PANORAMIC STREET VIEW

MAP NAVIGATION IN REAL WORLD ENVIRONMENT

By

Maj Yasir Khan Jadoon

NC Ruba Aslam

NC Mudasser Ajaz

Submitted to the Faculty of Computer Science, Military College of Signals
National University of Sciences and Technology, Islamabad in partial
fulfillment for the requirements of a B.E. Degree in Software Engineering
JUNE 2013

Panoramic Street View

This project has been completed under the supervision of Lt Col Dr Naveed Iqbal Rao, in Military College of Signals, NUST.

Supervisor

Lt Col Dr Naveed Iqbal Rao ________________________

ABSTRACT

Street View is an emerging concept in the world of online map navigation. Google developed it and have continuously improving it till date. It has wide variety of applications in different fields. The data gathered for its creation can be used for many information extraction purposes.

Panoramic Street View has been developed with the integration of images taken through Optical Cameras, IR sensors and GPS information. All these sensors are placed on a vehicle, which moves across the designated area, takes images and combine them into 360 panoramic street views.

This project aims in developing a framework by integrating all the sensors mounted at a vehicle/car and using Google's open source APIs. It is an interactive system which provides the users with the map of MCS and allows navigation just as in real world environment. Panoramas are created to generate the required navigational view. A vehicle is equipped with cameras, GPS and LRF sensor to gather the required data for the creation of walkthrough.

The system's graphical user interface (GUI) is much like a web page which can be viewed using a web browser. The user interface have same look and feel just as in Google Street View. Images and sensor data is stored on a separate storage device. The end application can run on any web browser running on any operating system. Google Maps API is used to get the maps and integrate panoramas into it to form a walkthrough.

COPYRIGHT NOTICE

DECLARATION

We hereby declare that no portion of the work referred to in this Project Thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning. If any act of plagiarism found, we are fully responsible for every disciplinary action taken against us depending upon the seriousness of the proven offence.

We dedicate this project to our parents and our beloved country Pakistan

ACKNOWLEDGEMENTS

In the name of Allah, the Most Gracious and the Most Merciful Alhamdulillah, all praises to Allah for the strengths and His blessing in completing this thesis.

Foremost, we would like to express our sincere gratitude to our advisor Lt Col Dr Naveed Iqbal Rao for the continuous support, patience, motivation, enthusiasm, and immense knowledge that he extended to us during our project work. His guidance helped us in all the time of research and writing of this thesis. We could not have imagined having a better advisor and mentor for our UG study.

Besides our advisor, we would like to thank Major General Raza Muhammad, HI(M) who remained a hidden support to us and facilitated our research and development with the best possible ways. In addition, we are thankful to Major General Amir Azeem Bajwa, HI(M) for showing generosity and financing our research work. Sincere thanks to Major General Suhail Abbas Zaidi, HI(M) for his technical support which was direly needed.

We are thankful to the Commandant Military College of Signals, Brigadier Ali Farhan for providing us with a learning environment which enabled us to carry out this project and thesis successfully.

Last but not the least, we also thank Google for all the research & development work they had made available of its Street View, without which this project would not have been possible.

Finally, we thank our parents and families for their many years of love and encouragement.

TABLE OF CONTENTS

LIST OF ANNEXURES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Chapter 1

Introduction

Panoramic Street View is an interactive application which allows its users to navigate in real world environment. This type of walkthrough is generated using images and positional data taken from cameras, GPS and LRF sensors which are assembled on a vehicle. Images are used to create panoramas which are integrated with the maps using Google Maps API. Navigation through these maps then provides the user with the required view.

1.1 Background

Google has developed Street View through integration of images taken through Optical Cameras, IR sensors and GPS information. All these sensors are placed on a vehicle, which moves across the city, takes images and combine them into 360 panoramic street views. This technology is of great military importance as well. These types of views generated during peace time may help law enforcement agencies to plan an operation in that area. This technology is available only with Google and is not available to our forces in a complete product form.

Figure 1.1 Google Street View of San Rafael, United States

Figure 1.2 Google Street View of Los Santos, United States

1.2 Problem Statement

As this technology is of great importance in generation of navigational walkthrough as well as 3D models, it can be used for efficient model generation in 2D and 3D with interactive navigational features.

This project aims to create an application which allows its users to see and navigate through the maps just as in Google Street View.

1.3 Objectives

Assembling the vehicle with the required cameras & censors, collecting data, creating 360° panoramas and creation of walkthrough are the main objectives of this project.

1.4 Organization of thesis

Rest of the thesis is organized as to provide a detail of literature review, design and development, analysis and evaluation, and future work.

CHAPTER 2

LITERATURE REVIEW

Chapter 2

Literature Review

2.1 Introduction

Google Street View is a technology that provides panoramic views of different countries of the world. Google launched its Street View on 25th May, 2007 in United States, and has since expanded. Panoramas are created using pictures taken from cameras assembled on car.

2.2 Background

Google aims to organize information and make it accessible and useful. Street level pictures taken for Street View contain a large amount of information. This project works on two principles that are central to the way Google operates. One is, for starting new projects, the hands-on equipment approach. The other is Google primarily solve really large problems. First Capturing, processing, and then providing street-level pictures at a global scale definitely qualify.

According to "*The World Factbook* ", the world contains around 50 million miles of roads, paved and unpaved, covering about 219 countries. Driving on all these roads once is equivalent to going around the globe 1,250 times. Such a mission can be daunting for anyone but Google attempted this by developing refined hardware, software, and operational processes, and persistently recuperating them. Street

View provides millions of its users with panoramic pictures captured in different cities of 20 countries across the globe.

2.3 Operations Overview

Accurately and efficiently using numerous vehicles for data-collection around the world is a core problem the Street View team has addressed. Google followed their iterative approach to engineering, and implemented and operated numerous vehicular platforms. Figure 2.1 shows several of the platforms.

Figure 2.1 Different Vehicular Platforms

In Street View's "garage phase," their experts mounted cameras, lasers, and a GPS on the roof of cars placed multiple computers in their trunks. Pictures were captured in abundance to create demos, which were so successful that it deemed adequately convincing to enter the next phase. They envisaged that the project operations would dominate overall project cost; they designed a vehicle competent of capturing everything possibly needed. The aim was to ensure not to drive anywhere more than once. A SUV was equipped with side- and front-facing laser scanners, eight high-resolution cameras two high-speed video cameras in a circular configuration, and a rack of computers recording data to an array of 20 hard drives at 500 Mbytes per second. The SUV was modified with the addition of special shock absorbers, a heavy-duty alternator and custom made axle with rotary encoders used for pose optimization. While this system enabled the engineers to collect initial data and develop end-to-end processing algorithms, it quickly became obvious that such a customization could not be done and operated at a large scale. The team, therefore, quickly shifted to a 3rd generation of vehicles, known as "lite" cars. This time the focus was on off-the-shelf components and reliability, even the image quality was kept secondary. These bare-bones vehicles had a low-resolution camera connected to a standard desktop PC with a single hard drive. Instead of relying on custom axles, they recorded wheel encoder messages from the existing antilock brake system. These vehicles proved to be very successful and provided recording of a vast amount of imagery worldwide. Primary drawback with this system was low image resolution. Thus, to enter yet another next generation design, the team developed a custom panoramic camera system dubbed R5. This system was mounted on a custom-h-inged mast, allowing the camera to be retracted when the vehicle passed under low bridges. The R5 design also allowed them to fix three laser scanners on the mast, thereby enabling the capture of 3D data all along the imagery.

These fourth generation vehicles are the reason of success and have captured the majority of pictures live in today's Street View.

Figure 2.2a Special data collection platform

Google's Street View team has designed and implemented numerous special data-collection platforms in parallel with the road vehicles. Few years back, a snowbike-based system was implemented to capture the 2010 Winter Olympics sites in Vancouver. Aside from the obvious mechanical differences, it's essentially a Trike without laser scanners. There are a number of challenges involved in recording data using moving platforms. Like hard drives are sensitive to shock, vibration, and extreme temperatures, while the vehicle is in operation and, to a lesser degree, while being shifted or shipped. Various techniques have been used by the team to minimize data loss, from custom-shipping wrapping with extra thick foam to shock-mounted disk enclosures. In extreme temperature and vibration conditions during collection of data, solid-state disk drives were used.

Figure 2.2b Special data collection platform

2.4 Navigation through Street View

From the different ways Google Maps surfaces Street View images, the 360-degree panorama is considered the most popular. Many mobile and desktop clients leverage these images to produce an immersive experience in which the user can virtually explore streets and cities.

Such an experience becomes even richer when we combine Street View imagery with other sources of data. For example, Street View could be used as an influential tool for finding local businesses, or doing a real estate search, getting driving directions etc. Building on the 3D data that is collected as Google Maps data, markers and overlays can be placed in the scene, resulting in 3D Street View images. With Google's belief to empower users, they recently opened Street View to user contributions. Now users can correct the location of businesses and places of interest just by dragging markers in Street View and snapping it to facades automatically. With this functionality, users can build map data with greater accuracy than ever before.

Figure 2.3a Google Street View

Figure 2.3b shows, users can navigate from Street View pictures to matching photos, and from there, directly within the file structure of user photos. This type of bridge between Google's images and user-contributed pictures is only a first step toward unifying navigation and browsing of geo-referenced images.

Figure 2.3b Google Street View

2.5 Conclusion

Street View has been and continues to be an stimulating adventure in global scale picture collection, processing, and serving. The idea initiating such a project and driving along streets in the world taking images of all the surroundings such as buildings and roadsides seemed impossible at first, but test and analysis showed that it was only possible with the organized effort at an affordable level, at least in those areas of the world where political systems allow it to happen, over a period of years. The wide availability of street-level image data has proved to be very popular with users, giving useful information that was previously never available.

CHAPTER 3

DESIGN & DEVELOPMENT

Chapter 3

Design & Development

3.1 Introduction

Panoramic Street View is designed and developed to get results in accordance of the Google Street View. This chapter explains the overall architecture and implementation of the system*.

3.2 Architecture**

System uses MVC architecture model. Controller handles change in view in the walkthrough. User actions are handled by the Controller and directed to the Model. Based on these actions, Model queries the database and gets the nearest panorama for the current location of the user in the map. View is notified of this change in state of Model and it updates the walkthrough.

The reason for using MVC architectural pattern is that Panoramic Street View is an interactive web application and MVC is one of the architectural patterns for Interactive System. Also, MVC pattern separates application logic from data and presentation (view). In Panoramic Street View, application logic works separate from data and presentation. This architecture is explained using Figure 3.1.

