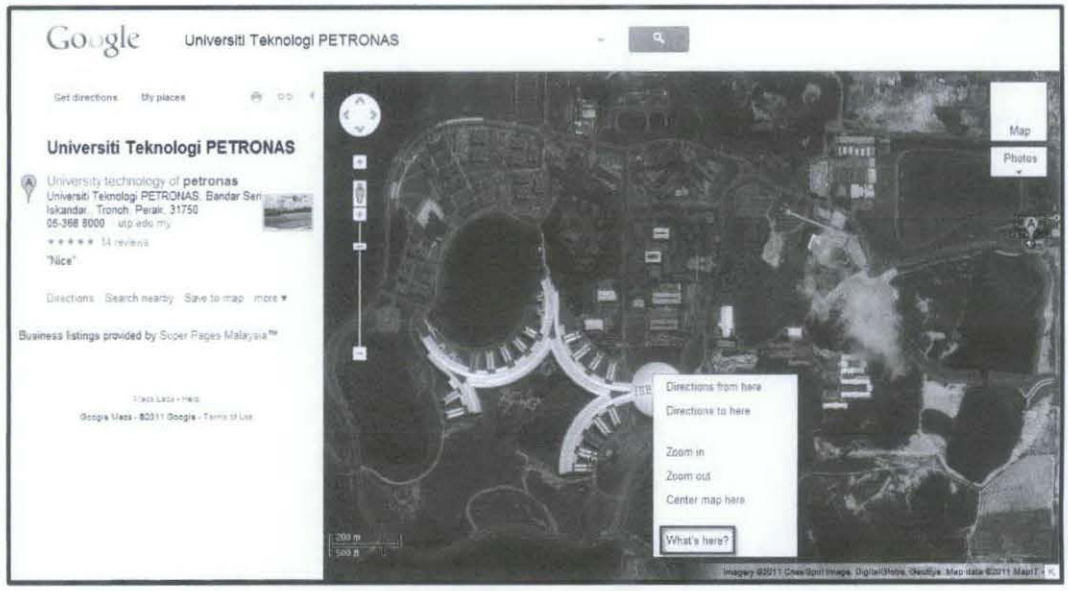


Figure 23.1: GPS coordinate using Google Map

Step 2

The next step is to right click on the icon on the map and press "What's here". The Geo coordinates (latitude, longitude) will be displayed in the search field. In this case the coordinates are (4.382409, 100.969446)

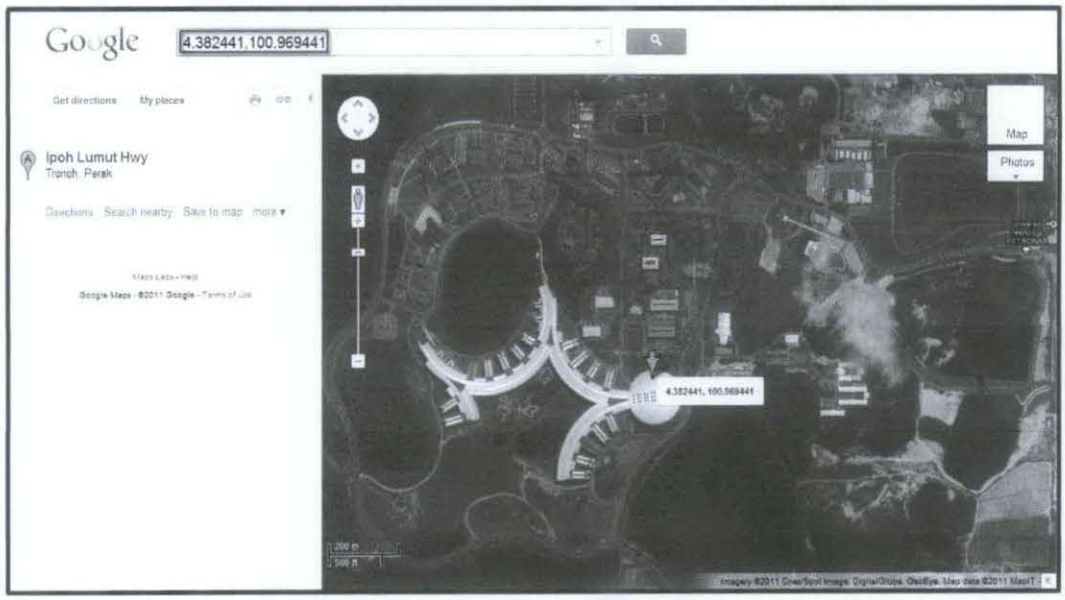


Figure 23.2: Getting the GPS Coordinate of IRC

Step 3

The last step is to add the POI information to the POI below by filling in the necessary fields which can now include the exact location of the IRC by entering the longitude and latitude coordinates. The remaining fields are left as default for now until further changes are made.

4.4.1.6 Step 6: Prepare a web service

Now that POI in the database has been created, the next step is to build up a web service to fetch the POI information and return it back to the Layar platform for further action of the layer to display.

The following php files are used to enable the web service:

1. config.inc.php - contains the database configuration information.
2. firstTutorial_simplified.php - the main php script file which is called by Layar server. It is the same file as defined under "API endpoint URL" field mentioned above.

