**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**
**JNANASANGAMA, BELAGAVI - 590018**


# "VirtuAlive"

Thesis submitted in partial fulfillment of the curriculum prescribed for the award of the degree of Bachelor of Engineering in Computer Science & Engineering by


| | |
|---|---|
| 1CR14CS069 | Kushal I |
| 1CR14CS105 | Rajat Agarwal |
| 1CR14CS119 | Ron Paul Jackson |
| 1CR14CS156 | Varun Kumar Saini |


Under the Guidance of


Dr. P N Singh
Professor
Department of CSE, CMRIT, Bengaluru




**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**
#132, AECS LAYOUT, IT PARK ROAD, BENGALURU - 560037


2017-18

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**
**JNANASANGAMA, BELAGAVI - 590018**


# Certificate

This is to certify that the project entitled "VirtuAlive" is a bonafide work carried out by Kushal I, Rajat Agarwal, Ron Paul Jackson, and Varun Kumar Saini in partial fulfillment of the award of the degree of Bachelor of Engineering in Computer Science & Engineering of Visvesvaraya Technological University, Belgaum, during the year 2017-18. It is certified that all corrections / suggestions indicated during reviews have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the Bachelor of Engineering Degree.

| | | |
|---|---|---|
| ---------- | ---------- | ---------- |
| Signature of Guide | Signature of HoD | Signature of Principal |
| **Dr. P N Singh** | **Dr. Jhansi Rani P** | **Dr. Sanjay Jain** |
| Professor | Professor & Head | Principal |
| Department of CSE | Department of CSE | CMRIT, |
| CMRIT, Bengaluru - 37 | CMRIT, Bengaluru - 37 | Bengaluru - 37 |

**External Viva**

| | Name of the Examiners | Institution | Signature with Date |
|---|---|---|---|
| 1. | ---------- | ---------- | ---------- |
| 2. | ---------- | ---------- | ---------- |

# Acknowledgement

We take this opportunity to thank all of those who have generously helped us to give a proper shape to our work and complete our project successfully. A successful project is fruitful culmination efforts by many people, some directly involved and some others indirectly, by providing support and encouragement.

We would like to thank Dr. SANJAY JAIN, Principal, CMRIT, for providing excellent academic environment in the college.

We would like to express our gratitude towards Dr. JHANSI RANI, Professor & HOD, Dept of CSE, CMRIT, who provided guidance and gave valuable suggestions regarding the project.

We consider it a privilege and honour to express our sincere gratitude to our Internal Guide Dr. P N SINGH, Professor, Department of Computer Science & Engineering, CMRIT, for her valuable guidance throughout the tenure of this project work.

We would also like to thank all the faculty members of CSE who have always been very co-operative and generous.

Kushal I
Rajat Agarwal
Ron Paul Jackson
Varun Kumar Saini

# Table of Contents

# List of Figures

# Abstract

Augmented reality (AR) has been used in the last years as a tool for enhancing collaboration between the real world and virtual environments. One of these fields where AR has been used is the touristic sector. The aim of this project is to identify the benefits of the use of AR in tourism mobile application through the development and evaluation of an AR tourist mobile application. The results of the implementation of this project show AR enhance tourist on-site experience in an innovative way in the real life. Moreover, AR could be applied in different industries as a method of improving the quality of service.

# Chapter 1

# PREAMBLE

## 1.1 Introduction

Augmented reality (AR) is a direct or indirect live view of a physical, real-world environment whose elements are "augmented" by computer-generated perceptual information, ideally across multiple sensory modalities, including visual, auditory, haptic, somatosensory, and olfactory. The overlaid sensory information can be constructive (i.e. additive to the natural environment) or destructive (i.e. masking of the natural environment) and is spatially registered with the physical world such that it is perceived as an immersive aspect of the real environment. In this way, Augmented reality alters one's current perception of a real world environment, whereas virtual reality replaces the real world environment with a simulated one. Augmented Reality is related to two largely synonymous terms: mixed reality and computer-mediated reality.

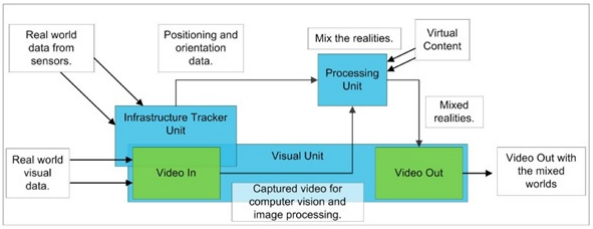


Figure 1.1: Structure of AR

Virtual reality (VR) is a computer-generated scenario that simulates a realistic experience. The immersive environment can be similar to the real world in order to create a lifelike experience grounded in reality or sci-fi. Augmented reality systems may also be considered a form of VR that layers virtual information over a live camera feed into a headset, or through a smartphone or tablet device.

Current VR technology most commonly uses virtual reality headsets or multi-projected environments, sometimes in combination with physical environments or props, to generate realistic images, sounds and other sensations that simulate a user's physical presence in a virtual or imaginary environment. A person using virtual reality equipment is able to"look around" the artificial world, move around in it, and interact with virtual features or items. The effect is commonly created by VR headsets consisting of a head-mounted display with a small screen in front of the eyes, but can also be created through specially designed rooms with multiple large screens.

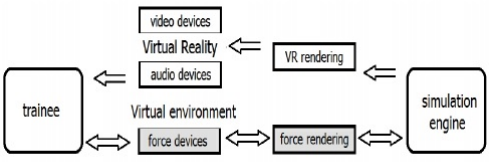


Figure 1.2: Structure of VR

## 1.2 Working of AR & VR

Augmented Reality(AR) refers to deploying virtual image over real-world objects. The overlay is executed simultaneously with the input received from a camera or another input device like smart glasses. This superimposition of virtual images over real-world objects creates an illusion that can effectively engage users in a virtual world. The Augmented Reality technology can work using one of the following three approaches:

1. SLAM(Simultaneous Localization and Mapping)
2. Recognition based
3. Location based

Virtual Reality(VR) simulation begins with putting on this headset. Users will usually start up the application or have someone get the game or application going on a PC or console. The VR headset is then plugged in and the user can switch their view over to the headset to immerse themselves in the experience. Once the headset is placed on the user's head and adjusted to fill their peripheral vision, they can use the motion controls to control the experience on screen or their own body movement to move around the scene. As the user looks around, motion controls in the headset will control the scene as it pans across the screen. As long as the user keeps the headset on, the scene will continue to move and interact with them as they move their head or use the controllers to look around and interact.

## 1.3 Characteristics of AR & VR

### 1.3.1 Characteristics of AR

- Combines real and virtual images
  - Both can be seen at the same time
- Interactive in real-time
  - The virtual content can be interacted with
- Registered in 3D
  - Virtual objects appear fixed in space

### 1.3.2 Characteristics of VR

- 3D stereoscopic display
- Wide field of view display
- Low latency head tracking

## 1.4 Types of AR & VR

There are five types of Augmented reality as mentioned below:-

- Marker-based Augmented reality
- Location-based Augmented reality
- Projection-based Augmented reality
- Outlining Augmented reality
- Superimposition-based Augmented reality

### 1.4.1 Marker-based Augmented reality

You must have heard about QR code. This is the most common type of Marker-based Augmented reality. 2D barcode is the simplest type of AR markers. And, more complex type consists of bright color and meaningful pictures. When a Smartphone having Marker-based AR application scans a pattern such as a bar-code or a symbol through the camera on it, the software recognizes it and superimposes a digital image on the screen. 3D or animated digital image is used for a better experience. The

Marker-based AR approach is also called as Recognition-based Augmented reality. The below figure can effectively illustrate the working principle of Marker-based type of AR applications.