**** Software Requirement and Specification document of this project is attached as ANX A***

***** Detailed Design of this project is attached as ANX-B***

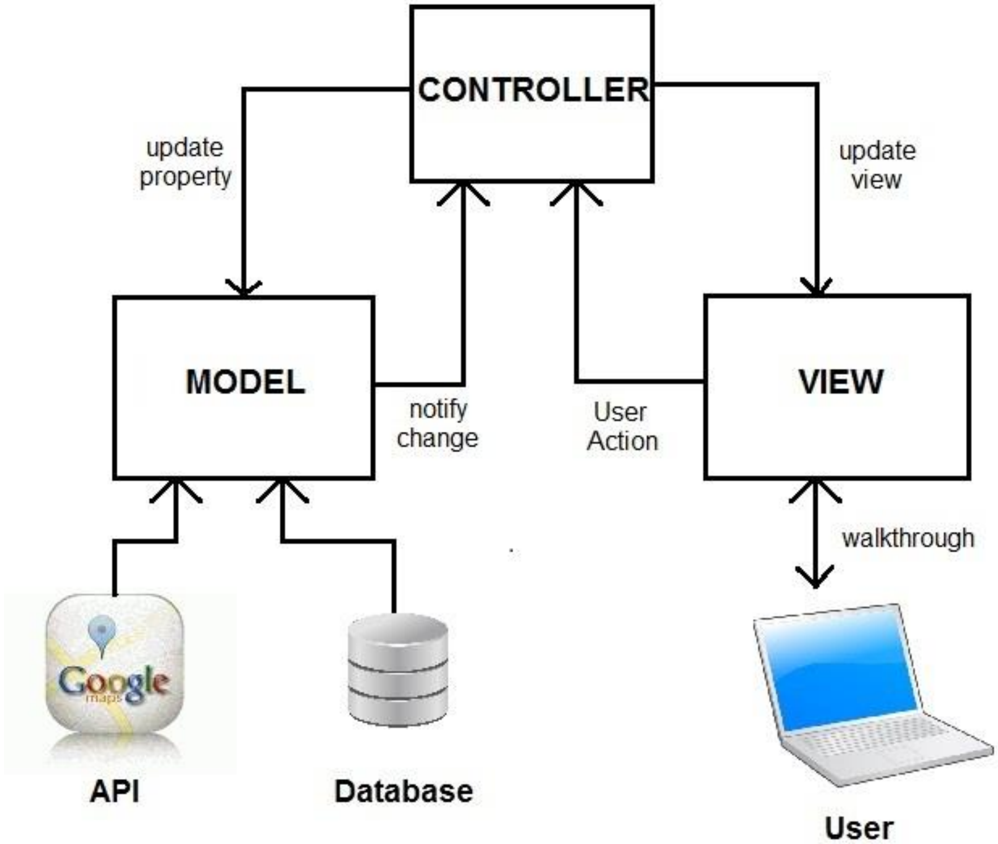


Figure 3.1 System Architecture

3.3 Implementation

This project involves the development of software and integration of hardware as well. The development follows a two phase process. One is data collection and the other is street view generation.

3.3.1 Assembling Vehicle & Data Collection

To collect data for street view, a 4-wheel drive vehicle is assembled with camera mount, GPS sensor and Laser Range Finder (LRF). Camera mount (Figure 3.2) consists of 8 Logitech QuickCam Sphere AF cameras. These cameras have high shutter speed which enables us to get images in a moving vehicle without any distortion in the image (blur effect). The camera mount is designed as such to cover 360° surrounding area so that images can be captured to create 360° panoramas. These cameras can be interfaced with computer using USB 2.0 ports. 2 x 7-port USB 2.0 HUBs are used to connect 8 cameras with the computer.

Figure 3.2 Camera Mount

A custom windows application is implemented to collect continuous images from cameras while driving the vehicle. Figure 3.3(a, b) shows the GUI of this application.

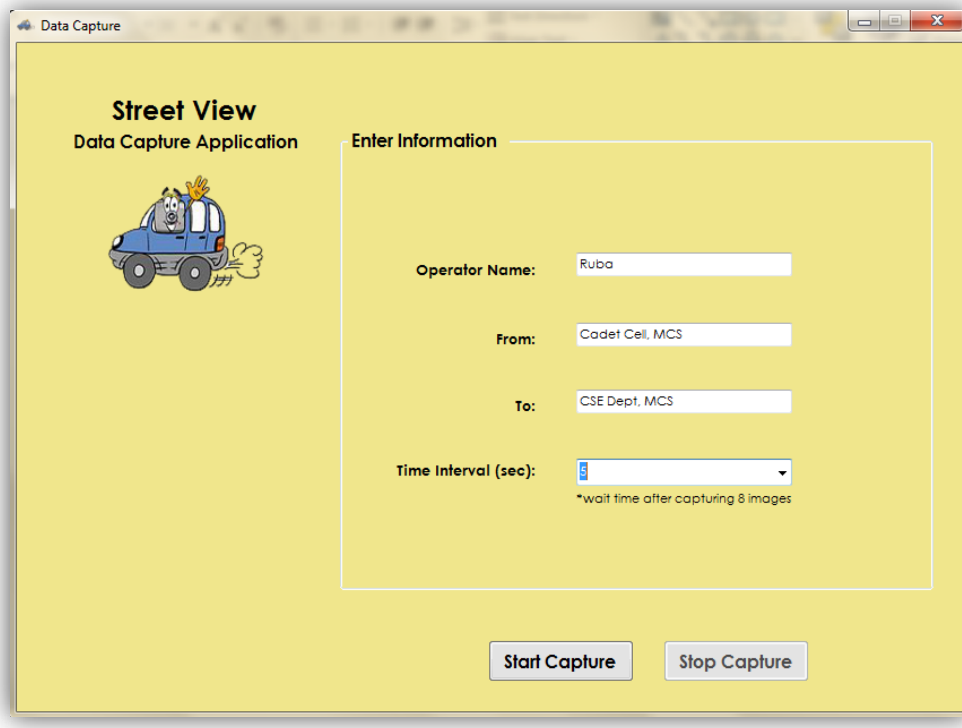


Figure 3.3a GUI Data Capture Application (First Screen)

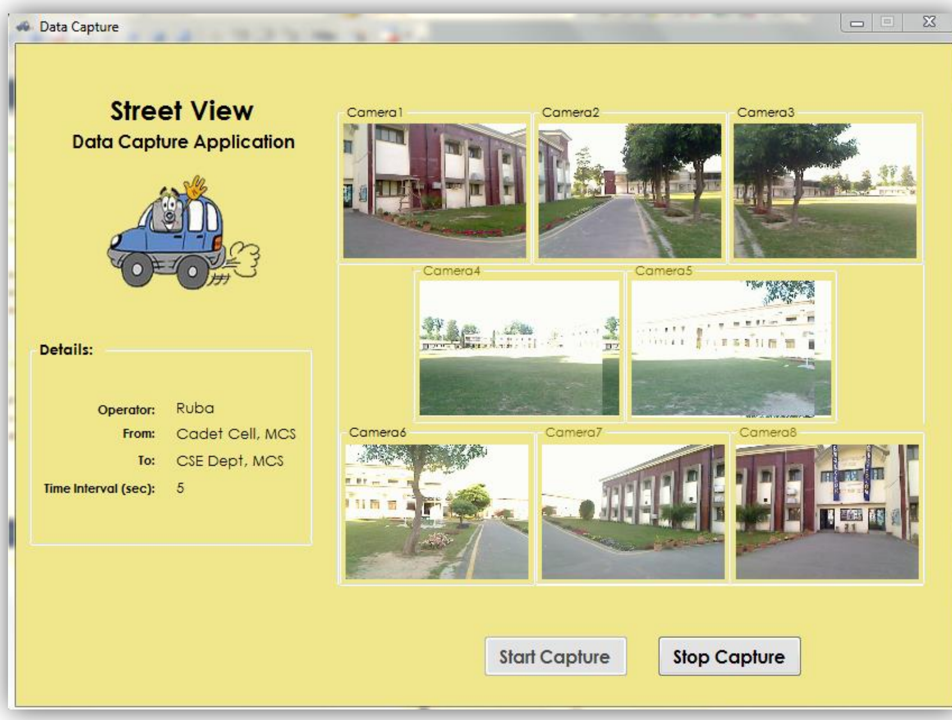


Figure 3.3b GUI Data Capture Application (Data Capturing screen)

This application automatically captures images from 8 cameras and saves them with time stamp. User has to enter some information on first screen (Figure 3.3a), his name, route in which he will drive the vehicle and capture data, and time interval. Time interval is the gap after capturing 8 images from all 8 cameras; it can be selected as 5, 8 or 10 seconds. After entering this information, user can click on Start Capture button. This will show the Data Capture Screen (Figure 3.3b) and system will start capturing images continuously until Stop Capture button is clicked. The Data Capturing Screen continuously shows the images in real time that are being captured by 8 cameras. Also, information entered in First Screen is shown in the left bottom and this information is saved in a text file for further record.

To collect the GPS information, GPS sensor of an Android mobile phone is used. A custom Android application is implemented that collects longitude & latitude coordinates and save it in a text file which is then used in the second phase of street view generation.

Laser Range Finder is used manually to collect and save distance information.

Figure 3.4 Laser Range Finder (LRF)

A portable computer is placed in the vehicle for the collection of data. The vehicle assembled with these sensors is then driven in the area whose Street View is to be generated.

Figure 3.5 describes the complete hardware interfacing of the system.

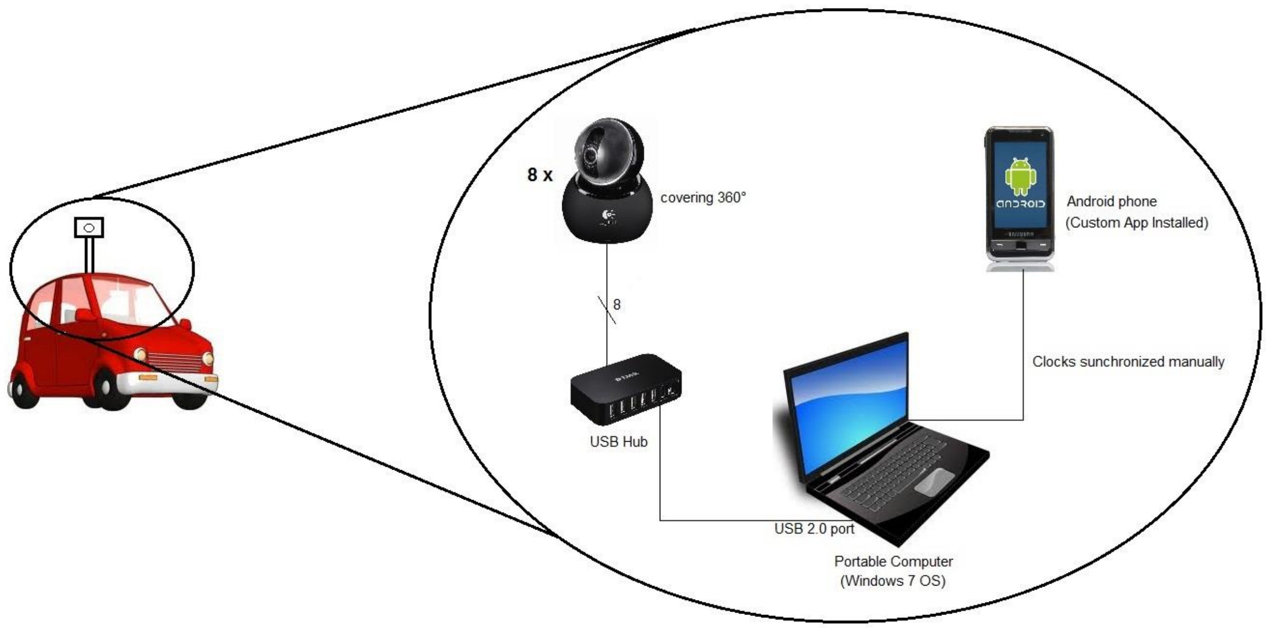


Figure 3.5 Hardware Interfacing

3.3.2 Street View Generation

Street View generation involves processing of data gathered using the vehicle. This includes creation of 360° panoramas and integration of these panoramas and GPS information with Google maps.

3.3.2.1 360° Panorama Creation

360° panorama is one that covers the complete surrounding area. This is created by stitching the overlapping images using image stitching algorithm or an off-the-shelf image stitching software. Microsoft Image Composite Editor (MS ICE) is an image stitching software that creates high definition panoramas efficiently. This software is used for panorama creation. Figure 3.6 & 3.7 shows the GUI of MS ICE and a 360° panorama created using this software respectively.