There are three options that is given for the developers. They can both create the web service from scratch and set up the entire required PHP files. The returned POI information should be formatted in JSON. If PHP version 5.2 or above is used, JSON is then supported natively. Otherwise, the JSON string needs to be constructed programmatically. This methods is takes the most time and is tedious for those who aren't familiar with the programming languages and database linking.

The other option is to use pre constructed php files which can be downloaded and re-used and edited to fit the uniqueness of the layer being created in this the UTP Layar. In the layer definition, there is already provided the POI URL which will be called to retrieve the POI information. So there is an option utilizing this ready prepared PHP files for personal use and it is an open source work so it possible for others to make use of the coding that is already prepared.

Another alternative option on preparing a web service, is to utilize third party tools that provides the developers with the chance of having their project hosted on an already developed and functioning web service for a rapid approach to develop and publish the AR projects for the public usage. This seems to the most reliable and less time consuming method.

For this phase, it is better to explore two of the methods above for completing the goal of the project. The chosen methods are to utilize the ready-made PHP files and modify the coding to localized the information and data used for UTP Layar. After many attempts of trying to link the database and the PHP files as well as coding the GetPois-JSON response, there were many errors and not functioning feature which hindered the advancement of this project completion on the scheduled completion time. For the purpose of research and better understanding of how the system actually fucntions, it is crucial that the PHP and database build is explored (Refer to Appendix C for PHP Coding samples).

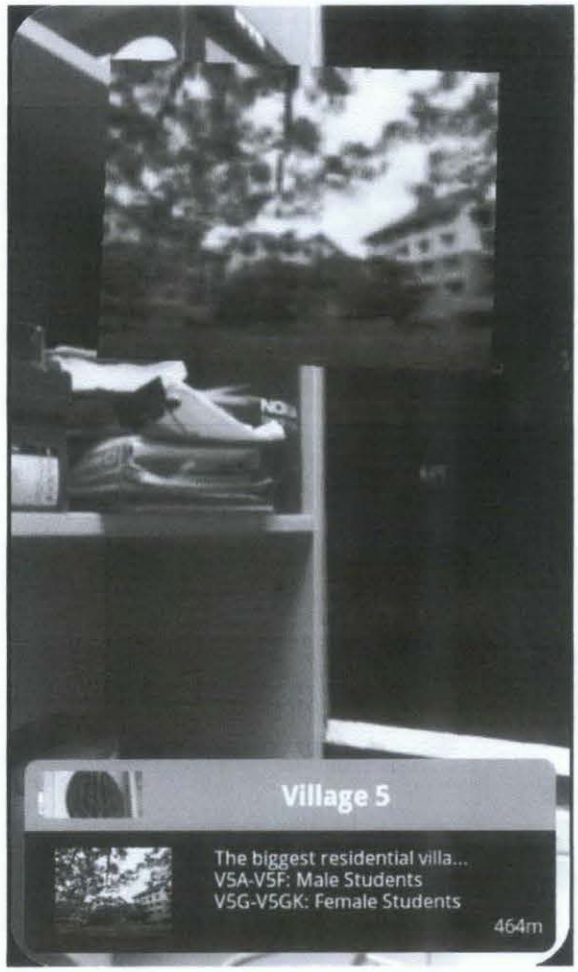
The next method that is explored since there is limited amount of time and applying the rapid prototype methodology adapted for this project is concluded that a different method for completing the projects objectives must be met. With that being said, to apply the third party tools for using a prepared and functioning web service is the most feasible action to be taken.

The third party tools enables the availability of ready configured web service and also provides a database service to easily store information, pictures and POIs for seamless fetching and integration between the Layar application and the layer for the layer to function with stability and efficiently.

4.5 SYSTEM TESTING

System testing is a crucial element for software quality assurance. The test process is implemented to execute the system with the intent to check the performance – the limitation and competency as well as for future references.

For this project, the testing done is the functionality and performance testing on a HTC phone running Android 2.2 and Layar AR Browser.




Accuracy: N/A
CLINIC
CLINIC

No	Function	Action Perform	Expected Test Result	Actual Test Result	Pass / Fail
1	UTP Layar functioning in Layar search browser	User searches for UTP Layar using tag	Layar application detects and displays UTP Layar	Successfully displayed UTP Layar	Pass
		UTP Layar displayed with correct info and icon	Displays correct info and icon in banner	Successfully displays info and icon	Pass
2	UTP Layar prescreen info and screenshot	User selects UTP Layar and displays summary of UTP Layar text	Displays summary	Successfully displays summary	Pass
		Display screenshot for user to see	Screenshot can be viewed	Successfully displays screenshot	Pass
3	UTP Layar launch	User clicks launch button to run UTP Layar	Application will start and detects surrounding POIs	Successfully detects all surrounding POIs	Pass
4	UTP Layar Functions	UTP Layar loads pictures and information of POIs	POIs picture and information is loaded in banner	Successfully loads some surrounding POIs	Pass (with some exception)
		Radar feature displays all nine POIs	Able to view nine dots on the radar	All nine dots successfully displayed	Pass
		Distance measurement	Distance of user from POI displays accurately and changes as user moves closer to POI	Successfully displays measurements and changing distance.	Pass
		POI reality marker is the location image	Displays AR picture of location image of specific POI instead of default disk image	Successfully displays location image instead of default image.	Pass

		User clicks 'More Information'	Action to open a web link to a different page containing more information	Successfully run action and opens a webpage.	Pass
		User clicks POI banner to lock location and unlock location target	POI is locked where it ignore other POI in view and unlocks for fixed navigation to one target.	Successfully locks on and unlocks target POI.	Pass
5	Closing the application	User clicks 'Back' button	Message box is prompted and application is closed.	Successfully exits UTP Layar itself.	Pass

Table 8: Functionality and Performance Test

As can be seen from the functionality and performance testing, proves that the basic functions of the UTP AR layer is running well. The only errors that happen are that some images of POIs fail to load. This is either due to the internet connectivity of the phone or an error in the database when story the data and some requirements are not met such as the size of the image.