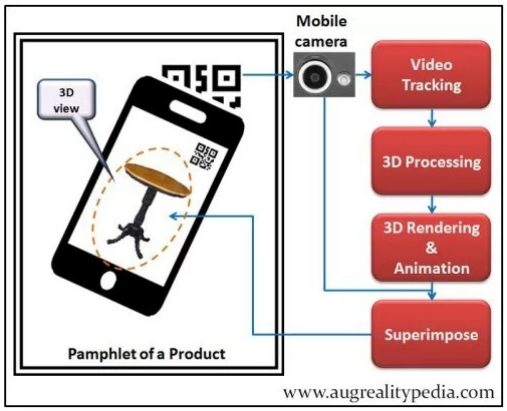


Figure 1.3: Marker-based Augmented reality

### 1.4.2 Location-based Augmented reality

When camera of a Smartphone having Location-based Augmented reality application is pointed towards a real scene, inbuilt GPS software recognizes the location of the device in the world. Based on this recorded location and orientation of device recorded through inbuilt sensors, like accelerometer and gyroscope of the device, the application offers data, relevant to that specific location, towards which user is looking for. These digital informative data are then superimposed to the real scene, visible to device camera. Most widely implemented AR applications are having location-based AR approach. Below figure can effectively illustrate the working principle of Location-based AR applications.

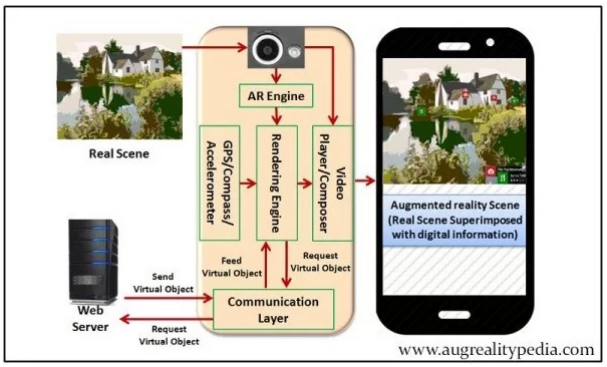


Figure 1.4: Location-based Augmented reality

### 1.4.3 Projection-based Augmented reality

Projection-based AR systems projects the virtual image onto physical objects. I have tried to explain the working of Projection-based AR using below figure. In this type of Augmented reality, the technique is to project a virtual image of a table clock on an actual table clock of the same size. The physical table clock is having different colour and visual texture than the virtual one. This projection leads to getting a feel of "how it feels" with the desired colour and visual texture of the virtual image. This physical model having a projection of the virtual image is called Projection Augmented model (PA model). We can touch, feel, and even grasp this PA model with our hands.

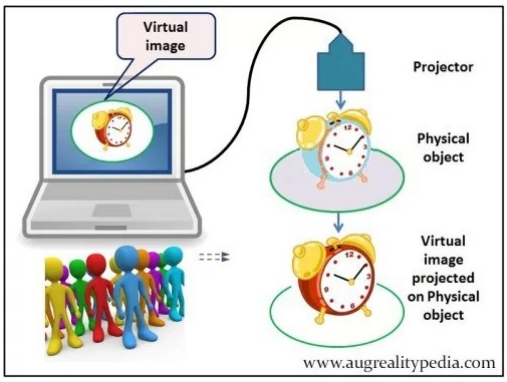


Figure 1.5: Projection-based Augmented reality

### 1.4.4 Outlining Augmented reality

Outlining-based AR application merges the outline of any object, so that you can pick up the same with your hand. You would also be able to manipulate it with a virtual object that does not exist in real view. The concept behind this type of Augmented reality is to track the outline of your hand with the help of a camera and then reorient the virtual object. A similar approach is also used for tracking a face. After detecting a face, AR software recognizes the position of the main features of the face, like the mouth, nose, eyes, etc. Using these positions as reference points, software overlays virtual object on the face. Once this recognition and connection between virtual object and face have established, the software can redraw the virtual object in the real-time with the actual movement of the face. Very recently, Shoppers Stop has launched AR based dressing room in Mumbai (INDIA).

### 1.4.5 Superimposition-based Augmented reality

In Superimposition-based AR, the augmented view is superimposed on the real view of any object. This augmented view could be either internal view of the object, or

it could be X-Ray/CT scan image. These types of AR can be used in various fields like defense, education, medical, architecture, etc. Below figure effectively describes the superimposition of an X-ray image of the ankle on the live video of a patient. Here, an augmented view has generated through AR overlay and displayed on the screen, enabling the doctor to look and operate the fracture in the ankle of a patient in real-time.

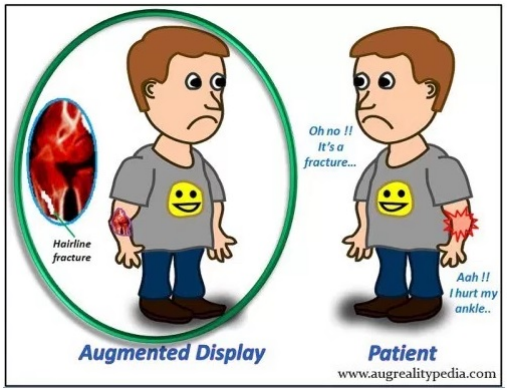


Figure 1.6: Superimposition-based Augmented reality

## 1.5 Advantages of AR & VR

AR is going to change the shape of commerce thoroughly shortly. The core advantage for business in augmented reality development for smartphones and tablets is that the hardware is available, and the usage is intuitive and understanding.

In the next few years, users will be able to try on clothes without actually wearing it or to check if the furniture fits the interior (IKEA actually already did it!). And all this just with a help of Android or iPhone augmented reality app. Moreover, with future technology evolution there will appear other human sensors as smelling, touching and feeling, AR brings truly unlimited possibilities for teaching and learning process. Additional reality provides the unique cognition path with immersive real-life simulations.

For the travel industry, virtual reality can be used to enhance travel experience and shape the behavior of travel consumers. What technology can do today is absolutely remarkable!

Using the VR, your tour or activity business can have a virtual walkthrough. Allow potential consumers to catch a glimpse of the atmosphere of the destination

they're about to book. Integrating virtual reality impacts the way you do business. Along with the storytelling, it makes a compelling case how customers opt for a certain destination before making a reservation. Its vivid. It showcases the best visual content. It has a larger scale effect on consumer choices, rather than using a standard destination description and tour itinerary. Decreasing consumer attention span is all about relevancy. This basically means fewer people have the patience to read through classical travel sales pitches and planned activities.

## 1.6 Existing System

The existing system is the presence of physical guides at tourist spot, the guide accompanies the tourist and gives his views/knowledge about the attraction that is being visited. The knowledge shared by the guide may not be accurate and may cause wrong information to propagate among tourists. Sometimes tourists don't prefer a guide because they charge extra money and they tend to miss the best attractions of the place because they are not guided properly, there are language barriers at times with the guide which may cause a communication gap.

Another existing system would be sign boards present at the location guiding people, but these sign boards require high maintenance and are subject to regular wear and tear because of the external weather conditions.

An existing solution for VR would be YouTube where people would watch videos of that place to get the feel of the location before actually visiting the place. The drawback of such a system is that a proper experience is not given to the user.

Google Maps gives us location in a map, but using AR we can find locations in a more convenient way which doesn't cause any ambiguity to the user, and provides the location in a very user friendly manner.

### 1.6.1 Disadvantages of existing system

- Authenticity: The information given by the guide may or may not be authentic.
- Cost: The financial compensation to be paid to the guide.
- Language: The guide may not be well-versed with all the languages of the tourists.

## 1.7 Proposed System

The proposed system is an Android application which would overcome the disadvantages of the existing system and it uses Augmented Reality to enhance the tourist on-site experience. The solution would be obtained by using AR as a one of a kind hybrid of a navigation cum information system, the user would not only be able to locate an attraction but also gain information about the same using the same interface. The navigation system would use the phones GPS data and map it against the database stored at the server side of the application. The UI of the navigation system would consist of the camera which would display markers which are clickable. On clicking the marker, relevant information pertaining to that marker will be displayed.

The Virtual Reality part of the proposed system is meant to be used to experience the attraction from anywhere in the world. The system comprises of 360 degree pictures and videos of the place which when viewed through a VR box provides a real lifelike experience for the user. This system is aimed at making the tourists believe that the particular location is a must visit destination and which would tend to improve the tourism industry.