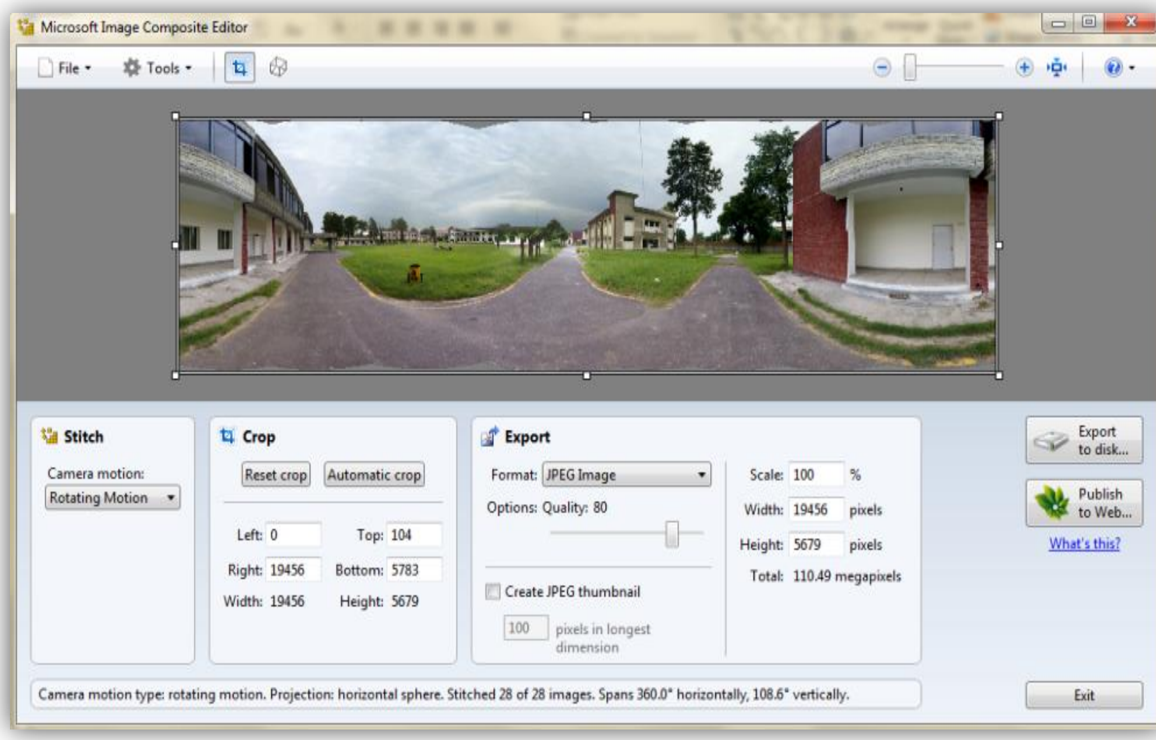


Figure 3.6 GUI of Microsoft Image Composite Editor

Figure 3.7 360° Panorama created using MS ICE

3.3.2.1.1 How MS ICE works?

Microsoft Image Composite Editor Stitch images in two steps. First it detects keypoints in the overlapping images and then compute homography.

1. Detect keypoints

There are local features in image which are considered as invariant for scaling and rotation of image. These features are needed to be extracted for stitching. There are 3 steps involved in extraction of above mentioned local features.

a. Scale-space extrema detection

In this step, interest points are found. To find these points all scales and locations are searched. For this purpose a multi-scale pyramid of Difference of Gaussian images is generated.

Equation for a Gaussian blurred image:

$$L(x,y,б) = G(x,y,б) * I(x,y) \qquad (3.1)$$

Where:

x,y - pixel position,

б - Denotes blur level,

L(x,y,б) - blurred image,

G(x,y,б) - Gaussian blurring function

I(x,y) - image being blurred.

$$G(x,y,б)=1/(2πб^2)\exp^{-(x^2 + y^2)/б^2} \quad (3.2)$$

Each pyramid's level consists of:

$$D(x,y,б) = L(x,y,kб)-L(x,y,б) \quad (3.3)$$

b. Key point localization

To determine accurate position of each candidate keypoint, interpolation is used with nearby data. The candidate point as origin is used for interpolation using Taylor expansion of difference of Gaussian scale space function $D(x, y, \sigma)$

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x \quad (3.4)$$

c. Orientation Assignment

Equation 3.5 and 3.6 are used for orientation assignment.

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (3.5)$$

$$\theta(x, y) = a\tan 2(L(x, y+1) - L(x, y-1), L(x+1, y) - L(x-1, y)) \quad (3.6)$$

$L(x, y, \sigma)$: Gaussian-smoothed image

$L(x, y)$: Image sample

2. Computing homography

Homography relates any two images (in the same planar space). It is computed for image registration. Registration is the conversion of different sets of data to a single coordinate system. For stitching purposes images are registered so that they can be integrated to form a panorama or mosaic.

It is computed using Direct Linear Transformation:

$$X' = HX \tag{3.7}$$

Where:

$$X' = \begin{bmatrix} w'u' \\ w'v' \\ w' \end{bmatrix}$$

$$H = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}$$

$$\begin{bmatrix} -u & -v & -1 & 0 & 0 & 0 & uu' & vu' & u' \\ 0 & 0 & 0 & -u & -v & -1 & uv' & vv' & v' \end{bmatrix} h = 0 \tag{3.8}$$

Where:

$$h = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ h_9 \end{bmatrix}$$

$$\begin{bmatrix} -u_1 & -v_1 & -1 & 0 & 0 & 0 & u_1u_1' & v_1u_1' & u_1' \\ 0 & 0 & 0 & -u_1 & -v_1 & -1 & u_1v_1' & v_1v_1' & v_1' \\ & & & & \vdots & & & & \\ 0 & 0 & 0 & -u_n & -v_n & -1 & u_nv_n' & v_nv_n' & v_n' \end{bmatrix} \mathbf{h} = \mathbf{0} \Rightarrow \mathbf{Ah} = \mathbf{0} \qquad (3.9)$$

Apply SVD:

$$UDV^T = A \qquad (3.10)$$

$$h = V_{smallest} \qquad (3.11)$$

(Column of V corr. to smallest singular value)

3.3.2.2 Walkthrough generation

These panoramas are then integrated with Google maps using Google Maps API. Figure 3.8 and 3.9 shows a code snippet of this integration and the walkthrough in end application respectively.

```
case "cs_dept1":
  // Description of this point.
  streetViewPanoramaData["location"] = {
    pano: 'cs_dept1',
    description: "out side of department",
    latLng: new google.maps.LatLng(33.578229271149276, 73.06172132492065)
  };
  return streetViewPanoramaData;

case "cs_dept2":
  streetViewPanoramaData["location"] = {
    pano: 'cs_dept2',
    description: "inside of cs dept",
    latLng: new google.maps.LatLng(33.57815105896953, 73.06156575679779)
  };
```

Figure 3.8 Code snippet of integrating panoramas and long, lat coordinates with Google maps

Figure 3.9 Walkthrough

Figure 3.10 shows the software interfacing of the system.

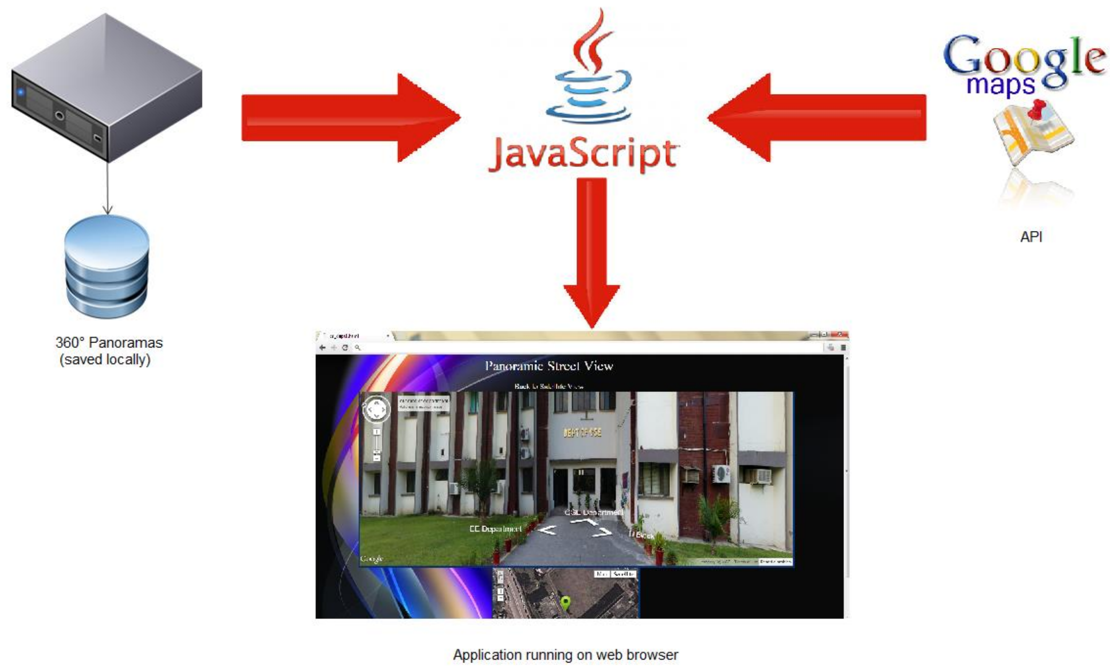


Figure 3.10 Software Interfacing

CHAPTER 4

ANALYSIS & EVALUATION

Chapter 4

Analysis & Evaluation

4.1 Introduction

Testing is an important part of system analysis and evaluation to ensure that system is in accordance of specified requirements*. This chapter includes the description and outcomes of test cases which were used to test the main features of the system.

4.2 Testing of main features

4.2.1 Test case: Collection of Data

Test Case id: 1

Summary:

8 images and long, lat coordinates will be acquired using desktop application, GPRS and 8 cameras. These images will be used to create panoramas.

Pre-condition/Test data:

1) 8 cameras should be mounted on a car, covering 360°.
2) Car shall move at speed of not more than 20km/hr.
3) All cameras should be in working condition and synchronized.
4) Destiny location for images must be specified with sufficient empty storage.

Software Requirement and Specification document of this project is attached as ANX A

Test Steps:

1) Start application. Enter required information and select time interval between capturing of images per coordinate.
2) Entered 5 seconds as time interval.
3) Clicked on start capturing.
4) Clicked on stop capturing after collecting images of all required coordinates.

Expected Results:

There should be 8 clear images for each coordinate with 30% overlap between each neighbor image and one log file in form of .txt having data of longitude and latitude.

Actual Results:

After executing all steps, data was obtained as expected i.e. 8 images per coordinate with 30% overlapping between images.

Test Case Status: Pass

4.2.2 Test Case: Navigational Walkthrough

Test Case id: 2

Test Case Title: Navigation through Walkthrough

Summary:

A web based application is developed which allows users to navigate through street view.

Pre-condition/Test data:

1) Internet should be available.

2) Any latest Web browser should be available (Firefox, Chrome, IE)
3) Panoramas for all coordinates should be available which are specified in application.

Test Steps:

1) Started application using Google chrome.
2) On screen there will be satellite view and markers showing places where street view is available.
3) Clicked on marker labeled as CSE department. Street view for that coordinate appeared.
4) Checked all navigation links of CSE department by clicking on them.
5) Checked all coordinates moving from one place to another.

Expected Results:

All coordinates must have 360° panorama and all panoramas must be referenced with accurate direction.

Actual Results:

All coordinates have 360° panorama and all panoramas are referenced with accurate direction.

Test Case Status: Pass

Conclusion

The test cases explained depicts the correct working of the system. Successful implementation of the system is done in accordance with research & development of Google.

CHAPTER 5

FUTURE WORK & CONCLUSION

Chapter 5

Future Work & Conclusion

Panoramic Street View has many useful applications not only in navigation but in other fields as well.

