Informatics Institute of Technology

In collaboration with

University of Westminster, UK.

ARmax – A Cross-Platform Augmented Reality Development Kit

A dissertation by

Mr. G. R. A. Aponso

Supervised By

Mr. S. Hettiarachchi

Submitted in partial fulfillment of the requirements for the
BEng (Hons) Software Engineering degree
Department of Computing

May 2013

Declaration

I hereby certify that this project report and all the artefacts associated with it is my own work and it has not been submitted before nor is currently being submitted for any degree program.

Full Name of Student: Goniyamalimage Rumesh Antony Aponso

Registration No.: 2008087

Signature: …………………………….   Date:…………………………….

To my parents…

Abstract

Augmented Reality has become a third eye to the day-to-day user providing a whole new dimension to the computing world. With the technological development of the mobile industry Augmented Reality concepts have soon been adapted to mobile devices.

However, due to the complexity of Augmented Reality features and the variation of mobile platforms, developers tend to use various native Augmented Reality Software Development Kits and Application Programming Interfaces when developing mobile Augmented Reality applications. Conversely, when it comes to the project level most of the clients require their applications to be run on a number of mobile platforms. Using different native Software Development Kits and Application Programming Interfaces is not an optimal solution because it forces a developer to do a substantial amount of re-work there by causing a lot of problems to the developer.

ARmax, a cross platform Augmented Reality Tool Kit is being proposed, which helps to develop Augmented Reality based applications and experiences for mobile platforms. ARmax is a JavaScript based Augmented Reality Tool Kit (framework) and it is introduced to create Augmented Reality objects/experiences to the modern web browsers which supports new technologies like HTML5 and CSS3.

In order to demonstrate the features of the AR framework a reference application was developed. A critical testing and evaluation was carried out to ensure the efficiency and ease of use of the ARmax framework as well as the prototype, where the testing verified 89.25% success rate. In addition, the evaluation feedback of the domain experts verified the novelty of the proposed solution and the benefits such as ease of use and platform independency.

Subject Descriptors:

- H.5.1 – Multimedia Information System
- H.5.2 – User Interface

Keywords:

- Augmented Reality, Hypertext Navigation and Maps, Input Devices and Strategies, Interaction Styles, Screen Design

Acknowledgments

I would not have been able to complete this project successfully, without the kind support and help from many individuals.

First and foremost I would like to thank my head of departments Ms. Naomi Krishnarajah, to my supervisor Mr. Saman Hettiarachchi and to the other lecturers, whose encouragement and guidance has been of invaluable assistance in preparing this report.

Also, I would like thank Zone24x7, Inc. for the support and guidance provided during the internship year.

Also, a big thank you to all my batch mates and friends, for their constant guidance and supervision at all times.

Last but not least, thank you to my parents and family for their unending support and encouragement through the duration of this project.

Table of Contents

List of Figures

List of Tables

List of Charts

List of Code Snippets

List of Test Plans

List of Abbreviations

Abbreviation	Definition
2D	Two-Dimensional
3D	Three-Dimensional

API	Application Programming Interface
AR	Augmented Reality
ARML	Augmented Reality Markup Language
AV	Augmented Virtuality
CDMA	Code Division Multiple Access
CSS	Cascading Style Sheets
DOM	Document Object Model
DOM	Document Object Model
FPS	Frames per Second
GPS	Geo Positioning System
HMD	Head-Mounted Display
HTML	HyperText Markup Language
IP	Internet Protocol
IT	Information Technology
JAD	Joint Application Development
JS	JavaScript
KML	Keyhole Markup Language
KML	Keyhole Markup Language
MAC	Media Access Control
MR	Mixed Reality
NDEF	NFC Data Exchange Format
NFC	Near Field Communication
OOA	Object-oriented Analysis
OOD	Object-oriented Design
OS	Operating System
PC	Personal Computer
PDA	Personal Digital Assistant
POI	Point of Interest
RFID	Radio-frequency Identification
RIM	Research In Motion
SDK	Software Development Kit
SRS	Software Requirement Specification
SVG	Scalable Vector Graphics
UI	User Interface

1. Chapter 1 – Introduction

1.1. Chapter Overview

This chapter will provide an overview to the project where it commences with a brief introduction about Augmented Reality and Mobile Cross-Platform Application Development with the problems faced by the developers today. Furthermore, this includes the aim, objectives, scope and the features of the proposed approach. It also includes the tentative project plan, which will be followed throughout the project. Finally this concludes the outline of chapters to follow.

1.2. Problem Domain

Augmented Reality (AR) has become a third eye to the day-to-day user providing a whole new dimension to both real world and computing world. With the technological development of the mobile industry AR concepts have been adopted to mobile devices in a very short period of time. Geo Location based and Image Processing based AR applications have astonished the world and have become popular among both IT developers as well as IT consumers around the globe.

However, the differentiation of mobile platforms leaves a problem to developers, which is to build applications using native languages for each platform or use a cross-platform framework.

"Smartphones and tablets are becoming ubiquitous, and, with them, mobile apps. But what does the proliferation of different devices mean for developers entering the market? Do you target a single platform, build your app twice for iOS and Android, or use a cross-platform framework?" (Chipperfield, 2012, http)

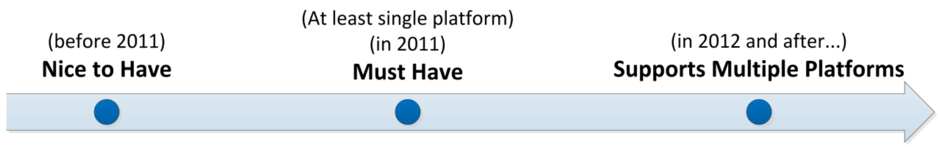


Figure 1.1 - Variation of Mobile Applications Strategy

As shown in Figure 1.1 in 2011, having a mobile applications strategy shifted from "nice to have" to "must have" for businesses of all sizes. In 2012, that same shift took place from "supports single mobile platform" to "supports multiple mobile platforms". Most of the companies/businesses/brand targets a number of platforms rather than having one. Android, iPhone, iPad, Windows and BlackBerry are a few of the most popular platforms among them. (Warren, 2012, http)

"Developing for all of these platforms is a challenge, especially for the developer or business with limited resources." (Warren, 2012, http)

What are the existing solutions?

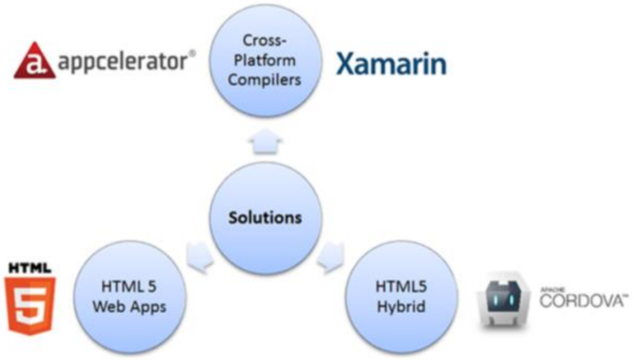


Figure 1.2 - Cross-Platform Mobile Application Development Solutions

As shown in Figure 1.2 above, there are three main aspects for these cross-platform solutions. The idea is to have a single codebase while deploying the application in multiple mobile platforms. As with any development strategy, this cross-platform approach is associated with pros and cons. The pros and cons are presented in the Table 1.1.

Criteria	Pro	Con
Code is reusable	✔	
Plugins support	✔	
Easy for web developers	✔	
Reduced development costs	✔	
Easy deployment	✔	
Device/OS features support		✔
Performance		✔
High-end graphics and 3D support		✔
Vendor lock-in	✔	

Table 1.1 - Cross-Platform Pros and Cons (Warren, 2012, http; Shapiro, 2012, http)

However, due to the complexity of AR features and the variation of mobile platforms, developers tend to use various native AR SDKs and APIs. However, when it comes to the project level most of the clients require their applications to run on a number of mobile platforms. Using different native SDKs and APIs is not an optimal solution because it forces a developer to do a substantial amount of re-work, thereby causing a lot of problems to the developer.

In September 2011 an Austrian company named Mobilizy has revealed a cross-platform AR developer kit called ARchitect. (Veronica, 2011, http) The ARchitect engine is Wikitude's technology framework for creating Augmented Reality content. It is the underlying platform of both the Wikitude app as well as Wikitude's SDK for third party application development based on common web technologies including HTML, CSS and JavaScript.

However, the idea of cross-platform AR development or browser based AR is still in its early stage, where there are lot of areas and functionalities/experiences to be covered such as image/marker recognition and tracking, geo-objects (icons, shapes) which appear in the AR view in specific geo-locations, etc. Therefore the need of an efficient, user-friendly and easy to use cross-platform AR tool kit is high.

ARmax, a cross platform AR Tool Kit is being proposed, which helps to develop AR based applications and experiences, for both mobile and desktop platforms. ARmax is a single code base solution. Having a single code based AR Tool Kit would be easier for developers rather than having many native AR SDKs and APIs.

In order to evaluate ARmax solution, a few Reference Applications are being developed to demonstrate features/functionalities of the ARmax Development Kit (framework) and present day AR experiences such as Geo Location (sensor based) based AR and QR Code based (Image Processing based) AR. The purpose of these applications is to evaluate ARmax in terms of efficiency and ease of use.

1.3. Problem Definition

From the above problem domain it is derived that the use of different native AR SDKs and APIs for different mobile platforms is not an optimal solution for developers.

1.4. Project Aim and Scope

To design, develop and evaluate a highly efficient and easy to use cross platform (such as Apple iOS and Google Android) augmented reality framework for developers, while also demonstrating this approach with a few reference applications.

Elaborating more on the aim, an AR framework named ARmax will be developed to enable developers to build cross-platform AR applications. In order to demonstrate the features of the AR framework a few reference applications will be developed. These reference applications will provide the AR experiences based on Geo Location (sensor based) and QR Code (Image

Processing based). The ARmax framework will be evaluated against Native AR APIs and SDKs in terms of efficiency and ease of use.

1.5. Project Objectives

In order to achieve the stated aim of the project the following objectives have been identified.

- Prepare TOR to identify the feasibility of the project.
- Carryout a literature survey in the following areas:
    - Elementary Augmented Reality concepts to understand the background of Augmented Reality.
    - Existing AR techniques and technologies in,
        - Sensor based (including Geo Location approaches)
        - Image Processing based

      areas to understand what and how the AR concepts have been implemented during the past and to gain an insight about the future.
    - Existing AR applications, tools and frameworks in above mentioned areas.
    - Determine the most optimal AR features and functionalities which should develop and evaluate in proposed ARmax framework.
    - Analyse features and functionalities of existing solutions.
- Conduct a survey to determine:
    - Limitations and improvement that should take place in existing AR frameworks.
    - Existing problems faced by developers while dealing with mobile cross platform application development and improvements expected.
    - Determine the tools, techniques and technologies which will be used in the prototype as well as in the project.
- Determine and formulate the requirement specification for the proposed ARmax framework with the use of the information gathered via survey and literature survey.
- Design and implement a prototype according to the requirement specification to demonstrate the features and functionalities of the ARmax framework and reference application for selected area(s).
- Carry out a critical testing on implemented prototype and ARmax framework via specified testing criteria and test cases.
- Carry out a critical evaluation of the ARmax framework as well as the prototype. The evaluation should cover all major requirements specified in the requirement

specification. The evaluation will include experts and research student’s opinion as well as a comprehensive testing on the prototype in terms of efficiency and ease of use.

- Identify areas for future enhancement.
- Document all the steps involved throughout the project to track the progress and for improved management.
- Submissions of Interim Project Report, Report on Prototype and Final Project Report to evaluate the project progress.

1.6. Features of the Prototype

In order to demonstrate features of the ARmax framework, a simple commercial reference application will be developed targeting a general category for multiple mobile platforms such as Apple iOS and Google Android. And also, the features of the reference application will be focused in the following scenario.

Scenario

Assume a developer is to develop a mobile application, which allows the user to subscribe to shops they often go to, or products and services which they often consume. The user will receive promotions, which are published by the owners of the shops/services. The promotions will be displayed as a geo-object around the user via AR, where the user will be able to interact with the geo-object by touching/clicking on it.

Key Features

I. Image/Marker recognition and tracking.
   - Users can subscribe to shops/products/services by scanning the image/marker which is created by shop owners through a reference application.

II. Geo-location based objects with customisable radar.
   - Users will have the ability to view promotions around him. Promotions will be displayed as geo-objects that will be displayed at the predefined geographical location in AR view. Geo-object will not tie to static visual representations such as icon, text labels or different shapes. It will support multimedia formats such audio and video.

To support these functionalities in the reference application, the following features of ARmax framework have been identified and categorised as follows,

General

- Supports a wide range of device features and sensors
    - An API will be provided to the developer to access device features and sensors.
- Supports geo-objects and geo-location
    - This will provide easy manipulation (such as create, update and delete) of geo-location based objects.
- Supports image/marker recognition
    - An API will be provided to the developer to implement image/marker recognition and tracking functions.
- Supports object layers and virtual object rendering on top of camera view.

Visualisation

- Drawables (shapes, paths, images, text, etc.)
- Text labels, images, animations, audio and video
- Customise radar and direction indicator
    - Radar will show the geo-objects around the user in customisable radar view.
    - Direction indicator will be used to guide the user to a specific geo-location.

Event/Interaction

Following features will capture the user inputs in order to trigger various functionalities.

- Binding events (click, touch, etc.)
- Event listener

1.7. Project Deliverables

Throughout the project the following will be produced.

- Terms of Reference
- Literature Review including the findings and results of survey as well as the questionnaires and the interview questions.
- Software Requirement Specification

- Software Design Specification
- Interim Project Report
- Test Plan and Test Results
- Prototype
- Report on Prototype
- Final Project Report including critical evaluation results as well as identified areas for future enhancement.

1.8. Resource Requirements

In order to successfully carry out this project, a number of hardware and software requirements have been identified. The identified resource requirements are in the Appendix VIII – A.

Note: Above requirements are subject to change.

1.9. Tentative Activity Schedule

Refer Appendix I.A to view the tentative project plan.

1.10. Outline of Chapters to Follow

Chapter 2 – Literature Review

Chapter 3 – Project Management

Chapter 4 – Requirements

Chapter 5 – Design

Chapter 6 – Implementation

Chapter 7 – Testing and Evaluation

Chapter 8 – Conclusion

1.11. Summary

This chapter provided an overview about the project and it was identified that the use of different native AR SDKs and APIs for different mobile platforms is not an optimal solution for developers. Thus, there was a need of a cross-platform AR framework. Accordingly, the aim and the scope of the ARmax project were defined. In order to achieve the specified aim, a number of objectives such as, preparing TOR to identify the project, carrying out a literature survey in AR and mobile application development, determining and formulating the requirement specification, designing and implementing a prototype, carrying out a critical testing on the implemented prototype, carrying out a critical evaluation of the ARmax framework, identifying areas for future enhancement and submission of various reports were defined. Furthermore, real time object identification through different sensors, virtual object rendering and interaction types were specified as features of the prototype with the use of an appropriate scenario. Also, the identified project deliverables and tentative resource requirements were briefly discussed. Lastly, a tentative project plan was presented, which will be followed throughout the project and the chapter concluded with an outline of chapters to follow.

The next chapter will discuss the study that has been carried out for this project. It will review the critical facts of current knowledge together with the substantive findings, theoretical and methodological contributions towards mobile augmented reality and mobile cross-platform application development.

2. Chapter 2 – Literature Review

2.1. Chapter Overview

Previous chapter discussed the problem domain and problem definition of the ARmax project, where it further elaborated the aims, objectives, scope and features of the ARmax framework and the prototype.

This chapter will provide an overview to the literary features of the problem domain. A study was carried out in augmented reality and mobile cross-platform application development, where it discussed approaches, techniques and technologies of particular areas. Accordingly, this chapter discusses the features that browsers support, and finally concludes with a review on an existing product related to the specified problem domain in section 1.2 (Problem Domain).

2.2. Augmented Reality (AR)

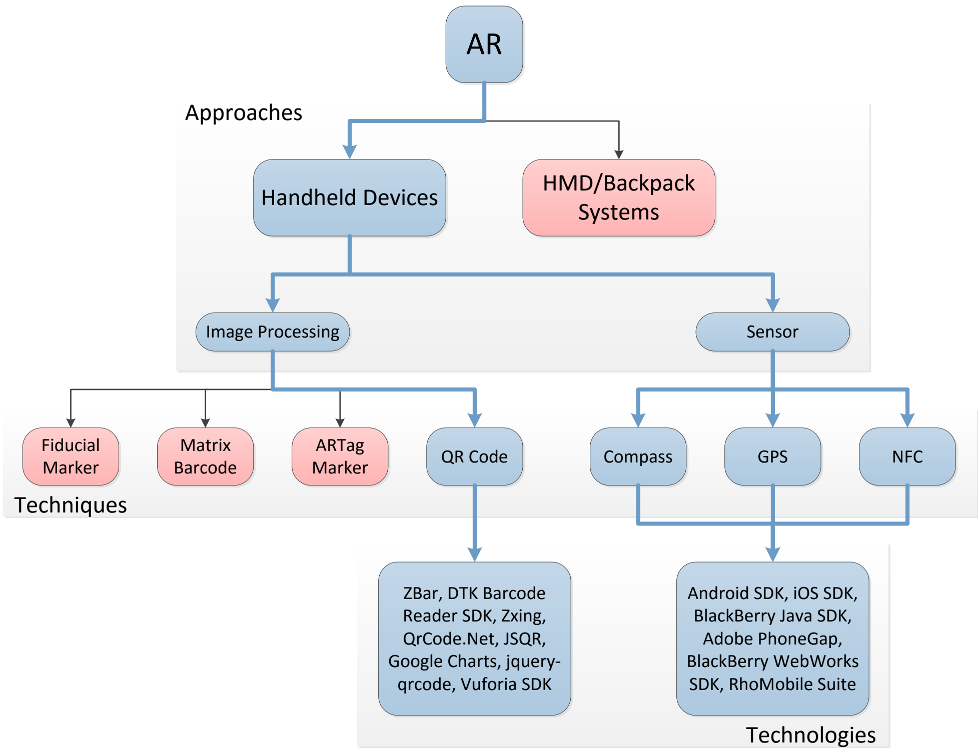


Figure 2.1 - Relevant Tree

The study in AR will follow the highlighted path in the “relevant tree” in Figure 2.1, where it discusses approaches, techniques and related technologies in corresponding area(s).

2.2.1. Background

In general overlaying virtual (computer graphics) objects on top of a real world environment is known as Augmented Reality (AR).

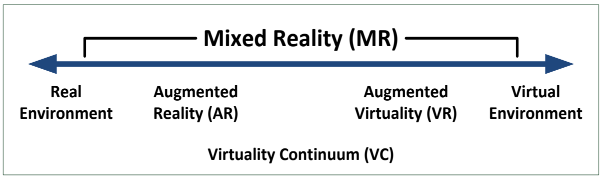


Figure 2.2 - Simplified Representation of a Virtuality Continuum (Milgram and Kishino, 1994)

As shown in Figure 2.2, Milgram and Kishino (1994) introduced the "virtuality continuum" which completely connects the real environment into a virtual one. Furthermore they proposed an operational definition for AR. In 1997, Azuma defined AR as a variation of Virtual Environment (VE) or Virtual Reality (VR) where the VE technologies completely immerse a user into an artificial world without allowing vision of the real world.

At present, Milgram's and Kishino's (1994) Virtuality Continuum and Azuma's (1997) definition are commonly accepted among the IT community.

However, long before Milgram's and Azuma's definitions, Ivan Sutherland created the first HMD AR system in 1968. Due to the limited processing power at that time, it could only display a very simple wireframe drawing which overlaid in the real world. (Sutherland, 1968)

Following Sutherland's revolutionary HMD AR system, Caudell and Mizell (1992) described the design and prototyping steps they have taken toward the implementation of the HUDset. In 1997, Feiner et al. presented The Touring Machine and in 1999, they created the first Mobile Augmented Reality System (MARS) which allowed the user to freely walk around while having all the necessary equipment mounted on his back. Platforms such as AR-Quake, BARS and Tinmith are similar to the MARS, but different in application areas. (Thomas et al. 2000; Julier et al. 2000; Piekarski and Thomas, 2002, p 1; Wagner, 2007, p 16)

But as is with any technology, Wagner and Schmalstieg (2003) described the pros and cons associated with these backpack/HMD systems (see Table 2.1). Thus, it is felt that these systems are not practical enough with the modern day-to-day life style.

Criteria	Pro	Con
Backpack/HMD systems are associated with higher performance with hands-free operation.	✔	
These systems are often designed to prove the concept of prototypes, thus it does not provide a practical form factor.		✔
All equipment of these systems is normally mounted to backpack, which is bulky and heavy.		✔
To operate these systems it needs a skilled user, therefore it is not practical and generally insupportable.		✔

Table 2.1 - Pros and Cons Associated with Backpack/HMD Systems (Wagner and Schmalstieg, 2003)

Soon after the innovation of the first commercial camera phone in year 2000, AR applications moved into mobile handheld devices such as cell phones and PDAs. Wagner and Schmalstieg (2003) also preferred very small form factor devices and display such as cell phones and handheld devices as an alternative for backpacks/HMDs.

In 2001, Newman et al. presented a PDA based wireless AR system named the BatPortal. Vlahakis et al. (2001) developed a mobile AR system for cultural heritage sites with a navigation interface, 3D models of ancient temples and statues and animated avatars. In 2003, Wagner and Schmalstieg developed an indoor AR guidance system which directly runs on a PDA. Wagner used a Windows Mobile port of ARToolKit for tracking and provided a 3D augmented view of the environment for the user. Mozzies, was the first commercial mobile phone AR camera game released with Siemens SX1 and it was awarded the best mobile game in 2003. Visual Codes (Rohs and Gfeller, 2004), The Invisible Train (Wagner et al. 2005) and AR-Tennis (Henrysson et al. 2005) are a few handheld device based applications which are highlighted in the field of mobile AR.

Furthermore, Wagner (2007) highlights a few main issues with many AR backpack/HMD systems such as high complexity of hardware setup, which often includes expensive commercial sensors, which are often bulky and fragile. Also, these devices were never meant to be operated by untrained users. For these reasons handheld devices might be better than backpacks/HMDs systems in terms of reliability and practical use. However when it comes to processing power, handheld devices have much to improve. Dr Wray of University of Kingston determines that by year 2020, a typical processing power of tablet or smartphone will be increased by 20 times or more than the current processing power (Wray, 2012, http). Therefore it seems that day-by-day handheld devices are becoming more powerful.

Thus it has been carefully chosen that handheld devices such as smartphones might be good to test the prototypes throughout the project.

2.3. Handheld AR Approaches

Throughout these years as discussed above, quite a few handheld AR approaches have been used. As shown in Figure 2.3, following approaches have been identified as main handheld AR approaches.

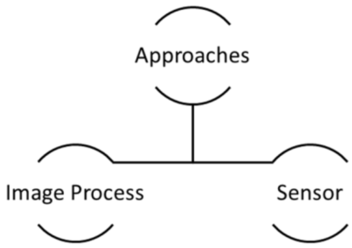


Figure 2.3 - Handheld AR Approaches

2.4. Image Processing Based AR Techniques

As shown in Figure 2.4, marker tracking was the leading image processing based techniques used by various researchers throughout these years.

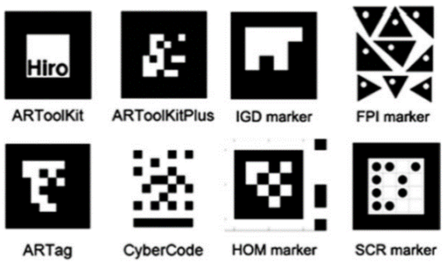


Figure 2.4 - Some Well-Known AR Markers (Furth, 2011, 346)

2.4.1. Matrix (2D) Barcode Tracking

Rekimoto (1996) introduced a new technique to generate AR systems over 2D matrix markers (square shaped barcode) (see Figure 2.4), which can identify a large number of objects.

2.4.2. Fiducial Marker Tracking

Fiducial markers (see Figure 2.4) are commonly known as “easy to identify markers” and it is being used in many vision-based tracking systems which are utilised in AR applications (Owen et al. 2002).

2.4.3. ARTag Marker Tracking

Fiala (2005) introduced the ARTag (see Figure 2.4), a fiducial marker system, which uses digital coding theory to get a very low false positive and inter-marker confusion rate with a small required marker size.

2.4.4. QR Code Tracking

QR code or quick response code (see Figure 2.5) is a trademark type of matrix (2D) barcode. It was invented by the Denso Wave Incorporated in 1994, where it has many advantages such as, (Furth, 2011, p 341 – 344)

- Symbol version – The symbol versions range from Version 1 to Version 40.
- Data Capacity – Table 2.2 shows the data capacities of typical 2D barcodes.

Criteria	QR Code	Data Matrix	Maxi Code
Developer (Country)	Denso (Japan)	RVSI Acuity CiMatrix (USA)	UPS (USA)
Type	Matrix	Matrix	Matrix
Data Capacity Numeric Alphanumeric Binary Kanji	 7,089 4,296 2,953 1,817	 3,116 2,355 1,556 778	 138 93 Not supported Not supported
Main Features	Large capacity Small printout size High scan speed	Small printout size	High scan speed
Main Usages	All categories	Factory automation	Logistics

Table 2.2 - Typical 2D Barcodes (Furth, 2011, 343)

- Character type – Table 2.3 shows the conversion efficiency of QR code.

Character Type	Data Capacity
Numeric only	3.3 cells/character
Alphanumeric	5.5 cells/character
Binary (8 bits)	8 cells/character
Kanji, full width Kana	13 cells/character
Chinese (UTF-8)	24.8 cells/character

Table 2.3 - Conversion Efficiency of QR Code (Furth, 2011, 343)

- Error correction level – Table 2.4 shows error correction capabilities of QR code.

Correction Level	Error Correction Capability
Level L	Approx. 7%
Level M	Approx. 15%
Level Q	Approx. 25%
Level H	Approx. 30%

Table 2.4 - QR Code Error Correction Capabilities (Furth, 2011, 343)

- Module size – 21 × 21 - 177 × 177 modules (size grows by 4 modules/side)

Furthermore, Wang et al. (2010) substituted the conventional marker (such as fiducial marker) of AR with QR code, in which they took the advantages of QR code, such as error correction capability and larger data capacity.

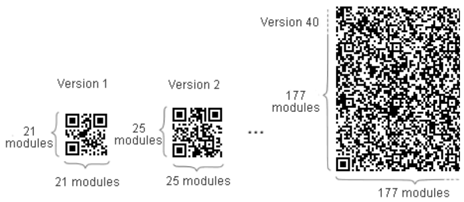


Figure 2.5 - A Sample QR Code

2.4.5. Conclusion on Image Processing Based AR Techniques

As per the discussion above, there were quite a few prominent marker based AR techniques as well AR SDKs such as ARToolKit, ARTag and ARToolKitPlus (Kato and Billinghurst, 1999; Fiala, 2005; Wagner and Schmalstieg, 2007).

However, Furth (2011) and Wang et al. (2010) pointed out the following advantages and disadvantages of QR code markers over conventional markers (such as fiducial marker and ARTag marker). Table 2.5 shows the comparison between QR code and general AR markers. Table 2.6 shows the advantages and disadvantages of conventional markers.