### 1.7.1 Advantages of proposed system

- Virtually free of cost making it highly affordable
- It is aimed at improving the tourism industry of a location, hence helping in generate more revenue from the tourism industry
- Provides a mode of experiencing the venue from any place in the world
- Helps in providing accurate information in a meaningful manner using technology

# Chapter 2

# LITERATURE SURVEY

Literature survey is mainly carried out in order to analyze the background of the current project which helps to find out flaws in the existing system and guides on which unsolved problems we can work out. So, the following topics not only illustrate the background of the project but also uncover the problems and flaws which motivated to propose solutions and work on this project.

## 2.1 Literature Survey

Literature survey is the documentation of a comprehensive review of the published and unpublished work from secondary sources data in the areas of specific interest to the researcher. The library is a rich storage base for secondary data and researchers used to spend several weeks and sometimes months going through books, journals, newspapers, magazines, conference proceedings, doctoral dissertations, master's theses, government publications and financial reports to find information on their research topic. Reviewing the literature on the topic area at this time helps the researcher to focus further interviews more meaningfully on certain aspects found to be important is the published studies even if these had not surfaced during the earlier questioning. So the literature survey is important for gathering the secondary data for the research which might be proved very helpful in the research. The literature survey can be conducted for several reasons. The literature review can be in any area of the business.

### 2.1.1 Emerging technologies of augmented reality by M. Haller, M. Billinghurst and B. Thomas

**Description:** Although the field of mixed reality has grown significantly over the last decade, there have been few published books about augmented reality, particularly the interface design aspects. Emerging Technologies of Augmented Reality:

Interfaces and Design provides a foundation of the main concepts of augmented reality (AR), with a particular emphasis on user interfaces, design, and practical AR techniques, from tracking algorithms to design principles for AR interfaces. Emerging Technologies of Augmented Reality: Interfaces and Design contains comprehensive information focusing on the following topics: technologies that support AR, development environments, interface design and evaluation of applications, and case studies of AR applications.

### 2.1.2 Interaction design by H. Sharp, Y. Rogers and J. Preece

**Description:** Interaction design can be defined as designing interactive products to support the way people communicate and interact in their everyday and working lives. To be successful, interaction designers will need a mixed set of skills drawn from psychology, human computer interaction, web design, computer science, information systems, and entertainment as well as an understanding of the desires and needs of people and the kinds of technology available. Interaction Design: beyond human computer interaction offers a cross-disciplinary, practical and process-oriented introduction to the field, showing not just what principles ought to apply to interaction design, but crucially how they can be applied.

### 2.1.3 Experimentation in software engineering by C. Wohlin

**Description:** Like other sciences and engineering disciplines, software engineering requires a cycle of model building, experimentation, and learning. Experiments are valuable tools for all software engineers who are involved in evaluating and choosing between different methods, techniques, languages and tools. The purpose of Experimentation in Software Engineering is to introduce students, teachers, researchers, and practitioners to empirical studies in software engineering, using controlled experiments. The introduction to experimentation is provided through a process perspective, and the focus is on the steps that we have to go through to perform an experiment. The book is divided into three parts. The first part provides a background of theories and methods used in experimentation. Part II then devotes one chapter to each of the five experiment steps: scoping, planning, execution, analysis, and result presentation. Part III completes the presentation with two examples. Assignments and statistical material are provided in appendixes. Overall the book provides indispensable information regarding empirical studies in particular for experiments, but also for case studies, systematic literature reviews, and surveys.

The book is self-contained and it is suitable as a course book in undergraduate or graduate studies where the need for empirical studies in software engineering is stressed. Exercises and assignments are included to combine the more theoretical material with practical aspects. Researchers will also benefit from the book, learning more about how to conduct empirical studies, and likewise practitioners may use it as a "cookbook" when evaluating new methods or techniques before implementing them in their organization.

### 2.1.4 Research methods in human-computer interaction by Lazar, J. Feng and H. Hochheiser

**Description:** A comprehensive research guide for both quantitative and qualitative research methods. Written by a team of authorities in human-computer interaction (HCI) and usability, this pedagogical guide walks you through the methods used in HCI and examines what are considered to be appropriate research practices in the field. Featuring a plethora of real-world examples throughout, you'll discover how these methods have been used in HCI research so that you can gain a stronger understanding of the subject matter. Serves as an authoritative, comprehensive resource on all things related to research methods in human-computer interaction. It addresses experimental research and design methods, statistical analysis, and time diaries. It shares authentic case studies, interviews, and focus group experiences. It contains reviews analyzing qualitative data, working with human subjects, handling automated computer data collection methods, and more.

# Chapter 3

# THEORETICAL BACKGROUND

Theoretical background highlights some topics on the area under study. The description contains several topics which are worth to discuss and also highlight some of their limitations and those that motivated for the proposed solution as well as highlights some of their advantages for which reason these topics and their features are used in this project.

A key measure of AR systems is how realistically they integrate augmentations with the real world. The software must derive real world coordinates, independent from the camera, from camera images. That process is called image registration, and uses different methods of computer vision, mostly related to video tracking. Many computer vision methods of augmented reality are inherited from visual odometry.

Usually those methods consist of two parts. The first stage is to detect interest points, fiducial markers or optical flow in the camera images. This step can use feature detection methods like corner detection, blob detection, edge detection or thresholding, and other image processing methods. The second stage restores a real world coordinate system from the data obtained in the first stage. Some methods assume objects with known geometry (or fiducial markers) are present in the scene. In some of those cases the scene 3D structure should be pre-calculated beforehand. If part of the scene is unknown simultaneous localization and mapping (SLAM) can map relative positions. If no information about scene geometry is available, structure from motion methods like bundle adjustment are used. Mathematical methods used in the second stage include projective (epipolar) geometry, geometric algebra, rotation representation with exponential map, kalman and particle filters, nonlinear optimization, robust statistics.

## 3.1 Feature Detection

In computer vision and image processing, feature detection includes methods for computing abstractions of image information and making local decisions at every image point whether there is an image feature of a given type at that point or not. The resulting features will be subsets of the image domain, often in the form of isolated points, continuous curves or connected regions.

There is no universal or exact definition of what constitutes a feature, and the exact definition often depends on the problem or the type of application. Given that, a feature is defined as an "interesting" part of an image, and features are used as a starting point for many computer vision algorithms. Since features are used as the starting point and main primitives for subsequent algorithms, the overall algorithm will often only be as good as its feature detector. Consequently, the desirable property for a feature detector is repeatability: whether or not the same feature will be detected in two or more different images of the same scene.

Feature detection is a low-level image processing operation. That is, it is usually performed as the first operation on an image, and examines every pixel to see if there is a feature present at that pixel. If this is part of a larger algorithm, then the algorithm will typically only examine the image in the region of the features. As a built-in pre-requisite to feature detection, the input image is usually smoothed by a Gaussian kernel in a scale-space representation and one or several feature images are computed, often expressed in terms of local image derivatives operations.

## 3.2 Types of Image features

### 3.2.1 Edges

Edges are points where there is a boundary (or an edge) between two image regions. In general, an edge can be of almost arbitrary shape, and may include junctions. In practice, edges are usually defined as sets of points in the image which have a strong gradient magnitude. Furthermore, some common algorithms will then chain high gradient points together to form a more complete description of an edge. These algorithms usually place some constraints on the properties of an edge, such as shape, smoothness, and gradient value. Locally, edges have a one-dimensional structure.

### 3.2.2 Corners/interest points

The terms corners and interest points are used somewhat interchangeably and refer to point-like features in an image, which have a local two dimensional structure. The name "Corner" arose since early algorithms first performed edge detection, and then analysed the edges to find rapid changes in direction (corners). These algorithms were then developed so that explicit edge detection was no longer required, for instance by looking for high levels of curvature in the image gradient. It was then noticed that the so-called corners were also being detected on parts of the image which were not corners in the traditional sense (for instance a small bright spot on a dark background may be detected). These points are frequently known as interest points, but the term "corner" is used by tradition.