An important application is model creation. Military Forces can develop models of training areas. With some change in hardware design and integration of surveillance module Military Forces can also create models of target areas.

The data collected for the creation of Street View can be used to extract useful information about an area using Data Mining methodologies. This useful information includes number of trees is an area, number of houses etc.

A useful commercial application of this project is the creation of walkthrough of housing societies. Developing housing societies can place their Street View on their website for advertisement purposes.

The complete system developed in this project has many extensions. There is a wide scope of applications that can be implemented using this system. Street View can be used to create 3D models as needed.

This work serves as an inspiration for others to develop more advanced modules integrated with Street View end application and its hardware components.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] https://developers.google.com/maps/documentation/streetview/. Google Street View, 20th June, 2012.

[2] http://maps.google.com/intl/en/help/maps/streetview/#utm_campaign=en&utm_medium=van&utmsource=en-van-na-us-gns-svn. Google Maps – Street View, 20th June, 2012.

[3] Fazal, Fawad, 3Dreams, 2011

[4] http://maps.google.com/help/maps/streetview/learn/turning-photos-into-street-view.html. Google Maps – Street View, 30th June, 2012.

[5] http://googlemaps.googlermania.com/google_maps_api_v3/en/custom_stre-etview/index.html. Google Maps Playground, 2nd July, 2012.

[6] Roy D. Ragsdale. DIY Street View, October 2009

[7] Dragomir Anguelov, Carole Dulong, Daniel Filip, Christian Frueh, Stéphane Lafon, Richard Lyon, Abhijit Ogale, Luc Vincent, and Josh Weaver, Google Street View: Capturing the world at street level, IEEE computer society, June 2010.

[8] http://blog.mikecouturier.com/2009/12/google-street-view-with-google-maps_27.html, Google Street View and Google Maps Interaction – Sample Showcase, 10th July, 2012.

[9] http://googlemaps.googlermania.com/google_maps_api_v3/en/map_example49.html, Synchronize Street View with Map, 14th July, 2012

[10] https://developers.google.com/maps/documentation/javascript/reference, Google Maps Javascript API V3 Reference, 14th July, 2012

[11] http://techcrunch.com/2013/03/08/inside-google-street-view-from-larry-pages-car-to-the-depths-of-the-grand-canyon/, Inside Google Street View, 10th March, 2013.

[12] http://research.google.com/archive/papers/cbprivacy_iccv09.pdf. A. Frome et al., “Large-Scale Privacy Protection in Google Street View,” Proc. 12th IEEE Int’l Conf. Computer Vision (ICCV 09), IEEE Press, 2009;

ANX-A

Software Requirement Specification (SRS)

ANX-A

Software Requirement Specification (SRS)

1. Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of Panoramic Street View. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external input.

1.2 Document Conventions

Every requirement statement in this document has equal priority and importance as this is an interactive system and every aspect has equal impact on the outcome.

1.3 Intended Audience and Reading Suggestions

This document is intended for developers, project managers, project supervisor and UG project evaluation committee.

1.4 Product Scope

Panoramic Street View is an interactive system which will provide the users with the map of MCS and allow navigation just as in real world environment. Panoramas will be created to create the required navigational view. A vehicle will be equipped with cameras, GPS and LRF sensor to gather the required data for the creation of walkthrough.

2. Overall Description

2.1 Product Perspective

Google has developed Street View through integration of images in Google maps. This technology is of great military importance as well. These types of views generated during peace time may help the Military forces to plan an operation in that area. This technology is available only with Google and is not available to our forces in a complete product form.

This project aims in developing a framework by integrating all the sensors mounted at a vehicle/car and using Google's open source APIs.

2.2 Product Functions

Application will perform following functionalities

- Create panorama
- Develop walkthroughs
- Let user to navigate through walkthrough maps.

Top level data flow diagram is given in Figure 1.

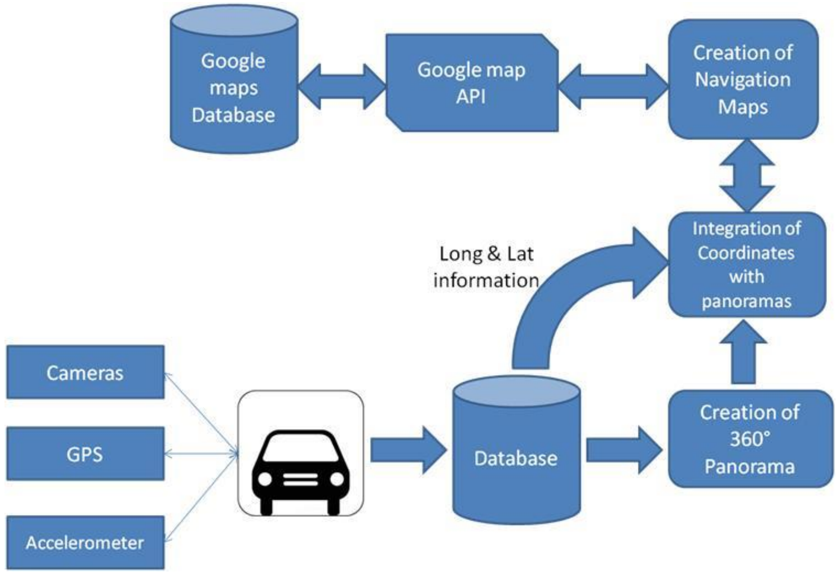


Figure 1 Block Diagram

2.3 User Classes and Characteristics

User classes of this system are:

- **Administrator**

  This user class can configure and control the software and hardware, both, of the system and has no limitation over the system's features.

- **General User**

  This user class can use the application running on any browser and can navigate through maps using the application's walkthrough.

2.4 Operating Environment

Environment required for application will be following:

- It should run on any web browser running on any Operating System.

- Minimum 500 KBps internet speed is required for accurate functioning of the web application.

2.5 Design and Implementation Constraints

The hardware constraints are:

- Cameras
- GPS sensor
- LRF
- Large Capacity Storage Device

Language to be used:

- JavaScript

2.6 User Documentation

The user documentation will be designed after the system is practically finalized, because since no such application currently exists, there is no documentation available for this. Application's functionality is understood by the user by going through the interface of the application, so thorough documentation is not needed for the application. But, for the hardware specifications and configuration documentation will be done thoroughly.

2.7 Assumptions and Dependencies

This project uses the Google Maps API, so it is assumed that Google Maps will be available to the system when it will be in operation i.e. internet connectivity and access key to API is needed. If any of these assumptions is incorrect, the system will not function at optimal performance level.

3. External Interface Requirements

3.1 User Interfaces

The system will have a graphical user interface. User interface will be much like a web page which can be viewed using a web browser.

3.2 Hardware Interfaces

The system requires cameras and sensors i.e. GPS and LRF. Images and sensor data will be stored on a separate storage device.

3.3 Software Interfaces

The system will operate on any web browser running on any operating system. Google Maps API will also be accessed to get the maps and integrate panoramas into it to form a walkthrough.

3.4 Communications Interfaces

The system will have some communications function to access the API. It will use communication protocol such as HTTP.

4. System Features

All the features of the system are of equal priority. The major features are as follows:

4.1 Collecting Data

4.1.1 Description and Priority

Data for the system, such as images, will be gathered using a vehicle assembled with cameras, GPS and LRF sensors. This data will then be used to create panoramas. As mentioned earlier, all features are of equal priority and are essential for correct functioning of the system.

4.1.2 Stimulus/Response Sequences

1. User (Administrator) will drive the vehicle in the area whose street view is to be created.

2. Cameras and sensors, assembled on the vehicle, will record images and geographical data.

3. This data will be stored in a separate storage device.

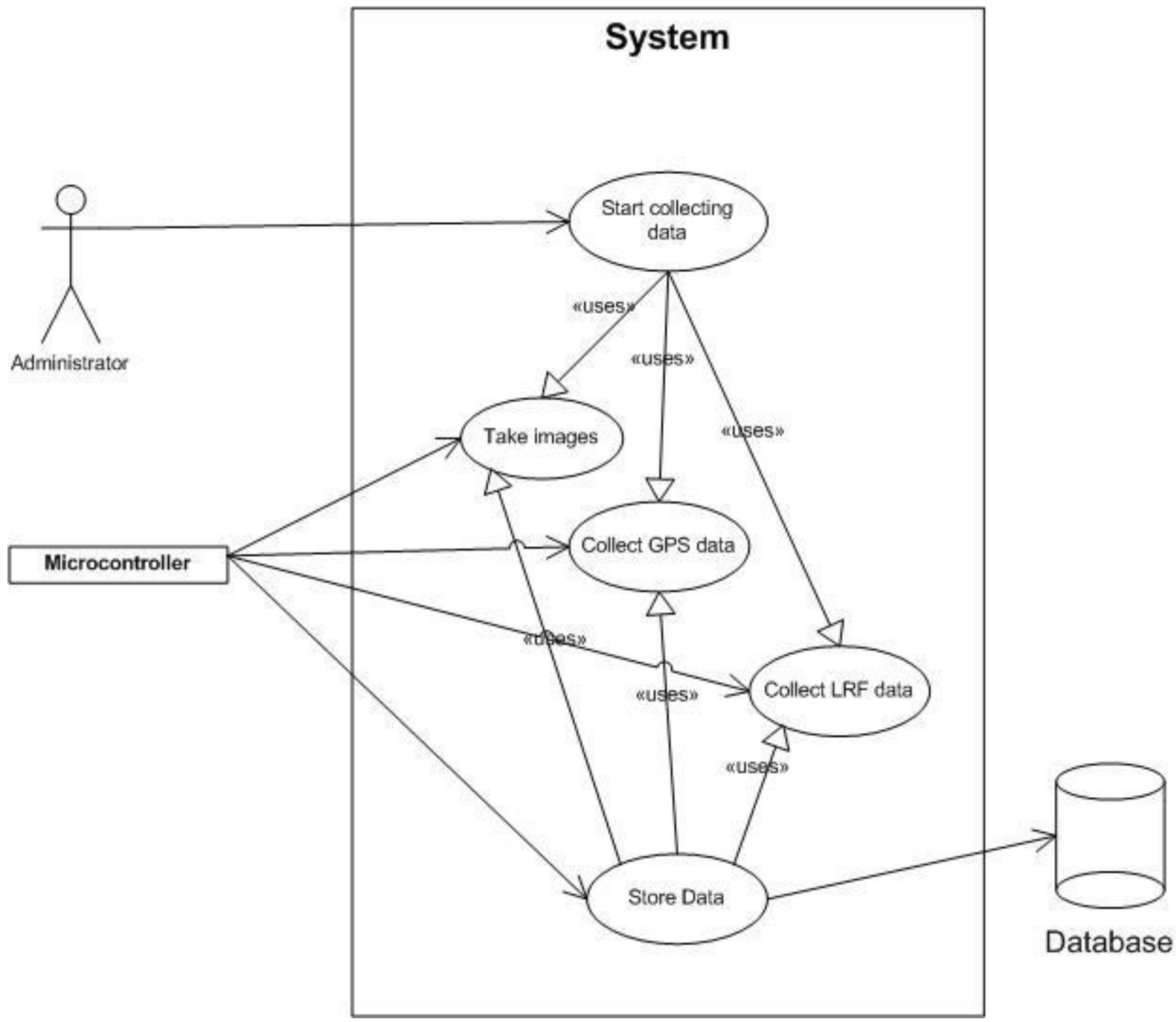
4.1.3 Functional Requirements

REQ-1: Vehicle should be equipped with the required sensors.

REQ-2: Vehicle should collect and store data in a structured form.

REQ-3: All the cameras and sensors should be synchronized using the microcontroller.

4.1.4 Use Case Diagram



4.2 Creating Panoramas

4.2.1 Description and Priority

Panoramas will be created using the data get from the camera and sensors assembled on a vehicle.