Criteria	QR Code	Webtag	Regular AR marker
Need to pre-register	No	Yes	Yes
Model storing	Internet	Internet	Local
Limited number of markers	$> 10^{7089}$	$4x10^{12}$	Smaller
Universality	Universal barcode	Stand-alone	Stand-alone

Table 2.5 - Comparison between QR Code and General AR Markers (Furth, 2011, 348)

Criteria	Advantage	Disadvantage
The black square border of conventional marker and the intrinsic pattern	The black border offers a system as a tracking indication. Therefore it is easier to track the marker for AR systems.	Once the border is occluded and become incomplete, the system is unable to detect the marker any more. The specification of the marker is not easy for the users to create their own markers. Marker specifications: • Thickness of black border • Gap between border • Intrinsic pattern
The amount of information which can be carried in a marker		The data capacity and data types supports are less compared to QR code. **ARTag** – The intrinsic pattern contains 10-bit binary code, which represents up to 1024 tag IDs. **ARToolKitPlus** – The intrinsic pattern contains 9-bit binary code, which represents up to 512 tag IDs.

Table 2.6 - Advantages and Disadvantages of Conventional Markers (Wang et al. 2010)

According to the above comparison done by Wang et al. (2010) ARTag marker is less error prone than ARToolKit and ARToolKitPlus fiducial markers. However, the amount of information ARTag marker can carry is limited compared to QR code.

Thus, as per the discussion in section 2.4.4 (QR Code Tracking) above, QR code seems to be less error prone and improved in many ways such as large capacity, small printout size and high scan speed over the conventional markers such as data matrix marker, fiducial marker and ARTag marker. Furthermore, QR code is universally used by consumers around the globe in various fields. Therefore QR code marker tracking might be more beneficial for AR systems than the conventional markers tracking in the future.

Moreover as a new trend, recognizing and tracking random natural images in the real world has turned out to be another image processing based technique recently. Qualcomm is one of the leading researchers in this area.

2.5. QR Code Related Technologies

Throughout these years, a lot of researches have been carried out and various QR code related projects such as QR code recognition based on image processing, Recognition of QR Code with mobile phones and Design and implementation of augmented reality system collaborating with QR code, where they implemented new QR code recognition algorithms/systems and modified existing ones (Chang et al. 2007; Liu et al. 2008; Wang et al. 2010; Gu and Zhang, 2011; Jin, 2012).

In addition, there are a number of open source and commercial libraries for QR code encoding and decoding such as ZBar, DTK Barcode Reader SDK, Zxing, QrCode.Net, JSQR, Google Charts and jquery-qrcode. Also, Qualcomm's AR Vuforia SDK is one of the top recognising and tracking library for both markers and natural images in real world. At present, these libraries are widely used by the developers. (Brown, 2010, http; DTK Software Company, 2012, http; Zxing, 2012, http; Microsoft, 2012, http; Duttke, 2012, http; Google, 2012, http; Etienne, 2012, http; Qualcomm, 2012, http)

Moreover, the QR code technologies which will be used in the ARmax framework will be decided later on in the implementation phase of the project.

2.6. Sensor Based AR Techniques

As per the discussion in section 2.2.1 (Background), there were a number of sensor based AR techniques being used in the past. However as a technique, geo location based applications such as Archeoguide, Human Pacman, MapLens, Nokia Mara, Wikitude, and Layar were highlighted in the mobile sensor based AR field, where it uses GPS and compass as primary sensors. (Vlahakis et al. 2001; Cheok et al. 2004; Morrison et al. 2009; MARA, 2012, http; Wikitude, 2012, http; Layar, 2012, http)

Also, it has already been started to experiment on new techniques such as NFC and how it could be combined with AR applications (Madden, 2011, http). Therefore, it is good, that the proposed ARmax framework could support these new techniques as well, while providing easy access to the developer.

2.7. Geo-Location and NFC Related Technologies

Many handheld devices come with developer SDKs such as Android SDK, iOS SDK and BlackBerry Java SDK, where it enables the developer to access built-in sensors (compass, GPS, accelerometer, nfc) and various features such as camera, contacts, notification settings in the device. Also there are tools and libraries such as Adobe PhoneGap, BlackBerry WebWorks SDK and RhoMobile Suite, which provide access to sensors and various features (mentioned above) in handheld devices.

Moreover, the sensor based technologies which will be used in the ARmax framework will be decided later on in the implementation phase of the project.

2.8. Mobile Application Development

“Mobile application development is a new animal, and IT leaders shouldn’t expect to tame it using their same old tricks.” (Finneran, 2011, http)

During the last decade, mobile application development has rapidly grown into one of the main aspect in the software industry. As per the discussion in the section 1.2 (Problem Domain), Warren (2012, http) describes how the mobile application strategy shifted throughout the last few years and at present it has become a new trend.

Due to various mobile platforms such as Apple iOS, Android, Windows Mobile, etc., software companies and developers face a huge challenge in planning mobile application development and deployment. As shown Figure 2.6, they have to consider the following areas with the entire management of the mobile application development including security, maintenance and support in order to satisfy the clients’ requirements. (Finneran, 2011, http)

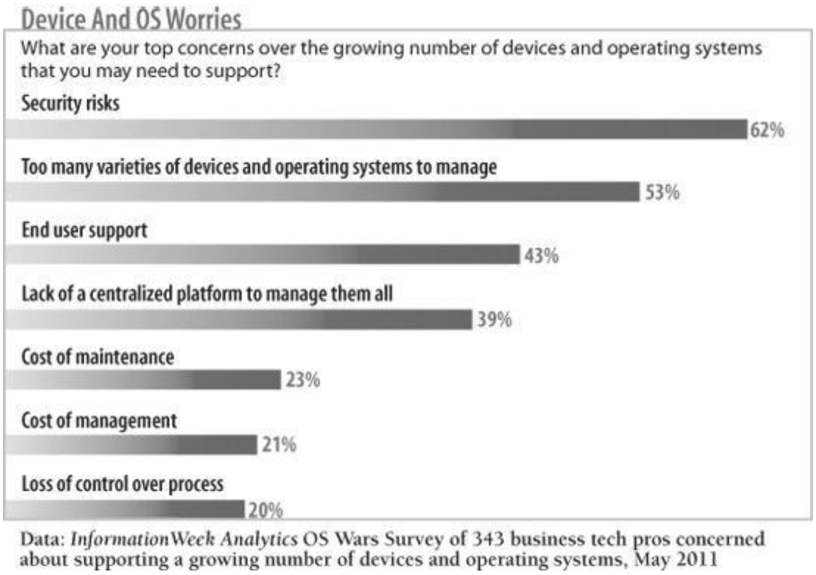


Figure 2.6 – Device and OS Worries (Finneran, 2011, http)

Thus, as shown in Figure 2.7, there are two main approaches in mobile application development, which is used by developers and software companies presently.

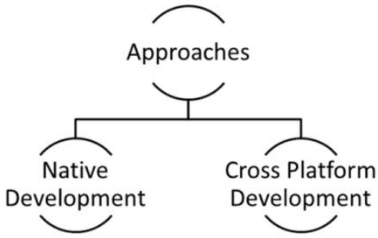


Figure 2.7 - Mobile Application Development Approaches

Most companies tend to select cross-platform approach to provide a mobile solution quickly while having single code base. Thus, it is less complex comparing with native languages. Furthermore, most of the cross-platform solutions are web based, where the look and feel of the applications could present similar to a native application (Finneran, 2011, http). On the other hand, native approach is been used when the application uses high-end graphics or 3D content, where it requires more processing power.

However, as with any approach, these approaches are also associated with pros and cons (see Table 2.7).

Pro – Yes, Con – No

Criteria	Native	Cross-Platform
Code is reusable	Somewhat	Yes
Plugins support	Yes	Yes
Easy for web developers	No	Yes
Reduced development costs	No	Yes
Easy deployment	No	Yes
Device/OS features support	Yes	Somewhat
Performance	Yes	Somewhat
High-end graphics and 3D support	Yes	Somewhat
Vendor lock-in	Yes	Somewhat

Table 2.7 - Native vs. Cross-Platform Pros and Cons (Warren, 2012, http; Shapiro, 2012, http)

As a result, cross-platform approach is better when considering a wide range of mobile devices and platforms, end user support, maintenance cost and management cost of applications. On the other hand, there might be cases where native approach is more appropriate than the cross-platform approach. However it is always good to use cross-platform approaches at all times where it is possible.

2.9. Mobile Cross-Platform Solutions

Figure 2.8 - Mobile Cross-Platform Approaches

As shown in Figure 2.8, there are three main cross-platform solutions, which are used by software companies and developers at the moment. Most of the solutions are web based, where all of them are single code based. For example, Adobe PhoneGap, RhoMobile, MoSync, Appcelerator Titanium and Widget Pad are a few frameworks which were developed under these solutions. Furthermore, above frameworks are written in dynamic programming languages such as JavaScript, Ruby or Python, where it will finally produce native applications for supported platforms including Android, iOS and other mobile platforms.

2.9.1. Cross-Platform Compilers

In general cross-platform compilers have the capability to convert executable code into other platforms. Likewise, there are mobile cross-platform solutions such as Appcelerator Titanium and Xamarin, where the common code such as JavaScript or C#, can be compiled into native languages while combining the native feature of device/OS (Chipperfield, 2012, http).

2.9.2. HTML5 Web Applications

The difference between mobile applications and web applications have become unclear with the arrival of HTML5 and CSS3 (Chipperfield, 2012, http). With the new features of HTML5 web applications such as multimedia, accelerated graphics, local storage, semantics and forms, it enabled the capability to provide progressively richer client-side experience to the consumers around the globe.

2.9.3. HTML5 Hybrid Applications

The idea of this solution is to develop mobile applications using a combination of web technologies such as HTML5, CSS3 and JavaScript libraries and then wrap it with native code wrappers (see Figure 2.9). In addition, this solution combines the advantages of both HTML5 and native applications (Chipperfield, 2012, http). Adobe PhoneGap is one of the leading frameworks in this area, where it supports many native features and web technologies.

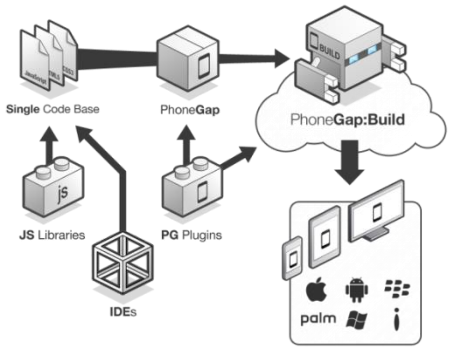


Figure 2.9 - PhoneGap Overview Diagram

2.9.4. Conclusion on Mobile Cross-Platform Solutions

As shown in Table 2.8, there are a number of mobile cross-platform solutions, which are primarily based on web technologies such as HTML, CSS and JavaScript.

Solution	Technologies	Supported Platforms	License
Adobe Air	HTML, JavaScript, Flex, Flash	Android, iOS, MeeGo, BlackBerry	Free
PhoneGap	HTML5, JavaScript, CSS	Android, iOS, webOS, BlackBerry, Symbian, Windows Phone 7, Bada	Free and open source under the apache license, version 2.0
OpenPlug	ActionScript, JavaScript, XML	Android, Symbian, iOS, Windows Phone	Free and commercial license available
WorkLight	HTML, JavaScript, CSS	Android, BlackBerry, iOS	Commercial license available
Appcelerator Titanium	JavaScript	Android, BlackBerry, iOS	Apache license, version 2.0 and commercial license available
RhoMobile	HTML, CSS, Ruby, JavaScript	Android, BlackBerry, iOS	Free and open source under the MIT license
WidgetPad	HTML, JavaScript, CSS	Android, iOS, WebOS	Free and open source

Table 2.8 – Cross-Platform Solutions - Web Technologies Based (Catariov, 2011, http)

Considering these cross-platform solutions, Adobe PhoneGap and Appcelerator Titanium are more popular among the developers. Both of these solutions are very different from each other, but then again they are implemented with similar web technologies such as JavaScript. Thus as shown in Table 2.9, they share different advantages and disadvantages when it comes to application development.

Criteria	Adobe PhoneGap	Appcelerator Titanium	Notes
JavaScript API	✓	✓	PhoneGap's API interacts as typical JS does in your web code. Appcelerator Titanium API is **NOT** web code; it is used to interact with native code.
Supports HTML5/CSS3	✓	✗	PhoneGap is a web application that runs in a native web browser view.
Supports web standards	✓	✗	PhoneGap looks, feels and develops like a standard web page. It is also subject to the same browser compatibility concerns.
Supports DOM based JS libraries	✓	✗	JS libraries that reference the DOM, like **jQuery, Prototype** or any of the new, **<canvas> based libraries** will only work with PhoneGap web views.
Native Code	✗	✓	Appcelerator Titanium creates a truly native application via a JS API that maps to native code.
Native UI/Performance	✗	✓	Appcelerator Titanium performance is limited only by the device. PhoneGap's is limited by the device web view.

Table 2.9 - PhoneGap vs. Appcelerator Titanium - Advantages and Disadvantages (Lukasavage, 2011, http)

As a result, Lukasavage (2011, http) suggests that, these solutions are not wrong choices when developing mobile applications because they have different and unique features. Therefore developers have a responsibility to select the correct solution depending on the scenario and the features it requires.

Thus, Adobe PhoneGap could be the choice for the implementation of the prototype of the ARmax framework, since it directly deals with graphical content. Also, new HTML5 features such as canvas, SVG and CSS3 2D/3D support and DOM based JavaScript library support might come in handy when developing the framework.

2.10. Mobile Browser Support

Even though it appears that all the web browsers that IT consumers encounter every day in devices such as personal computers, mobile phones, tablets, gaming consoles and televisions looks and perform in the same way, in reality they do not. There is a huge variety of browsers available and each of these browsers support a different set of features. Most of the browsers identify HTML, CSS and Scripts (i.e. JavaScript, VBScript, etc.) and is able to render the web pages without any problem.

The HTML 4.01 specification was published December 24, 1999 as a W3C Recommendation and it offered the same three variations as HTML 4.0. The last of its errata was published in May 12, 2001. Recently, the new HTML version that is HTML5 was published in January2008 as a working draft by W3C and it is still under development up-to-date. The W3C is developing a comprehensive test suite to achieve broad interoperability for the full specification by 2014, which is now the target date for Recommendation.

As a result, the browser creators such as Google, Microsoft, Mozilla, Palm, Apple Inc, Nokia and Research In Motion (RIM) are upgrading their browsers to support the features of HTML5 and CSS3. However, at present none of the browsers fully support every feature of the HTML5 partial specification published to date. Thus, it is important to recognise the features that mobile browsers such as Android browser, BlackBerry browser, Internet Explorer mobile and Safari support.

Moreover, it appeared that to test each browser manually was not practical due to constraints such as time and resources. Therefore, an online benchmark tool was used to test browser support. The online tool was “The HTML5 Test” (visit www.html5test.com). The tool gives points for every feature and if a browser passes all tests, it would receive a maximum score of 500 points and 15 bonus points. HTML5 defines an audio and video element, which allows the browser to play media files. The HTML5 specification does not define a specific codec. Hence, for each common codec that is supported it will be awarded bonus points. Similarly the tool awards bonus points for SVG and MathML support. Moreover, the bonus points are counted separately and do not count towards the maximum of 500 points.

However, it is true that anyone could argue that the scoring seems arbitrary and could question, who decides how many points are awarded for a particular feature. The site owners decided to award points for each feature depending on how important that feature is for web developers and how difficult it is to implement that feature. A small and simple feature would be worth less points than a large and complicated feature. They think that it is the most honest way to grade browsers, because otherwise a browser that only supports the small and simple features would score as high or higher than a browser that went the extra mile and decided to tackle the big features. However, in the end it is based on personal preference and all the browsers are given the same set of features and determine whether it supports that particular feature or not. This appears fair, because every browser is tested against same set of features.

Table 2.10 shows the statistics about HTML5 feature support of different desktop browsers.

Browser	Tested Devices	Score	Bonus
Maxthon 4.0	N/A	476	15
Chrome 26	N/A	468	13
Opera 12.10	N/A	419	9
Firefox 20	N/A	394	10
Safari 6.0	N/A	378	8
Internet Explorer 10	Microsoft Surface and others	320	6

Table 2.10 - Desktop Browser Benchmark Test by The HTML5 Test (Accessed Date: 2013-04-13)

Table 2.11 shows the statistics about HTML5 feature support of different mobile browsers.

Browser	Tested Devices	Score	Bonus
BlackBerry 10	BlackBerry Q10 or Z10	485	11
Chrome 25	All Android 4 devices	417	11
Opera Mobile 12.10	Multiple platforms	406	12
Firefox Mobile 19	Multiple platforms	399	14
iOS 6.0	Apple iPhone, iPad and iPod Touch	386	9
Windows Phone 8	Nokia Lumia 822, HTC 8X and others	320	6
Android 4.0	Samsung Galaxy Nexus	297	3
Bada 2.0	Samsung Wave and others	283	9
Nokia Belle FP 2	S60 5.5, Nokia 808 PureView and others	272	9
Android 2.3	Google Nexus S and others	200	1

Table 2.11 - Mobile Browser Benchmark Test by The HTML5 Test (Accessed Date: 2013-04-13)

Table 2.12 shows a comparison of HTML5 feature support on three different mobile browsers. The latest version of BlackBerry 10, iOS 6.0 and Android 4.0 browsers were compared. The results indicated that BlackBerry 10 browser is best, among browsers that were compared. Apart from the test data which has been provided by The HTML5 Test site, the benchmarking was carried out for a number of mobile phones. The benchmark results and the detailed feature support of the previous comparison are in the Appendix VII - Mobile Browser Support Benchmark Results.

Browser	Score	Bonus
BlackBerry 10	485	10
iOS 6.0	386	9
Android 4.0	297	3

Table 2.12 - Comparison between Mobile Browsers by The HTML5 Test (Accessed Date: 2013-04-13)

As a result, it appears that the design and the implementation phase of ARmax framework and the prototype should give more priority to these results when developing the proposed solution. Simultaneously, the results show that most of the modern browsers support the canvas element in the HTML5 specification, which is essential to the ARmax framework.

2.11. Existing Cross-Platform AR Product Review

2.11.1.ARchitect

In 2009 the creator of the Wikitude World Browser Mobilizy, has announced the idea of a standard AR Markup Language (ARML). The basic idea of the ARML is that the data can be viewed in any AR browser and also in Google Earth. This was a good approach to address the interoperability, which is one of the five barriers of AR according to Jolie O’Dell. Furthermore the proposed markup language is based on KML and it is a cross-platform development kit, which allowed the users to see many AR contents in many ways. (O’Dell, 2009, http) As per section 1.2 (Problem Domain), in September 2011, Mobilizy has publicized ARchitect, their new cross-platform AR developer kit which was announced in 2009.

2.11.2.Features of ARchitect

I. Full Support of HTML, JavaScript and CSS

Figure 2.10 - ARchitect Basic Concept of HTML, JavaScript, CSS (ARchitect Features, 2012, http)

Figure 2.10 shows ARchitect’s basic concept of HTML, JavaScript, and CSS. The core of the ARchitect is a JavaScript API. Also it supports HTML, CSS and other JavaScript libraries such as Google Analytics, jQuery, etc. (ARchitect Features, 2012, http)

It is good that ARchitect supports web technologies, because developers can quickly adapt to it due to less complexity of the code without having to learn new concepts or tools. With the support of other JavaScript libraries, UI of the applications could be built very easily and more attractively with many other features such as charts, maps, etc.

II. Geo-Objects and Geo-Location API

ARchitect provides an easy creation and manipulation over Geo-Location based objects with dynamic visual representation such as images, symbols, etc. and a number of event triggers. (ARchitect Features, 2012, http)

ARchitect does provide a good API for Geo-Location, but it does not provide enough API support to other native features of mobile devices such as Accelerometer, Camera, Contacts, Media, Network, Notification and other native features. Even though there are a number of open source APIs which support these features such as Adobe PhoneGap, ARchitect does not support such APIs. On the other hand in the proposed framework, the above stated feature has been considered, thus it is intended to support the open source APIs such as Adobe PhoneGap.

III. Image Recognition and Tracking

As shown in Figure 2.11, ARchitect supports image recognition and tracking marker along with natural images in the real world via image recognition capabilities of Vuforia SDK by Qualcomm. (ARchitect Features, 2012, http)

Figure 2.11 - ARchitect Feature Image Recognition (ARchitect Features, 2012, http)

Moreover, marker/image recognition and tracking capabilities of Vuforia SDK is quite impressive. However, to date it only supports Android and iOS platforms, and to use Vuforia in ARchitect, developers must register as a Qualcomm developer. This might lead to vendor lock-in, thus it is better to use alternative technologies as mentioned in section 2.5 (QR Code Related Technologies).

IV. Support for Shapes, Text, Images, Animations and Multimedia Formats

Figure 2.12 - ARchitect Multimedia Audio and Video Augmented Reality | ARchitect Drawables Images Text Augmented Reality | ARchitect Property Animations Augmented Reality (ARchitect Features, 2012, http)

As shown Figure 2.12, ARchitect supports a wide range of graphical content including multimedia formats such as audio and video. (ARchitect Features, 2012, http)

Via web technologies, it is easier to represent graphical content without putting much effort. ARchitect uses its own graphical content API to represent different shapes, images, text. Similarly there are open source JavaScript libraries such as KineticJS and Three.js, which combines the features of HTML5 canvas and WebGL. These libraries provide advanced features such as layering, styling, transitions, animations, geometries, lighting and materials than ARchitect graphic API.

V. Advanced Visualisations

As shown Figure 2.13, ARchitect provides pre-defined graphical elements such as radar and direction indicator, which enables a user to detect the Geo-Objects around him and guide them to it. (ARchitect Features, 2012, http)

Figure 2.13 - ARchitect Feature Radar | ARchitect Feature Direction Indicator (ARchitect Features, 2012, http)

These features are really helpful to the developer in terms of time saving and ease of use. It is good that proposed framework also could consider the stated features.

VI. Interactions

Figure 2.14 - ARchitect Feature Click Trigger Augmented Reality | ARchitect Features Field of Vision Trigger | ARchitect Feature Geo Trigger Augmented Reality (ARchitect Features, 2012, http)

As shown Figure 2.14, ARchitect has introduced a couple of new interaction methods such as Field of Vision (FOV), trigger and geo trigger (Geofence) along with the traditional interactions including click and touch (ARchitect Features, 2012, http).

Most of the traditional interaction methods can be done via earlier mentioned open source JavaScript libraries such as KineticJS and Three.js. However, the new vision based and geo location based interaction methods are quite innovative. FOV interaction method is unique to ARchitect. Thus, it is worth giving consideration to these new methods in the proposed framework.

2.11.3.Conclusion on ARchitect

According to the discussion above, the analysed features of ARchitect and ARmax (proposed framework) are listed as shown in Table 2.13.

Criteria	ARchitect	ARmax
Full support of HTML, JavaScript and CSS	Yes	Yes
Geo-objects and geo-location API	Yes	Yes
API support to other native features of mobile devices such as accelerometer, camera, contacts, media, network, notification, etc.	No	Yes
Marker/image recognition and tracking	Yes	Yes
Support for shapes, text, images, animations and multimedia formats	Somewhat	Yes
Advanced visualisations (radar and direction indicator)	Yes	Yes
Interactions	Yes	Yes
Vendor lock-in	Somewhat	No
Easy development and deployment	Yes	Yes

Table 2.13 - ARchitect vs. ARmax (proposed framework) - Features summary

ARchitect SDK covers the essentials of mobile cross-platform and browser based AR. However, as per the discussion above, there are some areas such as API support for handheld device features and graphic API that still need to be improved. For that reason, proposed framework might be a better alternative for ARchitect.

2.12. Summary

This chapter contains the study, which has been primarily carried out on mobile augmented reality and mobile cross-platform application development. The study commences with a background of augmented reality. The advantages and disadvantages of the two main approaches (i.e. Backpack/HMD Systems and Handheld Devices) are compared. The high complexity of hardware setup, often includes expensive commercial sensors, often bulky and fragile were highlighted as disadvantages of backpack/HMD systems. Thus, handheld approach was selected, because of its reliability and practical use.

Under handheld approach, two different sub approaches namely image processing based and sensor based were discussed. Under image processing based approach four different techniques were considered (i.e. Matrix (2D) Barcode, Fiducial Marker, ARTag Marker and QR Code). These techniques were compared against criteria such as data capacity, need for pre-registration, model storing, number of markers and universality. As a result, QR code seemed to be less error prone and improved in many ways such as large capacity, small printout size and high scan speed over the conventional markers (i.e. Matrix (2D) Barcode, Fiducial Marker and ARTag Marker). Thus, QR code technique was selected and the related technologies such as ZBar, DTK Barcode Reader SDK, Zxing, QrCode.Net, JSQR, Google Charts and jquery-qrcode were highlighted. Moreover new trends such as recognizing and tracking random natural images in real world were mentioned. Under sensor based technique, it was discovered that GPS and compass as primary sensors which were used in the past and as a new trend it has started to research on new sensors such as NFC and how it could be combined with AR applications. Accordingly, sensor based technologies were discussed and SDKs and libraries such as Android SDK, iOS SDK, BlackBerry Java SDK, PhoneGap, BlackBerry WebWorks SDK and RhoMobile Suite were highlighted. Correspondingly, it has been decided to select the technologies which will be used in the ARmax framework later on in the implementation phase of the project.

Moreover, two main mobile application development approaches (i.e. Native and Cross-Platform) were discussed. The cross-platform application development approach was reasoned and compared against the native application development approach. As a result, cross-platform approach was convinced over native approach, when considering the wide range of mobile devices and platforms, end user support, maintenance cost and management cost of applications.

Afterwards, three cross-platform solutions such as cross-platform compilers, HTML5 web application and HTML5 hybrid applications were discussed in detail. The features of each solution were compared. As a result, it seemed that Adobe PhoneGap which comes under HTML5 hybrid applications solution could be the choice for the implementation of the prototype of the ARmax framework.

Furthermore, the features that browsers support were discussed in detail. It appeared that to test each browser manually, is not practical due to constraints such as time and resources. Therefore, an online benchmark tool named “The HTML5 Test” was used to test browser support. The benchmarking was carried out for mobile browsers and desktop browsers. According to the results, it appeared that the design and the implementation phase of ARmax framework and the prototype should give more priority to these results when developing the proposed solution. Also, the results showed that most of the modern browsers supports canvas element in the HTML5 specification.

ARchitect SDK covers the essentials of mobile cross-platform and browser based AR. However As per the discussion above, there are some areas such as API support for handheld device features and graphic API that still needs to be improved. For that reason the proposed framework might be a better alternative for ARchitect.

Lastly, an existing cross-platform AR product was analysed. ARchitect was selected as the existing cross-platform AR product. The features such as full support of HTML, JavaScript and CSS, geo-objects and geo-location API, image recognition and tracking, support for shapes, text, images, animations and multimedia formats, advanced visualisations and interactions were discussed in detail. Furthermore, it was identified that the proposed ARmax framework was more enhanced over ARchitect in a number of areas such as API support for handheld device features and graphic API. Also, it was discovered that ARchitect has vendor lock-in to some extent, whereas ARmax does not.

The next chapter will discuss the project management aspect of the project. The selection of a development methodology, planning, risk management and monitoring criteria will be discussed in detail.

3. Chapter 3 – Project Management

3.1. Chapter Overview

Previous chapter discussed the study which was carried out in AR and mobile application development. The approaches, techniques and related technologies were discussed with provided evidence.

This chapter provides an overview to the project management phase of the ARmax project. The primary challenge of the project management is to achieve all of the project objectives and aim. Thus, this chapter will discuss the primary aspects such as planning, resource allocation, time allocation, risks identification and mitigation and monitoring of the ARmax project.