### 3.2.3 Blobs/regions of interest points

Blobs provide a complementary description of image structures in terms of regions, as opposed to corners that are more point-like. Nevertheless, blob descriptors may often contain a preferred point (a local maximum of an operator response or a center of gravity) which means that many blob detectors may also be regarded as interest point operators. Blob detectors can detect areas in an image which are too smooth to be detected by a corner detector.

Consider shrinking an image and then performing corner detection. The detector will respond to points which are sharp in the shrunk image, but may be smooth in the original image. It is at this point that the difference between a corner detector and a blob detector becomes somewhat vague. To a large extent, this distinction can be remedied by including an appropriate notion of scale. Nevertheless, due to their response properties to different types of image structures at different scales, the LoG and DoH blob detectors are also mentioned in the article on corner detection.

### 3.2.4 Ridges

For elongated objects, the notion of ridges is a natural tool. A ridge descriptor computed from a grey-level image can be seen as a generalization of a medial axis. From a practical viewpoint, a ridge can be thought of as a one-dimensional curve that represents an axis of symmetry, and in addition has an attribute of local ridge width associated with each ridge point. Unfortunately, however, it is algorithmically harder to extract ridge features from general classes of grey-level images than edge-,

corner- or blob features. Nevertheless, ridge descriptors are frequently used for road extraction in aerial images and for extracting blood vessels in medical images.

## 3.3 Feature Detection Methods

### 3.3.1 Corner Detection

A corner can be defined as the intersection of two edges. A corner can also be defined as a point for which there are two dominant and different edge directions in a local neighbourhood of the point.

An interest point is a point in an image which has a well-defined position and can be robustly detected. This means that an interest point can be a corner but it can also be, for example, an isolated point of local intensity maximum or minimum, line endings, or a point on a curve where the curvature is locally maximal.

In practice, most so-called corner detection methods detect interest points in general, and in fact, the term "corner" and "interest point" are used more or less interchangeably through the literature. As a consequence, if only corners are to be detected it is necessary to do a local analysis of detected interest points to determine which of these are real corners. Examples of edge detection that can be used with post-processing to detect corners are the Kirsch operator and the Frei-Chen masking set.

"Corner","interest point" and "feature" are used interchangeably in literature, confusing the issue. Specifically, there are several blob detectors that can be referred to as "interest point operators", but which are sometimes erroneously referred to as "corner detectors". Moreover, there exists a notion of ridge detection to capture the presence of elongated objects.

### 3.3.2 Blob Detection

In computer vision, blob detection methods are aimed at detecting regions in a digital image that differ in properties, such as brightness or color, compared to surrounding regions. Informally, a blob is a region of an image in which some properties are constant or approximately constant; all the points in a blob can be considered in some sense to be similar to each other. The most common method for blob detection is convolution.

Given some property of interest expressed as a function of position on the image, there are two main classes of blob detectors: (i) differential methods, which are based on derivatives of the function with respect to position, and (ii) methods based on local extrema, which are based on finding the local maxima and minima of the function. With the more recent terminology used in the field, these detectors can also be referred to as interest point operators, or alternatively interest region operators (see also interest point detection and corner detection).

### 3.3.3 Edge Detection

Edge detection includes a variety of mathematical methods that aim at identifying points in a digital image at which the image brightness changes sharply or, more formally, has discontinuities. The points at which image brightness changes sharply are typically organized into a set of curved line segments termed edges. The same problem of finding discontinuities in one-dimensional signals is known as step detection and the problem of finding signal discontinuities over time is known as change detection. Edge detection is a fundamental tool in image processing, machine vision and computer vision, particularly in the areas of feature detection and feature extraction.

## 3.4 Simultaneous localization and mapping (SLAM)

Simultaneous localization and mapping (SLAM) is the computational problem of constructing or updating a map of an unknown environment while simultaneously keeping track of an agent's location within it. While this initially appears to be a chicken-and-egg problem there are several algorithms known for solving it, at least approximately, in tractable time for certain environments. Popular approximate solution methods include the particle filter, extended Kalman filter, and GraphSLAM.

### 3.4.1 SLAM Algorithms

Statistical techniques used to approximate the above equations include Kalman filters, particle filters (aka. Monte Carlo methods) and scan matching of range data. They provide an estimation of the posterior probability function for the pose of the robot and for the parameters of the map. Set-membership techniques are mainly based on interval constraint propagation. They provide a set which encloses the pose of the robot and a set approximation of the map. Bundle adjustment is another popular technique for SLAM using image data, which jointly estimates poses and landmark positions, increasing map fidelity, and is used in commercialized SLAM systems such

as Google's Project Tango.

New SLAM algorithms remain an active research area, and are often driven by differing requirements and assumptions about the types of maps, sensors and models. Many SLAM systems can be viewed as combinations of choices from each of these aspects.

# Chapter 4

# SYSTEM REQUIREMENT SPECIFICATION

This chapter describes about the requirements. It specifies the hardware and software requirements that are required in order to run the application properly. The Software Requirement Specification (SRS) is explained in detail, which includes overview of dissertation as well as the functional and non-functional requirement of this dissertation.

A SRS document describes all data, functional and behavioral requirements of the software under production or development. SRS is a fundamental document, which forms the foundation of the software development process. It's the complete description of the behavior of a system to be developed. It not only lists the requirements of a system but also has a description of its major feature. Requirement Analysis in system engineering and software engineering encompasses those tasks that go into determining the need or conditions to meet for a new or altered product, taking account of the possibly conflicting requirements of the various stakeholders, such as beneficiaries or users. Requirement Analysis is critical to the success to a development project. Requirement must be documented, measurable, testable, related to in identified business needs or opportunities, and defined to a level of detail sufficient for system design.

The SRS functions as a blueprint for completing a project. The SRS is often referred to as the "parent" document because all subsequent project management documents, such as design specifications, statements of work, software architecture specification, testing and validation plans, and documentation plans, are related to it. It is important to note that an SRS contains functional and non-functional requirements only.

Thus the goal of preparing the SRS document is:

- To facilitate communication between the customer, analyst, system developers, maintainers.
- To serve as a contrast between purchaser and supplier.
- To firm foundation for the design phase.
- Support system testing facilities.
- Support project management and control.
- Controlling the evolution of the system.

## 4.1 Functional Requirements

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs or conditions. These may include calculations, data manipulation and processing and other specific functionality. In this system following are the functional requirements:

- The android phone uses its GPS device to fetch the latitude and longitude of the user from the satellite
- The GPS coordinates are then mapped with the server entries and probable tourist spots are shown on the map
- On selecting a particular spot ,the options of AR and VR of that location appear on the screen
- The AR markers are rendered based on the compass and gyroscope information provided by the phone
- The VR option is available anywhere in the world, and is rendered based on the gyroscope data provided.

## 4.2 Non-Functional Requirements

Non-functional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge the operation of a system rather than specific behaviors. They may relate to emergent system properties such as reliability, response time and store occupancy. Many non-functional requirements relate to the system as a whole rather than

to individual system features. This means they are often critical than the individual functional requirements. The following non-functional requirements are worthy of attention. The key non-functional requirements are:

- Security: The location of a user should not be breached and displayed to other users
- System Efficiency: The sensors in the phone should be managed efficiently so that the battery of the phone is conserved
- Reliability: The system is expected to work in all weather conditions and provide accurate results.

## 4.3 Performance

The Application performance can be determined by studying the number of hours the phone runs with the app running to the number of hours the phone runs ideally, this ratio was found to be optimum for VirtuAlive. The accuracy was found to be accurate upto 1m in good GPS conditions(evening with more satellites), 3m in normal conditions and 4m in bad conditions(morning with less satellites), the app is expected to give a response time of 3s to render the required information onto the screen. There is no evident lag experienced by the user as the application is not computationally heavy and is expected to run well in a 2Ghz processor phone.