4.2.2 Stimulus/Response Sequences

1. User (Administrator) selects the images from the storage device and gives to the system.

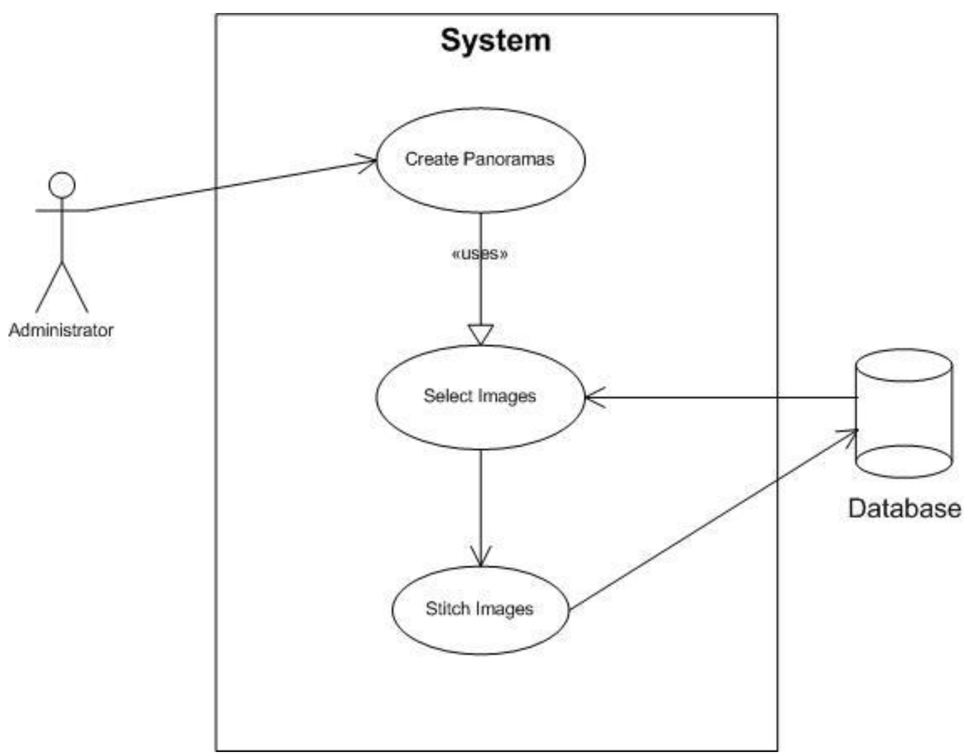
2. System will then stitch the images together to create panoramas.

4.2.3 Functional Requirements

REQ-1: System should be able to read the images stored in the storage device.

REQ-2: System should create panoramas of the selected images.

4.2.4 Use Case Diagram



4.3 Create Walkthrough

4.3.1 Description and Priority

Panoramas will be integrated with the map using Google Maps API and a walkthrough will allow the user to navigate through maps like real world environment.

4.3.2 Stimulus/Response Sequences

1. Administrator will integrate panoramas with maps, using GPS and LRF information. This will be done using Google Maps API.

2. Walkthrough will be created using JavaScript and HTML.

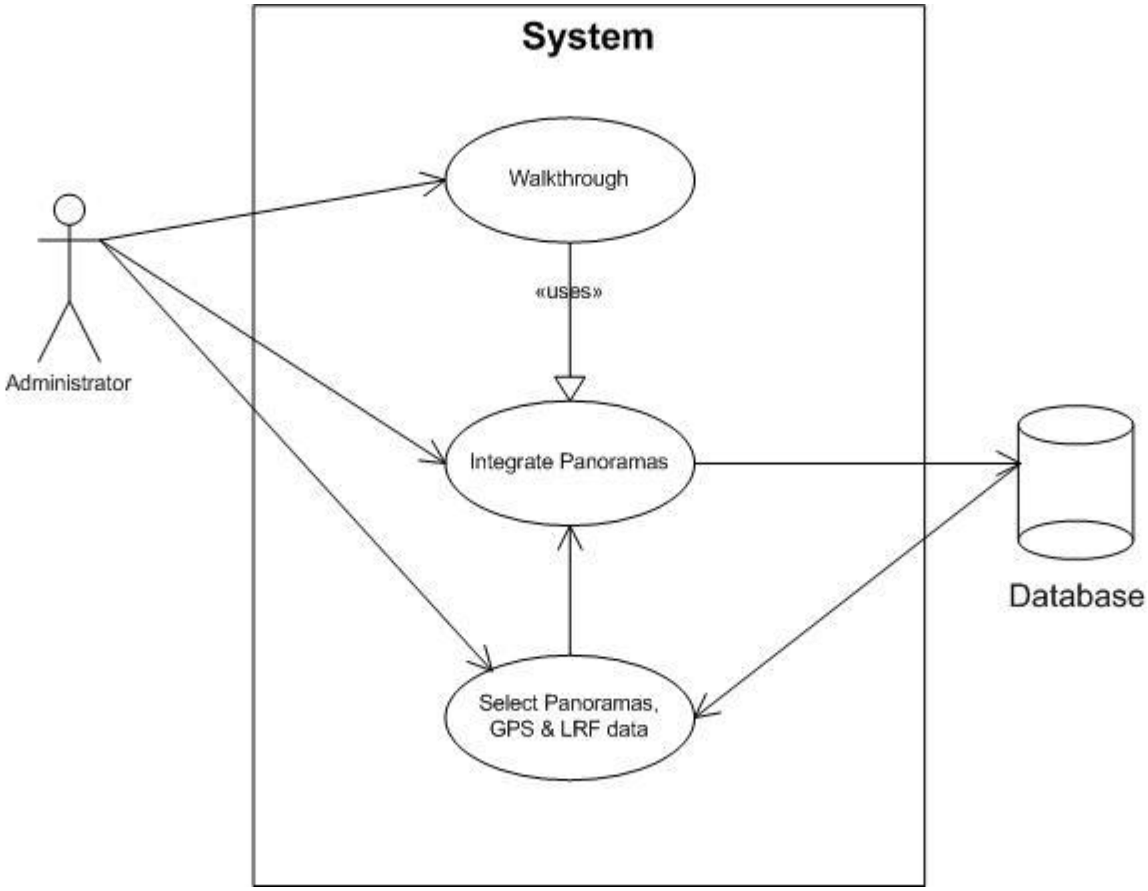
4.3.3 Functional Requirements

REQ-1: System should integrate panoramas onto the maps.

REQ-2: System should have a user interface with walkthrough for the navigation.

REQ-3: Walkthrough should provide user the real world environment view.

4.3.4 Use Case Diagram



4.4 Navigation

4.4.1 Description and Priority

Navigation functionality will be provided by the system in the form of a web application. This web application will be a simple web page which will allow the user to navigate through maps just as in real world environment.

4.4.2 Stimulus/Response Sequences

1. User will run the application using the provided HTML page.

2. The page will show the street view of MCS.

3. User will then be able to navigate through the street view of MCS.

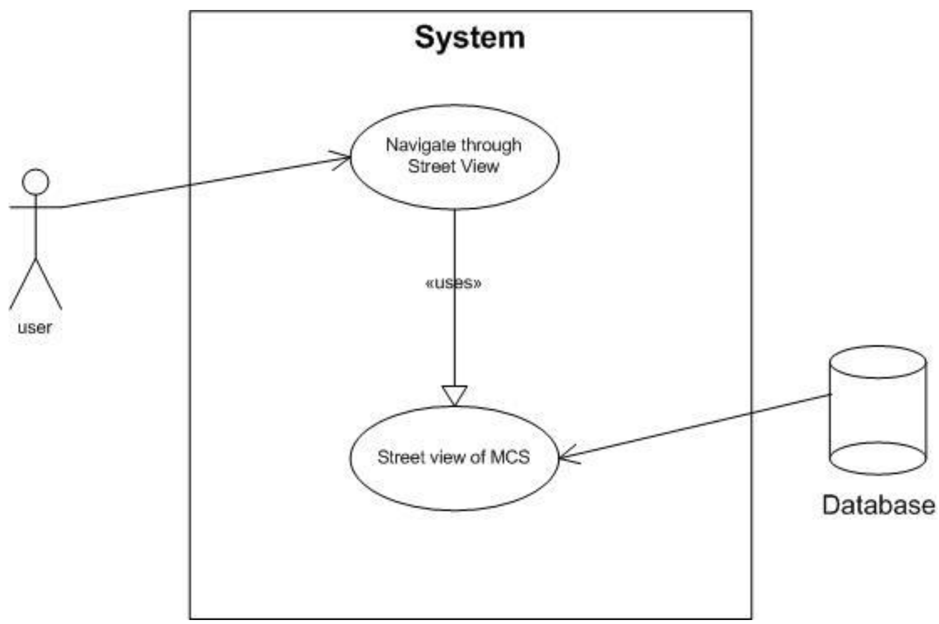
4.4.3 Functional Requirements

REQ-1: System should provide the street view of MCS for the user.

REQ-2: System should have a user interface with walkthrough for the navigation.

REQ-3: Walkthrough should provide user the real world environment view.

4.4.4 Use Case Diagram



5. Other Nonfunctional Requirements

5.1 Performance Requirements

1. The application should not exceed the Google's threshold of 100M API calls per day.

2. The application must load the panoramic view in less than 2 seconds.

5.2 Safety Requirements

1. System must not violate any law or the rights of any individual or entity.
2. System must not include functionality that proxies, requests or collects confidential information.
3. System must not circumvent (or claim to circumvent) intended limitations on core Google Maps API features and functionality.

4. Before data collection of public area, NOC must be acquired by local authorities.

5.3 Security Requirements

A. **Responsibility for Content** – System is responsible for all content of and within it.

B. **Confidentiality** – System must maintain the confidentiality of the data gathered for the creation of walkthrough.

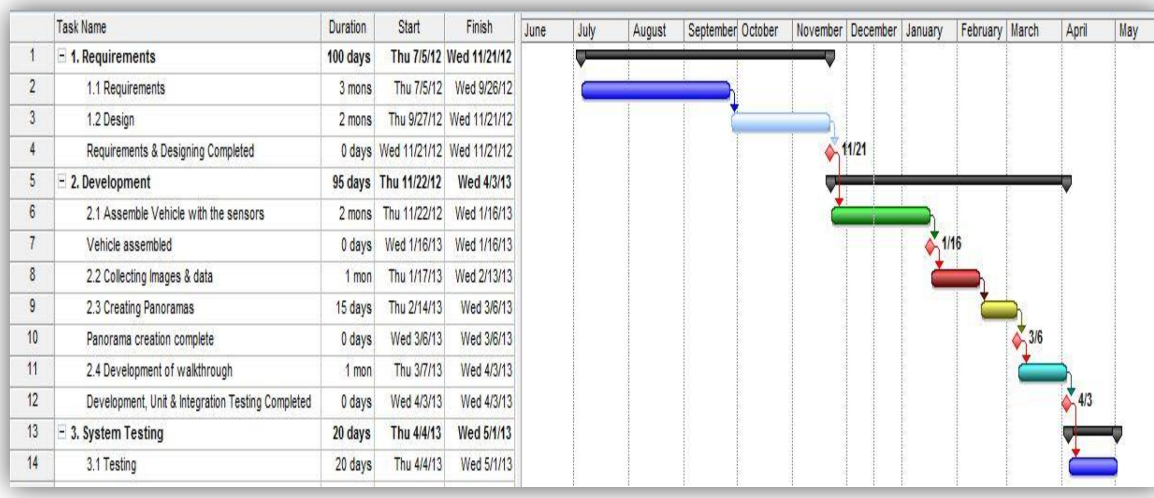
C. **Privacy –** Data base must be accessible only by authorized persons.