3.2. Planning

3.2.1. Selection of Development Methodology

Recently, Augmented Reality projects have started to get more attention by the day-to-day consumers'. Thus, the ARmax project has a very rapidly changing environment around it in terms of technology, concepts and user interaction. Therefore, a traditional waterfall or spiral methodologies are not suitable to manage and control changes. Waterfall development lifecycle has fixed phases and linear timelines. Thus, it is not capable of addressing the challenges in the modern software development domain. However, modern methodologies such as RAD or JAD could have been better for this project. Nevertheless, when it comes to gathering requirements, these process models use the workshop method or JAD sessions. For that reason, it is infeasible to carry out such sessions and RAD or JAD might not be the optimal development methodology for the ARmax project.

As a result, a hybrid approach combining both agile and prototyping methods are selected as the development methodology and will be followed during the project life cycle. By following such a method it will give the opportunity to the users and experts to give their feedback at any time of the project life cycle. In addition, it allows to gain more control over the change and to manage it.

Moreover, to gain more control and manage over the ARmax project, PRINCE2 is used as the project management methodology. Also, the experience gained in the 2nd year of the degree program was one of the main reasons to select PRINCE2 as the project management methodology. Furthermore, PRINCE2 is a process-driven project management method which contrasts with reactive/adaptive methods such as Scrum. Also, it is based on seven principles, seven themes and seven processes which lead to a quality product. Thus, some themes such as quality, risks, change and progress are given priority and will be discussed in the subsequent sections.

3.2.2. Selection of Tools, Techniques and Technologies

Microsoft Project will be used to create and record all the major activities and milestones during the project. The importance of a CM tool is highly recommended to make this project successful. Therefore, TortoiseSVN will be used as the CM tool for this project.

3.2.3. Finalised Project Plan

The finalised project plan of this project is in the Appendix I – Project Plan.

3.2.4. Resource allocation

This project is an individual project and the research student is the only human resource it has. Thus, all the tasks of this project are allocated to him.

3.2.5. Time allocation

At the start of the project, the project deliverables and the deadlines were defined. Therefore, the time allocation was done by considering the pre-defined deadlines. Some of the deadlines became milestones for particular tasks and some was defined separately. Refer Appendix I – B for refined project plan.

3.3. Risk Management

Risk management is a very important factor in project management, because poor risk management might lead to project failure. Thus, risk identification and mitigation plays a huge role in the path of project success.

3.3.1. Risk Identification and Mitigation

The identification of risks, assessment, prioritisation and the respective mitigations are listed down in the risk log in Appendix II – Risk Log.

3.4. Monitoring of the Project

Monitoring regularly and control according to the deviations plays a key role in the path of success of the project. Poor monitoring and control might lead to project failure. Therefore, project performance is observed and measured regularly to identify deviations from the project management plan. In order to overcome the deviations, necessary actions were taken.

3.4.1. Monitoring Criteria of the Project Progress

The log book is taken as the monitoring criteria throughout the project.

3.5. Summary

In this chapter, a discussion about the project management phase of this project was carried out. Planning, risk management and monitoring the project progress are the primary areas that this chapter considers. Under planning, a number of development methodologies such as waterfall, spiral, JAD, RAD, prototyping and agile were discussed and compared. As a result, a hybrid approach combining both agile and prototyping methods were selected as the development methodology, because it allows to gain more control over change and to manage the ARmax project. Accordingly, PRINCE2 was selected as the project management methodology for the ARmax project. The experience with the PRINCE2 methodology was one of the main reasons to select it. The risks which could occur during the time period of the project and the possible mitigations were discussed. Moreover, a risk log was produced to track and manage the risks. Finally, under monitoring, how the monitoring will be measured regularly to identify variances was discussed and a record book was selected as the monitoring criteria of the ARmax project.

The next chapter will discuss the requirements of this project, what requirement elicitation techniques were used to extract the requirements form the users and results and analysis of the results.

4. Chapter 4 – Requirements

4.1. Chapter Overview

Previous chapter provided an overview to the project management phase of the ARmax project, where it conversed about planning, resource allocation, time allocation, risks identification and mitigation and monitoring of the progress of ARmax project.

This chapter will provide an overview to the requirements gathering phase of the ARmax project. It initiates with identifying the stakeholders and the influence they have on this project. Then it discusses the requirements elicitation techniques that were used to gather requirements for this project. Afterwards, it analyses and discusses the results of the gathered requirements with the use of analysis models such as UML. Lastly it discusses the derived functional and non-functional requirements, risks, mitigations and scope refinement.

4.2. Project Title

ARmax Framework - A Cross-Platform Augmented Reality Development Kit

4.3. Stakeholders

4.3.1. Onion Model

Stakeholders are anyone who has an interest or a gain upon a successful completion of a project. They could be an individual, an organisation or another system and may have a positive or a negative influence on the particular project. The main stakeholders that have been identified of the ARmax framework are shown in Figure 4.1.

Figure 4.1 - Onion Model of the ARmax

4.3.2. Stakeholder Descriptions

The roles of the identified stakeholders and their viewpoints towards the proposed framework are listed down in Table 4.1.

Stakeholder	Viewpoint
Beneficiary	
Functional Beneficiary	
End User	Wants reliable and easy to use AR application
Developer (User/Consumer)	Wants an easy to use ARmax framework
Project Manager	Wants to finish the project successfully with customer satisfaction in order to make profit.
Financial Beneficiary	
Developer (Research Student)	Wants to make profit by selling the ARmax framework.
CEO / Shareholder	Wants to earn profit from the project.
Negative	
Competitor (Existing Products)	Wants to market their product by competitively improving features.
Regulatory	
Voluntary (Standardising)	

Regulators	Looks the standards of the ARmax framework and its functionalities are accorded with the conventional standards.
Enforcing	
Government	Wants the ARmax framework and its functionalities to meet the conventional standards.
Media	Wants to review the ARmax framework and its behaviour.
Operational Roles	
Human Operators	
Developer	Wants the ARmax framework functioning well with fewer errors.
Administrator	Wants to manipulate map information easily.
Neighbouring Systems	
Cross-platform mobile development systems	Wants to couple new features as plugins, into the existing systems.

Table 4.1 - List of Appropriate Stakeholder Roles of the ARmax Framework

In order to provide a good product, requirements elicitation and requirement gathering plays a key role in the project life cycle of every project. Thus, the following requirements elicitation techniques were used to capture the requirements of the stated stakeholders above.

4.4. Selection of Requirements Elicitation Techniques

As per the discussion in the chapter 2 (Literature Review), most of the theoretical aspects and requirements were considered. But to experience the problem which is faced in the real world and to capture the requirements they need, a practical approach is required. Therefore, in order to gather requirements for proposed ARmax framework, the following requirements elicitation techniques were considered.

4.4.1. Literature Review

The study, which was carried out in AR and mobile application development has covered most of the theoretical aspects. Moreover, it is a great source of knowledge to gather requirements for ARmax framework, because an analysis was carried out to identify the pros and cons of the projects that have been carried out in the past and the present. Thus, literature review can be considered as a requirement elicitation technique to gather requirements for the ARmax framework.

4.4.2. Questionnaire

First and foremost this is an individual research project and it is constrained time, cost and resources. Therefore a quicker and reliable requirement gathering method is required with less expense. Distributing questionnaires is a classic requirements gathering method, where it can be used to collect requirements from a large number of people in a short amount of time with less expense.

On the other hand, interviews would have been the best method to gather requirements if there were no such constraints as time and cost. Also, within the given time constraint it is not possible to meet many experts. Gunda (2008, p 11-12) suggests that when the available resources and funds are less, questionnaires could be the best way to gather requirements from a large number of people in a very short amount of time. Also, he highlights that the effectiveness and the design of the questionnaire, target audience of the questionnaire and honesty of the respondents could affect the results.

When considering the above facts, distributing questionnaires was the best technique to gather requirements for proposed ARmax framework.

4.4.3. Prototyping

A prototype is a visual representation of the actual system, where it delivers the main features/functionalities and the work flow.

Gunda (2008, p 13-14) points out that requirements gathering should be completed first to continue with the development. But then again a question arises, how will the additional requirements be discovered before the development? Brainstorming or JAD could have been a better solution for this matter. However, brainstorming requires a group of people who has the knowledge that is specific to the technological areas and the problem domain. Furthermore, in JAD sessions it requires stakeholders and users apart from the mentioned brainstorming group. Therefore, it is not possible to conduct a brainstorm or a JAD session within the given time period and the availability of the resources. Thus, the solution is prototyping, which provides both functional and graphical aspects of the product or the system which is going to be developed. Furthermore, the prototyping technique is associated with pros such as reduced time and cost of development, provides a visualisation of the system, user satisfaction and areas for future enhancements can be identified. As cons, it could lead to an unfinished product or system and the final product could turn out to be the same as the prototype.

Accordingly, considering the facts above, prototyping was selected to extract the additional requirements of the user and to reduce the time and the cost for development. Thus prototyping could be considered an improved technique to gather requirements for proposed ARmax framework.

4.4.4. Social Analysis / Observation

The notion of observation is collecting requirements by observing people or existing systems. This method is being used to extract additional requirements from the user, where the user is unable to express the needs of the new system or the problems existing in the current system (Gunda, 2008, p 12-13). As per the discussion in the Questionnaires and Prototyping, interviews or brainstorming could have been better solutions if there were no constraints as mentioned. Thus, observations technique was used to fill the additional requirements gaps between the system and the user, because it is possible with given constraints.

Since prototyping and observation techniques are linked, these techniques were used to gather requirements for proposed ARmax framework.

4.5. Execution of Requirements Elicitation Techniques

A real survey was conducted among the professionals in the software industry. The target audience of the survey were professionals such as mobile application developers and project managers in different companies, who have experience in both native and cross-platform mobile application development. The questions of the survey aimed to extract the requirements and opinions on different development techniques, problems they face with existing tools and features which they would expect from a new tool such as ARmax. The questionnaire was built using as online survey tool (Google Docs) and the survey link was distributed among the professionals. The results gathered from the survey is analysed and discussed in the section 4.6 (Results of the Survey).

Furthermore, the main parameters that were used to this survey are presented in the Appendix III. The online survey can be found in Appendix XI – A.

Apart from the online survey, two other requirements elicitation techniques were executed by providing a paper prototype to a selected group of users (users includes normal users and professionals) and observing how they would interact with it. The feedback and the opinions of the users were recorded. Also, the data and requirements gathered from these techniques were used to analyse results. The results are presented in the next section.

4.6. Results of the Survey

In this section, the results gathered from an online survey are listed below. Graphs are used for representation and easy understanding.

The survey was conducted among fifty professionals employed in approximately five different software companies in Sri Lanka. The participants were mainly developers and project managers, who has years of experience in the mobile application development field. Furthermore, forty-two participants answered the questions, where eight had not answered. The results are as follows,

4.6.1. Mobile Platforms

Chart 4.1, shows the mobile platforms that are often used among professionals and to which mobile platforms the applications are being developed. According to the results shown, Android, iOS and Symbian were the most frequently used mobile platforms. Similarly, Android, iOS and Windows Phone were the mobile platforms that most of the applications are being developed for. This is an important factor, because the proposed solution should support the mobile platforms that are frequently used.

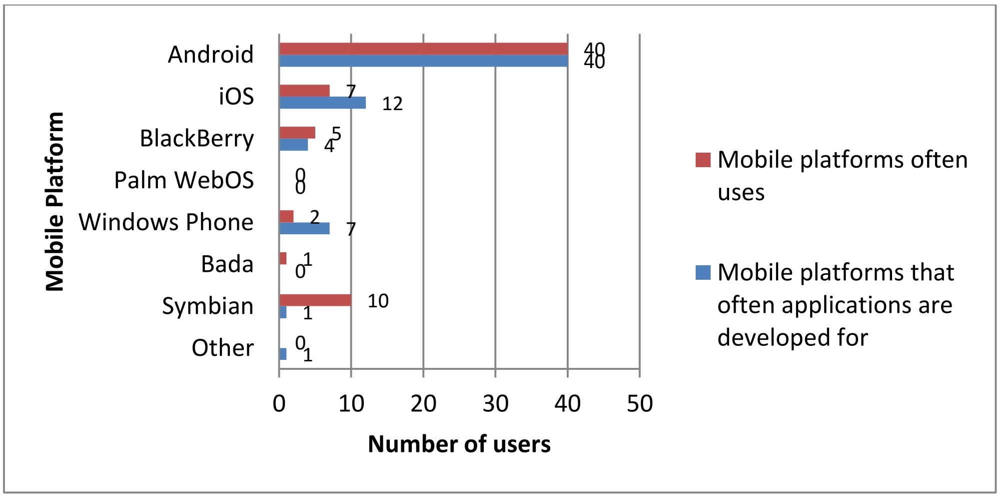


Chart 4.1 - Mobile Platforms Often Used Vs. Mobile Platforms that Often Applications are Developed For

4.6.2. Native vs. Cross-Platform Mobile Application Development

As shown in Chart 4.2 and Chart 4.3, the respondents were asked to rate different criteria with the experience in respective areas. It was interesting to find out that the results plotted in the graphs are similar to the findings of chapter 2 (Literature Review). As per the discussion in chapter 2 (Literature Review) and according to the results below, cross-platform development approach could be considered a better approach.

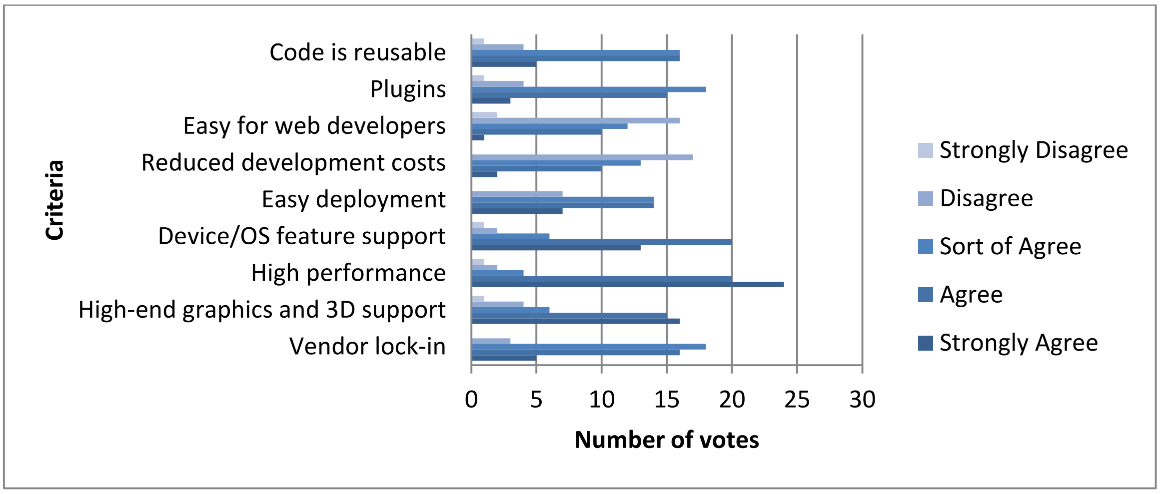


Chart 4.2 - Native Mobile Application Development

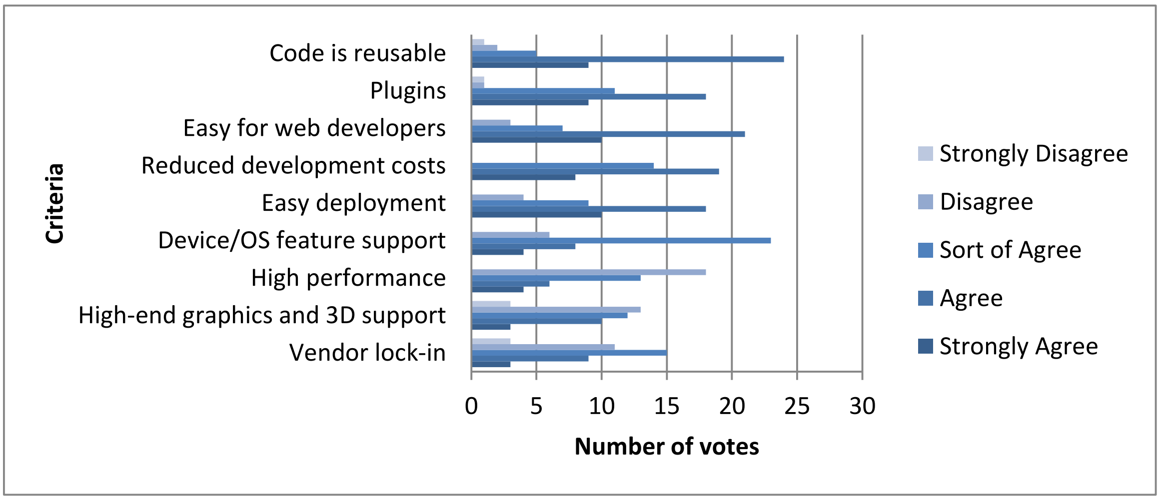


Chart 4.3 - Cross-Platform Mobile Application Development

4.6.3. Mobile AR Application Development

As shown in Chart 4.4, the respondents were asked, what would be the optimal approach to develop mobile AR applications according to their experience. As expected, majority of the respondents selected the native approach. However, surprisingly 45% of the total respondents selected the cross-platform approach. This indicates that the idea of cross-platform is still new among the professionals and there is more opportunity to discover and to improve.

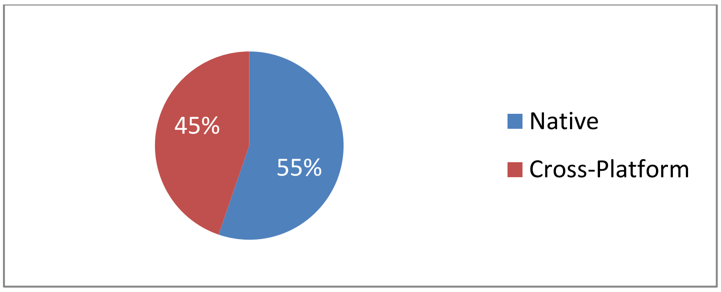


Chart 4.4 - What approach do you often choose to develop mobile augmented reality applications?

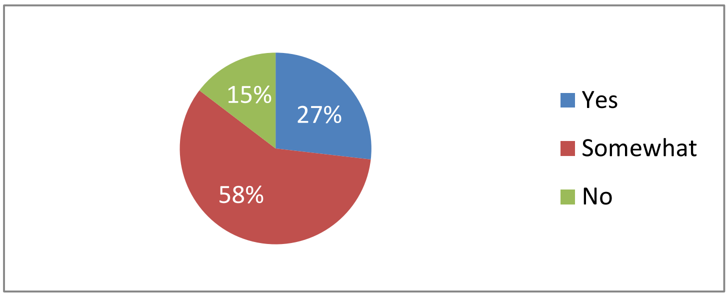


Chart 4.5 - Do you think cross-platform approach would be better to develop mobile augmented reality applications?

The respondents were asked if they think cross-platform approach would be better to develop mobile AR applications. As shown in Chart 4.5, most of the respondents are in between yes and no, where approximately a quarter of the respondents are confident about the cross-platform approach.

4.6.4. Mobile Cross-Platform Frameworks

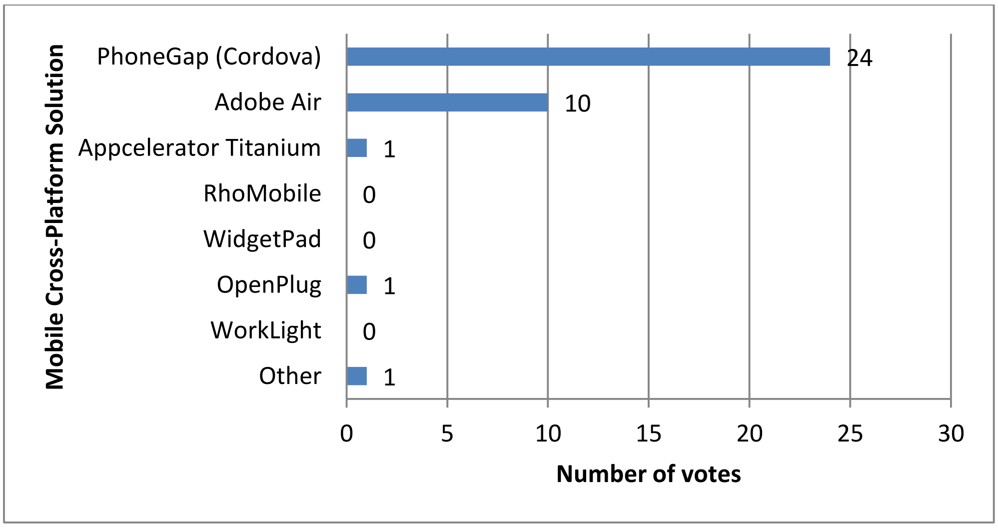


Chart 4.6 - Mobile Cross-platform Frameworks Often Uses

The respondents were asked what was the framework or tool that they often use when developing cross-platform mobile applications. As shown in Chart 4.6, most of the professional are familiar with PhoneGap and Adobe Air.

Apart from results discussed above, the other findings of the survey are listed down in the Appendix IV.

4.7. Analysis

4.7.1. Analysis Techniques and Tools

Object-oriented analysis (OOA) applies object-modelling techniques to analyse the functional requirements for a system. In order to analyse the gathered requirements and to extract the functional requirements, object-oriented analysis is selected as the analysis technique. Experience with the OOA technique for the past few years was one of the main reasons to select OOA as the analysis technique. And also, there are number of different notations to represent different elements of different analysis models. Unified Modelling Language (UML) is a visual language, which is widely used in the software industry. Also, one main reason for selecting UML is it’s the standard language to analyse and design OO systems. Thus, UML is selected as the model to represent the analysed requirements.

Furthermore, there are a number of different tools to create UML models such as Microsoft Visio, Rational Rose XDE and StarUML. The experience gained working with Microsoft Visio and StarUML for the past few years and StarUML being open source and free, were the reasons to select Microsoft Visio and StarUML as the modelling tools.

4.7.2. Analysis of the Results

4.7.2.1. Experimental Survey

In the survey conducted, there were interesting results to be observed. They are:

- Android and iOS are the mobile platforms that are often used and the applications that are often developed (see Chart 4.1).
- Advantages of using cross-platform approach is that code is reusability, easy for web developers, reduced development cost, easy deployment and reduced vendor lock-in (see Chart 4.3).

- Advantages of using native approach are device/OS feature support, high performance, high-end graphics and 3D support (see Chart 4.2).
- Most of the professionals are familiar with mobile application development. But, when it comes to cross-platform mobile application development, it appears that the professionals are not too familiar with it. Similarly, as expected most of the professionals tend to select native approach for mobile AR development (see Chart 4.4). However, the results indicate that a considerable amount of professionals are interested in cross-platform mobile AR development (see Chart 4.5).
- Most professionals highlighted that performance issues, poor graphic quality and user interface issues are common with regard to cross-platform mobile application development (refer Appendix IV - Chart 10.8).

4.7.2.2. Literature Review

The results from the Literature Review were found to be more or less similar to the results of the survey conducted. They are:

- PhoneGap is the most familiar cross-platform framework among professionals. It appears that PhoneGap provides as easy implementation and easy deployment for a mobile application. Therefore, mobile cross-platform framework being chosen in the Literature Review was correct (see Chart 4.6).
- Most of the professionals are familiar with NFC technology and all of them think that combining NFC with AR applications may improve the user interaction/experience (refer Appendix IV - Chart 10.10 and Chart 10.11).

4.7.3. Analysis Models

In order to demonstrate the features of ARmax framework a prototype will be developed. The following subsections will discuss the features and functionalities of that prototype.

4.7.3.1. Use Case Model

The use case diagram shown in Figure 4.2, defines the high-level view of the functionalities which will be implemented in the prototype of ARmax framework. It includes two main categories such as test features and a day-to-day reference application. Test features mainly tests the elements such as compass and direction indicator, various user interactions and the device web view's performance. The reference application will demonstrate a real life scenario, where it displays the places as geo-objects around the user, in an augmented reality view. The end user will be able to interact with geo-objects by touching it. Also, features such as navigating to a particular place, set a place, scanning barcode/QR code and scanning NFC tags will be provided to the end user.

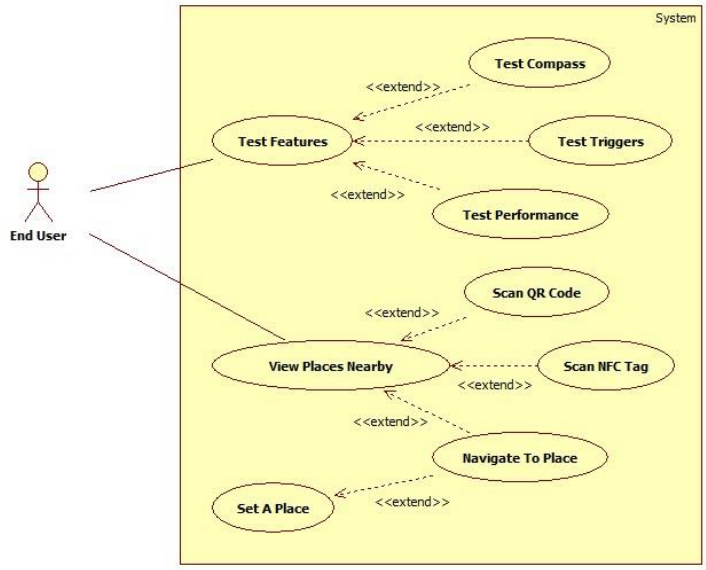


Figure 4.2 - Use Case Diagram of the ARmax Prototype

4.7.3.2. Use Case Descriptions

The use case specification for "View Places Nearby" is shown below in Table 4.2. As well, the additional use case specifications are in the Appendix V.

Use Case ID:	UC-5	**Use Case Name:**	View Places Nearby
Description:	User can explore places around him/her. The places are displayed by a small icon. Also, user is able to interact with these places and perform actions such as view information or navigate to place.		
Actor(s):	• End User		
Preconditions:	• Mobile application should be operational. • Mobile device should have the internet connectivity. • Mobile device's GPS should be operational. • User should select the particular "View Places Nearby" view by choosing respective option.		
Flow of Events			
Basic Flow:	1. User view places nearby through "View Places Nearby" view. 2. User touches one place (i.e. place's icon) in the view. 3. The application responds by displaying the information about the selected place (note: the information will be retrieve via a web service). 4. User presses "Navigate Me" image button. (i.e. Use Case: Navigate To Place) 5. The application responds by displaying the map view with highlighted path to the selected destination.		

Alternative Flow(s):	1. In step 2, user scans a QR code and then the application responds by displaying the information of that scanned QR code. (i.e. Use Case: Scan QR Code) 2. In step 2, user scans a NFC tag and then application responds by displaying the information of that scanned NFC tag. (i.e. Use Case: Scan NFC Tag)
Exception(s):	1. In step 2, if the system crashes when user touches the place's icon, then application will notify the user by raising an error message. 2. In step 3, if the information retrieve fails, then application will notify the user by raising an error message. 3. In step 4, if the system crashes when user presses the "Navigate Me" image button, then application will notify the user by raising an error message. 4. In step 5, if the map and the directions loading fail, the system will notify the user by raising an error message.
Post Conditions:	• The application provides correct navigation information to the user.