## 4.4 Hardware Requirements

- System : Android Phone
- Internal Storage : 2 GB
- RAM : 512MB
- CPU : 2Ghz processor

## 4.5 Software Requirements

- Operating system : Windows 10.
- Coding Language : JAVA
- IDE : Android Studio

# Chapter 5

# SYSTEM ANALYSIS

System Analysis refers to the process of examining a business situation with the intent of improving through better procedures and methods. System Analysis and Design relates to shaping organizations, improving performance and achieving objectives for profitability and growth. The emphasis is on systems in action, the relationships among subsystems and their contribution to meet a common goal.

## 5.1 Feasibility Study

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are:

- Economical Feasibility
- Technical Feasibility
- Social Feasibility

### 5.1.1 Economical Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most

of the technologies used are freely available.

The necessary technology to implement the system is freely available. The system can be developed and operated in the existing hardware and software infrastructure. A country's economy is highly dependent on its tourism industry. Thus, enhancing the tourist experience benefits the economy of the country.

### 5.1.2 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

The technology being used to implement the system is one of the latest trends, which is best suited for the proposed system. The proposed equipment for the system to be implemented is an Android phone with GPS and gyroscope sensors, which is very common nowadays.

### 5.1.3 Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

This system can be implemented in the tourism sector due to the boom in this sector in recent years. The system is expected to operate properly irrespective of external conditions with the exceptions being poor hardware and power constraints.

# Chapter 6

# SYSTEM DESIGN

Design is a meaningful engineering representation of something that is to be built .It is the most crucial phase in the developments of a system. Software design is a process through which the requirements are translated into a representation of software. Design is a place where design is fostered in software engineering. Based on the user requirements and the detailed analysis of the existing system, the new system must be designed. This is the phase of system designing. Design is the perfect way to accurately translate a customer's requirements in the finished software product. Design creates a representation or model, provides details about software data structure, architecture, interfaces and components that are necessary to implement a system. The logical system design arrived at as a result of system analysis is converted into physical system design. It consists of:

- Development Methodology
- Data Flow Diagram
- Class Diagram
- Use Case Diagram
- Sequence Diagram
- Activity Diagram

## 6.1 Development Methodology

Software development methodology or system development methodology in software engineering is a framework that is used to structure, plan, and control the process of developing an information system. The software development methodology (also known as SBM) framework didn't emerge until the 1960's. According to Elliott (2004)

the system's development life cycle (SDLC) can be considered to be the oldest formalized methodology framework for holding information systems. The main idea of the SDLC has been "to pursue the development of information systems in a very deliberate, structured and methodical way, requiring the each stage of the life cycle from inspection of the idea to delivery of the final system, to be carried out rigidly and sequentially" within in the context of the framework being applied. The main target of this methodology framework in the 1960's was "to develop large scale functional business systems in an age of large scale business conglomerates. Information systems activities revolved around heavy data processing and number crunching routines".

## 6.2 Data Flow Diagram

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

- The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
- The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
- DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
- DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

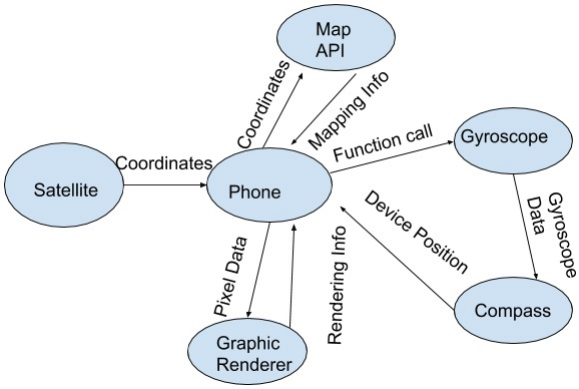


Figure 6.1: Data Flow Diagram

## 6.3 UML Diagram

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

- The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.
- The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.
- The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects. The primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.

5. Support higher level development concepts such as collaborations, frameworks, patterns and components.

### 6.3.1 Class Diagram

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

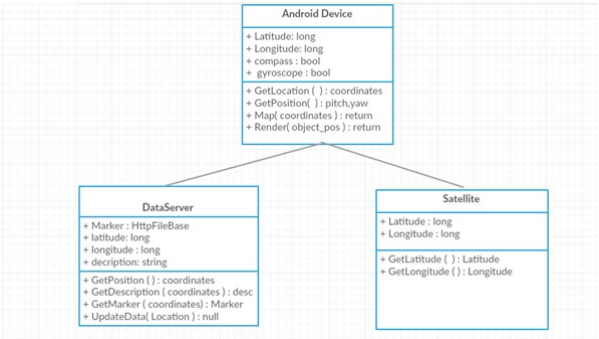


Figure 6.2: Class Diagram

### 6.3.2 Use Case Diagram

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

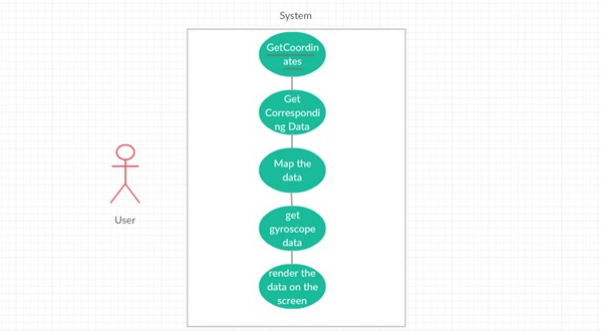


Figure 6.3: Use Case Diagram

### 6.3.3 Sequence Diagram

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

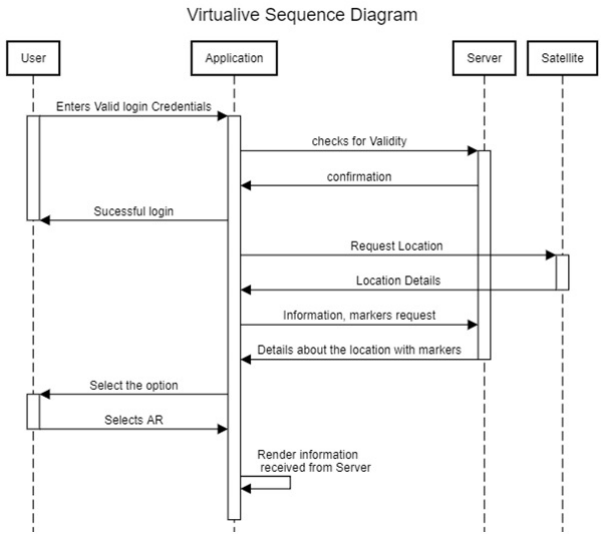


Figure 6.4: Sequence Diagram

### 6.3.4 Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

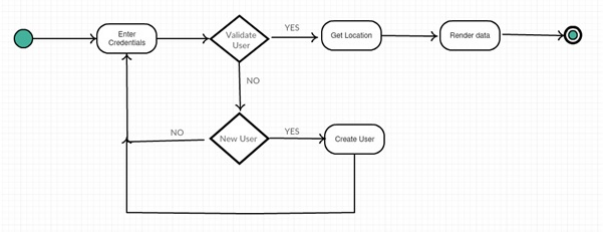


Figure 6.5: Activity Diagram

# Chapter 7

# IMPLEMENTATION

Implementation is a stage of the project where the theoretical design is turned into a working system. This phase involves the actual materialization of the ideas, which are expressed in the analysis document and developed in the design phase. Implementation should be perfect mapping of the design document in a suitable programming language in order to achieve the necessary final product. If the implementation is not carefully planned and controlled, it can cause chaos and confusion.

The implementation stage requires the following tasks:

- Careful Planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.
- Evaluation of the changeover method.
- Correct decisions regarding selection of the platform.
- Appropriate selection of the language for application development.

## 7.1 Implementation

Implementation is the stage of the project when the theoretical design is turned into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. The implementation stage involves careful planning, investigation of system and constraints, design of methods to achieve the changeover and evaluation of the changeover method.

Implementation of any software is always preceded by important decisions regarding selection of the platform, the language used, etc. These decisions are often infused by several factors such as the real environment in which the system works the speed that is required, the security concerns, other implementation specific details, etc. There are three major implementation decisions that have been made before the implementation of this project.

They are as follows:

- Selection of programming language for development of the system.
- Selection of platform (OS).
- Coding standards.