5.4 Software Quality Attributes

1. System should be flexible, so that it can accommodate future modifications.
2. System should be reliable in the sense that it should provide the accurate content to the users.

System should respect privacy e.g. it should not expose the identities of the people visible in the panorama.

6. Project Plan



	Task Name	Duration	Start	Finish
1	1. Requirements	100 days	Thu 7/5/12	Wed 11/21/12
2	1.1 Requirements	3 mons	Thu 7/5/12	Wed 9/26/12
3	1.2 Design	2 mons	Thu 9/27/12	Wed 11/21/12
4	Requirements & Designing Completed	0 days	Wed 11/21/12	Wed 11/21/12
5	2. Development	95 days	Thu 11/22/12	Wed 4/3/13
6	2.1 Assemble Vehicle with the sensors	2 mons	Thu 11/22/12	Wed 1/16/13
7	Vehicle assembled	0 days	Wed 1/16/13	Wed 1/16/13
8	2.2 Collecting Images & data	1 mon	Thu 1/17/13	Wed 2/13/13
9	2.3 Creating Panoramas	15 days	Thu 2/14/13	Wed 3/6/13
10	Panorama creation complete	0 days	Wed 3/6/13	Wed 3/6/13
11	2.4 Development of walkthrough	1 mon	Thu 3/7/13	Wed 4/3/13
12	Development, Unit & Integration Testing Completed	0 days	Wed 4/3/13	Wed 4/3/13
13	3. System Testing	20 days	Thu 4/4/13	Wed 5/1/13
14	3.1 Testing	20 days	Thu 4/4/13	Wed 5/1/13