Table 4.2 - Use Case Specification of View Places Nearby

4.7.3.3. Domain Model

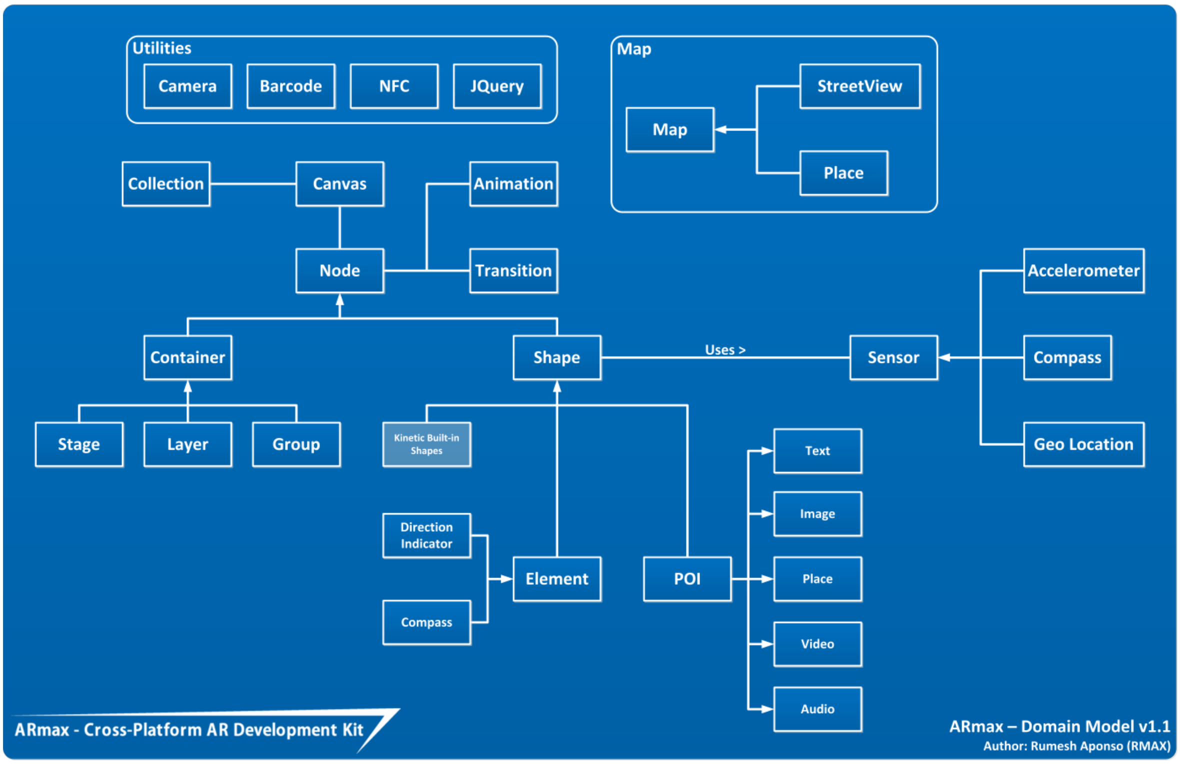


Figure 4.3 - ARmax Domain Model v1.1

Domain model of the proposed ARmax framework is shown in Figure 4.3. Moreover, it is be categorised into four sections which are graphic layer, sensor layer, map layer and utilities layer. Utility layer consists of 3rd party JavaScript libraries and plugins. Map layer consists of modified Google Map API classes. Sensor layer is used to get the sensor inputs from the device and graphic layer handles all the visualisation and animations in the framework.

4.7.3.4. Test Specifications

The test specifications which were derived from the above analysis models will be discussed later on in the testing phase of the ARmax framework and the prototype.

4.7.4. Functional Requirements

The functional requirements identified for the prototype are listed down in Table 4.3 with a unique ID and a small description. The requirements are prioritised into three priority levels known as High, Medium and Low along with the corresponding use case for a particular requirement which is mapped using the Use Case ID.

<table>
<tr><th>ID</th><th>Requirement</th><th>Description</th><th>Priority</th><th>Use Case</th></tr>
<tr><td rowspan="2">FR1</td><td rowspan="2">Test Compass</td><td>End user is provided with the ability to interact with different advance visualization objects such as compass, direction indicator, animations, etc. in the AR view.</td><td rowspan="2">Medium</td><td rowspan="2">UC-2</td></tr>
<tr><td>Input: Placing the mobile device pointing towards different angles.
Process: Accessing the compass sensor and getting the heading.
Output: Displaying north using graphic elements on a layer.</td></tr>
<tr><td rowspan="2">FR2</td><td rowspan="2">Test Triggers</td><td>End user is provided with the ability to interact with different triggering methods which the framework supports (i.e. tap, click and drag). The features of the framework such as node nesting, layering, caching and transitions will be tested as well in this section.</td><td rowspan="2">Medium</td><td rowspan="2">UC-3</td></tr>
<tr><td>Input: User tap the screen of the mobile device.
Process: Listen to the events (i.e. tap, drag and drop) and trigger the appropriate function(s).
Output: Updating and displaying the graphic according to the triggered function.</td></tr>
<tr><td rowspan="2">FR3</td><td rowspan="2">Test Performance</td><td>The performance of the web view in the respective mobile device will be tested.</td><td rowspan="2">High</td><td rowspan="2">UC-4</td></tr>
<tr><td>Input: Tapping the “Start” icon.
Process: Calculating the FPS and time to render the graphic.
Output: Refreshing and displaying the graphic and the simple information box.</td></tr>
<tr><td rowspan="2">FR4</td><td rowspan="2">View Places Nearby</td><td>End user can explore the places around him/her in an AR view with the support of the compass and radar. Moreover, the location of the user will be captured as the current location via GPS and the places such as shops, cafés and pharmacies within a 1km radius will be shown to the user using a map with directions or in AR view with the support of a compass and a direction indicator.</td><td rowspan="2">High</td><td rowspan="2">UC-5</td></tr>
<tr><td>Input: Placing the mobile device pointing towards different angles.
Process: Accessing the compass sensor and GPS sensor, getting the heading and the location data with nearby places.
Output: Displaying places using graphic elements on a layer.</td></tr>
</table>

FR5	Scan QR Code	End user is provided with the ability to scan QR codes. Input: Placing the mobile device in front of a QR code. Process: Capturing the QR code and processing the information. Output: Displaying the QR code information.	High	UC-6
FR6	Scan NFC Tag	End user is provided with the ability to scan NFC tags. Input: Placing the mobile device into a NFC tag. Process: Capturing the NFC tag and processing the information. Output: Displaying the NFC information.	Low	UC-7
FR7	Navigate To Place	End user can navigate to a place, shops, cafés and pharmacies or to a custom location with the help of the guided direction indicator in as an AR view. Also user is has the ability to switch into map view to set a place or a location. Input: Tapping the “Navigate To Place” icon. Process: Accessing the GPS, getting current location data and calculating the distance and path to the destination. Output: Displaying the directions and the highlighted path in the map view.	Medium	UC-8
FR8	Set A Place	User can store the current position of him/her in the application. Input: Tapping the “Set a Place” icon. Process: Accessing the GPS, getting the location data and storing the location in the mobile device. Output: Displaying the location storing status message (i.e. success of error message).	Low	UC-9

Table 4.3 - Functional Requirements of the Prototype

Moreover, Table 4.4 shows the released information of the functional requirements. The high priority and medium priority functional requirements will be released in ARmax v1.1, where the low priority functional requirements will be released in the next version of ARmax.

ID	Priority	Requirement ID	Release 1	Release 2
R1	High	FR3 – Test Performance FR4 – View Places Nearby FR5 – Scan QR Code	✔	---
R2	Medium	FR1 – Test Compass FR2 – Test Triggers FR7 – Navigate To Place	✔	---
R3	Low	FR6 – Scan NFC Tag FR8 – Set A Place	---	✔

Table 4.4 - Release Information of the Functional Requirements.

4.7.5. Non-Functional Requirements

The non-functional requirements that were identified in the prototype are listed below in Table 4.5 with a unique ID and a small description. The requirements are prioritised into three priority levels known as High, Medium and Low. Also, the corresponding non-functional requirement type has been added for a better understanding for the reader.

ID	Requirement	Description	Priority	Type
NFR1	Response time should be high	• It should take less than 5s to retrieve location information from GPS. • It should take less than 5s to process and display the QR code information. • It should take less than 1s to store a new place. • The update rate of a graphic should be more than 5. (FPS of graphic >= 5) • The time taken to draw one graphic in the canvas should be less than or equal to 200ms.	High	Performance
NFR2	Resource consumption for given load should be low	• It should be able to process a minimum number of 25 visual objects (i.e. images, shapes, animations, radar, etc.) in the AR view.	High	Efficiency
NFR3	Should support adding features, and future enhancements	• It should support new features and future enhancements. • The designs and codes should be properly documented.	Medium	Modifiability Extensibility
NFR4	Should support multiple mobile platforms	• It should support mobile platforms such as Android and iOS. • It should support desktop browsers such as Google Chrome, Mozilla Firefox and Safari.	High	Platform compatibility
NFR5	The user interface should be attractive and user-friendly	• The user interface should be attractive and easy to use. • The learnability of the application should be high even for a non-technical user. • The respective success and error messages should guide the user properly.	High	Usability
NFR6	It should run without crashing	• The application should not crash and should provide useful error messages for any failures including exception scenarios.	High	Reliability

Table 4.5 - Non-Functional Requirements of the Prototype

4.7.6. Other Requirements

The other requirements that were identified in the prototype are listed below in Table 4.6 with a unique ID and a small description. The requirements are prioritised into three priority levels known as High, Medium and Low.

ID	Requirement	Description	Priority
OR1	API Documentation	The API specification document for ARmax framework.	Medium
OR2	User Manual	The technical communication document of the prototype, which will be given to the user to provide assistance.	Low

Table 4.6 - Other Requirements of the Prototype

In section 1.8 (Resource Requirements) a number of resource requirements have been identified. However, due to the findings in chapter 2 and chapter 4, some of the resource requirements have been changed. The refined resource requirements are in the Appendix VIII – B and the sources of the resource requirements are specified under "Source" column.

4.7.7. Risks and Mitigation

The risks related with the functional requirements have been identified and listed below in Table 4.7 with a unique risk ID and a small description. The risks are prioritised into three priority levels known as High, Medium and Low. And an appropriate mitigation is included to overcome or avoid that particular risk.

ID	Risk	Description	Priority	Mitigation	Use Case
R1	Poor performance of the application	The performance of the application could be minor.	High	Use mobile optimised technologies and external libraries. e.g. Google Maps JavaScript API v3.11 Use a mobile device, which has enhanced processing power.	UC-4 UC-5
R2	Poor graphic quality	The quality of the graphics could be poor when comparing to native graphics.	Medium	Use quality graphics (higher resolution) with less file size.	UC-2 UC-5
R3	User interface issues	There could be UI issues from one mobile platform to another.	Low	Use coding standards which helps to avoid coding errors. Moreover, through unit testing the issues can be resolved.	UC-1 UC-5

Table 4.7 - Risks related to Functional Requirements

Apart from the risks stated above, there are risks related to other requirements such as testing on iOS. To test an application in an iOS, the tester should have an Apple OS running PC, Xcode IDE and an Apple Developer Account. However, the cost of creating a new Apple Developer Account is high. Furthermore, the availability of resources such as Apple OS running machine is minor. Thus, it could affect testing the prototype on iOS platform.

4.7.8. Scope Refinement

As per the discussion in section 1.4 (Project Aim and Scope), the aim of the project is to design, develop and evaluate a highly efficient and easy to use cross-platform augmented reality framework for developers. In order to achieve this aim, the scope of the project has been refined as follows:

ARmax is a JavaScript based AR Development Kit (framework) and it is introduced to create the AR objects and experiences to the modern mobile web browsers which supports new technologies such as HTML5 and CSS3. In order to demonstrate the features of ARmax framework, a simple reference application (prototype) will be designed, developed and evaluated.

Moreover, some of the features of the prototype which were discussed in section 1.6 (Features of the Prototype) are the same, where some are changed due to various constraints such as time, cost and availability of resources. The finalised functional requirements of the proposed prototype are clearly specified in section 4.7.4 (Functional Requirements).

As a result, the prototype will be developed to Android and iOS mobile platforms. The testing will be carried out mainly for these platforms. Finally, the prototype will be evaluated against two main parameters, which are performance and ease of use.

4.8. Summary

In this chapter, the stakeholders and their roles and viewpoints regarding the proposed ARmax framework were discussed. This was followed by gathering requirements from identified stakeholders. Therefore, requirements elicitation techniques such as literature review, interviews, questionnaire, prototyping, JAD sessions, brainstorming and observation were discussed. Literature review is a great resource of knowledge to gather requirements, because it has analysed the pros and cons of the projects that have been carried out in the past and the present. Thus, it was selected to gather requirements for the ARmax framework and the prototype. Interviews would have been the best method to gather requirements if there were no such constraints as time and cost. Also, within the given time constraint it was not possible to meet many experts. Thus, a quicker and reliable requirement gathering method was required with less expense. Thus, distributing questionnaire was chosen as a requirements gathering method, because it can be used to collect requirements from a large number of people in a short amount of time with less expense. Similarly, prototyping was selected over brainstorming and JAD sessions, because of the reduced time and cost of development, provides a visualisation of the system, user satisfaction and it helps to identify the areas of future enhancements. Accordingly, observations used to extract the additional requirements from the user, where the user is unable to express the needs of the new system or the problems existing in the current system.

The execution of the selected requirements elicitation methods were described in detail and graphs were used to illustrate the results. The results were analysed using OOA technique. Among the analysed results, Android and iOS were the mobile platforms that were often used and the platforms that applications were often developed for. The advantages of using a cross-platform approach were code is reusable, easy for web developers; reduced development cost, easy deployment and reduced vendor lock-in. Device/OS feature support, high performance, high-end graphics and 3D support were identified under native approach. Also, Adobe PhoneGap was the most familiar cross-platform framework among professionals. It appears that PhoneGap provides an easy implementation and easy deployment for a mobile application. Most of the professionals are familiar with NFC technology and all of them think that combining NFC with AR applications may improve the user interaction/experience.

Afterwards, the extracted requirements were modelled using UML models with support of UML tools such as Microsoft Visio and StarUML. As a result, functional requirements were modelled into a use case diagram and the proposed ARmax framework was modelled into a domain model. Performance, efficiency, usability, reliability, modifiability and extensibility were discussed under non-functional requirements. Need of an API documentation and a user manual were identified under other requirements. Moreover, all of these requirements were prioritised into three priority levels known as High, Medium and Low. The possible risks and mitigations were presented. Finally, the scope refinement was done according to the findings of the requirements gathering processes.

Since the requirements were identified, the next chapter will be discussing the design of the ARmax framework. The discussion includes the selection of tools and development methodology for this project. Moreover, it will outline a solution for the problem which is specified in section 1.2 (Problem Definition) with appropriate high level and low level designs.

5. Chapter 5 – Design

5.1. Chapter Overview

Previous chapter provided a detailed description about the stakeholders, requirements elicitation techniques used and the analysis of the gathered requirements. And, from the analysis it derived the functional requirements, non-functional requirements and other requirements of the ARmax framework and its prototype.

This chapter provides an overview to the design of the ARmax framework and its prototype, where it adheres to the derived requirements (i.e. functional requirements, non-functional requirements and other requirements) in the previous chapter. This chapter commences with selection of design methodology and tools. Accordingly, it will provide a detailed description of high level and low level designs of the ARmax framework including UML models such as class diagrams and sequence diagrams. Finally, it will discuss the features of the ARmax prototype with a mock UI which has been designed.

5.2. Design Methodology and Tools Selection

The development of a system or method for a unique situation is referred by a design methodology. There are a number of development methodologies that have been introduced in the past and each of them addresses a different type of problem. Top down design, bottom up design, structured design, object-oriented design are a few design methodologies among them.

Object-oriented design is the process of planning a system of interacting objects for the purpose of solving a problem. By considering the characteristics of this project, the object-oriented design could address the problems such as change, debug and maintain more effectively than other design methodologies. For example, structured design is an industry standard and it features a top-down, hierarchical approach that tends to generate well-organized systems. The technique starts by identifying inputs and desired outputs to create a graphical representation. The step-by-step design philosophy makes this methodology inflexible. System and data requirements must be frozen at the beginning of the life cycle, so the actual systems developed may not reflect the current data and system requirements. Compared with other methodologies, structured design is not very designer-friendly. On the other hand, object-oriented software or systems are highly modular, thus they are easier to change, debug, and maintain than the traditional structured software and systems. Hence, the modules reflect natural classifications.

Because of that, they tend to be more independent, supports code reusability and more stable than the somewhat arbitrary modules suggested by traditional structured techniques.

Furthermore, the experience gained by the object-oriented design methodology for the past few years was one of the main reasons to select object-oriented design as the design methodology for this project. As per the discussion in section 4.7.1 (Analysis Techniques and Tools) there were a number of tools which were used to create UML models. The same tools were used in this section as well to create class diagrams, sequence diagrams and other models.

5.3. High Level Designs

5.3.1. Design Overview

Figure 5.1 shows the high level design overview of the proposed ARmax cross-platform augmented reality development kit (framework). The design comprises of two layers named application layer and AR layer. The application layer consists of two demonstration applications, which has been developed using the ARmax framework. One application will demonstrate the features of the ARmax framework, while the other application demonstrates a real world scenario. The AR layer consists of the entire graphic, map and sensor classes and the corresponding logics in order to create AR elements and objects. Thus, it could be considered as the backbone of the ARmax framework. Moreover, ARmax framework uses a set of utilities such as barcode scan/encode and NFC tag read/write.

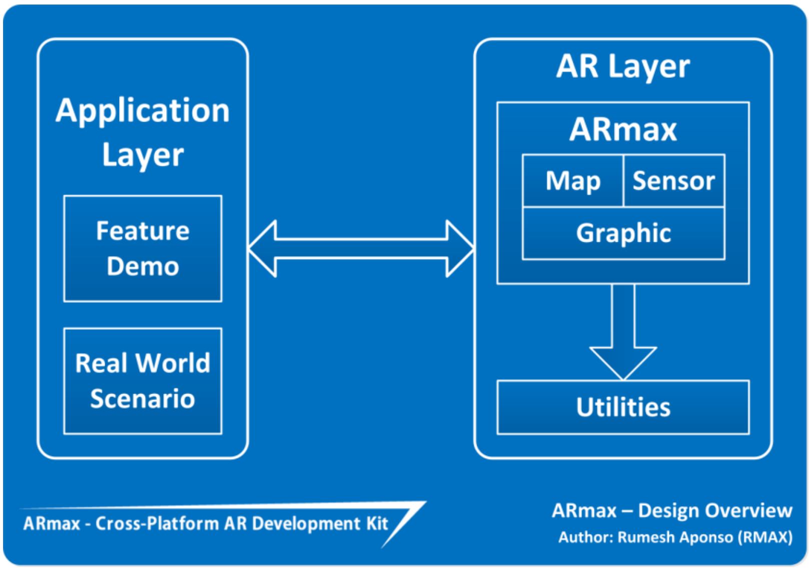


Figure 5.1 – High Level Design Overview

5.4. Low Level Designs

This section discusses the AR layer and application layer in detail. In AR layer, it will discuss four main sections (i.e. Graphic layer, sensor layer, map layer and utilities). In the application layer, it will discuss the two demonstration applications, features and their UI designs.

5.4.1. AR Layer

Graphic layer is responsible for the entire graphical objects such as shapes, texts, elements (compass and direction indicator), POIs (Point of Interest objects) and animations, which are visible to the end-user. Sensor layer manages and controls the sensor data. Map layer manages and controls the map related functions. Utilities contain the common functionalities such as geo location based calculation functions (i.e. distance and azimuth). In addition, it contains 3rd party libraries and plugins, which provide features such as barcode scan/encode, NFC tag read/write and access camera view. Moreover, these sub layers support intercommunication among them and has the capability to work as independent layers as well.

5.4.1.1. Graphic Layer

The class diagram showed in Figure 5.2 is only the graphical layer excluding the “shape classes” of the ARmax framework. It outlines the core classes, functions and attributes of the graphic layer. According to the class diagram below, canvas contains stages and stages are made up of user defined layers. Each layer can contain shapes, groups of shapes, or groups of other groups. The stage, layers, groups, and shapes are virtual nodes, similar to DOM nodes in an HTML page. Figure 5.3 shows an example node hierarchy.

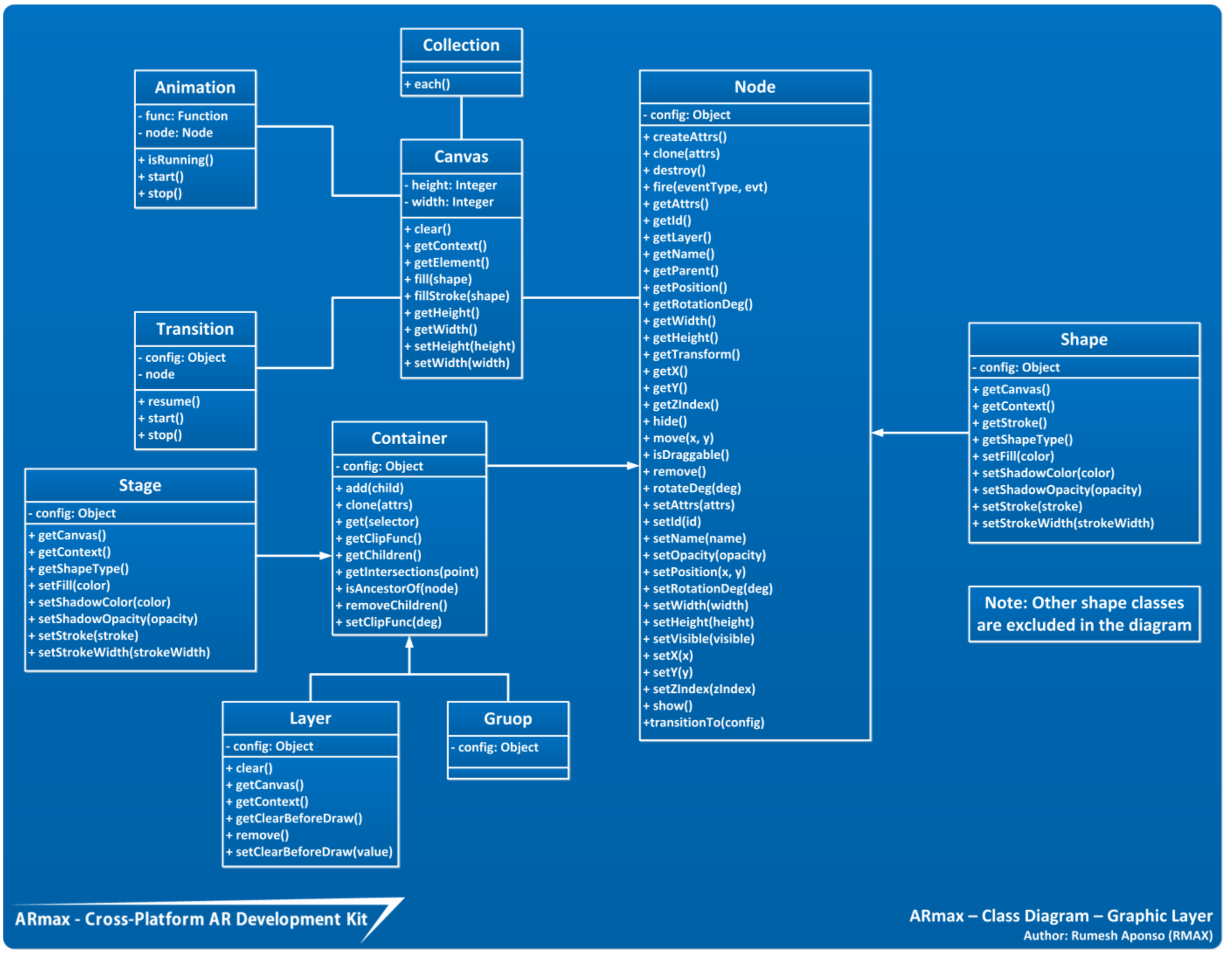


Figure 5.2 - ARmax Class Diagram - Graphic Layer (excluding shapes)

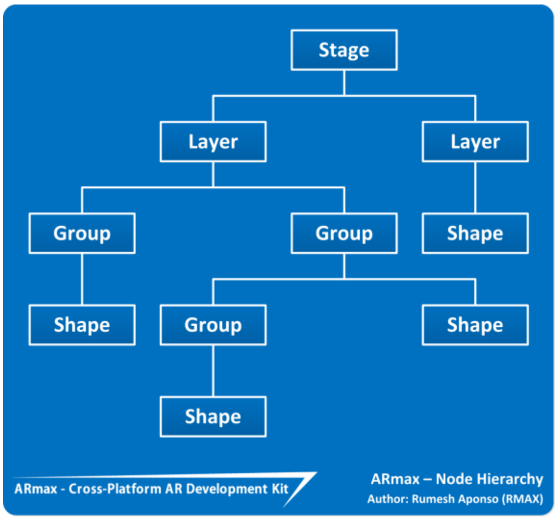


Figure 5.3 - Example Node Hierarchy

Once a stage is set up with layers and shapes, a developer can do a variety of functions such as bind event listeners, transform nodes, run animations and apply filters. In addition, all nodes can be styled and transformed. Although ARmax has prebuilt shapes available such as rectangles, circles, images, sprites, text, lines, polygons, regular polygons, paths and stars the developer can also create custom shapes by instantiating the Shape class and creating a draw function. Following is a set of features that above design intends to support.

Features:

- Object Oriented API
- Node nesting and event bubbling
- Layering support
- Node caching to improve draw performance
- Animation support
- Transition support
- Drag and drop with configurable constraints and bounds
- Filters
- Ready to use shapes including rectangles, circles, images, text, lines, polygons, SVG paths
- Custom shapes
- Pre-defined element such as compass and direction indicator
- Serialization & de-serialization
- Selector support e.g. stage.get('#foo') and layer.get('.bar');
- Desktop and mobile events (mousedown, mouseup, mouseover, mouseout, mouseenter, mouseleave, mousemove, click, dblclick, touchstart, touchend, touchmove, tap, dbltap, dragstart, dragmove, dragend, draw, beforeDraw)
- Pixel ratio optimizations for sharp text and image rendering

As per the discussion in section 4.7.3.3 (Domain Model), the shapes section of the domain model mainly comprises of three types of classes. They are normal shape classes (i.e. circle, ellipse, line, path, polygon, rectangle and star), pre-defined element classes and POI classes.

Figure 5.4 shows the POI class diagram of the ARmax framework. The common attributes and methods are generalised into POI class. There are five types of POI classes and each of them represents different types of objects in the real world. For example place POI object holds information about places of real world and it has a street address and a phone number. By integrating this place POI with Map layer (this will be discussed in section 5.4.1.3) it will enable the developer to access features such as navigation and street view.