### 7.1.1 Implementation Requirements

The implementation decisions are mainly concerned with the ease of future enhancement of the system and a distributed environment. The requirements for implementation are as follows:

- The language chosen for this project is Java- JDK.
- The language chosen for GUI is XML.
- IDE for development chosen is Android Studio.
- Operating System used to implement this project is Windows XP.

## 7.2 Programming Language Selection

The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

- Simple
- Architecture neutral
- Object oriented Portable
- Distributed
- High performance
- Interpreted

- Multithreaded
- Robust
- Dynamic
- Secure

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called Java byte codes the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works.

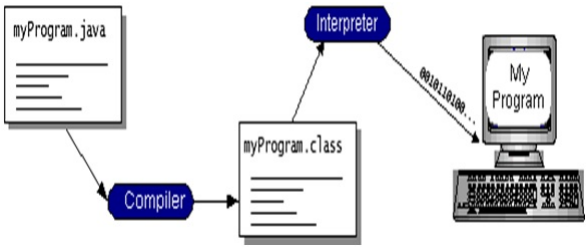


Figure 7.1: Java Program Compilation

You can think of Java byte codes as the machine code instructions for the Java Virtual Machine (Java VM). Every Java interpreter, whether it's a development tool or a Web browser that can run applets, is an implementation of the Java VM. Java byte codes help make "write once, run anywhere" possible. You can compile your program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on Windows 2000, a Solaris workstation, or on an iMac.

### 7.2.1 The Java Platform

A platform is the hardware or software environment in which a program runs. We've already mentioned some of the most popular platforms like Windows 2000, Linux, Solaris, and MacOS. Most platforms can be described as a combination of the operating system and hardware. The Java platform differs from most other platforms in

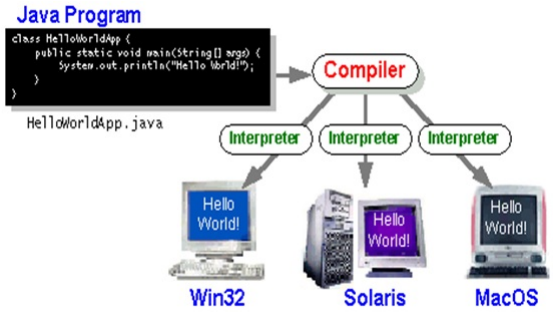


Figure 7.2: JVM

that it's a software-only platform that runs on top of other hardware-based platforms.

The Java platform has two components:

- The Java Virtual Machine (Java VM)
- The Java Application Programming Interface (Java API)

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as packages. The next section, What Can Java Technology Do? Highlights what functionality some of the packages in the Java API provide.

The following figure depicts a program that's running on the Java platform. As the figure shows, the Java API and the virtual machine insulate the program from the hardware.

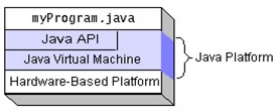


Figure 7.3: Java Platform

Native code is code that after you compile it, the compiled code runs on a specific hardware platform. As a platform-independent environment, the Java platform can be a bit slower than native code. However, smart compilers, well-tuned interpreters, and just-in-time bytecode compilers can bring performance close to that of native code without threatening portability.

### 7.2.2 Java Technology

The most common types of programs written in the Java programming language are applets and applications. If youve surfed the Web, you're probably already familiar with applets. An applet is a program that adheres to certain conventions that allow it to run within a Java-enabled browser.

However, the Java programming language is not just for writing cute, entertaining applets for the Web. The general-purpose, high-level Java programming language is also a powerful software platform. Using the generous API, you can write many types of programs.

An application is a standalone program that runs directly on the Java platform. A special kind of application known as a server serves and supports clients on a network. Examples of servers are Web servers, proxy servers, mail servers, and print servers. Another specialized program is a servlet. A servlet can almost be thought of as an applet that runs on the server side. Java Servlets are a popular choice for building interactive web applications, replacing the use of CGI scripts. Servlets are similar to applets in that they are runtime extensions of applications. Instead of working in browsers, though, servlets run within Java Web servers, configuring or tailoring the server.

How does the API support all these kinds of programs? It does so with packages of software components that provides a wide range of functionality. Every full implementation of the Java platform gives you the following features:

- The essentials: Objects, strings, threads, numbers, input and output, data structures, system properties, date and time, and so on.
- Applets: The set of conventions used by applets.
- Networking: URLs, TCP (Transmission Control Protocol), UDP (User Data gram Protocol) sockets, and IP (Internet Protocol) addresses.
- Internationalization: Help for writing programs that can be localized for users worldwide. Programs can automatically adapt to specific locales and be displayed in the appropriate language.
- Security: Both low level and high level, including electronic signatures, public and private key management, access control, and certificates.
- Software components: Known as JavaBeansTM, can plug into existing component architectures.

- Object serialization: Allows lightweight persistence and communication via Remote Method Invocation (RMI).

- Java Database Connectivity (JDBCTM): Provides uniform access to a wide range of relational databases.

The Java platform also has APIs for 2D and 3D graphics, accessibility, servers, collaboration, telephony, speech, animation, and more.

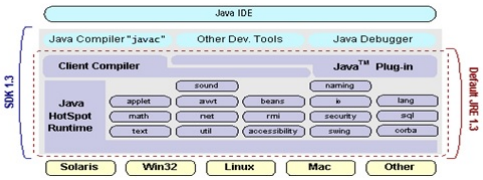


Figure 7.4: Java SDK

### 7.2.3 Role of Java Technology

It is likely to make your programs better and requires less effort than other languages. We believe that Java technology will help you do the following:

- Get started quickly: Although the Java programming language is a powerful object-oriented language, its easy to learn, especially for programmers already familiar with C or C++.

- Write less code: Comparisons of program metrics (class counts, method counts, and so on) suggest that a program written in the Java programming language can be four times smaller than the same program in C++.

- Write better code: The Java programming language encourages good coding practices, and its garbage collection helps you avoid memory leaks. Its object orientation, its JavaBeans component architecture, and its wide-ranging, easily extendible API let you reuse other peoples tested code and introduce fewer bugs.

- Develop programs more quickly: Your development time may be as much as twice as fast versus writing the same program in C++. Why? You write fewer lines of code and it is a simpler programming language than C++.

- Avoid platform dependencies with 100% Pure Java: You can keep your program portable by avoiding the use of libraries written in other languages. The 100% Pure JavaTM Product Certification Program has a repository of historical process manuals, white papers, brochures, and similar materials online.

- Write once, run anywhere: Because 100% Pure Java programs are compiled into machine-independent byte codes, they run consistently on any Java platform.
- Distribute software more easily: You can upgrade applets easily from a central server. Applets take advantage of the feature of allowing new classes to be loaded on the fly, without recompiling the entire program.

## 7.3 GUI Selection

A layout defines the structure for a user interface in your app, such as in an activity. All elements in the layout are built using a hierarchy of View and ViewGroup objects. A View usually draws something the user can see and interact with. Whereas a ViewGroup is an invisible container that defines the layout structure for View and other ViewGroup objects.

The View objects are usually called "widgets" and can be one of many subclasses, such as Button or TextView. The ViewGroup objects are usually called "layouts" can be one of many types that provide a different layout structure, such as LinearLayout or ConstraintLayout. You can declare a layout in two ways:

- **Declare UI elements in XML.** Android provides a straightforward XML vocabulary that corresponds to the View classes and subclasses, such as those for widgets and layouts. You can also use Android Studio's Layout Editor to build your XML layout using a drag-and-drop interface.
- **Instantiate layout elements at runtime.** Your app can create View and ViewGroup objects (and manipulate their properties) programmatically.

Declaring your UI in XML allows you to separate the presentation of your app from the code that controls its behavior. Using XML files also makes it easy to provide different layouts for different screen sizes and orientations.

The Android framework gives you the flexibility to use either or both of these methods to build your application's UI. For example, you can declare your application's default layouts in XML, and then modify the layout at runtime.