Figure 2 Gantt chart

ANX-B

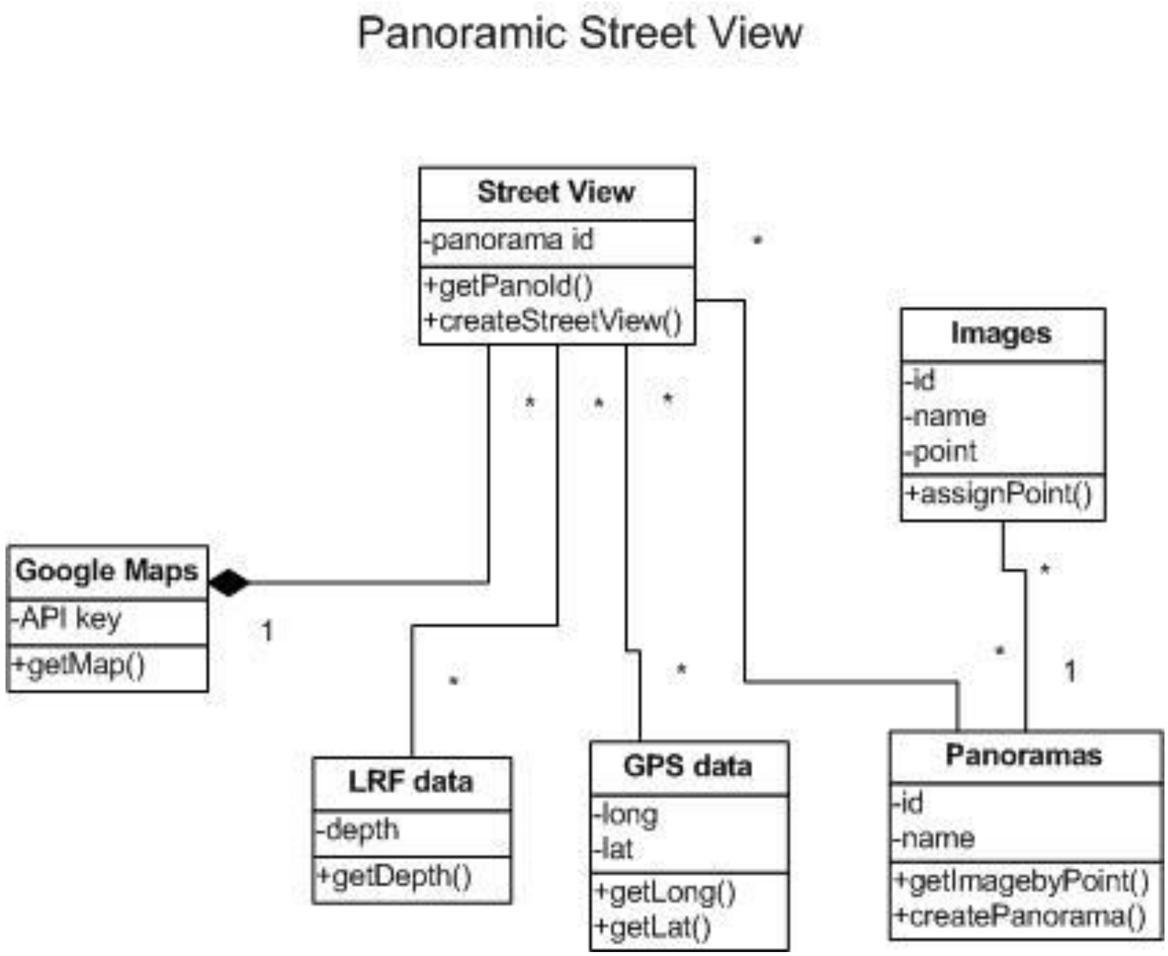
Detailed Design

ANX-B

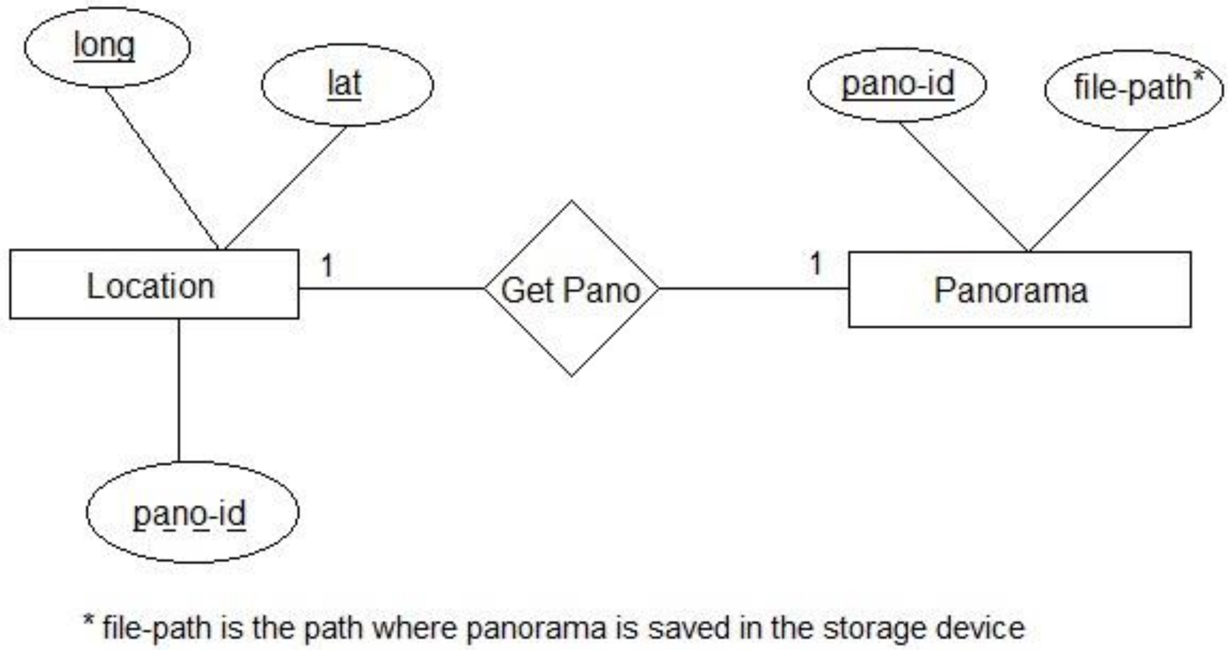
Detailed Design

1. Data Design

1.1 Class Diagram

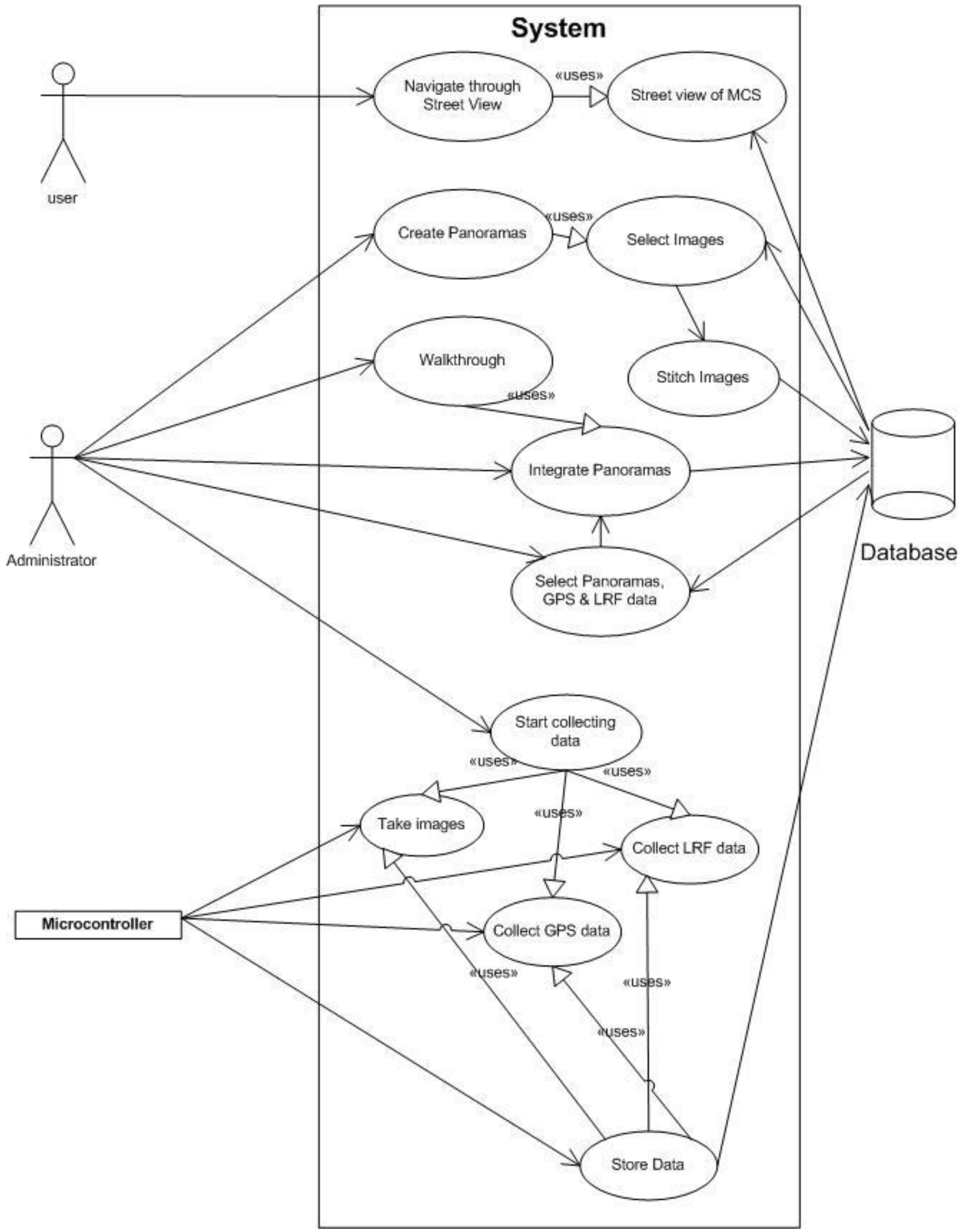


1.2 ER Diagram



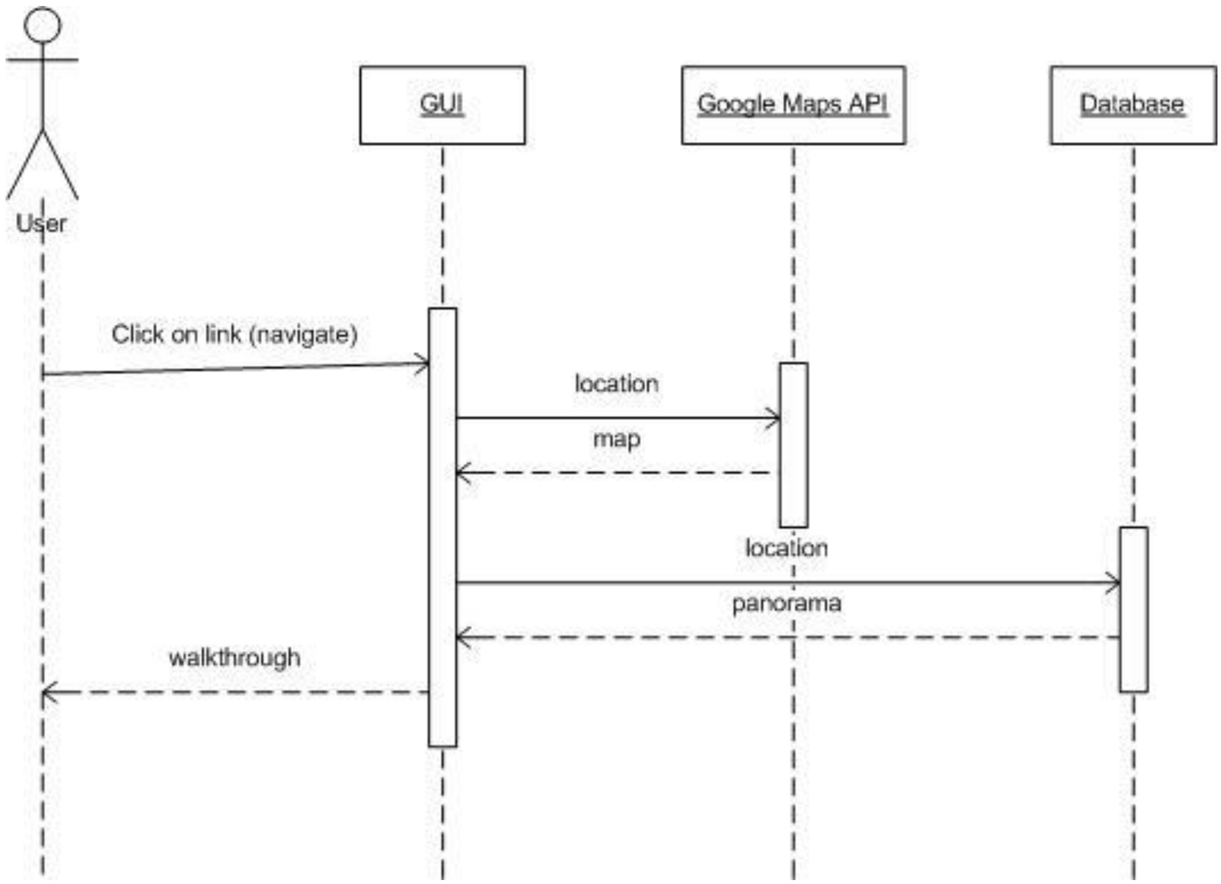
Location and Panorama are two tables of the database. Location stores the long & lat values gathered from sensors assembled on the vehicle. Pano-id links Panorama and Location tables to get the panorama of any location.

1.3 Use Case Diagram:



2. Logical View

2.1 Sequence Diagram:

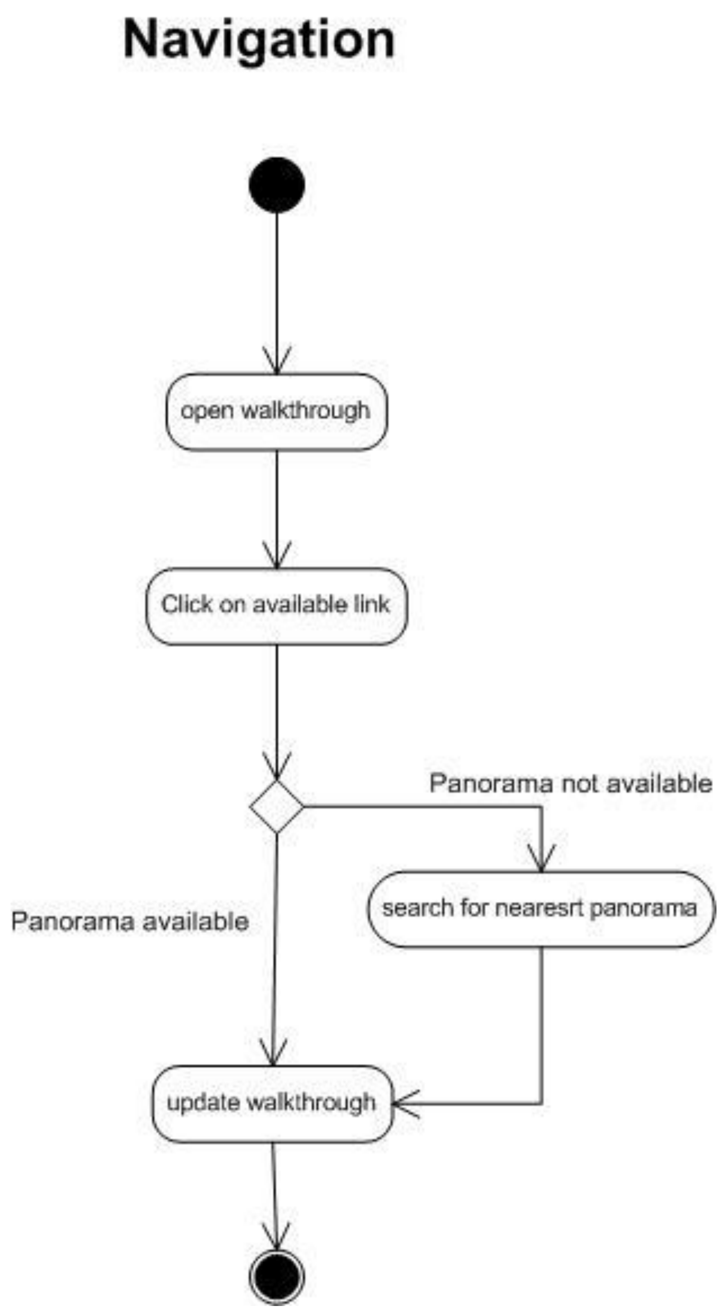


Just as user navigate i.e., click on the link available on the walkthrough, System is informed about the change in the location. For this location, map and panorama are get from Google maps API and database respectively. This fetched data is then used to update the walkthrough.

3. Dynamic View

3.1 Activity Diagram

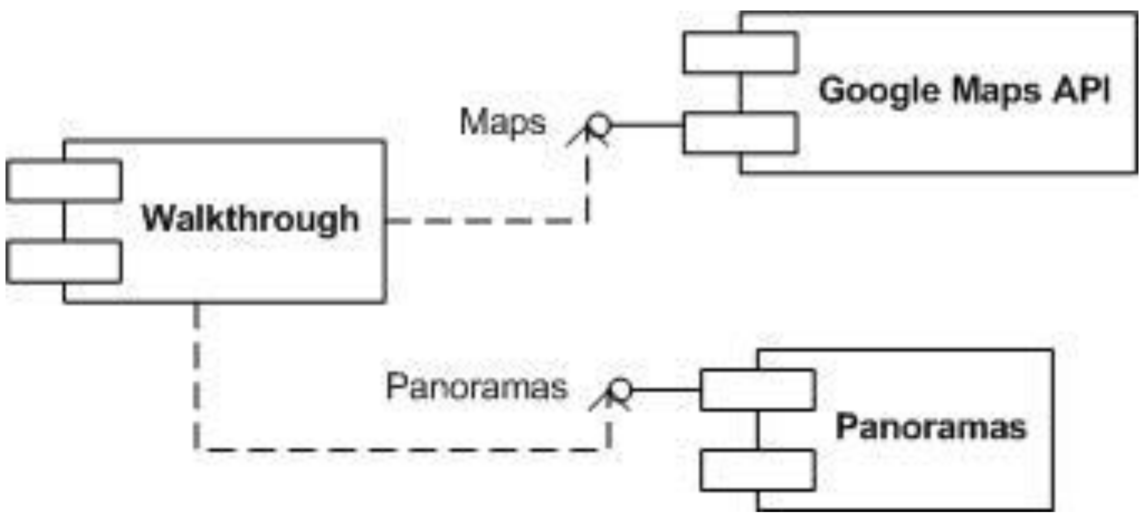
This activity diagram describes the navigation feature of the system:



4. Implementation View

4.1 Component Diagram

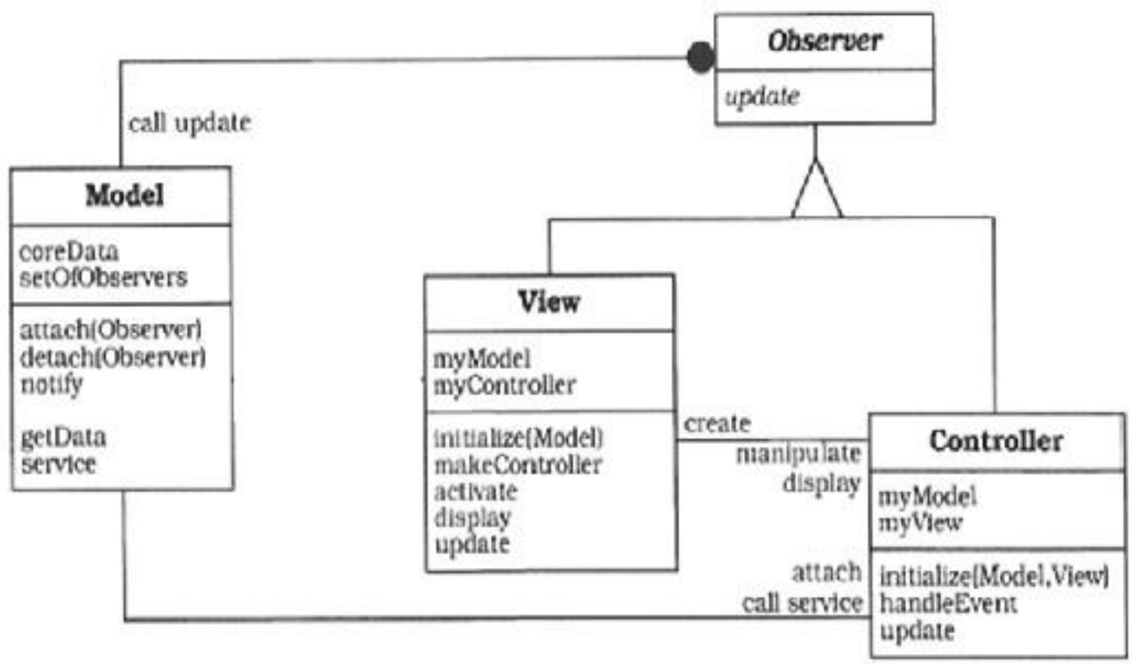
Component diagram shows the three main components of the system. Walkthrough consist of the map and panoramas integrated together using the Google Maps API.