Also at times there is a chance for the user to not be in an accurate position. This is due to the geographical location of UTP which makes GPS lock on the phone hard due to its uneven geographical level.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 CONCLUSION

The answer to information technology is a theoretical and an empirical, hands-on discipline. The goal of this project, which aims to expose Universiti Teknologi PETRONAS Computer Information Sciences undergraduate students through the practical experience of working on a large project, has given a chance to put into practice all the software development concepts that have only been studied in theories.

In addition, this project combines the research on the Augmented Reality technology and prototype building aspect of the UTP AR Layer which goal is to allow visitors and new students to attain information regarding the orientation of the univeristy's campus as to increase navigation and increase in the satisfactory of knowing the many places of interest that is found within this Universiti Teknologi PETRONAS campus.

5.2 FUTURE WORK

5.2.1 Further improve on the features of the AR layer

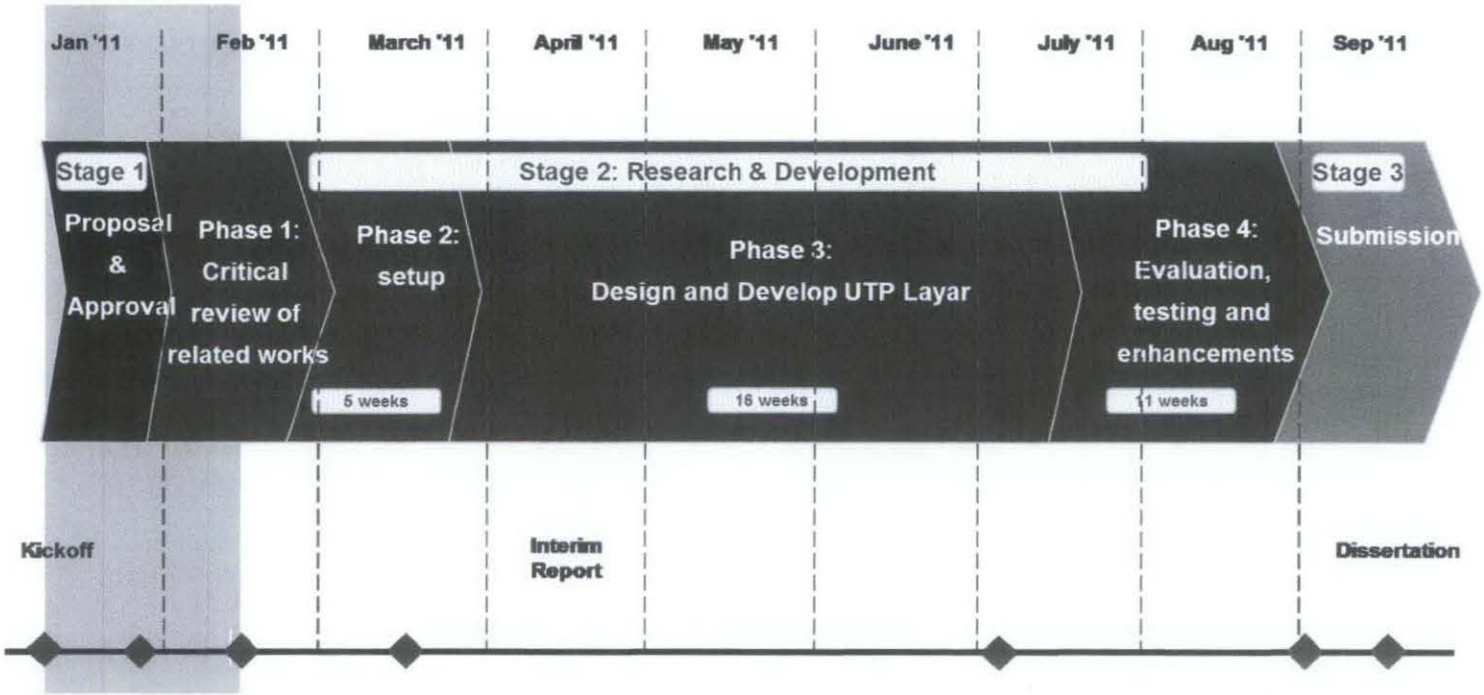
Further improvements to the functions of the UTP Layar is needed so that it's basic use of being a GPS guide can be instead expanded to serve as knowledge sharing tool where information regarding places and locations in UTP are made known for the public. These functions include such the ability to share video and music when clicking POI, be more interactive such as displaying more options and displaying 3D objects to further gain more interest. A possibility to create an AR billboard around UTP which can be used for putting advertisements and posters by the various clubs and events held in UTP so it eliminates the need to post physical posters around the campus.