### 7.3.1 Integrated Development Environment

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains IntelliJ IDEA software and designed

specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as primary IDE for native Android application development. The following features are provided in the current stable version:

- Gradle-based build support
- Android-specific refactoring and quick fixes
- Lint tools to catch performance, usability, version compatibility and other problems
- ProGuard integration and app-signing capabilities
- Template-based wizards to create common Android designs and components
- A rich layout editor that allows users to drag-and-drop UI components, option to preview layouts on multiple screen configurations
- Support for building Android Wear apps
- Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine
- Android Virtual Device (Emulator) to run and debug apps in the Android studio.

## 7.4 Platform Selection

Even though there is no dependence of the working of the project on a particular platform, Windows XP is chosen to be the platform. However the backend of the software works in any of the OS, it may be the Linux platform or the Windows platform with the limitation of the database used. Windows XP is popular for its new features. Also, it comes from the master maker of OS, Microsoft. First and foremost thing is that Windows XP doesnt need any external drivers to run the hardware. It has universal support for hardware. There is no need to install programs before using a disk or USB. All these features are easily repairable and that makes them popular.

Windows XP can be used for a range of computers. This is quality which lacks in other OS. Most of them are designed to function in certain computers. Windows XP can boot on any computer available today. It suits well for 1998 model processors to the new ones. It is a direct advantage from Windows XP. Windows XP has only a low price when we account its innovative features. Windows XP became popular in a short span of time. It exceeded some OS like Linux, even though Linux is a freeware.

This is because of the user friendly interface of Windows XP. It is very easy to use and feels same for an expert and a common man. Anyone can directly use this OS. It won't take any study time before installing it. It has wizards for doing each task setup. Wizards are helpful and give step by step information for a process.

Microsoft redesigned whole GUI for Windows XP. One reason for the popularity of Windows XP is its GUI. Windows XP supports wireless networking. Windows XP is easily available for users.

Windows XP provides tools and programs designed to keep computer safe, manage systems, and perform regularly scheduled maintenance that keeps computer running at optimum performance. Highly appreciated feature in Windows XP is its plug-and-play features. the information about an active network connection like the duration of a connection and the speed at which initially connected can be obtained by using the Status menu command.

## 7.5 Code Conventions

Code conventions are important to programmers for a number of reasons. 80% of the lifetime of piece software goes to maintenance. Hardly any software is maintained for its whole life by the original author. Code conventions improve the readability of the software, allowing engineers to understand new code more quickly and thoroughly. If source code has a product, there is a need to make sure it is as well packaged and clean as any other product created.

- Code should be robust and error free.
- Code should be easy to use and understand
- Code should be easy to maintain.

### 7.5.1 Naming Conventions

Naming conventions make programs more understandable by making them easier to read. They can also give information about the functions of the identifier-For example, whether its a constant, package or class-which can be helpful in understanding the code. The conventions given in this section are high level. The naming rules for the identifiers are explained below:

- Methods: Methods should be verbs, in mixed case with the first letter lower case, with the first letter of each internal word capitalized. Example: generateE().

- Variables: Variable name should be short yet meaningful. The choice of a variable name should be mnemonic-i.e, designed to indicate to the casual observer the intent of its use. Example: ServerSocket ss: JTextField jtf;

### 7.5.2 File Organization

File name should be same as the main class in the package with the extension .java for java source files, .class for byte code files, .jar for binary files, .txt for text files, .sql for sequel files. The files name should reflect the functionality defined/implemented in that file. Files with logical connection should reflect that connection in their names wherever possible. Files belonging to the same module should reflect that dependency by a file name.

### 7.5.3 Declaration

- Function Declaration: All functions should be preceded by a documentation comment describing the function, its arguments and return code(s). Open brace "" appears at the beginning of the line following the declaration statement Closing brace "" starts a line by itself.
- Variable Declarations: Initialization of a variable should be done in a separate statement to its declaration, but as soon after the declaration as possible.

### 7.5.4 Class Declaration

Class names should be noun in mixed case with the first letter of each internal word capitalized. The class name must be simple and descriptive and can use whole words, avoid acronyms and abbreviations.

### 7.5.5 Comments

Comments should be used to give overviews of code and provide additional information that is not readily available in the code itself. Comment should contain only information that is relevant to reading and understanding the program.

## 7.6 Modules

- Login & Registration Module
- Augmented Reality Module
- Virtual Reality Module

### 7.6.1 Login & Registration Module

This module is used to implement the login and registration of users. The details of the users are stored in a SQL database. Encryption techniques are used to encrypt passwords of users.

### 7.6.2 Augmented Reality Module

This module makes use of libraries required to implement the Augmented Reality feature and performs location-based mapping using GPS coordinates. Depending on the user's location, the Augmented Reality view, which is the markers are displayed.

### 7.6.3 Virtual Reality Module

This module implements the Virtual Reality feature and makes use of Google Cardboard, which is a virtual reality (VR) platform developed by Google. We used gyroscope and compass to find the user's location and change the rendered environment accordingly. The user is presented with a 360 degree view of the tourist spot selected.

# Chapter 8

# TESTING

Testing is an important phase in the development life cycle of the product this was the phase where the error remaining from all the phases was detected. Hence testing performs a very critical role for quality assurance and ensuring the reliability of the software. Once the implementation is done, a test plan should be developed and run on a given set of test data. Each test has a different purpose, all work to verify that all the system elements have been properly integrated and perform allocated functions. The testing process is actually carried out to make sure that the product exactly does the same thing what is suppose to do. Testing is the final verification and validation activity within the organization itself. In the testing stage following goals are tried to achieve:-

- To affirm the quality of the project.
- To find and eliminate any residual errors from previous stages.
- To validate the software as the solution to the original problem.
- To provide operational reliability of the system.

During testing the major activities are concentrated on the examination and modification of the source code. The test cases executed for this project are listed below. Description of the test case, steps to be followed; expected result, status and screenshots are explained with each of the test cases.

## 8.1 Testing Methodologies

There are many different types of testing methods or techniques used as part of the software testing methodology. Some of the important types of testing are:

- Functional Testing

- Performance Testing
- Security Testing
- Usability Testing
- Compatibility Testing
- Recoverability Testing

### 8.1.1 Functional Testing

The functional testing of Mobiles normally consists in the areas of testing user interactions as well as testing the transactions. The various factors which are relevant in functional testing are:

- Type of application based upon the business functionality usages (banking, gaming, social or business)
- Target audience type (consumer, enterprise, education)
- Distribution channel which is used to spread the application (e.g. Apple App Store, Google play, direct distribution)

The most fundamental test scenarios in the functional testing can be considered as :

1. To validate whether all the required mandatory fields are working as required.
2. To validate that the mandatory fields are displayed in the screen in a distinctive way than the non-mandatory fields.
3. To validate whether the application works as per as requirement whenever the application starts/stops.
4. To validate whether the application goes into minimized mode whenever there is an incoming phone call. In order to validate the same we need to use a second phone, to call the device.
5. To validate whether the phone is able to store, process and receive SMS whenever the app is running. In order to validate the same we need to use a second phone to send sms to the device which is being tested and where the application under test is currently running.
6. To validate that the device is able to perform required multitasking requirements whenever it is necessary to do so.

7. To validate that the application allows necessary social network options such as sharing, posting and navigation etc.

8. To validate that the application supports any payment gateway transaction such as Visa, Mastercard, Paypal etc as required by the application.

9. To validate that the page scrolling scenarios are being enabled in the application as necessary.

10. To validate that the navigation between relevant modules in the application are as per the requirement.

11. To validate that the truncation errors are absolutely to an affordable limit.

12. To validate that the user receives an appropriate error message like "Network error. Please try after some time" whenever there is any network error.

13. To validate that the installed application enables other applications to perform satisfactorily, and it does not eat into the memory of the other applications.

14. To validate that the application resumes at the last operation in case of a hard reboot or system crash.

15. To validate whether the installation of the application can be done smoothly provided the user has the necessary resources and it does not lead to any significant errors.

16. To validate that the application performs auto start facility according to the requirements.

17. To validate whether the application performs according to the requirement in all versions of Mobile that is 2g, 3g and 4g.