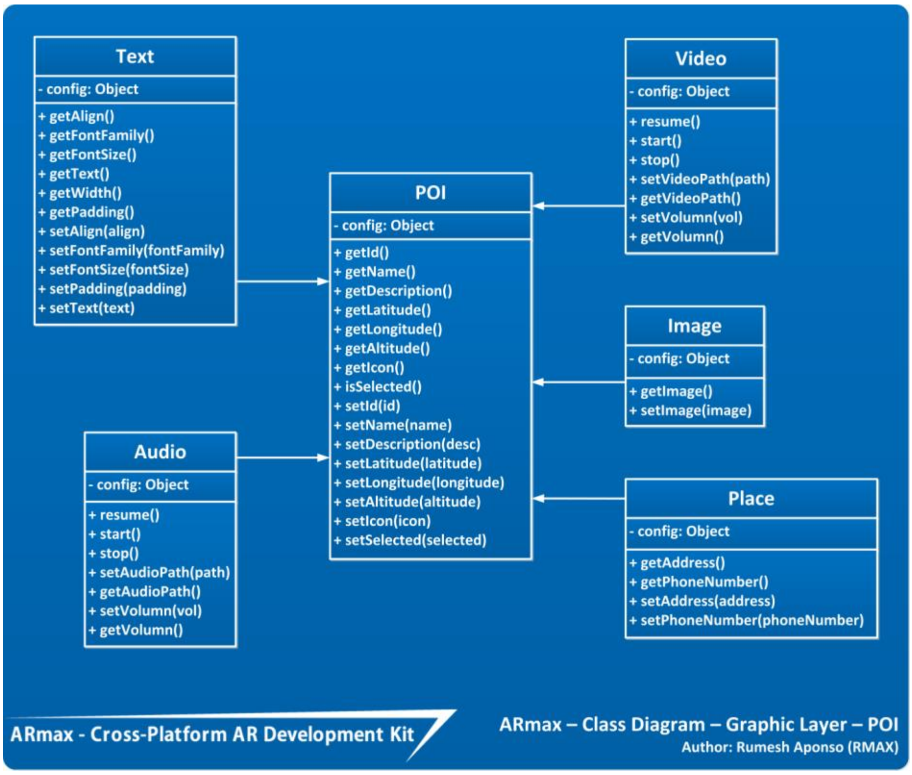


Figure 5.4 - ARmax Class Diagram - Graphic Layer - Shapes - POIs

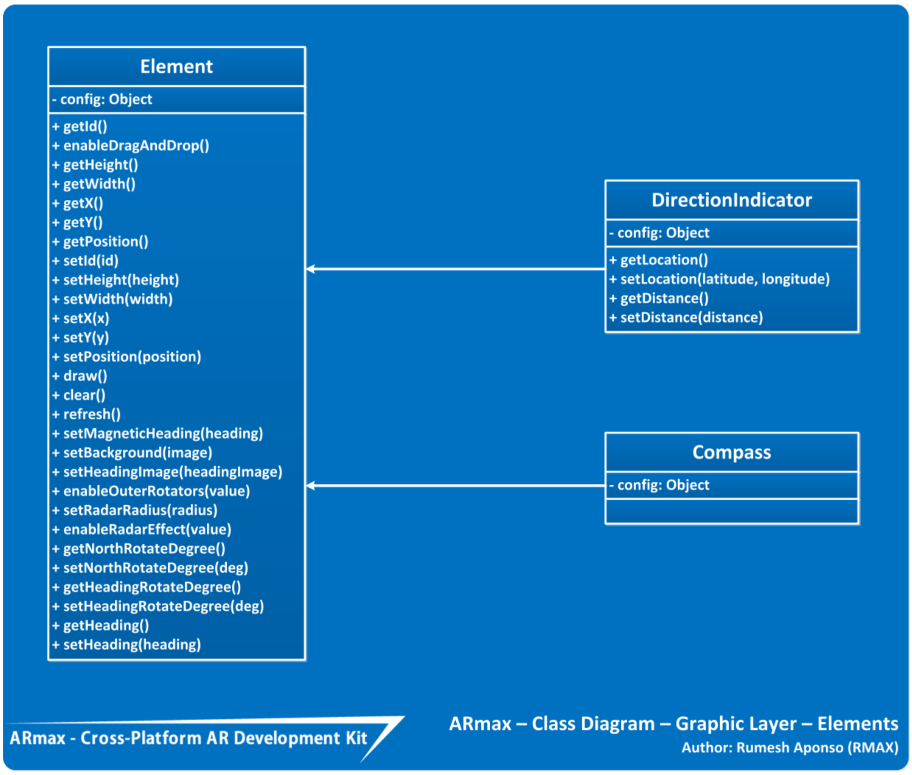


Figure 5.5 - ARmax Class Diagram - Graphic Layer - Shapes – Elements

Figure 5.5 shows the element class diagram of the shape classes. Generally, there are two pre-defined elements such as compass and direction indicator. The common attributes and methods are generalised into Element class. In order to draw these elements a set of pre-defined images are used and the developer will be able to customise these images to satisfy his/her requirements. In order for this to function well, these elements need to integrate with the corresponding sensor in the sensor layer (this will be discussed in section 5.4.1.2). By rotating these images according to sensor data it will create a compass or a direction indicator.

Moreover, there are a number of theories used to calculate the rotation angle and the distance between two points on the surface of a spheroid. Those are azimuth and Vincenty's Formulae. An azimuth is an angular measurement in a spherical coordinate system. The vector from an observer (origin) to a point of interest is projected perpendicularly onto a reference plane, the angle between the projected vector and a reference vector on the reference plane is called the azimuth. Azimuth is usually measured in degrees. Vincenty's formulae are two related iterative methods used in geodesy to calculate the distance between two points on the surface of a spheroid. They are based on the assumption that the figure of the Earth is an oblate spheroid, and hence are more accurate than methods such as great-circle distance which assume a spherical Earth.

Furthermore, Figure 5.6 shows the process that was followed to create elements such as a compass or a direction indicator.

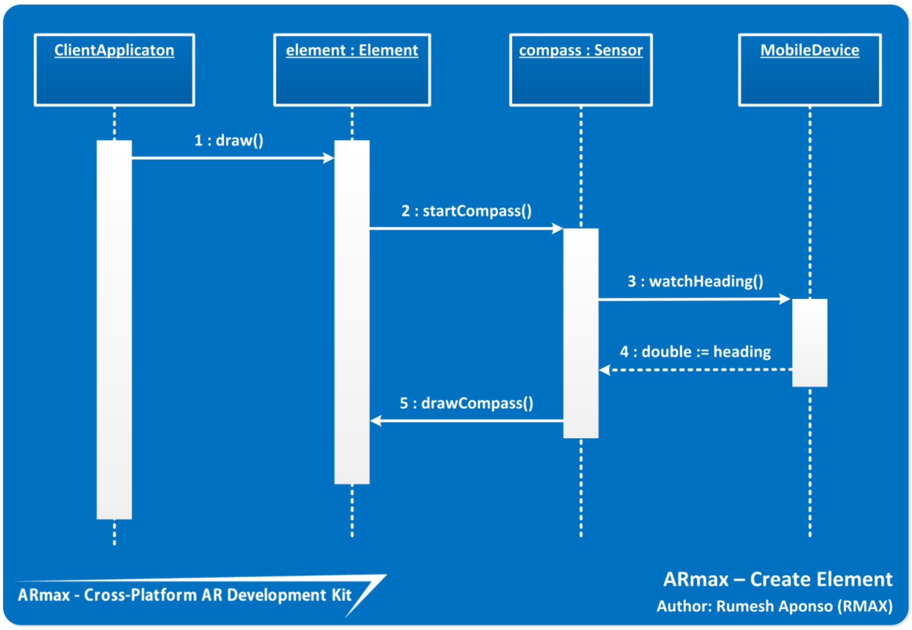


Figure 5.6 - ARmax - Create Element - Sequence Diagram

5.4.1.2. Sensor Layer

The class diagram showed in Figure 5.7 is the sensor layer of the ARmax framework. It outlines the core classes, functions and attributes of the sensor layer. The common attributes and methods are generalised into Sensor classes, where the unique attributes and methods are contained within the respective sensor class. There are three main sensors such as compass, geolocation and accelerometer, which is outlined in this section. The compass is a sensor that detects the direction or heading that the device is pointed towards. It measures the heading in degrees from 0 to 359.99. The accelerometer is a motion sensor that detects the change (delta) in movement relative to the current device orientation. The accelerometer can detect 3D movement along the x, y, and z axis. Geolocation provides location information for the device, such as latitude and longitude. Common sources of location information include Global Positioning System (GPS) and location inferred from network signals such as IP addresses, RFID, Wi-Fi and Bluetooth MAC addresses, and GSM/CDMA cell IDs.

Moreover, there are two types of ways to access and get the sensor data. The first type gets the current state's sensor data from a particular sensor. This is a single function call. The other type gets the sensor data at a regular interval. This is often called a watch function. The interval is specified in milliseconds via the frequency parameter in the Sensor options object. The frequency will be measured in milliseconds in this context. Once the watch is initiated, the sensor will retrieve data according to the specified time interval. In order to stop the watch function, the "stopwatch" method needs to call and it will clear the sensor watch function.

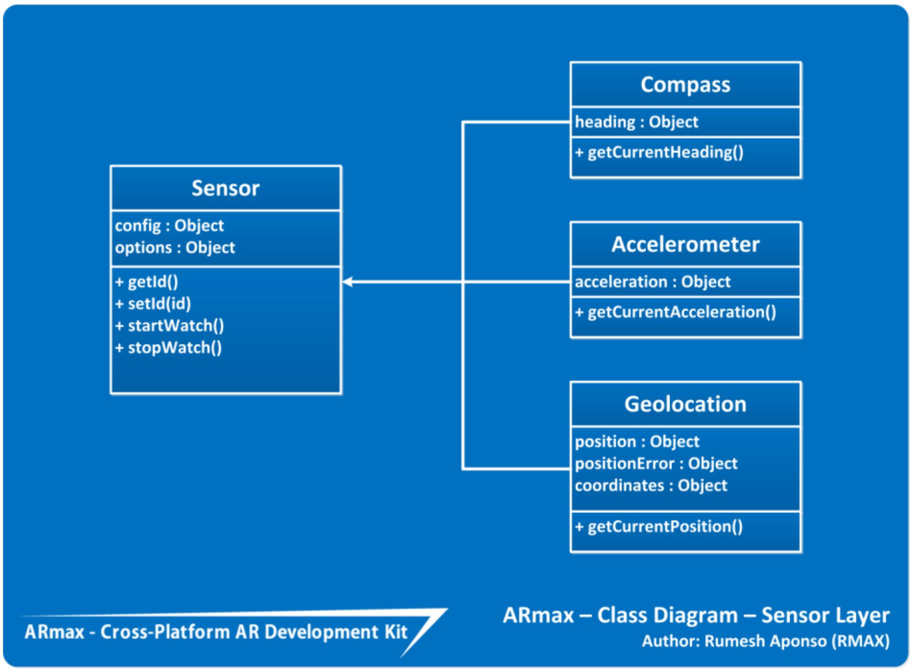


Figure 5.7 - ARmax Class Diagram - Sensor Layer

Furthermore, by integrating a compass watch function with a pre-defined compass element in the graphic layer (see Section 5.4.1.1) it would create a complete compass object with regular heading update. Similarly, a direction indicator object could be created by integrating compass and geolocation sensors.

Figure 5.8 shows the process that is followed to retrieve the geolocation information from the geolocation sensor. The sequence diagram below illustrates the single function call type of the geolocation sensor.

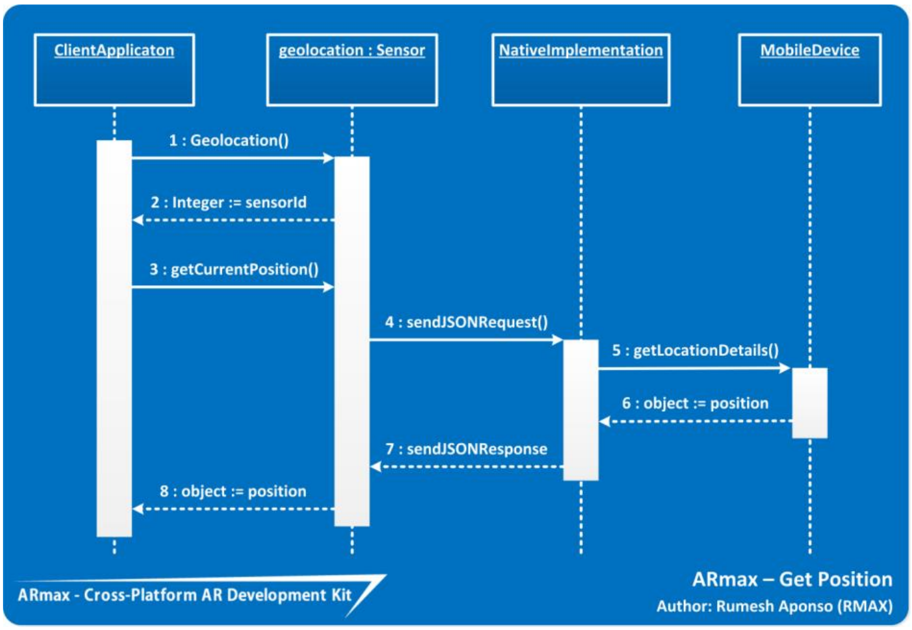


Figure 5.8 - ARmax - Get Geo Location Details - Sequence Diagram

5.4.1.3. Map Layer

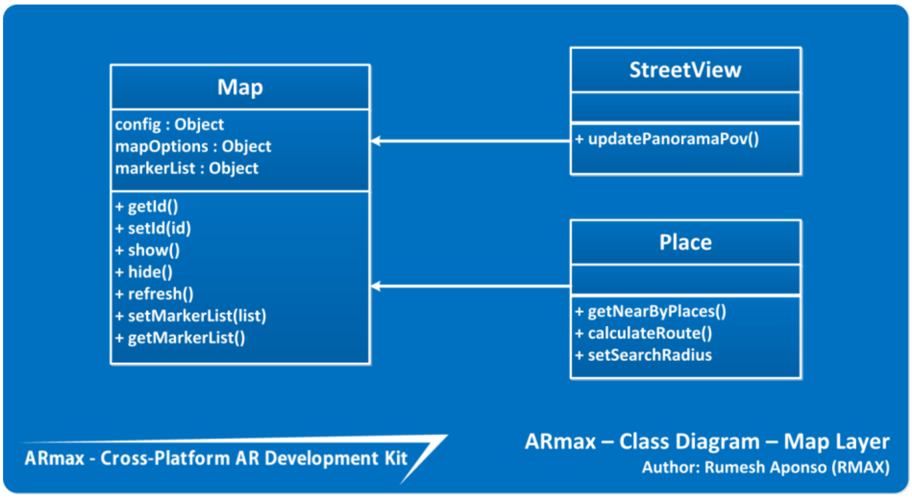


Figure 5.9 - ARmax Class Diagram - Map Layer

The class diagram shown in Figure 5.9 is the map layer of the ARmax framework. It outlines the core classes, functions and attributes of the map layer. The common attributes and methods are generalised into Map class. There are two types of maps such as StreetView and Place.

StreetView map gives a 360 degrees panorama view of a location, whereas place map gives a regular map with places around it in a defined radius. The radius is measured in meters. Moreover, these maps could be integrated with the POIs in the graphic layer (see Section 5.4.1.1). In this case, POI object acts like a marker in the map.

5.4.1.4. Utilities

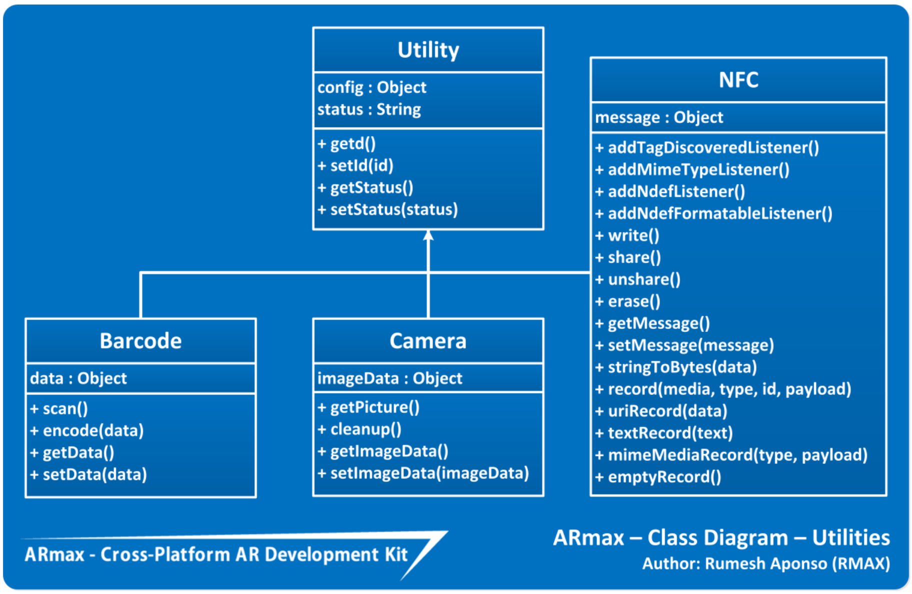


Figure 5.10 - ARmax Class Diagram – Utilities

The class diagram shown in Figure 5.10 are the utilities of the ARmax framework. It outlines the core classes, functions and attributes of the utilities which will be provided in the ARmax framework. The common attributes and methods are generalised into the Utility class, where the unique functions and attributes are enclosed within the respective utility class. There are three main utilities such as Barcode, NFC and Camera. All of these utilities are 3rd party utilities. The core functions of these utilities will capture and will be delivered in the ARmax framework, while encapsulating the excessive functions.

The barcode utility enables a user to scan or encode a barcode. The following barcode types will be supported by this utility.

- QR Code
- EAN 8
- Code 93
- PDF417
- Data Matrix
- EAN 13
- Codabar
- RSS Expanded
- UPC E
- Code 128
- ITF
- UPC A
- Code 39
- RSS14

Moreover, the NFC utility provides access to Near Field Communication (NFC) functionality, allowing applications to read NDEF message in NFC tags. A “tag” may actually be another device that appears as a tag.

Furthermore, the camera utility takes a photo using the camera or retrieves a photo from the device's album. The image is returned as a base64 encoded String or as the URI of an image file.

Apart from these utility classes, there is a Commons class, which includes the functions such as geo location based calculation functions (i.e. distance and azimuth). This Commons class is not included in the Utility class diagram above (see Figure 5.10).

5.4.2. Application Layer

5.4.2.1. Prototype

As per the discussion in section 5.3.1 (Design Overview), the application layer comprises of two demonstration applications. One is to demonstrate the features of the ARmax framework and the other application is to demonstrate the real world scenario. However, considering constraints such as time, it appears that more time will be needed to build two separate applications. As a result, it has been decided to build a single prototype comprising both of these demonstration applications. The features of this prototype are clearly defined in the functional requirements of the previous chapter (see Section 4.7.4).

Moreover, there is a set of non-functional requirements (see Section 4.7.5) such as platform compatibility and usability, which the prototype should achieve. Thus, the prototype design should be attractive and easy to use. The learnability of the prototype should be high. It should support mobile platforms such as Android and iOS. When considering these requirements and the nature of the application, which is augmented reality, it is obvious that the UI design of the prototype is vital. Furthermore, as per the discussion in section 2.10 (Mobile Browser Support), it appears that different mobile browsers support different features. Thus, this has been taken into consideration, when designing the UI for the prototype.

Correspondingly, in order to create the UI designs, Adobe Fireworks CS6 was used. One of the main reasons to use Adobe Fireworks CS6 was the experience gained during the internship year at Zone24x7 Inc. Also, the designing task itself could take more time. However, time is very important in this project, thus the use of a professional tool like Adobe Fireworks CS6 could save time and effort taken to design the UIs of the prototype. The next section will discuss the mock UIs that were created using Adobe Fireworks CS6.

5.4.2.2. Features and UI Designs

Figure 5.11 shows the home screen of the prototype. The ARmax prototype logo is placed at the top left corner. There are three small buttons in the upper right corner and those buttons are static throughout the application. The buttons from left to right respectively are back button, setting button and application close button. In the bottom left corner there are five icons. Starting from left to right they are home icon, test compass icon, test triggers icon, test performance icon and view places nearby icon. The home icon simply represents the home screen and it will navigate the user to home screen. The test compass icon will navigate the user to test compass and direction indicator screens (see Appendix VI - Figure 10.3), where the pre-defined graphic element will be tested. The test triggered icon will navigate the user to test triggers screen (see Appendix VI - Figure 10.4), where the features such as events, layering and node caching of the ARmax framework will be tested. The test performance icon will navigate the user to test performance screen (see Appendix VI - Figure 10.5), where the device's browser performance will be tested. Accordingly, these test compass, test triggers and test performance features are first demonstration application, which were discussed in the section 5.3.1 (Design Overview). Finally, the view places nearby icon will navigate the user to view places nearby screen, where the places which are near to the user will be shown in an AR view (see Appendix VI - Figure 10.7) This is the real world scenario which is addressed in section 5.3.1 (Design Overview) as the second demonstration application.

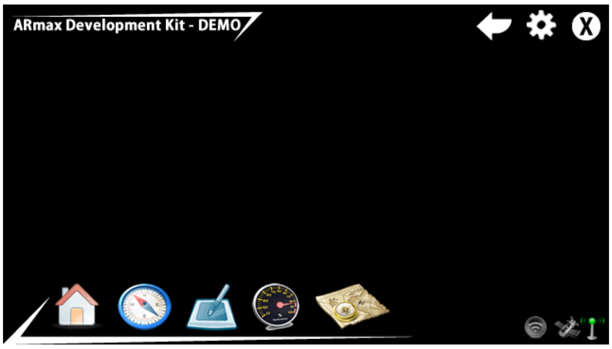


Figure 5.11 - Home Screen of the ARmax Prototype

Note: The mock UIs of the other screen are in the Appendix IV.

5.5. Summary

This chapter commences with the selection of a design methodology and tools. A few design methodologies such as top down design, bottom up design, structured design and object-oriented design were discussed and object-oriented design methodology was selected as the design methodology for the ARmax project. Object-oriented design addresses the problems such as change, debug and maintain more effectively than other design methodologies. The experience gained by working with object-oriented design for the past few years was the reason to select it as the design methodology for this project.

Accordingly, it discussed the high level architecture of the ARmax cross-platform augmented reality development kit (framework). Then it discussed the two layers, which are application layer and AR layer. AR layer comprises of four sub layers named graphic layer, sensor layer, map layer and utilities. These sub layers are the backbone of the ARmax framework and these were discussed in detail using UML models such as class diagrams and sequence diagrams.

Moreover, it was decided for the application layer to have demonstration applications, where one application demonstrated the features of the ARmax framework, while the other demonstrated a real world scenario. However, due to the time constraint, these two applications were combined into a single prototype. The features of this prototype were discussed and the non-functional requirements such as platform compatibility and usability were given higher priority. It appeared that different mobile browsers support different features. Thus, this was taken into consideration, when designing the UI for the prototype. The importance of the UI designs was discussed and Adobe Fireworks CS6 was selected as tool to create mock UIs. Among the reasons the experience gained in the internship year, the time and effort that was taken to design the UIs of the prototype was less and this was highlighted. Finally it discussed the mock UIs, which were created using Adobe Fireworks CS6.

The next chapter will discuss the implementation phase of the ARmax framework and the prototype. The tools and technology selection, core functions of the prototype, problems faced and the solutions found will be discussed in detail.

6. Chapter 6 – Implementation

6.1. Chapter Overview

Previous chapter provided an overview to the design of the ARmax framework and its prototype. The high level and low level designs were made using UML models such as class diagrams and sequence diagrams. Also, it discussed the features of the prototype with mock UIs which have been designed.

This chapter provides an overview to the implementation of the ARmax framework and its prototype, where it adheres to the designs which were made in the previous chapter. It commences with the selection of HTML5 and JavaScript as the primary programming languages. The selection of IDE and mobile devices will be discussed in detail. It will also discuss the coding standards and external libraries and technologies, which have been used during the implementation of the ARmax framework and prototype. Accordingly, the features of the prototype will be discussed in detail including respective code snippets and screenshots. Lastly, the problems faced and the solutions found will be presented.

6.2. Selection of Technologies

Since the proposed ARmax framework is a mobile cross-platform solution, it should support many mobile platforms such as Android, iOS and BlackBerry. However, languages such as C#, Objective C and Java are not platform independent. Therefore, selecting a language that is independent throughout the mobile platforms and supports mobile application development was compulsory. As a result, in order to build the proposed ARmax framework HTML5 and JavaScript were selected as the programming language.

Moreover, HTML specification is a global web standard, which every browser understands in each device such as personal computers, mobile phones and televisions. Also, it is a DOM based specification. Thus, it is cross-platform and language-independent and browser creators such as Google, Mozilla, Apple Inc. and Research In Motion are adheres to this specification. The very next thing that comes to mind when talking about HTML and browsers is scripting languages. There are scripting languages such as JavaScript and VBScript. For this project JavaScript has been selected. JavaScript was originally implemented as a part of web browsers so that client-side scripts could interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed. It is understood that JavaScript is also a platform independent language. The results gathered in section 2.10 (Mobile Browser Support) shows that every browser supports JavaScript. Among the features introduced in the new HTML5 specification, most of the new features are integrated with the JavaScript. Thus, it was another reason to select JavaScript as the scripting language among the other scripting languages.

As per the discussion in section 2.9.4 (Conclusion on Mobile Cross-Platform Solutions), Adobe PhoneGap was selected as the mobile cross-platform solution in order to build the proposed prototype of this project. Adobe PhoneGap is an improved cross-platform solution, which uses HTML5 and JavaScript as the primary programming language. This was one of the main reasons to select HTML5 and JavaScript as the programming language for this project.

6.3. Selection of Tools

Since HTML5 and JavaScript were chosen as the primary programming languages, it is obvious that any text editor tool such as notepad or notepad++ or any IDE such as netbeans, eclipse or visual studio would support the requirement of an IDE. However, Adobe Dreamweaver CS6 was selected as the IDE to build the ARmax framework and the proposed prototype. Adobe Dreamweaver was mainly developed to build and manage websites. Thus, it supports the basic requirements such as easy build and manages over web projects, widgets and extensions, where other IDEs are more complicated and only supports fewer web development technologies compared to Adobe Dreamweaver.

On the other hand, Adobe Dreamweaver supports most leading web development technologies, including HTML, XHTML, CSS, XML, JavaScript, Ajax, PHP, Adobe ColdFusion® software, and ASP. It also supports jQuery Mobile, PhoneGap, HTML5 and CSS3 for mobiles, tablets and computers, while other stated text editors and IDEs do not.

Because of these reasons, it appears that the best IDE to build this project and the prototype is Adobe Dreamweaver CS6. Furthermore, the experience gained working with Adobe Dreamweaver IDE during the internship year at Zone24x7 Inc was one of the main reasons to select it as the IDE for this project.

Apart from the Adobe Dreamweaver IDE, eclipse and Xcode IDE were used to deploy the application into mobile devices.

6.4. Selection of Mobile Devices

As per the discussion in section 2.8 (Mobile Application Development), the performance of the cross-platform applications are less compared to native applications. Hence, PhoneGap was selected to build the prototype for this project and section 2.9.4 (Conclusion on Mobile Cross-Platform Solutions) shows that the performance of the PhoneGap applications are limited to the mobile device web view. Thus, this could affect the project implementation and testing phases. Therefore, as a precaution two mobile devices were selected with high processing power. The mobile devices are Samsung Galaxy SIII i9300 and iPhone 4. Samsung Galaxy SIII i9300 was used to test the Android version and iPhone 4 was used to test the iOS version of the proposed ARmax prototype.

6.5. Coding Standards Used

The primary programming languages of this project are HTML5 and JavaScript. The importance of using coding standards was identified and use of proper coding standards could lead to easy debug and maintenance of the project. Moreover, it increases the quality of the code and the overall solution. Google is one of the leading companies in web technologies such as HTML and JavaScript and they have published a few coding standards on HTML and JavaScript. Thus, Google HTML/CSS Style Guide and Google JavaScript Style Guide were used in this project, in order to improve the quality of the code. The coding standard guides which referred are specified in Table 6.1 with the source.

Guide	Source
Google JavaScript Style Guide	Available at: http://google-styleguide.googlecode.com/svn/trunk/javascriptguide.xml
Google HTML/CSS Style Guide	Available at: http://google-styleguide.googlecode.com/svn/trunk/htmlcssguide.xml

Table 6.1 - Google Coding Standard Guides

6.6. External Libraries and Technologies used in the Implementation

This section briefly discusses the external libraries and technologies used in the implementation phase.