REFERENCES

- Comai S., Tanca L., 2003. *Termination and Confluence by Rule Prioritization*, IEEE Transactions on Knowledge and Data Engineering, Vol. 15, No. 2, March 2003, pg 257-270.
- Dennis A., Wixom B.H., Tegarden D. (2005), *Systems Analysis and Design with UML Version 2.0*. John Wiley & Sons, Inc.
- Milgram, P., Kishino, F, 1994. "Augmented Reality: A Class of Displays on the Reality-virtuality Continuum." SPIE, Telemanipulator and Telepresence Technologies, 2351 (34). 42-48.
- Hayes, Gary. "16 Top Augmented Reality Business Models." Jan. 26, 2010. http://www.personalizemedia.com/16-top-augmented-reality-business-models
- Hayes, Gary. "Augmented Reality Business Ideas – AR The Next Big Thing in Enterprise Computing?" Sept. 14, 2009. http://laridon.com/augmented-reality-business-ideas-ar-the-next-big-thing-in-enterprise-computing
- Vallino, James R, 1998. "Interactive Augemented Reality" National Science Foundation & National Institutes of Health, 18-22.
- Ensha, Azadeh. "Another Augmented-Reality App for the iPhone." New York Times. Oct. 15, 2009. http://gadgetwise.blogs.nytimes.com/2009/10/15/augmented-reality-apps-continute-to-roll-out
- Parr, Ben. "Easter Egg: Yelp Is the iPhone's First Augmented Reality App." Mashable. Aug. 27, 2009. http://mashable.com/2009/08/27/yelp-augmented-reality
- Ganapati, Priya. "How it Works: Augmented Reality." Wired. Aug. 25, 2009. http://www.wired.com/gadgetlab/2009/08/total-immersion
- Berlin, Leslie. "Kicking Reality Up a Notch." New York Times. July 11, 2009. http://www.nytimes.com/2009/07/12/business/12proto.html

APPENDICES

APPENDIX A – PROJECT MILESTONES



Stage 1: Proposal & Approval

Stage 2: Research & Development

Stage 3: Submission

APPENDIX B – WORK B REAKDOWN STRUCTURE (WBS)

TASK NO.	TASK NAME	STATUS
PLANNING		
1.0	Conduct meetings with supervisor to confirm scope and topic	Complete
1.1	Make initial contact with colleagues and visitors	Complete
1.3	Identify project objective, purpose and scope	Complete
1.4	Perform Feasibility Analysis	Complete
1.4.1	Perform Technical Feasibility Analysis	Complete
1.4.2	Perform Organizational Feasibility Analysis	Complete
ANALYSIS		
2.0	Requirements determination and gathering	Complete
2.0.1	Collect existing reports and system documentation	Complete
2.0.2	Conduct interview	Complete
2.1	Perform literature review	Complete
2.2	Produce Preliminary and Progress Report	Complete
DESIGN		
3.0	Business Process Modeling	Complete
3.0.2	Consult with Supervisor	Complete
3.1	Use Case Modeling	Complete
3.1.1	Draft Use Case Diagram	Complete
3.1.2	Consult with Supervisor	Complete
3.1.3	Produce final Use Case Diagram	Complete
3.2	Structure Modeling	Complete
3.2.1	Draft Class Diagram	Complete
3.2.2	Consult with Supervisor	In Progress
3.3	Design preliminary system interface	In Progress
IMPLEMENTATION		
4.0	Build prototype.	Open
4.0.1	Develop finalized layer definition	Open
4.0.2	Develop database using MySQL	Open
4.0.2.1	Build Web Service using PHP	Open
4.0.2.2	Enable radar and distance features capabilities	Open
4.1	Continuous refinement	Open
TESTING		
5.0	Evaluate prototype	Open
5.1	Perform User Acceptance Test (UAT)	Open
5.2	Identify and fix bugs	Open
5.3	If requirements are unmet, revisit Planning, Analysis and Design Phase	Open

APPENDIX C: WEB SERVICE PHP CODING SAMPLE

config.inc.php coding

```
<?php

// Copyright (c) 2011, Layar B.V.
// All rights reserved.

// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
met:
//     * Redistributions of source code must retain the above copyright
//       notice, this list of conditions and the following disclaimer.
//     * Redistributions in binary form must reproduce the above copyright
//       notice, this list of conditions and the following disclaimer in the
//       documentation and/or other materials provided with the distribution.
//     * Neither the name of the <organization> nor the
//       names of its contributors may be used to endorse or promote products
//       derived from this software without specific prior written
permission.

// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
IS"
// AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
// IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE
// ARE DISCLAIMED. IN NO EVENT SHALL LAYAR B.V BE LIABLE FOR ANY DIRECT,
// INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
// (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES;
// LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
AND
// ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR
TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS
// SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

/* Pre-define connection to the MySQL database, please specify these fields
based on your database configuration.*/
  define('DBHOST', 'localhost');
  define('DBDATA', 'database_name');
  define('DBUSER', 'database_username');
  define('DBPASS', 'database_password');
?>
```