5. Design Pattern

System uses **Observer** design pattern. The intent of the Observer pattern is to define dependency between objects so that when one object changes state, all its dependents are notified and updated automatically. In Panoramic Street View, View and Controller observes the state of the Model and update automatically whenever a change occurs in the Model based on user actions.

The diagram explains this description:



ANX-C

USER GUIDE - DATA CAPTURE

<u>**ANX C**</u>

User Guide – Data Capture

Panoramic Street View's Data Capture application is easy to use. Its easily understandable GUI helps its user to capture data efficiently.

1. Place camera mount on a 4 wheel drive vehicle such that cameras height from car's roof should be 4 feet.
2. Connect cameras, using 2 x 7-port USB hubs, with the laptop.
3. Install driver for Logitech cameras using lws251.exe.
4. Make sure all the cameras are working properly.

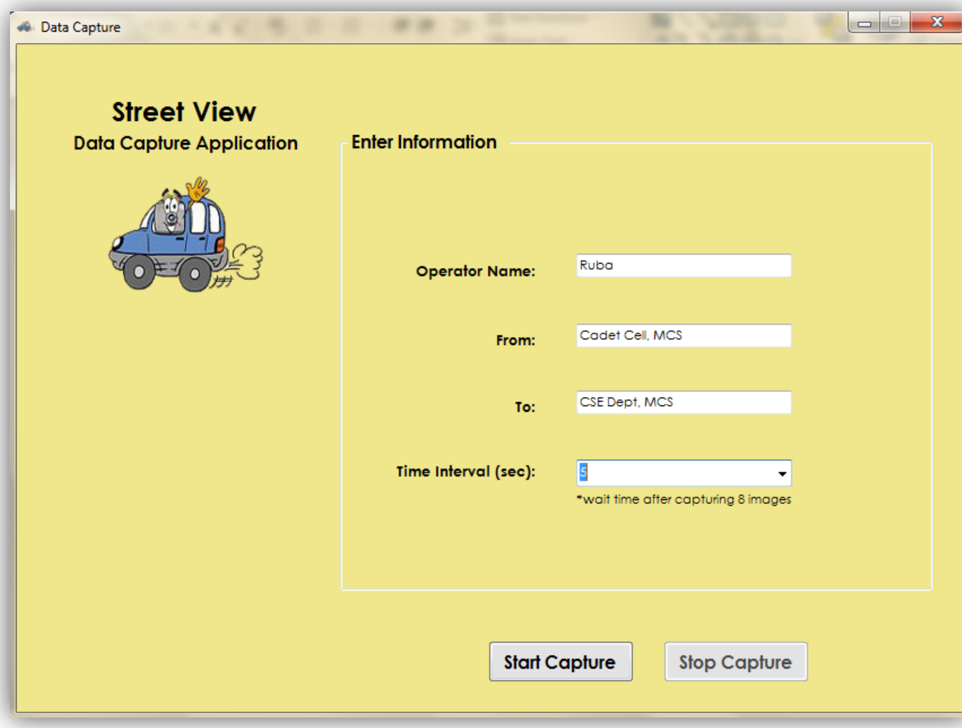


Figure 1 Street View Data Capture Application (First Screen)

5. Place the provided OpenCV folder in C:\Program Files\.
6. Run StreetView.exe, First Screen of the data capture application will appear (Figure 1).
7. Enter required information in the First Screen and select the time interval. This interval is the gap after capturing 8 images for a single panorama.
8. Click the start capture button (also start android application on android phone to capture long & lat coordinates), Data Capture Screen will appear (Figure 2) and cameras will start capturing the images.
9. Drive the car in the area whose street view is to be implemented.
10. Click Stop Capture button to stop the capture and stop android application as well.

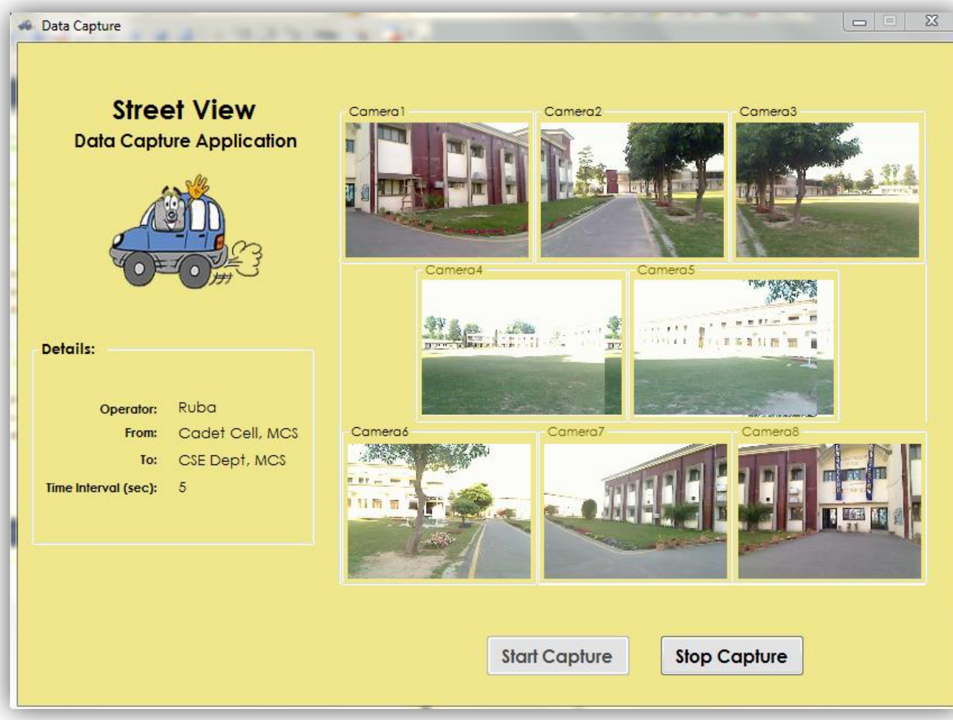


Figure 2 Street View Data Capture Application (Data Capturing screen)

11. All the images captured will be saved in the application's folder with timestamp as image names.
12. Long & lat coordinates will be saved in a text file in Android phone's memory.
13. All this captured data will be used in JavaScript code to create the walkthrough for Street View.

ANX-D

USER GUIDE - NAVIGATION THROUGH WALTHROUGH

<u>ANX-D</u>

User Guide – Navigation through walkthrough

Panoramic Street View's walkthrough is easy to use. GUI provides an easy to use walkthrough for the user.

7. Double click home.html file, the GUI will appear in default web browser of the system (Figure 1).
8. Select any location, by clicking on one of the markers, to view that location's street view.

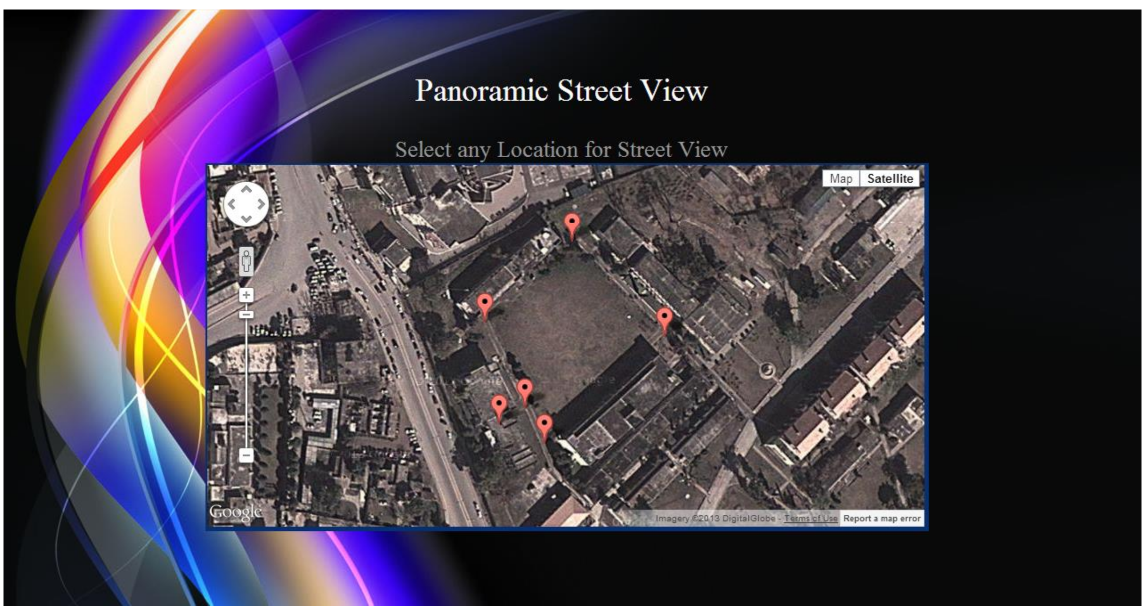


Figure 1 GUI of Panoramic Street View

9. Walkthrough will appear with a small map in bottom (Figure 2).
10. Navigate through the walkthrough using the links provided.
11. With every movement through links, location in map also changes due to synchronization between Map and Street View.

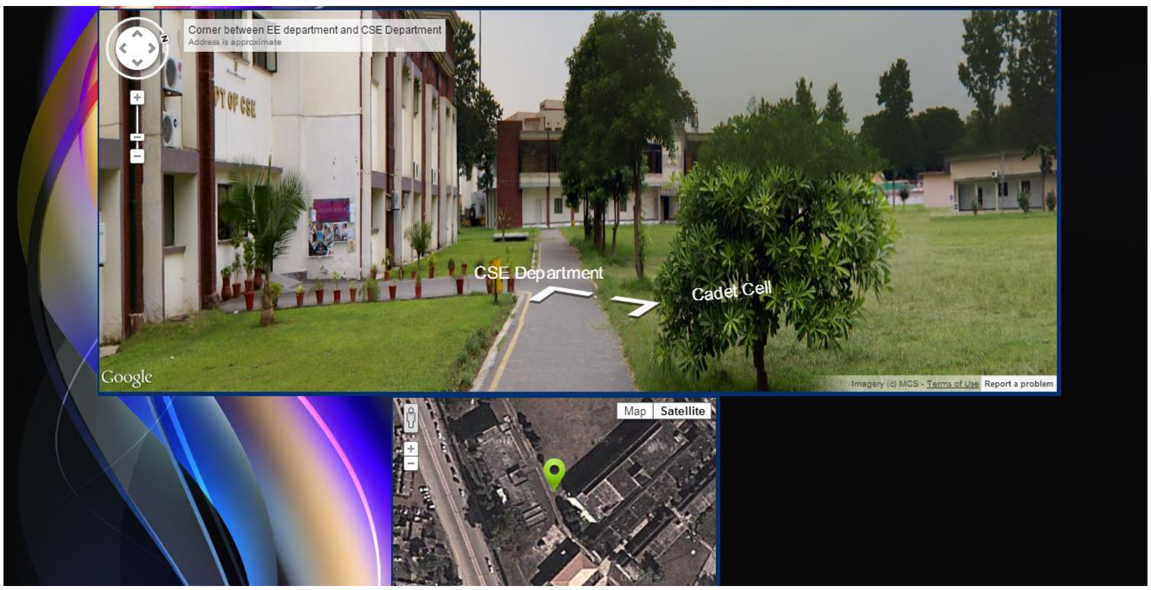


Figure 2 GUI of Panoramic Street View

ANX-E

HARDWARE AND SOFTWARE COMPONENTS

ANX E

Hardware and Software Components

Hardware Components:

- 8 x Logitech QuickCam Sphere AF cameras assembled in a camera mount
- 1 x Android phone with GPS sensor
- 1 x Trupulse 200 Laser Range Finder
- 1 x stand for LRF
- 1 x 4 wheel drive vehicle

Software Components

- Windows 7
- Microsoft Image Composite Editor
- OpenCV 2.4