18. To perform Regression Testing to uncover new software bugs in existing areas of a system after changes have been made to them. Also rerun previously performed tests to determine that the program behavior has not changed due to the changes.

19. To validate whether the application provides an available user guide for those who are not familiar to the application.

### 8.1.2 Performance Testing

This type of testing's fundamental objective is to ensure that the application performs acceptably under certain performance requirements such as access by a huge number

of users or the removal of a key infrastructure part like a database server.

The general test scenarios for Performance Testing in a Mobile application are:

1. To determine whether the application performs as per the requirement under different load conditions.
2. To determine whether the current network coverage is able to support the application at peak, average and minimum user levels.
3. To determine whether the existing client-server configuration setup provides the required optimum performance level.
4. To identify the various application and infrastructure bottlenecks which prevent the application to perform at the required acceptability levels.
5. To validate whether the response time of the application is as per as the requirements.
6. To evaluate product and/or hardware to determine if it can handle projected load volumes.
7. To evaluate whether the battery life can support the application to perform under projected load volumes.
8. To validate application performance when network is changed to WiFi from 2G/3G or vice versa.
9. To validate each of the required the CPU cycle is optimization
10. To validate that the battery consumption, memory leaks, resources like GPS, Camera performance is well within required guidelines.
11. To validate the application longevity whenever the user load is rigorous.
12. To validate the network performance while moving around with the device.
13. To validate the application performance when only intermittent phases of connectivity is required.

### 8.1.3 Security Testing

The fundamental objective of security testing is to ensure that the application's data and networking security requirements are met as per guidelines. The following are the most crucial areas for checking the security of Mobile applications:

1. To validate that the application is able to withstand any brute force attack which is an automated process of trial and error used to guess a person's username, password or credit-card number.
2. To validate whether an application is not permitting an attacker to access sensitive content or functionality without proper authentication.
3. To validate that the application has a strong password protection system and it does not permit an attacker to obtain, change or recover another user's password.
4. To validate that the application does not suffer from insufficient session expiration.
5. To identify the dynamic dependencies and take measures to prevent any attacker for accessing these vulnerabilities.
6. To prevent from SQL injection related attacks.
7. To identify and recover from any unmanaged code scenarios.
8. To ensure whether the certificates are validated, does the application implement Certificate Pinning or not.
9. To protect the application and the network from the denial of service attacks.
10. To analyze the data storage and data validation requirements.
11. To enable the session management for preventing unauthorized users to access unsolicited information.
12. To check if any cryptography code is broken and ensure that it is repaired.
13. To validate whether the business logic implementation is secured and not vulnerable to any attack from outside.
14. To analyze file system interactions, determine any vulnerability and correct these problems.
15. To validate the protocol handlers for example trying to reconfigure the default landing page for the application using a malicious iframe.
16. To protect against malicious client side injections.
17. To protect against malicious runtime injections.
18. To investigate file caching and prevent any malicious possibilities from the same.

19. To prevent from insecure data storage in the keyboard cache of the applications.
20. To investigate cookies and preventing any malicious deeds from the cookies.
21. To provide regular audits for data protection analysis.
22. Investigate custom created files and preventing any malicious deeds from the custom created files.
23. To prevent from buffer overflows and memory corruption cases.
24. To analyze different data streams and preventing any vulnerabilities from these.

### 8.1.4 Usability Testing

The usability testing process of the Mobile application is performed to have a quick and easy step application with less functionality than a slow and difficult application with many features. The main objective is to ensure that we end up having an easy-to-use, intuitive and similar to industry-accepted interfaces which are widely used.

1. To ensure that the buttons should have the required size and be suitable to big fingers.
2. To ensure that the buttons are placed in the same section of the screen to avoid confusion to the end users.
3. To ensure that the icons are natural and consistent with the application.
4. To ensure that the buttons, which have the same function should also have the same color.
5. To ensure that the validation for the tapping zoom-in and zoom-out facilities should be enabled.
6. To ensure that the keyboard input can be minimized in an appropriate manner.
7. To ensure that the application provides a method for going back or undoing an action, on touching the wrong item, within an acceptable duration.
8. To ensure that the contextual menus are not overloaded because it has to be used quickly.
9. To ensure that the text is kept simple and clear to be visible to the users.
10. To ensure that the short sentences and paragraphs are readable to the end users.

11. To ensure that the font size is big enough to be readable and not too big or too small.

12. To validate the application prompts the user whenever the user starts downloading a large amount of data which may be not conducive for the application performance.

13. To validate that the closing of the application is performed from different states and verify if it re-opens in the same state.

14. To ensure that all strings are converted into appropriate languages whenever a language translation facility is available.

15. To ensure that the application items are always synchronized according to the user actions.

16. To ensure that the end user is provided with a user manual which helps the end user to understand and operate the application who may be not familiar with the application's proceedings.

Usability testing is normally performed by manual users since only human beings can understand the sensibility and comfort ability of the other users.

### 8.1.5 Compatibility Testing

Compatibility testing on mobile devices is performed to ensure that since mobile devices have different size, resolution, screen, version and hardware so the application should be tested across all the devices to ensure that the application works as desired. The following are the most prominent areas for compatibility testing:

1. To validate that the user Interface of the application is as per the screen size of the device, no text/control is partially invisible or inaccessible.

2. To ensure that the text is readable for all users for the application.

3. To ensure that the call/alarm functionality is enabled whenever the application is running. The application is minimized or suspended on the event of a call and then whenever the call stops the application is resumed.

### 8.1.6 Recoverability Testing

1. Crash recovery and transaction interruptions

2. Validation of the effective application recovery situation post unexpected interruption/crash scenarios.

3. Verification of how the application handles a transaction during a power failure (i.e. Battery dies or a sudden manual shutdown of the device)

4. The validation of the process where the connection is suspended, the system needs to re-establish for recovering the data directly affected by the suspended connection.

# Chapter 9

# CONCLUSION & FUTURE SCOPE

## 9.1 Conclusion

The purpose of this project has been to identify the benefits of using Augmented Reality in tourism mobile applications. Due the use of a hybrid platform approach of implementation of the system, the iterative process of developing was carried out without any problem. The obtained results have demonstrated that potential users of this type of application would accept and use in their touristic experiences. Moreover, this kind of systems would encourage people to visit unusual touristic places by themselves because, with the Use of Augmented reality and Location-Based systems, it is possible to get in touch with the environment and at the same time enhance the on-site tourist experience.

## 9.2 Future Scope

Future work could use Augmented Reality in mobile application to implement smart cities concept over the world. However, one identified limitation of the developed application is the use of internet as requisite; hence, it is important to implement further functionality that allows the use of Augmented Reality in Off-line environments. The Virtual Reality feature can be extended to include live feeds at tourist spots.

# References


[1] M. Haller, M. Billinghurst and B. Thomas:*Emerging technologies of augmented reality, 1st ed. Hershey*, Idea Group Pub., 2007

[2] *National Development Plan / National Plan for Good Living*, 2013-2017 Summarized Version, 1st ed. Quito, 2013

[3] Lazar, J. Feng and H. Hochheiser:*Research methods in human-computer interaction 1st ed. Chichester, West Sussex, U.K.*, Wiley, 2010

[4] C. Wohlin:*Experimentation in software engineering, 1st ed. Berlin*, Springer, 2012

[5] H. Sharp, Y. Rogers and J. Preece:*Interaction design, 1st ed. Chichester*, Wiley, 2007

[6] K. Leichtenstern and E. Andre:*User-centred development of mobile interfaces to a pervasive computing environment, pp. 114–119*, 2008

[7] G. Booch, J. Rumbaugh and I. Jacobson:*The Unified Modeling Language user guide, 1st ed. Reading, Massachusetts*, Addison-Wesley, 1999

[8] Developer.android.com, 'Design — Android Developers', 2014. Available: `https://developer.android.com/design/index.html`

[9] S. Pousty and K. Miller:*Getting started with OpenShift, 1st ed. Sebastopol*, CA: O'Reilly Media, 2014