fifthTutorial_Filters_simplified.php coding

```
<?php
// Created by Xuan Wang
// Layar Technical Support
// Email: xuan@layar.com
// Website: http://layar.com

/*** Include some external files ***/

// Include database credentials. Please customize these fields with your own
// database configuration.
require_once('config.inc.php');


/*** Specific Custom Functions ***/

// Put needed getPOI request parameters and their values in an associative
array
//
// Arguments:
//  array ; An array of needed parameters passed in getPOI request
//
// Returns:
//  array ; An associative array which contains the request parameters and
//  their values.
function getRequestParams($keys) {

  $paramsArray = array();
  try {
    // Retrieve parameter values using $_GET and put them in $value array
with
    // parameter name as key.
    foreach( $keys as $key ) {
      if ( isset($_GET[$key]) )
        $paramsArray[$key] = $_GET[$key];
      else
        throw new Exception($key .' parameter is not passed in GetPOI
request.');
    }
    return $paramsArray;
  }
  catch(Exception $e) {
    echo 'Message: ' .$e->getMessage();
  }
}//getRequestParams

// Connect to the database, configuration information is stored in
// config.inc.php file
function connectDb() {
  try {
    $dbconn = 'mysql:host=' . DBHOST . ';dbname=' . DBDATA ;
    $db = new PDO($dbconn , DBUSER , DBPASS ,
array(PDO::MYSQL_ATTR_INIT_COMMAND => 'SET NAMES utf8'));
    // set the error mode to exceptions
    $db->setAttribute(PDO::ATTR_ERRMODE , PDO::ERRMODE_EXCEPTION);
```

```
    return $db;
  }// try
  catch(PDOException $e) {
    error_log('message:' . $e->getMessage());
  }// catch
}// connectDb

// Change a string value to float
//
// Arguments:
//   string ; A string value.
//
// Returns:
//   float ; If the string is empty, return NULL.
//
function changetoFloat($string) {
  if (strlen(trim($string)) != 0)
    return (float)$string;
  return NULL;
}//changetoFloat

// Change a string value to integer.
//
// Arguments:
//   string ; A string value.
//
// Returns:
//   Int ; If the string is empty, return NULL.
//
function changetoInt($string) {
  if (strlen(trim($string)) != 0)
    return (int)$string;
  return NULL;
}//changetoInt

// Convert a string into an array.
//
// Arguments:
//  string ; The input string
//  separater, string ; The boundary string used to separate the input
string
//
// Returns:
//  array ; An array of strings. Otherwise, return an empty array.
function changetoArray($string, $separator){
  $newArray = array();
  if($string) {
    if (substr_count($string,$separator)) {
      $newArray= array_map('trim' , explode($separator, $string));
        }//if
    else
      $newArray[0] = trim($string);
  }
  return $newArray;
}//changetoArray

// Convert a TinyInt value to a boolean value TRUE or FALSE
```

```
//
// Arguments:
//  int  value_Tinyint ; The Tinyint value (0 or 1) of a key in the
database.
//
// Returns:
//   boolean ; The boolean value, return 'TRUE' when Tinyint is 1. Return
//     'FALSE' when Tinyint is 0.
//
function changetoBool($value_Tinyint) {
  if (strlen(trim($value_Tinyint)) != 0) {
    if ($value_Tinyint == 0)
      return FALSE;
    else
      return TRUE;
   }
  return NULL;
}//changetoBool

// Put fetched actions for each POI into an associative array.
//
// Arguments:
//   db ; The database connection handler.
//   poi ; The POI array.
//
// Returns:
//   array ; An associative array of received actions for this
POI.Otherwise,
//   return an empty array.
//
function getPoiActions($db , $poi) {
  // Define an empty $actionArray array.
  $actionArray = array();

  // A new table called 'POIAction' is created to store actions, each action
  // has a field called 'poiID' which shows the POI id that this action
belongs
  // to.
  // The SQL statement returns actions which have the same poiID as the id
of
  // the POI($poiID).
  $sql_actions = $db->prepare('
      SELECT label,
             uri,
             contentType,
             activityType,
             autoTriggerRange,
             autoTriggerOnly,
             params
      FROM POIAction
      WHERE poiID = :id ');

  // Binds the named parameter marker ':id' to the specified parameter value
  // '$poiID.
  $sql_actions->bindParam(':id', $poi['id'], PDO::PARAM_STR);
  // Use PDO::execute() to execute the prepared statement $sql_actions.
  $sql_actions->execute();
```

```
  // Iterator for the $actionArray array.
  $count = 0;
  // Fetch all the poi actions.
  $actions = $sql_actions->fetchAll(PDO::FETCH_ASSOC);

  /* Process the $actions result */
  // if $actions array is not empty.
  if ($actions) {
    // Put each action information into $actionArray array.
    foreach ($actions as $action) {
      // Change 'activityType' to Integer.
      $action['activityType'] = changetoInt($action['activityType']);
      $action['autoTriggerRange'] =
changetoInt($action['autoTriggerRange']);
      $action['autoTriggerOnly'] = changetoBool($action['autoTriggerOnly']);
      $action['params'] = changetoArray($action['params'] , ',');
      // Assign each action to $actionArray array.
      $actionArray[$count] = $action;
      $count++;
    }// foreach
  }//if
  return $actionArray;
}//getPoiActions

// Put fetched icon dictionary for each POI into an associative array.
//
// Arguments:
//  db ; The database connection handler.