6.6.1. Adobe PhoneGap v2.4.0

License	Free and open source under the Apache License, Version 2.0
API Reference	Available at: http://docs.phonegap.com/en/2.5.0/index.html
Download	Available at: http://phonegap.com/download/

PhoneGap is an open source solution for building cross-platform mobile applications with standard-based web technologies such as HTML, JavaScript, and CSS. It is an open source implementation of open standards and is free, which means developers and companies can use PhoneGap for mobile applications that are free, commercial, open source, or any combination of these.

Moreover, in order to gain access over the sensors such as compass, geolocation and accelerometer of a mobile device, Adobe PhoneGap was used in the sensor layer of the design which was discussed under section 5.4.1.2 (Sensor Layer).

6.6.2. KineticJS v4.3.1

License	MIT or GPL Version 2
API Reference	Available at: http://kineticjs.com/docs/
Download	Available at: http://kineticjs.com/

KineticJS is an HTML5 canvas JavaScript library that extends the 2D context by enabling canvas interactivity for desktop and mobile applications. The graphic layer which was discussed under section 5.4.1.1 (Graphic Layer) of the ARmax framework uses the already implemented features such as animations, transitions, node nesting, layering, filtering, caching and event handling of the KineticJS library.

6.6.3. Google Maps JavaScript API v3.11

License	Free service
API Reference	Available at: https://developers.google.com/maps/documentation/javascript/reference

The Google Maps JavaScript API enables to embed Google Maps in to web pages. Version 3 of this API is especially designed to be faster and more applicable to mobile devices, as well as traditional desktop browser applications. The Google Maps JavaScript API was used in the map layer which was discussed under section 5.4.1.3 (Map Layer) of the ARmax framework in order to access the map data, place information and streetview data.

6.6.4. PhoneGap Barcode Plugin

License	MIT License
Download	Available at: https://github.com/phonegap/phonegap-plugins/tree/master/Android/BarcodeScanner

The PhoneGap barcode plugin enables a user to scan or encode a barcode. Barcode types which were discussed in the design section 5.4.1.4 (Utilities) will be supported by this plugin. Therefore, it is used to provide functionalities such as scan and encode to the developers.

6.6.5. PhoneGap NFC Plugin

License	MIT License
Download	Available at: https://github.com/chariotsolutions/phonegap-nfc

The PhoneGap NFC plugin provides access to Near Field Communication (NFC) functionality, allowing applications to read NDEF messages in NFC tags. The PhoneGap NFC plugin was used in the utilities layer in the ARmax framework (see Section 5.4.1.4) in order to provide functionalities such as write, listen, share and erase.

6.6.6. jQuery v1.8.2

License	MIT License
API Reference	Available at: http://api.jquery.com/
Download	Available at: http://jquery.com/download/

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. jQuery is used to implement the features in the ARmax prototype.

6.6.7. stats.js

License	MIT License
Download	Available at: http://jquery.com/download/

stats.js is a JavaScript performance monitor utility, where it provides a simple information box that will help to monitor the code performance. It uses two monitoring criteria to decide the code performance. One is FPS (i.e. Frames rendered in the last second. Higher the number the better it is) and the other is time needed to render a frame. The time is measured in milliseconds, and lower the time the better it is.

Moreover, this utility is used in FR3 functional requirement of the ARmax prototype to determine the performance of the mobile device's web view.

6.7. Features of the Prototype

This section discusses the features of the prototype, where it describes how the implementation was carried out for each feature. The explanations will include respective code snippets and screenshots.

6.7.1. Test Compass and Direction Indicator

As per the discussion in section 4.7.4 (Functional Requirements), this feature relates to the FR1 functional requirement and it is supposed to provide the ability to interact with different advance visualization objects such as compass, direction indicator in an AR view to the user.

In order to draw a compass or direction indicator, ARmax framework uses a HTML5 canvas layer and a set of images. These images are used to draw a compass or direction indicator on the canvas layer and "drawImage" method, which comes under HTML5 canvas, is used to convert the image into a vector. The pseudocode showed in Code Snippet 6.1 shows the high level steps that is involved drawing a compass in a canvas. This "drawCompass" function needs four parameters such as h, n, direction and degree.

```
        h :  Current pointing direction of the device in degrees relative to the actual north
        n :  Actual north in degrees
direction :  Current pointing direction of the device in text
   degree :  Current pointing direction in degrees
```

```
function drawCompass(h, n, direction, degree) {
    Set heading and north rotation degrees
    Save the canvas state and clear the canvas
    Draw compass
    Decide radar radius
    Draw radar
    Draw compass highlighted area
    Draw heading pointer
    Decide outer circles rotation degree
```

```
    Draw outer rotator three
    Draw outer rotator two
    Draw outer rotator one
    Draw direction and degree notes
    Restore the canvas state
}
```

Code Snippet 6.1 - Pseudocode of drawCompass Function

The Code Snippet 6.2 shows the actual code of the 3rd step of the above pseudocode, which is used to draw compass.

```
// Translate canvas
canvas_ctx.translate(
    COMPASS_X + (img_compass.width / TWO),
    COMPASS_Y + (img_compass.height / TWO)
);

// Save canvas state
canvas_ctx.save();

// Rotate canvas
canvas_ctx.rotate(NORTH_ROTATE_DEGREE * ROTATE_CONST_2);

// Draw compass image
canvas_ctx.drawImage(
    img_compass,
    -(img_compass.width / TWO),
    -(img_compass.height / TWO)
);

// Restore canvas state
canvas_ctx.restore();
```

Code Snippet 6.2 - Draw Compass Image into a Canvas Code Snippet

The pseudo code provided in Code Snippet 6.1, is only capable of drawing a static compass in the canvas. However, in order for it to function similarly to an actual compass, this draw method should be called in at a regular interval. As per the discussion in section 5.4.1.1 (Graphic Layer - Figure 5.6), it shows the process that was followed to create a complete compass element. Accordingly, when the “draw” method of a compass object is called by the user a compass watch function will be started with a default frequency of 10. This means compass heading will be retrieved for every 10ms. Once the heading information is received, the “drawCompass” method will be called inside the compass watch function, which means that for every 10ms the compass will be redrawn according to the parameters passed. Thus, the compass will appear to the user as an animation with the correct heading and north, until the watch function is stopped. Similarly, the direction indicator can also be drawn by integrating “drawDirectionIndicator” method with a compass watch function. The Code Snippet 6.3, shows the compass watch function which is stated above. Moreover, Figure 6.1 shows the screenshots of the implemented compass and direction indicator elements.

```
/* Draws compass element in the predefined canvas layer
 * H  - Magnetic heading receives by the device's compass sensor
 * N  - Actual north relative to the magnetic heading
 * HM - Modified magnetic heading
 */
function startCompass() {

    var options = null;

    if (compassWatchId == ZERO) {

        options = {
            frequency: DEFAULT_COMPASS_FREQUENCY
        };

        compassWatchId = navigator.compass.watchHeading(function (heading) {

            H = heading.magneticHeading;
            N = ((THREE_SIXTY_DEGREE - H) - NINTY_DEGREE);
            HM = (H + NINTY_DEGREE);

            // Draw compass
            drawCompass(ZERO, N,
                getDirectionByHeading(HM.toFixed(0) % THREE_SIXTY_DEGREE),
                ((HM.toFixed(0) % THREE_SIXTY_DEGREE) + "°")
            );

            // Update DOM
            $('#compass_test_info_heading').html("Heading: " +
                ((HM.toFixed(0) % THREE_SIXTY_DEGREE) + "°") + " " +
                getDirectionByHeading(HM.toFixed(0) % THREE_SIXTY_DEGREE)
            );

            $('#compass_test_info_heading_meter').val((HM.toFixed(0) % THREE_SIXTY_DEGREE));

        }, function (error) {

            alert(COMPASS_ERROR);

        }, options);

    } else {

        navigator.compass.clearWatch(compassWatchId);
        compassWatchId = ZERO;

    }
};
```

Code Snippet 6.3 - Compass Draw Watch Function

Figure 6.1 - Screenshot of Compass and Direction Indicator Elements

6.7.2. Test Triggers

As per the discussion in section 4.7.4 (Functional Requirements), this feature relates to the FR2 functional requirements and it demonstrates the features such as node nesting, layering, caching, transitions and event handling which the framework supports.

The design of the ARmax framework supports a number of events (see Section 5.4.1.1). These events could bind with stages, layers, groups or shapes in order to perform a task. Moreover, built-in events in KineticJS were used along with features such as node nesting, layering, caching and transitions in order to implement the following features of the ARmax prototype.

Figure 6.2 - Screenshots of Test Triggers - Touch Event Example - Listen Event Example

Figure 6.2 shows the event handling example, which test the touch/tap event and a touch listener event. According to the user's action a message layer will be updated (i.e. "x: -104, y: 88" in the left screenshot and "Mouseover oval" in the right screenshot). When a user touches the triangle in the left screenshot, the corresponding triangle coordinates of the touch point will be updated. Similarly, if the user touches the circle, the message will be updated as "Touch Start Circle", otherwise the message will be updated as "Touch End Circle". The screenshot at the right side, demonstrate a touch listener. If the user touches the oval when the listener is on, the message will be updated as "Mouseover oval", otherwise the message will be empty.

Code Snippet 6.4 shows the above event handling demonstration of the screenshots specified in Figure 6.2.

```
Left Screenshot Code
-------------------------------------------------------------------------------------------
// Triangle touch event
triangle.on('touchmove', function () {
    var touchPos = testTriggersCanvas_ctx.getTouchPosition();
    var x = touchPos.x - 190;
    var y = touchPos.y - 40;
    writeMessage(messageLayer, 'x: ' + x + ', y: ' + y);
});

// Circle touch event
```

```
circle.on('touchstart', function () {
    writeMessage(messageLayer, 'Touch Start Circle');
});
circle.on('touchend', function () {
    writeMessage(messageLayer, 'Touch End Circle');
});


Right Screenshot Code
-------------------------------------------------------------------------------------------
// Oval touch event
oval.on('mouseover', function () {
    writeMessage(messageLayer, 'Mouseover oval');
});
oval.on('mouseout', function () {
    writeMessage(messageLayer, '');
});

// Oval listener
document.getElementById('test_triggers_listen_button').addEventListener('touchstart', function () {
    oval.setListening(true);
    shapesLayer.drawHit();
}, false);

document.getElementById('test_triggers_dontListen_button').addEventListener('touchstart', function ()
{
    oval.setListening(false);
    shapesLayer.drawHit();
}, false);
```

Code Snippet 6.4 - Event Handling Example Code Snippet

Similarly, Figure 6.3 shows a shape caching (i.e. left screenshot) and a layering (i.e. right screenshot) example. Furthermore, Figure 6.4 shows a drag event (i.e. left screenshot) and a simple easing (i.e. right screenshot) example.

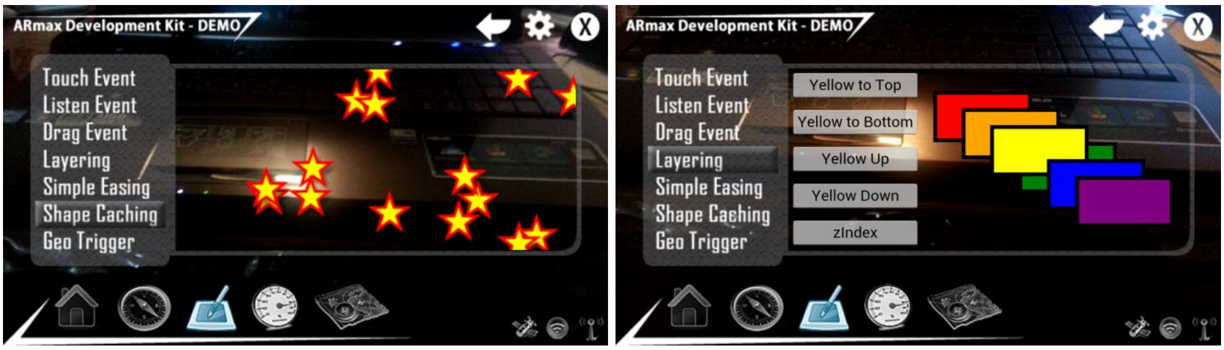


Figure 6.3 - Screenshots of Test Triggers - Shape Caching Example - Layering Example

Figure 6.4 - Screenshots of Test Triggers - Drag Event Example - Simple Easing Example

6.7.3. Test Performance

As per the discussion in section 4.7.4 (Functional Requirements), this feature relates to the FR3 functional requirements and it demonstrates a performance test of the web view in the respective mobile device.

The performance test is based on two criteria, which are FPS and the time needed to render a frame. The higher the FPS value or lower the time needed to render a frame means that the performance is greater. The time will be measured in milliseconds.

Figure 6.5 shows the example screenshot of the performance test which demonstrates the ARmax prototype. In order to test the performance a standard animation was drawn in a HTML5 canvas layer. The FPS and the time needed to render a frame were determined using stats.js JavaScript performance monitor utility. The corresponding performance criterion was shown using a timing diagram.

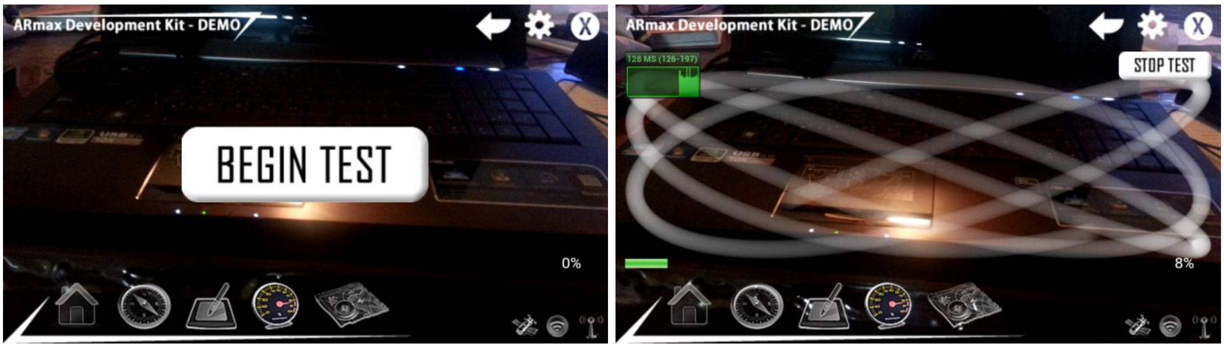


Figure 6.5 - Screenshots of Performance Test

Furthermore, Code Snippet 6.5 shows the performance test function which used in the above example.

```
function doAnimation() {
    var stats = new Stats();
    stats.setMode(1);

    var fps_stat_element = document.getElementById('fpsStat');
    fps_stat_element.appendChild(stats.domElement);

    var performance_canvas = document.getElementById('test_performance_canvas_layer');
    performance_canvas.width = 640;
    performance_canvas.height = 255;

    var context = performance_canvas.getContext('2d');
    context.fillStyle = 'rgba(255,255,255,0.05)';
    context.clearRect(0, 0, 640, 255);
    setTimeout(function () {
        context.clearRect(0, 0, 640, 255);
        var myVar = setInterval(function () {
            if (performance_test_timer_is_on == 0) {
                clearInterval(myVar);
            }
```

```
            var time = Date.now() * 0.001;

            context.clearRect(0, 0, 640, 255);

            stats.begin();

            for (var i = 0; i < 2000; i++) {

                var x = Math.cos(time + i * 0.01) * 300 + 320;
                var y = Math.sin(time + i * 0.01234) * 90 + 130;

                context.beginPath();
                context.arc(x, y, 10, 0, Math.PI * 2, true);
                context.fill();

            }

            stats.end();

        }, 1000 / 60);
    }, 250);
}
```

Code Snippet 6.5 - Performance Test Function Code Snippet

6.7.4. View nearby Places

As per the discussion in section 4.7.4 (Functional Requirements), this feature relates to the FR4, FR5, FR6, FR7 and FR8 functional requirements. It demonstrates a real world scenario with combination of the above functional requirements.

Scenario: When a user invokes this feature, the location of the user will be captured as the current location via GPS and the places such as shops, cafés and pharmacies within a 1km radius will be shown to the user using a map with directions. There are two modes to this feature, one is map mode and the other is camera mode. Showing place around the user in a map is called map mode, whereas the places around the user displayed as POIs via an AR view is called camera mode. The user is able to switch between these modes at any time. In the camera mode, the places are displayed as POIs and the user is able to view more information such as place description, distance and phone number by touching the particular POI icon. With more information of that particular POI, the user will be provided with a set of sub features such as get direction to a place and view streetview of that place. If a user selects get directions, then the directions to that particular place will be shown in the map mode, whereas, if a user selects view streetview of the place, then the particular streetview of the place will be shown. Apart from the main functionality of this feature stated above, there are a few experimental features such as scanning a barcode, scanning NFC tag and setting a place. And these are provided to the user in the camera mode.

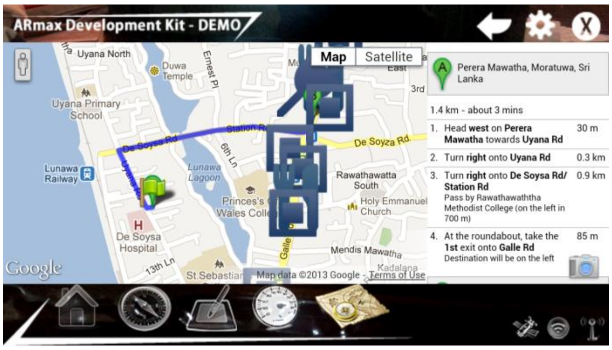


Figure 6.6 - View Places Nearby - Map Mode

Moreover, Figure 6.6 shows the map mode of the view place nearby feature. In order to get nearby places Google Maps JavaScript API v3.11 was used in the map layer of the ARmax framework. Code Snippet 6.6 shows the code, which is used to retrieve nearby places. As in the "getNearByPlaces_callback" function, the places which are retrieved are added to a POI object array.

```
function getNearByPlaces() {
    var myLatlng = new google.maps.LatLng(CURRENT_LAT, CURRENT_LON);

    var request = {
        location: myLatlng,
        radius: '1000',
        types: ['store']
    };

    service_VPN = new google.maps.places.PlacesService(directionsMap_VPN);
    service_VPN.nearbySearch(request, getNearByPlaces_callback);
}

function getNearByPlaces_callback(results, status) {
    if (status == google.maps.places.PlacesServiceStatus.OK) {
        poiList = new Array();

        for (var i = 0; i < results.length; i++) {
            var place = results[i];
            var temp_poi = new POI(
                                   place.id,
                                   place.name,
                                   place.html_attributions,
                                   place.geometry.location.lat(),
                                   place.geometry.location.lng(),
                                   null,
                                   place.icon,
                                   place.formatted_address,
                                   place.international_phone_number,
                                   null
                                  );
            poiList[i] = temp_poi;
        }

        selectFirstPOI();
    }
}
```

Code Snippet 6.6 - Get Nearby Places

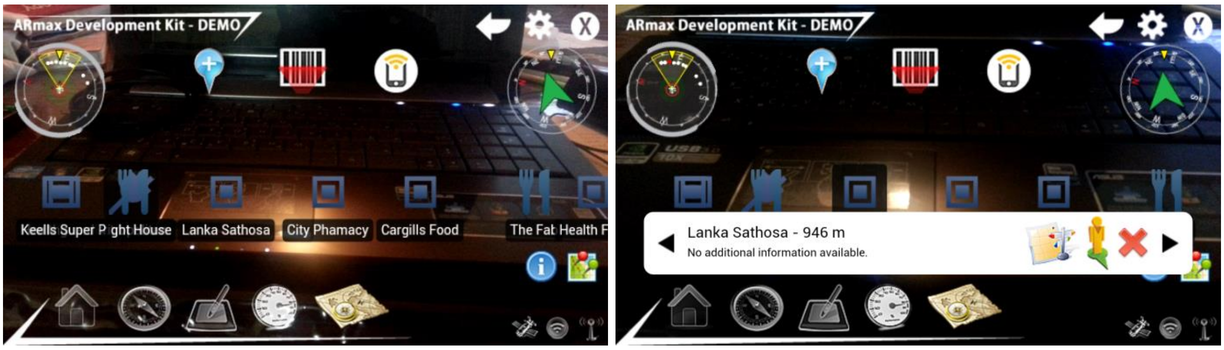


Figure 6.7 – View Places Nearby – Camera Mode

Afterwards, the POIs will be drawn in to a HTML5 canvas layer using a method called "drawResource". This method requires three parameters named userLocation, userAzimuth and poi. The "drawResource" method will determine the POI icon coordinates of the mobile device screen according to the parameters passed. Then an event listener will be bonded to the canvas layer in order to capture the user interaction. Figure 6.7 shows the implemented camera mode screenshots of the view nearby places feature.

Accordingly, Figure 6.8 shows the screenshots of the sub features such as get direction to the place and view streetview of the place. Furthermore, Figure 6.9 shows the screenshot of the implemented barcode scan experimental feature.

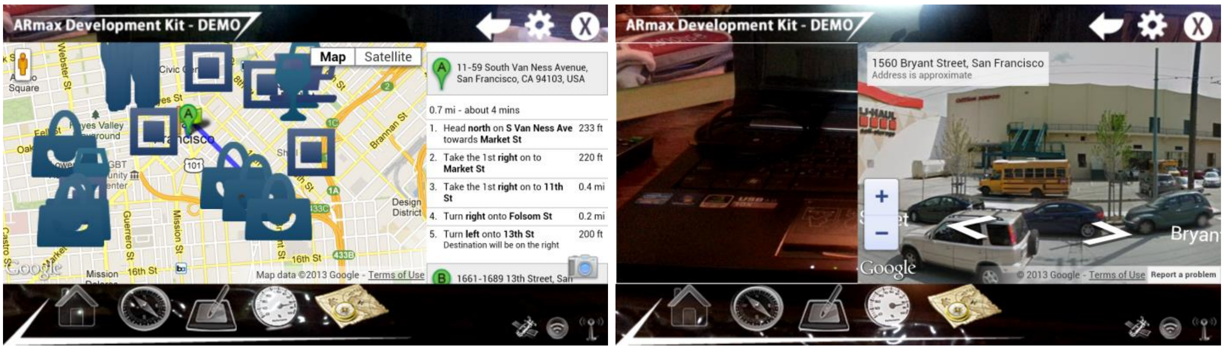


Figure 6.8 - View Places Nearby - Sub Features - Left: Directions to a Place - Right: StreetView of a Place

Figure 6.9 - View Places Nearby - Barcode Scan Feature

In order to scan a barcode, the PhoneGap barcode plugin was used in the utilities of the ARmax framework. Figure 6.9 shows the scan function, which opens a new activity to scan barcodes. Upon a successful scan of a barcode, the results will be shown using a notifications in the camera mode. Code Snippet 6.7 shows the barcode “scan” function.

```
function scan() {
    window.plugins.barcodeScanner.scan(
        function (result) {
            alert("We got a barcode\n" +
                  "Result: " + result.text + "\n" +
                  "Format: " + result.format + "\n" +
                  "Cancelled: " + result.cancelled);
        }, function (error) {
            alert("Scanning failed: " + error);
        }
    );
}
```

Code Snippet 6.7 - Barcode Scan Function

6.8. Problems Faced and Solutions Found

The problems encountered while implementing ARmax prototype and the solutions found are presented in Table 6.2. Apart from stated problems in Table 6.2, there were small issues regarding coding and managed to solve them by researching on the internet.

Problem Encountered	Solution Found	Comments
Getting camera view via JavaScript into browser view seemed to be a problem.	Currently it is handled for Android via natively by adding a separate camera view layer behind the html layer.	As for another solution in HTML5, it is possible to get a camera view into the browser. However, it is very slow and not powerful enough at the time. Another reason for this matter is, it is still in the development stage, and therefore accessing camera view natively is a better solution at the moment. At the moment accessing camera view is done for only Android. Therefore it is being checked in iOS and BlackBerry, whether the camera views is accessible as expected.
Canvas layers of the prototype sometimes didn’t refresh as expected.	This problem is handled in Android by clearing the application cache each time the application loads. super.init(); super.clearCache();	The solution has a progressive outcome when comparing to earlier versions. But rarely the mentioned problem is still encountered.
A proper performance test tool hasn’t been found so far in order to test the prototype.	stats.js is JavaScript performance monitor utility found.	With the use of this utility the ability to measure the FPS and time taken to draw one frame in the canvas was possible.

Table 6.2 - Problems Encountered and Solutions Found

6.9. Summary

This chapter commenced with the selection of technologies and tools for the implementation of the ARmax framework and the prototype. Since, HTML5 is a DOM based specification it is cross-platform and language-independent. Also most of the new features of the HTML5 specification are integrated with JavaScript. Thus, HTML5 and JavaScript were selected as the primary programming languages of the ARmax framework and the prototype.

Accordingly, Adobe Dreamweaver CS6 was selected as the IDE to implement ARmax framework and the prototype, because it supports most leading web development technologies, including HTML, XHTML, CSS, XML, JavaScript, Ajax, PHP, Adobe ColdFusion® software, and ASP. It also supports jQuery Mobile, PhoneGap, HTML5 and CSS3 for mobiles, tablets and computers, where other standard text editors and IDEs does not. Similarly, it discussed the selection of mobile devices and the coding standards that were used in the implementation phase.

Moreover, it briefly discussed the external libraries and technologies such as Adobe PhoneGap, KineticJS, Google Maps JavaScript API, PhoneGap Barcode Plugin, PhoneGap NFC Plugin, jQuery and stats.js which were used in the implementation phase of the ARmax framework and the prototype. Afterwards, the features of the prototype were discussed in detail. The explanations included the respective code snippets and screenshots of the prototype. Lastly, the problems faced during the implementation and the solutions found were presented.

The next chapter will discuss the testing and evaluation phase of the project. It will discuss the test criteria such as module testing, integration testing functional testing and non-functional testing and the corresponding test plans will be specified. The results will be discussed and it includes the bug report and quantitative test results. Accordingly, the product evaluation and the process evaluation will be discussed. Product evaluation will be done under two categories, namely, qualitative (i.e. experts and critical evaluation) and quantitative. Finally, process evaluation will be done under the following categories that is objectives, milestones, deliverables and project aim.

7. Chapter 7 – Testing and Evaluation

7.1. Chapter Overview

Previous chapter provided an overview to the implementation phase of the ARmax framework and its prototype. The selection of programming languages, IDE, mobile devices and coding standards used was conversed. The external libraries and the technologies, which have been used in the ARmax framework and the prototype, were discussed in brief. Finally, the features of the prototype were discussed in detail.

This chapter provides an overview to the testing phase and the evaluation phase of the ARmax framework and its prototype. The testing of ARmax framework and the prototype will be done according to four criterions such as module testing, integration testing, functional testing and non-functional testing. The appropriate test plans and test cases will be presented. Moreover, a power consumption of the ARmax prototype will be tested and compared against the existing AR applications. Formerly, the bug report and the quantitative test results will be presented. Furthermore, the evaluation of the ARmax framework and the prototype will be discussed in detail. Evaluation by the external evaluators, critical evaluation of the ARmax framework and prototype and quantitative evaluation of the testing results will be discussed under product evaluation. Finally, the objectives, deliverables, milestones and the aim of the project will be evaluated under process evaluation.