//  iconID, integer ; The iconID value  which is stored in this POI.
//
// Return:
//  array ; An associative array of retrieved icon dictionary for this POI.
//  Otherwise, return NULL.
function getIcon($db, $iconID) {
  // If no icon object is found, return NULL.
  $icon = NULL;

  // Run the query to retrieve icon information for this POI.
  $sql_icon = $db->prepare( '
            SELECT url, type
              FROM Icon
             WHERE id = :iconID
            ' );
  $sql_icon->bindParam(':iconID', $iconID, PDO::PARAM_INT);
  $sql_icon->execute();
  $rawIcon = $sql_icon->fetch(PDO::FETCH_ASSOC);

  // Assign returned values to $icon array.
  if($rawIcon){
    $rawIcon['type'] = changetoInt($rawIcon['type']);
    $icon = $rawIcon;
  }
  return $icon;
}//getIcon

// Put fetched object parameters for each POI into an associative array.
//
```

```
// Arguments:
//   db ; The database connection handler.
//   objectID, integer ; The object id assigned to this POI.
//
// Returns:
//   associative array or NULL ; An array of received object related
parameters
//   for this POI. otherwise, return NULL.
//
function getObject($db , $objectID) {
  // If no object object is found, return NULL.
  $object = NULL;

  // A new table called 'Object' is created to store object related
parameters,
  // namely 'url', 'contentType', 'reducedURL' and 'size'. The SQL statement
  // returns object which has the same id as $objectID stored in this POI.
  $sql_object = $db->prepare(
    ' SELECT contentType,
             url,
             reducedURL,
             size
      FROM Object
      WHERE id = :objectID
      LIMIT 0,1 ');

  // Binds the named parameter marker ':objectID' to the specified parameter
  // value $objectID.
  $sql_object->bindParam(':objectID', $objectID, PDO::PARAM_INT);
  // Use PDO::execute() to execute the prepared statement $sql_object.
  $sql_object->execute();
  // Fetch the poi object.
  $rawObject = $sql_object->fetch(PDO::FETCH_ASSOC);

  /* Process the $rawObject result */
  // if $rawObject array is not empty.
  if ($rawObject) {
    // Change 'size' type to float.
    $rawObject['size'] = changetoFloat($rawObject['size']);
    $object = $rawObject;
  }
  return $object;
}//getObject


// Put fetched transform related parameters for each POI into an associative
// array. The returned values are assigned to $poi[transform].
//
// Arguments:
//   db ; The database connection handler.
//   transformID , integer ; The transform id which is assigned to this POI.
//
// Returns: associative array or NULL; An array of received transform
related
// parameters for this POI. Otherwise, return NULL.
//
function getTransform($db , $transformID) {
```

```
  // If no transform object is found, return NULL.
  $transform = NULL;
  // A new table called 'Transform' is created to store transform related
  // parameters, namely 'rotate','translate' and 'scale'.
  // 'transformID' is the transform that is applied to this POI.
  // The SQL statement returns transform which has the same id as the
  // $transformID of this POI.
  $sql_transform = $db->prepare('
      SELECT rel,
             angle,
             rotate_x,
             rotate_y,
             rotate_z,
             translate_x,
             translate_y,
             translate_z,
             scale
      FROM Transform
      WHERE id = :transformID
      LIMIT 0,1 ');

  // Binds the named parameter marker ':transformID' to the specified
parameter
  // value $transformID
  $sql_transform->bindParam(':transformID', $transformID, PDO::PARAM_INT);
  // Use PDO::execute() to execute the prepared statement $sql_transform.
  $sql_transform->execute();
  // Fetch the poi transform.
  $rawTransform = $sql_transform->fetch(PDO::FETCH_ASSOC);

  /* Process the $rawTransform result */
  // if $rawTransform array is not  empty
  if ($rawTransform) {
    // Change the value of 'scale' into decimal value.
    $transform['scale'] = changetoFloat($rawTransform['scale']);
    // organize translate field
    $transform['translate']['x']
=changetoFloat($rawTransform['translate_x']);
    $transform['translate']['y'] =
changetoFloat($rawTransform['translate_y']);
    $transform['translate']['z'] =
changetoFloat($rawTransform['translate_z']);
    // organize rotate field
    $transform['rotate']['axis']['x'] =
changetoFloat($rawTransform['rotate_x']);
    $transform['rotate']['axis']['y'] =
changetoFloat($rawTransform['rotate_y']);
    $transform['rotate']['axis']['z'] =
changetoFloat($rawTransform['rotate_z']);
    $transform['rotate']['angle'] = changetoFloat($rawTransform['angle']);
    $transform['rotate']['rel'] = changetoBool($rawTransform['rel']);
  }//if

  return $transform;
}//getTransform

// Prepare the search value which will be used in SQL statement.
```

```
// Arguments:
//   searchbox ; the value of SEARCHBOX parameter in the GetPOI request.
//
// Returns:
//   searchbox_value ; If searchbox parameter has an empty string, return a
//   string which is  a combination of numbers, letters and white spaces.
//   Otherwise, return the value of searchbox parameter.

function getSearchValue ($searchbox) {

        // if $searchbox exists, prepare search value.
        if (isset($searchbox)) {

    // initiate searchbox value to be any string that consists of numbers,
    // letters and spaces.
          $searchbox_value = '[0-9a-zA-Z\s]*';