7.2. Testing Criteria and Test Plan

"Testing is the process of establishing confidence that a system does what it is supposed to." (Hetzel, 1973)

As per quote above it is obvious that testing is an important phase in the software lifecycle, which is very essential to every software solution. Thus, it holds a very significant spot in the overall research, which has been carried out thus far. Also, it was conducted with the intension of finding defects and to measure the quality of the ARmax framework and its prototype in terms of quantitative data. The testing phase was carried out under a number of different types of testings' such as module testing, integration testing, functional testing and non-functional testing. The module testing and integration testing was carried out in order to confirm whether the design goals of the ARmax framework have been met and to check the practical usability of the ARmax framework. Moreover, functional and non-functional testing confirms that the prototype of the ARmax framework adheres to the requirements which were made in chapter 4 (Requirements), which is critical to assess the overall progress of the research.

The module, integration, functional and non-functional testing was carried out according to a test plan. Thus, the test plan was an important aspect of the overall testing phase. Furthermore, the test cases of the test plan focused on code coverage, boundary conditions, logical conditions, hardware limitations and platform compatibility. Accordingly, the test cases were made and will be discussed in the subsequent sections.

7.3. Functional Testing

Under this section, the functional requirement which were set in section 4.7.4 (Functional Requirements) were tested. The testing of functional requirements was very essential, because it provides an overall impression of the implementation phase and the status of the ARmax prototype. The test results which are shown in Table 7.1 summarise the final implementation status of the ARmax prototype.

ID	Priority	Implementation/Design Status	Requirement	Use Case	Test Status
FR1	Medium	Implemented	Test Compass	UC-2	Tested
FR2	Medium	Implemented	Test Triggers	UC-3	Tested
FR3	High	Implemented	Test Performance	UC-4	Tested
FR4	High	Implemented	View Places Nearby	UC-5	Tested
FR5	High	Implemented	Scan QR Code	UC-6	Tested
FR6	Low	Designed	Scan NFC Tag	UC-7	Not Tested
FR7	Medium	Implemented	Navigate To Place	UC-8	Tested
FR8	Low	Implemented	Set A Place	UC-9	Tested

Table 7.1 - Functional Requirements Test Summary

7.4. Module Testing

In these sections different modules such as graphic, sensor, map, and QR code/barcode recognition will be tested. In addition, the test plans include purpose, equivalent class, expected output, actual output and the status. Most of the time each test case was tested several times. Thus, the pass rate (i.e. the pass number of times / total number of times tested) of test case has been added to the test plan.

7.4.1. Graphic Module

Under this module test, the features of the graphic layer such as node nesting, layering, caching, animations, transitions, shapes and events will be tested. The test cases showed in Appendix XII – A – Test Plan 10.1 were used to test the features of the graphic layer. Moreover, there were no preconditions associated with these test cases.

7.4.2. Sensor Module

The sensor layer of the ARmax framework was tested under this module test. Primarily, retrieval of sensor data was tested. The compass and the accelerometer testing did not need an internet connection in order to retrieve data. However, geolocation sensor testing needed an internet connection to retrieve location data. Therefore, retrieval of location information was tested on different internet connection types such as 3G and Wi-Fi. Thus it was a precondition which was associated with the geolocation sensor in every mobile device. Accordingly, sensor data retrieval was tested among a number of mobile devices such as Samsung Galaxy SIII, Apple iPhone 4 and HTC Wildfire S. Selection of these mobile devices were associated with a precondition, which was these mobile devices should have sensors such as compass and accelerometer and a built-in GPS module. Moreover, the test cases shown in Appendix XII – B – Test Plan 10.2 was used to test the sensor layer of the ARmax framework.

7.4.3. Map Module

The features such as add marker(s), get directions for a specific place/location and streetview of the ARmax framework were tested under this module. The test cases shown in Appendix XII – C – Test Plan 10.3 were used to test the map module. Moreover, there were no preconditions associated, apart from the internet connectivity.

7.4.4. QR Code/Barcode Recognition Module

The QR code/barcode scan utility of the ARmax framework was tested under this module. The testing was done with the use of a Samsung Galaxy SIII mobile phone. The tests included different QR code information types (i.e. URL, text, phone number, SMS and contact), different QR code sizes (i.e. small, normal, large and very large) and different barcode types (i.e. Code 128, UPC-A, Data Matrix and EAN_8). Apart from that, the lighting condition of the environment was a key testing area, because it could affect the recognition process of the QR code/barcode. Thus, scanning of QR codes was carried out under different lighting conditions (i.e. low and normal light environment). Finally, a number of custom QR codes with different colours (i.e. red, green and black) were tested. The test cases showed in Appendix XII – D – Test Plan 10.4 were used to test QR code/barcode recognition module.

7.5. Integration Testing

Integration testing is a mechanism to expose defects in the interfaces and interaction between integrated modules. Thus, this section was used to test the compass and direction indicator elements, which consists of graphic, sensor and map modules which were tested above. The tests mainly cover the compass functionality in both the compass and direction indicator elements which is a combination of graphic and sensor modules. Apart from that, the unique functionalities such as setting a place, getting path and directions and calculating distance of the direction indicator will be tested under integration testing. The test cases shown in Test Plan 7.1 were used to test the stated functionalities. Moreover, there were no preconditions associated with the test cases, apart from the internet connectivity, because the geolocation sensor and map module needed the internet connectivity to function properly.

No	Purpose	Expected Output	Actual Output	Pass Rate	Status
EC1 Compass functionality on compass element and direction indicator element					
1.1	Turn device to north direction	Compass north should point to north and the heading should be 0°	Compass north pointed to north and the heading was 0° (accuracy +-2°)	20/20	Pass
1.2	Turn device to east direction	Compass north should point to north and the heading should be 90°	Compass north pointed to north and the heading was 90°(accuracy +-3°)	20/20	Pass
1.3	Turn device to south direction	Compass north should point to north and the heading should be 180°	Compass north pointed to north and the heading was 180° (accuracy +-2°)	20/20	Pass
1.4	Turn device to west direction	Compass north should point to north and the heading should be 270°	Compass north pointed to north and the heading was 270° (accuracy +-2°)	20/20	Pass
1.5	Animation of the element	Animation of the element should be smooth	Animation is smoothly rendered	18/20	Fail
EC2 Direction Indicator					
2.1	Tap map icon	Map of the direction indicator should be show	Map showed	20/20	Pass
2.2	Set a place	Directions and path to the new location should be update in the map	The directions and path updated in the map	20/20	Pass
2.3	Pointing arrow of the direction indicator	The direction indicator's pointing arrow should point to the destination's coordinates.	The direction indicator's pointing arrow is pointed to the destination's coordinates.	20/20	Pass

2.4	Straight-line Distance	The distance from end user's location to the destination should be updated regularly.	The distance is updating regularly	20/20	Pass

Test Plan 7.1 - Compass and Direction Indicator Integration Test Plan

7.6. Non-Functional Testing

As per the discussion in section 4.7.5 (Non-Functional Requirements), a set of non-functional requirements were identified. Moreover, non-functional requirements tend to be those that reflect the quality of the product, particularly in the context of the suitability perspective of its users. Thus, testing non-functional requirements was vital for the ARmax prototype. As a result, non-functional requirements such as performance, efficiency and platform compatibility were given priority and thoroughly tested. The stated requirements in section 4.7.5 (Non-Functional Requirements) such as, it should take less than 5s to retrieve location information from GPS, it should take less than 5s to process and display the QR code information and the update rate of a graphic should be more than 5 were taken as the parameters when testing. The tests were carried out a number of times depending on the test case priority and severity. Moreover, some performance tests were carried out in mobile and desktop platforms. The test cases and the results are shown in Test Plan 7.2 which used to test the non-functional requirements.

No	Purpose	Expected Output	Actual Output	Pass Rate	Status
EC1 Performance					
1.1	Time taken to retrieve location information.	Time taken to retrieve location information <= 5s	Time = approximately 4s - 13s, Avg = 4s	36/40	Fail
1.2	Time taken to process and display the QR code information.	Time taken to process and display <= 5s	Time = approximately 3s - 5s, Avg = 3s	20/20	Pass
1.3	The update rate of the graphics				
1.3.1	Mobile platforms				
1.3.1.1	Android browser 4.1.2 in a Samsung Galaxy SIII	FPS >= 5 Time taken to draw one graphic >= 200ms	FPS = 5 – 8, Avg = 7 Time = 125ms – 183ms	5/5	Pass
1.3.1.2	Safari browser in a iOS Simulator - iPhone (Retina 4-inch) / iOS 6.1		FPS = 10 – 19, Avg = 18 Time = 27ms – 139ms	5/5	Pass
1.3.1.3	Safari on iPhone 4 / iOS 6		Not Tested	N/A	N/A
1.3.1.4	Chrome Nightly 26.0.1410.58 in a LG Nexus 4		FPS = 4 – 6, Avg = 6 Time = 151ms – 262ms	0/5	Fail
1.3.2	Desktop browsers				
1.3.2.1	Google Chrome 26	FPS >= 5 Time taken to draw one graphic >= 200ms	FPS = 40 – 58, Avg = 57 Time = 4ms – 18ms	5/5	Pass
1.3.2.2	Firefox 21.0		FPS = 14 – 24, Avg = 23 Time = 32ms – 48ms	5/5	Pass

1.3.2.3	Internet Explorer 10.0		FPS = 13 – 23, Avg = 22 Time = 35ms – 64ms	5/5	Pass
1.3.2.4	Safari 6.0		FPS = 19 – 26, Avg = 23 Time = 12ms – 46ms	5/5	Pass
1.3.2.5	Opera 12.15		FPS = 25 – 35, Avg = 34 Time = 25ms – 43ms	5/5	Pass
EC2 Platform Compatibility					
2.1	Mobile platforms				
2.1.1	Android browser 4.1.2 in a Samsung Galaxy SIII	Prototype of the ARmax framework should work in the particular mobile device	Prototype worked	20/20	Pass
2.1.2	Safari on iPhone 4 / iOS 6		Not tested	N/A	N/A
2.1.3	iOS Simulator - iPhone (Retina 4-inch) / iOS 6.1		Prototype worked. Since it's a simulator some of the sensors data were not received.	20/20	Pass
2.2	Desktop browser support to the graphic layer and map layer of ARmax framework				
2.2.1	Google Chrome 26	Canvas based graphics should show	Graphics rendered well	20/20	Pass
2.2.2	Firefox 21.0		Graphics rendered well	20/20	Pass
2.2.3	Internet Explorer 10.0		Graphics rendered well	20/20	Pass
2.2.4	Safari 6.0		Graphics rendered well	20/20	Pass
2.2.5	Opera 12.15		Graphics rendered well	20/20	Pass
EC3 Efficiency					
3.1	The canvas should be able to process minimum number of 25 visual objects (i.e. images, shapes, animations, radar, etc.) in the AR view.	Canvas should show minimum number of 25 visual objects	Canvas should showed 25 visual objects	20/20	Pass

Test Plan 7.2 - Non-Functional Requirements Test Plan

7.7. Power Consumption of Different AR Applications

It was recommended to test and compare the power consumption of the ARmax prototype against other existing AR applications. Thus, a power consumption test was carried out in Samsung Galaxy SIII mobile device with the use of an open source power testing tool named "Power Tutor". Wikitude Browser and Layar were taken as existing AR applications. Each application was opened for 5 minutes and the power consumption of each application was recorded. The power consumption was measured in J (Joule). The approximate power consumption test results are shown in Chart 7.1.

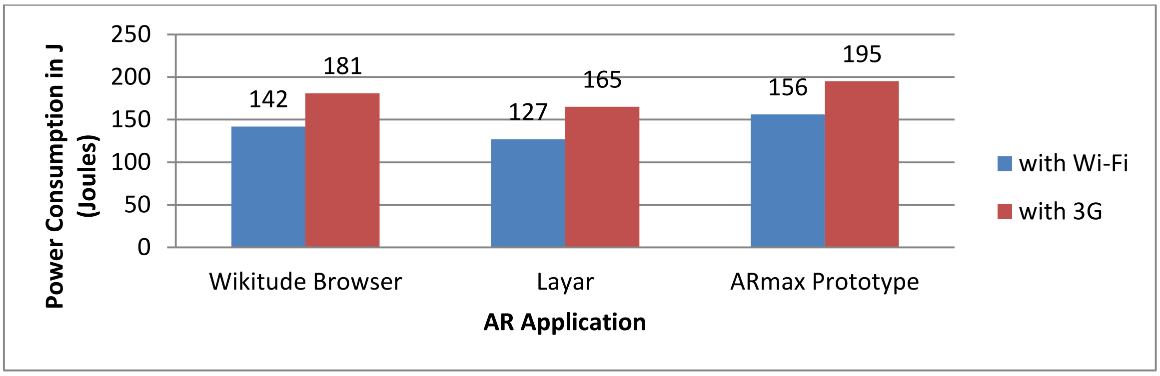


Chart 7.1 - Power Consumption of Different AR Applications

7.8. Test Results

7.8.1. Bug Report

The bugs which were discovered in the implementation phase and during testing phase are presented in Appendix IX – Table 10.14. The bug details such as id, name, build number, severity, priority, reason, status and environment were recorded in order to provide a solution.

7.8.2. Test Results Summary

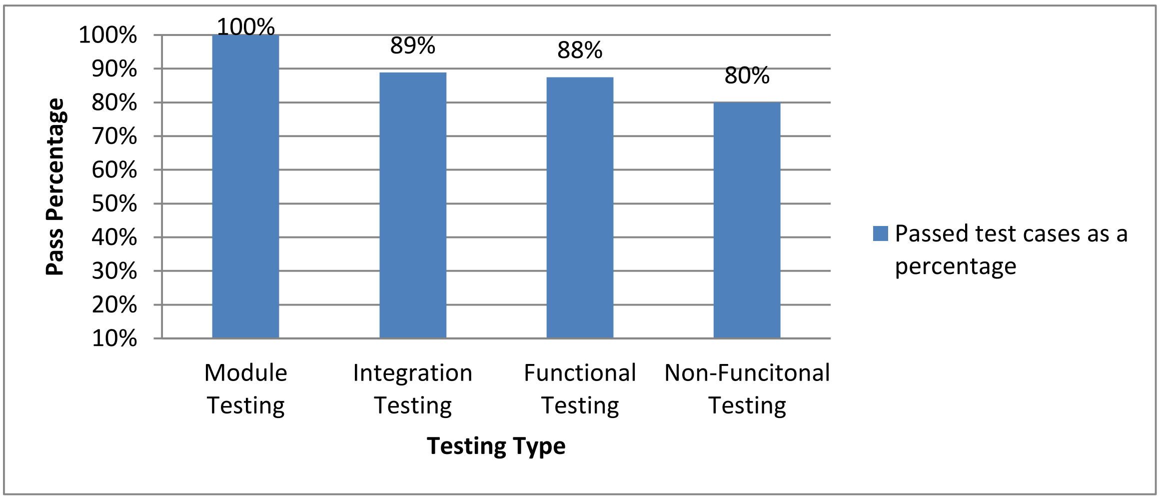


Chart 7.2 - Testing Summary

The summary of the testing phase is presented in Chart 7.2, where it shows the pass percentages of the test cases in module testing, integration testing, functional testing and non-functional testing. In module testing, all of the test cases were passed successfully. Hence, 89% of the test cases successfully passed in integration testing. However, one functional requirement was not tested, thus 88% of the test cases passed under functional testing. Moreover, 80% of the test cases passed under non-functional testing. Furthermore, a detailed evaluation of the test results will be discussed in section 7.10.4 (Quantitative Evaluation of the Test Results).

The next section will discuss the evaluation phase of the ARmax framework and its prototype.

7.9. Evaluation

The evaluation of the ARmax project is done primarily in two stages. First the ARmax project is evaluated using a combination of qualitative and quantitative methods, which will be discussed under product evaluation. Secondly, the objectives, milestones, deliverables and aim of the ARmax project will be discussed under process evaluation.

7.10. Product Evaluation

7.10.1. Evaluation Methodology

The initial survey which was carried out under chapter 4 (Requirements) validated the concept and approach of the ARmax project. However, the main focus of this part is to qualitatively analyse the whole ARmax project using a questionnaire at the end of the project. A YouTube video was made to demonstrate the prototype online. The link of the video was then forwarded together with a questionnaire to a group of people ranging from end-users to domain experts such as web developers, software engineers, software architects, project managers and researchers. The questionnaire can be found in Appendix XI - B.

7.10.2. Evaluation by External Evaluators

As per the discussion above, the evaluation by external evaluators was carried out with the use of a questionnaire. The questionnaire was created by considering a number of areas of the project such as the concept of the solution, selection of tools and technologies, ARmax prototype in terms of features and usability, application areas and the potential of the solution, drawbacks, improvements and future enhancements.

The Solution:

"Concept of using JavaScript is novel. Present AR SDK's requires to use compiled binaries for execution. New websites need to be based on HTML5 technology. Therefore using these for future AR applications is the correct way to go. Native development solutions would be faster, but the cost benefit of using a cross-platform SDK vs. speed of execution is high. Since these systems are not used for time critical applications, combining cross-platform technologies is ok."
– Cassim Farook – Software Consultant

"The concept of use of AR with mobile devices is a good idea that has taken the current moment in the industry. AR and mobile device technologies such as HTML5 are used in the industry now. It does provide an advantage to the developer and enable stable development strategies." – Kulari Lokuge – Senior Learning Technologist, Swinburne University of Technology

"I believe the proposed concept addresses a key problem lot of mobile application developers faces in today's world. Due to the many number of mobile operating systems, it is sometimes financial non-viable for development companies to invest on implementing augmented reality based solutions due to platform dependency which demands for different implementations for different platforms, which can be very costly. The proposed solution addresses this particular problem. In my opinion, the use of latest www standards such as HTML5/CSS3 is a great move. This opens up lot of opportunities for many developers who wish to use the proposed framework. As of today, almost every smartphone and computer operating system supports HTML5/CSS3/JS rendering in great extent and is moving towards making that the primary platform to develop applications (Firefox mobile is a good example for this). Having a framework to work around AR in a web application can make it lot easier for developers to implement AR solutions to variety of platforms with ease. Despite popular belief, Frameworks like PhoneGap is not a great solution for every application. PhoneGap, for example, does not have sufficient support for every function/feature natively supported by iOS SDK. If the developer to use such feature, he/she will first have to write a phonegap wrapper for that specific feature and then use that wrapper through phonegap. This pretty much makes the effort of the developer an unproductive one. But for some large projects, phone gap may actually be useful. So as I said, it's subjective and not great for every application. For some apps, native dev solutions are better. For some, phonegap like solutions are better." – Roshan Amadoru – CEO – Codespark / Visiting Lecturer - IIT

All of the evaluators point out that the concept of the ARmax solution was novel, where it does provide an advantage to the developer and enable stable development strategies. Moreover, they point out that the cost benefit of using a cross-platform SDK is high compared to native SDKs. The selected technologies such as HTML5 and JavaScript were conversed and accepted as the correct choice.

The ARmax Prototype:

"I can see that it is smooth." – Cassim Farook – Software Consultant

"This is a good example, I would have liked to see other applications." – Kulari Lokuge – Senior Learning Technologist, Swinburne University of Technology

"I believe the prototype has been well thought of and designed. It performs well in a resource constrained mobile device environment, and provides various different user interactions/gestures, that can be used by developers." – Roshan Amadoru – CEO – Codespark / Visiting Lecturer - IIT

In general, the ARmax prototype was accepted among both end-users to domain experts. It had a significant attractiveness and a proper user interaction. The features such as compass element, direction indicator, view places nearby and street view got more attention and were conversed.

Benefits of the Proposed Solution:

"Easy to use and deploy" – Cassim Farook – Software Consultant

"This would encourage devise diversity and enable platform independence" – Kulari Lokuge – Senior Learning Technologist, Swinburne University of Technology

In section 1.4 (Project Aim and Scope), the aim of this project was to build a highly efficient and easy to use cross platform augmented reality framework for developers. According to the above feedback from the evaluators the aim of the project could be considered as achieved.

Application Areas and the Potential of the Solution:

"Travel, advertising, sports, hiking. It could be adopted once the SDK is put up for the public." – Cassim Farook – Software Consultant

"Media, safety, sports, medical. Yes, will require performance and accuracy justification evidence" – Kulari Lokuge – Senior Learning Technologist, Swinburne University of Technology

"Travel, advertising, GIS. Yes, definitely. The world is heading to an era of cutting edge wearable computing solutions, and augmented reality has a key focus in the wearable computing field. The proposed solution, due to it's platform neutrality, can be of great value to develop augmented reality." – Roshan Amadoru – CEO – Codespark / Visiting Lecturer - IIT

Most of the evaluators have seen and commented on the commercial potential of the ARmax framework. Moreover, they suggested that it will require performance and accuracy justification evidence, which was presented in section 7.8 (Test Results). Also, they suggested that there are many application areas such as travel, advertising, media, sports and medical.

Drawbacks of the Solution:

"Depends on JavaScript and CSS browser support. Sometimes certain browsers actually do not conform to standard specifications." – Cassim Farook – Software Consultant

"What support will you provide for the two different O/S? Ability to use graphic, media with equal performance in both systems, ability to support other functionalities that are specific to a device or O/S while the proposed system is used for development" – Kulari Lokuge – Senior Learning Technologist, Swinburne University of Technology

"One key drawback of the implemented solution is the non-existence of object identification support. In many AR applications, object identification plays a key role, and helps the developers to have a reference point to provide enhanced user interaction. However, since this is only a B.Sc. final thesis project, it is understandable the student will not have sufficient time to implement the entire framework and the object identification during the project timeline." – Roshan Amadoru – CEO – Codespark / Visiting Lecturer - IIT

As per the discussion in section 2.10 (Mobile Browser Support), it is true that certain browsers do not conform to standard specifications. However, this project primarily uses the canvas features such as canvas element, 2D context and text in HTML5 specification. As shown in Table 7.2, most of the browsers support HTML5 canvas features. Thus, it seems the raised drawback might not affect the proposed solution at all.

Canvas Feature	Browser Category – Support Percentage (No. of Browsers Support / Total Browsers)				
	Desktop	Tablet	Mobile	Television	Gaming
Canvas Element	93.02% (40/43)	100% (25/25)	95.65% (44/46)	100% (11/11)	100% (8/8)
2D context	93.02% (40/43)	100% (25/25)	93.48% (43/46)	100% (11/11)	100% (8/8)
Text	88.37% (38/43)	100% (25/25)	91.30% (42/46)	100% (11/11)	75% (6/8)

Table 7.2 - Canvas support of different browsers (The HTML5 Test, 2013, http)

Improvements and Future Enhancements:

"A debug mode or trace log facility for the developer" – Cassim Farook – Software Consultant

"Ability to define efficiency/performance criteria" – Kulari Lokuge – Senior Learning Technologist, Swinburne University of Technology

"As I said about, the object identification support must be given to the framework. Further, the map must be enhanced to provide a more augmented feeling to the user than just a opaque map which sits behind the user and the environment. A proper turn-by-turn navigable map implementation can greatly help to improve this implementation." – Roshan Amadoru – CEO – Codespark / Visiting Lecturer - IIT

As future enhancements above suggestions have been suggested by the evaluators. One suggestion was to define efficiency/performance criteria. However, to some extent this has been covered by the performance test feature of ARmax prototype, where it measures the browser's refresh rate (FPS). However, it has been taken in to consideration in order to improve the quality of the ARmax framework.

7.10.3.Critical Evaluation of ARmax Framework and Prototype

"All truths are easy to understand once they are discovered; the point is to discover them." Galileo Galilei (1564 – 1642)

As per the quote above, it is easy to understand that web technologies such as HTML5 and JavaScript could be used to develop augmented reality experiences for the day-to-day consumers around the globe in a variety of fields such as travelling, healthcare, music, advertising, art, safety and navigation. However, ARmax project was about discovering the truth by providing a working prototype as proof-of-concept. The ARmax project's primary goal was to design and develop a cross-platform augmented reality framework for developers, while demonstrating this approach with a few reference applications. The objectives set initially were met at the end of the project, thus the project is a huge success. Features such as test compass and view nearby places of the ARmax prototype and the design of the ARmax framework could be considered as key achievements of this project.

Moreover, the variation of mobile devices and platforms was a main problem, which is faced by developers at present in mobile augmented reality application development. However, this problem was solved by proving that the prototype of the ARmax framework is a cross-platform solution, which can be successfully deployed in multiple mobile platforms (i.e. Android and iOS) without any code change. Accordingly, the cost of maintenance and management of the prototype is less, where it provides an improved end user support. These are the benefits that a company could gain by using ARmax framework.

In addition, most of the developers are familiar with web technologies such as HTML and JavaScript and ARmax framework is mainly written in HTML5 and JavaScript. Therefore the skills and the effort that is needed to understand the code is less compared to native languages such as Objective C, Java and C#. Similarly, the test results in section 7.6 (Non-Functional Testing) shows that the graphic layer and the map layer of the ARmax framework could be used in desktop browsers as well. Thus, these facts can be considered as plus points of the ARmax framework.

However, as with any project, a number of limitations have been identified in ARmax framework and the prototype. One limitation is the performance of the prototype. It is true that the ARmax uses the features of the Adobe PhoneGap cross-platform solution. Thus, the performance of the prototype is limited to the mobile device's web view. However, according to the test results in section 7.6 (Non-Functional Testing), the performance of the web view could be considered as acceptable. Accessing the mobile device's camera view was the other limitation of the ARmax framework. At the moment, the camera view is placed natively into the background of the ARmax prototype. Nevertheless, the HTML5 specification for accessing camera view is still in progress and according to the results in section in 2.10 (Mobile Browser Support) not many browsers support that feature thus far. Thus, it could be considered as a future enhancement of the framework.

Finally, the ARmax framework was the first step of a long journey in mobile cross-platform augmented reality application development. Thus, there will always be room for further improvement of the ARmax framework

7.10.4.Quantitative Evaluation of the Test Results

Most of the test cases of the module testing and integration testing which were discussed in section 7.3 and 7.4 were passed successfully. The technologies and libraries (i.e. HTML5, KineticJS and PhoneGap) used in the ARmax framework and the prototype was already thoroughly tested by the respective parties. Thus, it was one of the main reasons to gain such results. Moreover, the non-functional requirements of the ARmax prototype were mainly evaluated against the use of the test results which were discussed in section 7.6 (Non-Functional Testing).

The test results show that approximately 4s – 13s (avg. = 4s) was taken by the geolocation sensor to retrieve location information. The acceptable range was defined as less than or equal to 5s, therefore the test case failed. Nonetheless, during the testing it was noticed that the time taken for retrieval of location information was depending on the internet connection speed. Which means higher internet connection speed retrieves location information quickly. Accordingly, the results show that time taken to process and display the QR code information took 3s – 5s (avg. = 3s) and the acceptable range was defined as less than or equal to 5s. Therefore the test case passed. Also it was discovered that the recognition time of the QR code might take more than 5s depending on the size of the QR code and the amount of information which is encoded in the QR code. Large amount of information means that it takes more time to recognise the QR code. However, once recognised the time taken to process and display information was always in the acceptable range.