          // if $searchbox is not an empty string, return the $searchbox
value.
          if (!empty($searchbox))
                $searchbox_value = $searchbox;

                return $searchbox_value;
        } //if
        else { // If $searchbox does not exist, throw an exception.
                throw new Exception("searchbox parameter is not passed in
GetPOI request.");
        }//else

}// getSearchValue

// Prepare radiolist value which will be used in SQL statement. In this
// function, we convert the returned value into the ones that are stored in
the
// database.
//
// Arguments:
// radiolist ; the integer value of RADIOLIST parameter in the GetPOI
request.
//
// Returns:
// radio_value ; the value that can be used to construct the right SQL
// statement.
function getRadioValue ($radiolist) {
  // if $radiolist exists, prepare radio_value.
  if(isset($radiolist)) {

    $radio_value = '';
        // if $radiolist == 1, return $radio_value ="sale";
        // if $radiolist == 2, return $radio_value ="rent";
        switch ($radiolist) {
          case '1':
                $radio_value = "sale" ;
                break;
          case '2':
                $radio_value = "rent" ;
                break;
```

```
          default:
                throw new Exception("invalid radiolist value:" . $radiolist);
          } //switch

         return $radio_value;
        }//if
        else {
          throw new Exception("radiolist parameter is not passed in GetPOI
request.");
        }//else

}// getRadioValue

// Prepare checkbox value which will be used in SQL statement.
// In this function, we add all the numbers in $checkboxlist parameter. If
// $checkboxlist is empty, then we return 0.
//
// Arguments:
// checkboxlist ; the value of CHECKBOXLIST parameter in the GetPOI request.
//
// Returns:
// checkbox_value ; the value that can be used to construct the right SQL
// statement.

function getCheckboxValue ($checkboxlist) {

  // if $checkboxlist exists, prepare checkbox_value.
  if(isset($checkboxlist)) {

    // Initialize returned value to be 0 if $checkboxlist is empty.
        $checkbox_value = 0;

  // If $checkboxlist is not empty, return the added value of all the
numbers
  // splited by ','.
        if (!empty($checkboxlist)) {

                if (strstr($checkboxlist , ',')) {

                        $checkbox_array = explode(',' , $checkboxlist);

                        for( $i=0; $i<count($checkbox_array); $i++ )
                                $checkbox_value+=$checkbox_array[$i];

                }//if
                else
                        $checkbox_value = $checkboxlist;
        }//if

        return $checkbox_value;
  } //if
  else {
    throw new Exception("checkboxlist parameter is not passed in GetPOI
request.");
  }//else

}//getCheckboxValue
```

```
// Prepare custom_slider value which will be used in SQL statement.
// In this function, we simply return the value of $customslider defined in
the
// GetPOI request.
//
// Arguments:
// customslider ; the value of CUSTOM_SLIDER parameter in the GetPOI
request.
//
// Returns:
// customslider ; the value that can be used to construct the right SQL
// statement.
//
function getSliderValue ($customslider) {

    // if $customslider exists, return its value.
        if(isset($customslider))

          return $customslider;

        else
          throw new Exception("custom slider parameter is not passed in
GetPOI request.");

}//getSliderValue

// Put received POIs into an associative array. The returned values are
// assigned to $reponse['hotspots'].
//
// Arguments:
//   db ; The handler of the database.
//   value , array ; An array which contains all the needed parameters
//   retrieved from GetPOI request.
//
// Returns:
//   array ; An array of received POIs.
//
function getHotspots( $db, $value ) {
  // Define an empty $hotspots array.
  $hotspots = array();
/* Create a SQL query to retrieve POIs which meet the criterion of filter
settings returned from GetPOI request.
   Returned POIs are sorted by distance and the first 50 POIs are selected.
   - The distance is caculated based on the Haversine formula.
     Note: this way of calculation is not scalable for querying large
database.
   - searchbox filter, find POIs with title that contains the search term.
     If the searchbox is empty, all POIs are returned.
   - radiolist filter, find POIs with value from "Radiolist" column that
equals to the prepared
     radiolist value from GetRadioValue function.
   - checkbox filter, find POIs which don't return 0 after comparing the
value from "Checkbox" column
     and prepared checkbox value (from GetCheckboxValue function) using
Bitwise operations.
```

```
    http://en.wikipedia.org/wiki/Bitwise_operation. if CHECKBOX parameter
is empty, then no POIs are returned.
   - custom_slider filter, find POIs with value from "Custom_Slider" column
that is not bigger than
    the CUSTOM_SLIDER parameter value passed in the GetPOI request.
*/

  // Use PDO::prepare() to prepare SQL statement. This statement is used due
to
  // security reasons and will help prevent general SQL injection attacks.
  // ':lat1', ':lat2', ':long' and ':radius' are named parameter markers for
  // which real values will be substituted when the statement is executed.
  // $sql is returned as a PDO statement object.
  $sql = $db->prepare( '
                        SELECT id,
              imageURL,
              title,
              description,
              footnote,
              lat,
              lon,
              (((acos(sin((:lat1 * pi() / 180)) * sin((lat * pi() / 180)) +
                        cos((:lat2 * pi() / 180)) * cos((lat * pi() / 180))
*
                      cos((:long  - lon) * pi() / 180))
                      ) * 180 / pi()
              )* 60 * 1.1515 * 1.609344 * 1000
              ) as distance,
              iconID,
              objectID,
              transformID
                        FROM POI_RealEstate
         WHERE poiType = "geo"
           AND title REGEXP :search
           AND Radiolist = :radiolist
           AND (Checkbox & :checkbox) != 0
           AND Custom_Slider <= :slider
       HAVING distance < :radius
     ORDER BY distance ASC
        LIMIT 0, 50 ' );

  // PDOStatement::bindParam() binds the named parameter markers to the
  // specified parameter values.
  $sql->bindParam(':lat1', $value['lat'], PDO::PARAM_STR);
  $sql->bindParam(':lat2', $value['lat'], PDO::PARAM_STR);
  $sql->bindParam(':long', $value['lon'], PDO::PARAM_STR);
  $sql->bindParam(':radius', $value['radius'], PDO::PARAM_INT);

  // Custom filter settings parameters. The four Get functions can be
  // customized.
  $sql->bindParam(':search', getSearchValue($value['SEARCHBOX']),
PDO::PARAM_STR);
  $sql->bindParam(':radiolist', getRadioValue($value['RADIOLIST']),
PDO::PARAM_STR);
  $sql->bindParam(':checkbox', getCheckboxValue($value['CHECKBOXLIST']),
PDO::PARAM_INT);
```

```
  $sql->bindParam(':slider', getSliderValue($value['CUSTOM_SLIDER']),
PDO::PARAM_INT);

  // Use PDO::execute() to execute the prepared statement $sql.
  $sql->execute();
  // Iterator for the response array.
  $i = 0;
  // Use fetchAll to return an array containing all of the remaining rows in
  // the result set.
  // Use PDO::FETCH_ASSOC to fetch $sql query results and return each row as
an
  // array indexed by column name.
  $rawPois = $sql->fetchAll(PDO::FETCH_ASSOC);

  /* Process the $pois result */
  // if $rawPois array is not  empty
  if ($rawPois) {

    // Put each POI information into $hotspots array.
        foreach ( $rawPois as $rawPoi ) {
        $poi = array();
      $poi['id'] = $rawPoi['id'];
      $poi['imageURL'] = $rawPoi['imageURL'];
      // Get anchor object information
      $poi['anchor']['geolocation']['lat'] = changetoFloat($rawPoi['lat']);
      $poi['anchor']['geolocation']['lon'] = changetoFloat($rawPoi['lon']);
      // get text object information
      $poi['text']['title'] = $rawPoi['title'];
      $poi['text']['description'] = $rawPoi['description'];
      $poi['text']['footnote'] = $rawPoi['footnote'];
      //User function getPOiActions() to return an array of actions
associated
      //with the current POI
      $poi['actions'] = getPoiActions($db, $rawPoi);
      // Get object object information if iconID is not null
      if(count($rawPoi['iconID']) != 0)
        $poi['icon'] = getIcon($db , $rawPoi['iconID']);
      // Get object object information if objectID is not null
      if(count($rawPoi['objectID']) != 0)
        $poi['object'] = getObject($db, $rawPoi['objectID']);
      // Get transform object information if transformID is not null
      if(count($rawPoi['transformID']) != 0)
        $poi['transform'] = getTransform($db, $rawPoi['transformID']);
      // Put the poi into the $hotspots array.
      $hotspots[$i] = $poi;
      $i++;
    }//foreach
  }//if
  return $hotspots;
}//getHotspots

/*** Main entry point ***/

/* Put parameters from GetPOI request into an associative array named
$requestParams */
// Put needed parameter names from GetPOI request in an array called $keys.
```

```
$keys = array('layerName', 'lat', 'lon', 'radius', 'RADIOLIST',
'CHECKBOXLIST', 'CUSTOM_SLIDER', 'SEARCHBOX');

// Initialize an empty associative array.
$requestParams = array();
// Call funtion getRequestParams()
$requestParams = getRequestParams($keys);
/* Connect to MySQL server. We use PDO which is a PHP extension to formalise
database connection.
        For more information regarding PDO, please see
http://php.net/manual/en/book.pdo.php.
 */
// Connect to predefined MySQl database.
$db = connectDb();

/* Construct the response into an associative array.*/

// Create an empty array named response.
$response = array();

// Assign cooresponding values to mandatory JSON response keys.
$response['layer'] = $requestParams['layerName'];

// Use Gethotspots() function to retrieve POIs with in the search range.
$response['hotspots'] = getHotspots($db, $requestParams);

// if there is no POI found, return a custom error message.
if (!$response['hotspots'] ) {
        $response['errorCode'] = 20;
        $response['errorString'] = 'No POI found. Please adjust the range.';
}//if
else {
  $response['errorCode'] = 0;
  $response['errorString'] = 'ok';
}//else

        /* All data is in $response, print it into JSON format.*/

        // Put the JSON representation of $response into $jsonresponse.
        $jsonresponse = json_encode( $response );

        // Declare the correct content type in HTTP response header.
        header( 'Content-type: application/json; charset=utf-8' );

        // Print out Json response.
        echo $jsonresponse;

?>
```