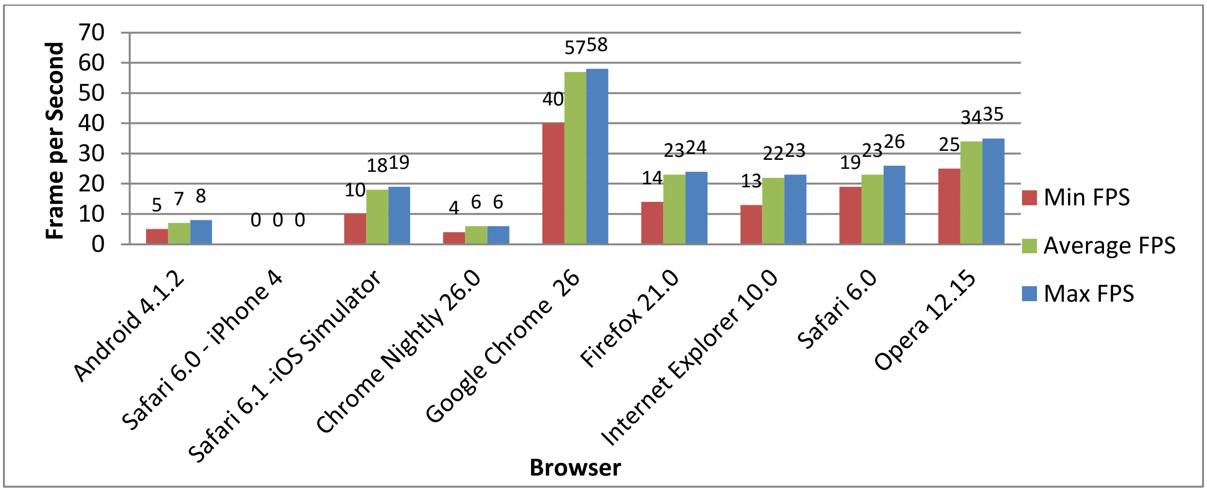


Chart 7.3 - The Update Rate of the Graphics in Mobile Browsers and Desktop Browsers

The update rate of the graphics was tested and the results are shown in Chart 7.3. The test was initially carried out in four mobile browsers and five desktop browsers. The performance test was conducted 5 times and each time it was done for 42s. According to the results, Google Chrome 26 desktop browser had a maximum FPS of 58 and minimum FPS of 40 (avg. = 57). Opera 12.15 was the second best and Safari 6.0 was the third. Approximately, Firefox 21.0 and Internet Explorer 10.0 were shown the same FPS. From the mobile browsers Safari was the best, where it had a maximum FPS of 19 and a minimum of 10 (avg. = 18). However, the Safari browser was tested in an iOS simulator - iPhone (Retina 4-inch) / iOS 6.1. Therefore, the FPS values received will leave a doubt, which is the actual mobile device (i.e. iPhone 4) would also perform likewise. Nevertheless, Android browser 4.1.2 in a Samsung Galaxy SIII was the second best mobile browser with maximum FPS of 5 and a minimum FPS of 8 (avg. = 7). Chrome Nightly browser 26.0.1410.58 in a LG Nexus 4 was the mobile browser with the lowest FPS, which was a maximum FPS of 6 and a minimum FPS of 4 (avg. = 6). However, 5 FPS and 200ms per frame was the defined acceptable range, therefore except Chrome Nightly browser 26.0.1410.58 in a LG Nexus 4, all the browsers which were tested have passed successfully.

Moreover, the platform compatibility was tested with the use of the same desktop browsers and mobile devices excluding Google Nexus 4. The prototype was successful in Samsung Galaxy SIII. The prototype could not be tested on an iPhone due to limited resources. Thus, as an alternative solution the testing was done with the use of an iOS simulator - iPhone (Retina 4-inch) / iOS 6.1. Since it is a simulator there were no sensors, which mean there were no any sensor data. In order to test the prototype a set of sensor data were generated and used. Similarly, the same set of generated sensor data was used to test the desktop browsers, and graphic layer and map layer was given priority. The test cases passed successfully. This means that even without the sensor layer, ARmax framework is capable and could be used to create advance graphics and map related applications.

Furthermore, the efficiency was tested as specified in section 7.6 (Non-Functional Testing), and as expected according to the previous tests which were carried out earlier, it passed without leaving any doubts. Finally, a power consumption test was carried out among existing AR applications such as Wikitude Browser and Layar and ARmax prototype. Layar consumed approximately 127J and it was the lowest. Wikitude Browser consumed approximately 142J and became second. ARmax prototype was the highest power consumed application with an approximate value of 152J. The camera view is always open in ARmax prototype, where in other applications it is not. Thus, it might be the reason for the extra power consumption. However, it was noticed that the display of the mobile device consumes 75% or higher amount of power from the total power consumption of the application. Similarly, it was noticed that 3G data connection consumed more power comparing to a Wi-Fi connection.

7.11. Process Evaluation

In this section the objectives, deliverables, milestones and project aim will be evaluated.

7.11.1. Objectives

At the start of ARmax project, a number of objectives have been set. The evaluation of those objectives will be discussed in this section. The first objective was to prepare the TOR to identify the feasibility of the ARmax project. The TOR was prepared successfully after a proper feasibility study without having any problems. The next objective was to carry out a literature survey in augmented reality concepts, existing AR techniques and technologies, existing AR applications, tools and frameworks to understand what and how the AR concepts have been implemented during the past, and to gain an insight about future. A proper literature review was carried out mainly in AR and mobile application development, and it was helpful to determine the most optimal AR features and functionalities which should be supported by the ARmax framework. In addition, it was helpful to determine the tools, techniques and technologies which could be used in the ARmax prototype. Then, an online survey was carried out among professional such as project managers, mobile application developers and AR researchers in order to identify the limitations of the existing AR frameworks. This also revealed the problems with mobile cross platform application development which are faced by developers. Correspondingly, the online survey was useful to gather the improvements and new requirements, which the ARmax framework should provide. Also, it was better to conduct a number of interviews face to face. However, due to the limited time it was not possible.

The next objective was to determine and to formulate the requirement specification for the ARmax framework. The requirements gathered via online survey and the literature survey was very useful to produce a proper SRS. Moreover, OOA was used to analyse the requirements and UML models such as use case diagram and domain model was used to model the functional requirements and the high level designs of the ARmax framework and the prototype. This was followed by the design and implementation phases of the ARmax framework and the prototype. Furthermore, the SRS was beneficial in these phases. However, a number of problems were encountered during the implementation phase of the ARmax framework and the prototype and was able to manage those issues with the guidance of the project supervisor and through self-research.

Afterwards, a critical testing was carried out on implemented prototype and ARmax framework. The testing was carried out in four criterions such as module testing, integration testing, functional testing and non-functional testing. Also, the 4th year subject (i.e. Software Quality Assurance) was very helpful to specify the test plan and test cases for the testing. Subsequently, a critical evaluation was carried out of the ARmax framework and the prototype. A product and process evaluation was done. However, the quality of the evaluation could have been increased more, if it were able get more opinions from the external evaluators. However, due the limited time it was not feasible to meet many experts and end users. Finally, the areas for future enhancement were identified with the use of the evaluation process.

7.11.2.Deliverables and Milestones

As per the discussion in section 1.7 (Project Deliverables), a set of deliverable were set initially of the ARmax project. The stated deliverables were successfully produced and submitted. Along the way there were a number of shortcomings due to the limited time and resources. But, was able to manage through proper time management and with the help of the project supervisor and fellow undergraduates. Apart from that, the knowledge and experience gained during the past years and in the internship year was very helpful to produce the deliverables. Moreover, each deliverable submission was a milestone of the ARmax project. As mentioned earlier, most of the submissions were able to submit in time, whereas some didn't. The Appendix X – Table 10.15 shows a complete summary of the deliverable, milestones and the submission dates of the ARmax project.

7.11.3.Project Aim

As per the discussion above, it appears that the objectives, deliverable and the milestone were achieved. There were a number of ups and downs along the way. However, those were managed and resolved systematically. Thus, the aim of the ARmax project could be considered as having been accomplished successfully.

7.12. Summary

This chapter commenced with defining testing criteria and test plans which was targeted to test the implemented ARmax framework and the prototype. The testing criterions were module testing, integration testing, functional testing and non-functional testing. The core functionalities of the ARmax framework were tested mainly under the module testing, where it included areas such as graphic, sensor, map and QR code/barcode recognition modules. The module testing was 100% successful. The integration testing was targeted to test the compass and direction indicator elements, where the different modules such as graphic, sensor and map layer are integrated with each other. The integration testing was 89% successful. The features of the ARmax prototype were tested under functional testing. The stated features in the section 4.7.4 (Functional Requirements) were mainly tested and the testing was 88% successful. The non-functional features such as performance, platform compatibility and efficiency were tested under non-functional testing. The testing was carried out in a number of mobile devices and desktop browsers, where 88% of the test cases were successfully passed. Afterwards, a power consumption test of the ARmax prototype and the existing AR applications was carried out with the use of an open source power testing tool named "Power Tutor". Then the bug report and the test results summary were discussed.

The evaluation of the ARmax framework and prototype were discussed in detail under two criterions. Those were product and process evaluation. An online survey was used to evaluate the ARmax framework and the prototype by external evaluators. Also, an online video was published to demonstrate the features of the ARmax prototype. The evaluators pointed out that the concept of using JavaScript for mobile augmented reality applications is novel. Also, they pointed out that the cost benefit of using a cross-platform SDK is high compared to native SDKs. Similarly, they pointed out that the prototype worked smoothly and was easy to use and deploy and these were highlighted as main benefits. Furthermore, they have commented on the potential of commercial use of the ARmax framework. Travel, sports, hiking and advertising are highlighted as application areas that this solution is most useful. Apart from that, they have commented on the drawbacks, improvements and future enhancements of the ARmax framework. Afterwards, the critical evaluation of the ARmax framework and the prototype by the research student was discussed. The features of ARmax framework and the prototype were considered as the key achievements of the project. It was highlighted, that use of ARmax framework benefits a company by reducing cost of maintenance and management with an improved end user support. However, performance of the application was identified as one of the main limitations of the framework. Then a detailed quantitative evaluation was carried out on the test results. The test results proved that the ARmax prototype are more or less similar to the existing applications and in some cases more improved than them.

Moreover, the objectives, deliverables, milestones and the aim of the project were discussed under process evaluation. Each objective was described in detail and how it was carried out. The shortcomings due to the limited time and resources and how it is managed was discussed under deliverables and milestones. The planned and the actual submission dates were presented with the use of a table. Finally, the evaluation of the project aim was discussed briefly.

The next chapter will conclude the project carried out thus far. It will briefly discuss the problems and challenges faced during the project, the learning outcomes of the project, and limitations of the ARmax framework and future enhancements. Finally, the concluding remarks will be presented.

8. Chapter 8 – Conclusion

8.1. Chapter Overview

The testing and evaluation of the ARmax framework and its prototype were discussed in detail in the previous chapter.

This chapter will conclude the project which has been carried out thus far. Also, it will be briefly discuss the problems and challenges faced during the project, the learning outcomes, and limitations of the ARmax framework and future enhancements. Finally, the concluding remarks will be stated.

8.2. Problems and Challenges Faced During the Project

During the course of the project, there were a number of problems and challenges that were encountered. Managing time and resources was the main challenge of the project, because those constraints were limited. Apart from that, there were a few other problems that were encountered in the implementation phase of the project. However, those problems were solved in time with the help of the supervisor and through research on the internet. In the testing phase, the main problem was the resources. In order to test the ARmax prototype, a number of mobile devices were needed in different platforms such as iOS and Android. It was easy to find Android devices, whereas iOS devices were not that easy.

8.3. The Learning Outcomes of the Project

By undertaking this project, it enabled the research student to develop and improve both his soft and technical skills. Among the soft skills, time management abilities, problem-solving skills, communication skills and work well under pressure were highlighted and the project helped to improve these skills further. Also, it enabled the research student to gain a thorough knowledge in AR and mobile cross-platform development. Also, the experience gained through working with different technologies such as HTML5, JavaScript and PhoneGap could be considered as the primary technical skills gained during the course of this project. Apart from these, report writing skills made a significant improvement over the period of this project. Moreover, the experience and the learning gained by preparing the literature review, SRS, design and implementation, testing and evaluation were great. As a result, it helped to gain a valued self-confidence over the contents which are covered in this research project and that could be considered as the most significant learning outcome of this project.

8.4. Limitations of the ARmax Framework

The ARmax framework fully supports the geo location and sensor based AR features and experiences. The framework supports to recognise QR code, yet it is the basic image recognition technique. Thus, the support for the image recognition based AR features and experiences are limited. Since, the HTML5 specification for accessing camera is not finalised yet, accessing camera view could be considered another limitation of the framework.

8.5. Future Enhancements

There were a number of improvements that were identified during the implementation, testing and evaluation phases of this project. The external evaluators made a few suggestions in order to improve the ARmax framework. However, due to limited time those improvements were not possible. Thus, the identified improvements and suggestions were taken as future enhancements.

Future Enhancements:

- Support more pre-defined AR elements (i.e. similar to compass and direction indicator).
- Trace log facility for the developer.
- Support for image recognition based techniques.
- NFC based interactions.

8.6. Concluding Remarks

This project mainly addressed the problem of the development of mobile augmented reality applications into a number of different mobile platforms. The solution was to build a cross-platform AR development kit (framework), where developers just have to build one application and deploy it in different mobile platforms. The project has its ups and downs and it required a significant sum of innovative ideas and solutions which does not comes in a line of vision. Carrying out this project for nine months helped in gaining a valuable amount of knowledge and experience through each phase of the project life cycle. Thus, the project was a notable success and it has the scope and the content needed to qualify as an academic research project. Moreover, the outcome of the project, which is the ARmax framework, has the potential of being used commercially and adds value to the areas of mobile cross-platform application development and mobile augmented reality.

9. References

ARchitect Engine. (2012). [Online]. Wikitude GmbH.
Available at: http://www.wikitude.com/developer/architect [Accessed: 8 September 2012].

ARchitect Features. (2012). [Online]. Wikitude GmbH.
Available at: http://www.wikitude.com/developer/architect/features [Accessed: 14 October 2012].

ARToolKit. [Online]. Human Interface Technology Laboratory University of Washington. Available at: http://www.hitl.washington.edu/artoolkit/projects/ [Accessed: 26 September 2012].

Azuma, R.T. (1997) 'A Survey of Augmented Reality', *Presence: Teleoperators and Virtual Environments*, Vol. 6, No 4, pp 355 – 385.

Brown, J. (2010). ZBar bar code reader. [Online]. SourceForge.
Available at: http://zbar.sourceforge.net/download.html [Accessed: 7 October 2012].

Catariov, A. (2011, May 19). Cross-platform solutions for Mobile App Development. [Online]. Available at:
http://www.google.lk/url?sa=t&rct=j&q=&esrc=s&source=web&cd=4&ved=0CDUQFjAD&url=http%3A%2F%2Fwww.idt.mdh.se%2Fkurser%2Fct3340%2Fht11%2FMINICONFERENCE%2FFinalPapers%2Fircse11_submission_16.pdf&ei=szN4UKrxHoGNrgfvtIG4CQ&usg=AFQjCNFv41UBOs2UIdkd8vXnJtJv9epN8w&cad=rja [Accessed: 12 October 2012].

Caudell, T. P. and Mizell, D. W. (1992). Augmented Reality: An Application of Heads-Up Display Technology to Manual Manufacturing Processes, Proceedings of 1992 IEEE Hawaii International Conference on Systems Sciences [Online]. *2*, pp 656 – 669. Available at:
http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=183317&isnumber=4717&punumber=378&k2dockey=183317@ieeecnfs [Accessed: 21 September 2012].

Chang, Y., Chu, C. and Chen, M. (2007). A General Scheme for Extracting QR Code from a Non-uniform Background in Camera Phones and Applications, *Proceedings of Ninth IEEE International Symposium on Multimedia (ISM 2007)* [Online], pp 123 – 130. Available at: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=4412365&contentType=Conference+Publications&queryText%3DQR+Code+Recognition [Accessed: 6 October 2012].

Cheok, A. D., Goh, K. H., Liu, W., Farbiz, F., Fong, S. W., Teo, S. L., li, Y. and Yang, X. (2004). Human Pacman: a mobile, wide-area entertainment system based on physical, social, and ubiquitous computing, *Proceedings of the 2nd Workshop on Network and System Support For Games (NetGames '03)* [Online], *8*:2, pp 71 – 81. Available at:
http://dl.acm.org/citation.cfm?id=992039.992041&CFID=170094111&CFTOKEN=50840343 [Accessed: 7 October 2012].

DTK Software Company. (2011). DTK Barcode Reader SDK 4.2. [Online]. DTK Software Company. Available at: http://www.dtksoft.com/barreader.php [Accessed: 7 October 2012].

Duttke, J. Home Page. [Online]. Available at: http://www.jsqr.de/ [Accessed: 7 October 2012].

Etienne, J. (2012). jquery.qrcode. [Online]. GitHub, Inc.
Available at: https://github.com/jeromeetienne/jquery-qrcode [Accessed: 7 October 2012].

Feiner, S., MacIntyre, B., Hollerer, T. and Webster, A. (1997). A Touring Machine: Prototyping 3D Mobile Augmented Reality Systems for Exploring the Urban Environment, *Proceedings of first IEEE International Symposium on Wearable Computers (ISWC '97)* [Online], pp 74 – 81.

Finneran, M. (2011). Mobile App Development Needs A New Approach. InformationWeek [Online], pp 1 – 4. Available at: http://www.informationweek.com/development/mobility/mobile-app-development-needs-a-new-appro/231300153?pgno=1 [Accessed: 15 August 2012].

Furth, B. (2011). *Handbook of Augmented Reality*. [Online]. Available at: http://books.google.lk/books?id=fG8JUdrScsYC&lpg=PP1&dq=Handbook+of+Augmented+Reality&pg=PA341&redir_esc=y#v=onepage&q&f=true [Accessed: 6 October 2012].

Google. (2012). Google Chart Tools: Infographics. [Online]. Google, Inc. Available at: https://developers.google.com/chart/infographics/docs/qr_codes [Accessed: 7 October 2012].

Gu, Y. and Zhang, W. (2011). QR code recognition based on image processing, *International Conference on Information Science and Technology (ICIST)* [Online], pp 733 – 736. Available at: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=5765349&contentType=Conference+Publications&queryText%3DQR+Code+Recognition [Accessed: 6 October 2012].

Henrysson, A., Billinghurst, M. and Ollila, M. (2005).Face to Face Collaborative AR on Mobile Phones, *Proceedings of the 4th IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR 05)* [Online], pp 80 – 89. <http://dl.acm.org/citation.cfm?id=1105185&CFID=114958659&CFTOKEN=52293392> [Accessed: 21 October 2012].

Jin, C., Yuan, J., Li, L., Chen, E., Han, G. and Tang, T. (2012). One of the image correction algorithms of QR code and its implementation based on wavelet transform, *IEEE 3rd International Conference on Software Engineering and Service Science (ICSESS)* [Online], pp 93 – 98. Available at: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6269413&contentType=Conference+Publications&pageNumber%3D2%26queryText%3DQR+Code+Recognition [Accessed: 6 October 2012].

Julier, S., Baillot, Y. Lanzagorta, M., Brown, D. and Rosenblum, L. (2000).BARS: Battlefield Augmented Reality System, *NATO Information Systems Technology Panel Symposium on New Information Processing Techniques for Military Systems* [Online]. Available at: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.1.5064 [Accessed: 21 September 2012].

Kato, H. and Billinghurst, M. (1999). Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System,*Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR99)* [Online], pp 85 – 94. Available at: http://www.computer.org/portal/web/csdl/doi?doc=abs/proceedings/iwar/1999/0359/00/03590085abs.htm [Accessed: 8 September 2012].

Layar. (2012). [Online]. Layar. Available at: http://www.layar.com/ [Accessed: 7 October 2012].

Liu, Y., Yang, J. and Liu M. (2008). Recognition of QR Code with mobile phones. *Chinese Control and Decision Conference (CCDC 2008)* [Online], pp 203 – 206. Available at: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=4597299&contentType=Conference+Publications&queryText%3DQR+Code+Recognition [Accessed: 6 October 2012].

Lukasavage, T. (2011, May 25). Review: PhoneGap is Web-based, Appcelerator is Pure Javascript. [Online]. SAVAGELOOK.COM. Available at: http://savagelook.com/blog/portfolio/phonegap-is-web-based-appcelerator-is-pure-javascript [Accessed: 12 October 2012].

Madden L. (2011, Sep. 8). Augmented Reality and NFC. The wave of the future?. [Online]. Augmented Planet. Available at: http://www.augmentedplanet.com/2011/09/augmented-reality-and-nfc-the-wave-of-the-future/ [Accessed: 7 October 2012].

Milgram, P. and Kishino, F. (1994).A Taxonomy of Mixed Reality Visual Displays, *IEICE Transactions on Information Systems* [Online],*E77-D*:12, n. pag. Available at: http://etclab.mie.utoronto.ca/people/paul_dir/IEICE94/ieice.html [Accessed: 8 September 2012].

Morrison, A., Oulasvirta, A., Peltonen, P., Lemmelä, S., Jacucci, G., Reitmayr, J., Näsänen, J. and Juustila, A. (2009). Like Bees Around the Hive: A Comparative Study of a Mobile Augmented Reality Map, *Proceedings of the 27th international conference on Human factors in computing systems (CHI 2009)* [Online], pp 1889 – 1898. Available at: http://mi.eng.cam.ac.uk/~gr281/docs/MorrisonCHI09.pdf [Accessed: 7 October 2012].

Newman, J., Ingram, D. and Hopper, A. (2001). Augmented Reality in a Wide Area Sentient Environment, *Proceedings of the 2nd IEEE and ACM International Symposium on Augmented Reality (ISAR 2001)* [Online], pp 77 – 86. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=970517 [Accessed: 21 September 2012].

Nokia Research Centre - MARA. (2012). [Online]. Nokia. Available at: http://research.nokia.com/page/219 [Accessed: 7 October 2012].

O'Dell, J. (2009). Mobilizy Proposes Open, Cross-Platform Markup Language for Augmented Reality. [Online]. SAY Media, Inc.
Available at: http://www.readwriteweb.com/archives/arml.php [Accessed: 8 September 2012].

Owen, C.B., Xiao, F. and Middlin, P. (2002). What is the best fiducial?, *The First IEEE International Workshop - Augmented Reality Toolkit* [Online]. Available at: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=1107021&contentType=Conference+Publications&searchField%3DSearch_All%26queryText%3DWhat+is+the+best+fiducial%3F [Accessed: 24 September 2012].

Piekarski, W. and Thomas, B.H. (2002) *The Tinmith System – Demonstrating New Techniques for Mobile Augmented Reality Modelling*.[Online]. Available at: http://www.tinmith.net/papers/piekarski-auic-2002.pdf [Accessed: 21 September 2012].

QrCode.Net. (2012). [Online]. Microsoft, Inc. Available at: http://qrcodenet.codeplex.com/ [Accessed: 7 October 2012].

Qualcomm. (2012). Vuforia. [Online]. Qualcomm Inc. Available at: http://www.qualcomm.com/solutions/augmented-reality [Accessed: 13 October 2012]

Raskar, R., van Baar, J., Beardsley, P., Willwacher, T., Rao, S. and Forlines, C. (2005). iLamps: geometrically aware and self-configuring projectors, *Proceedings of SIGGRAPH '05 ACM SIGGRAPH 2005 Courses* [Online]. Available at: http://dl.acm.org/citation.cfm?id=1198555.1198715&coll=DL&dl=GUIDE&CFID=170094111&CFTOKEN=50840343 [Accessed: 7 October 2012].

Rekimoto, J. (1996).Augmented Reality Using the 2D Matrix Code, *In Proceedings of the Workshop on Interactive Systems and Software (WISS'96)*. Available at: http://www.sonycsl.co.jp/person/rekimoto/papers/wiss96.pdf [Accessed: 23 September 2012].

Rekimoto, J. (1998). Matrix: a real-time object identification and registration method for augmented reality, *In Proceedings of the 3rd Asia Pacific Computer Human Interaction* [Online], pp 63 – 68. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=704151 [Accessed: 23 September 2012].

Rohs, M. and Gfeller, B. (2004).Using Camera-Equipped Mobile Phones for Interacting with Real-World Objects, Advances in Pervasive Computing, pp 265 – 271. Available at: http://www.vs.inf.ethz.ch/res/papers/rohs-gfeller-visualcodes-2004.pdf [Accessed: 21 September 2012].

Shapiro, G. (2012, Feb. 10). Native or Cross Platform Mobile Development?. [Online]. Available at: http://www.garethshapiro.com/item/native-or-cross-platform-mobile-development/?page=3 [Accessed: 10 October 2012].

Sutherland, E. (1968). 'A Head-Mounted Three Dimensional Display', Proceedings of Fall Joint Computer Conference [Online], pp 757 – 764. Available at: http://dl.acm.org/citation.cfm?id=1476686 [Accessed: 21 September 2012].

Thomas, B., Close, B., Donoghue, J., Squires, J., De Bondi, P., Morris, M. and Piekarski, W. (2000).ARQuake: An Outdoor/Indoor Augmented Reality First Person Application, *Proceedings of the 4th International Symposium on Wearable Computers* [Online], pp 139 – 146. Available at: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.19.6863 [Accessed: 21 September 2012].

Veronica. (2011, May 17). Wikitude revolutionizes the creation of Augmented Reality content with new "ARchitect" platform. [Online]. Wikitude GmbH. Available at: http://www.wikitude.com/enwikitude-reveals-architect-developer-kit [Accessed: 8 September 2012].

Vlahakis, V., Karigiannis, J., Tsotros, M., Gounaris, M., Almeida, L., Stricker, D., Gleue, T., Christou, I., Carlucci, R. and Ioannidis, N. (2001). ARCHEOGUIDE: First results of an Augmented Reality, Mobile Computing System in Cultural Heritage Sites, *Mobile Computing System in Cultural Heritage Sites", Proceedings of Virtual Reality, Archaeology, and Cultural Heritage International Symposium (VAST01)* [Online], pp28 – 30. Available at:

http://archeoguide.intranet.gr/papers/publications/ARCHEOGUIDE-VAST01-1.pdf [Accessed: 8 September 2012].

Wagner, D. and Schmalstieg, D. (2003). First Steps Towards Handheld Augmented Reality, *Proceedings of the 7th IEEE International Symposium on Wearable Computers (ISWC 03)* [Online], pp 127 – 135. Available at: http://www.ims.tuwien.ac.at/media/documents/publications/HandheldAR_ISWC03final.pdf [Accessed: 21 September 2012].

Wagner, D. and Schmalstieg, D. (2007). ARToolKitPlus for Pose Tracking on Mobile Devices, *Proceedings of 12th Computer Vision Winter Workshop (CVWW'07)* [Online]. Available at: http://www.icg.tu-graz.ac.at/Members/daniel/Publications/ARToolKitPlus [Accessed: 30 September 2012].

Wagner, D., Schmalstieg, D., Pintaric, T. and Ledermann, F. (2005). Towards MassivelyMulti-User Augmented Realityon Handheld Devices, *Proceedings of the Third International Conference on Pervasive Computing (Pervasive 2005)* [Online]. Available at: http://studierstube.icg.tugraz.at/invisible_train/documents/wagner05_pervasive.pdf [Accessed: 21 September 2012].

Wagner, D. (2007). *Handheld Augmented Reality*.[Online]. Available at: http://studierstube.icg.tugraz.at/thesis/Wagner_PhDthesis_final.pdf [Accessed: 19 September 2012].

Wang, J., Shyi, C., Hou, T.W.and Fong, C.P. (2010). Design and implementation of augmented reality system collaborating with QR code, *2010 International Computer Symposium (ICS)* [Online], pp 414 – 418. Available at: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=5685477&contentType=Conference+Publications&queryText%3DQR+code+and+AR [Accessed: 25 September 2012].

Warren, C. (2012). The pros and cons of cross-platform app design. *Mashable Tech* [Online], pp 1. Available at: http://mashable.com/2012/02/16/cross-platform-app-design-pros-cons/ [Accessed: 8 September 2012].

Wikitude. (2012). [Online]. Wikitude GmbH. Available at: http://www.wikitude.com/?lang=en [Accessed: 7 October 2012].

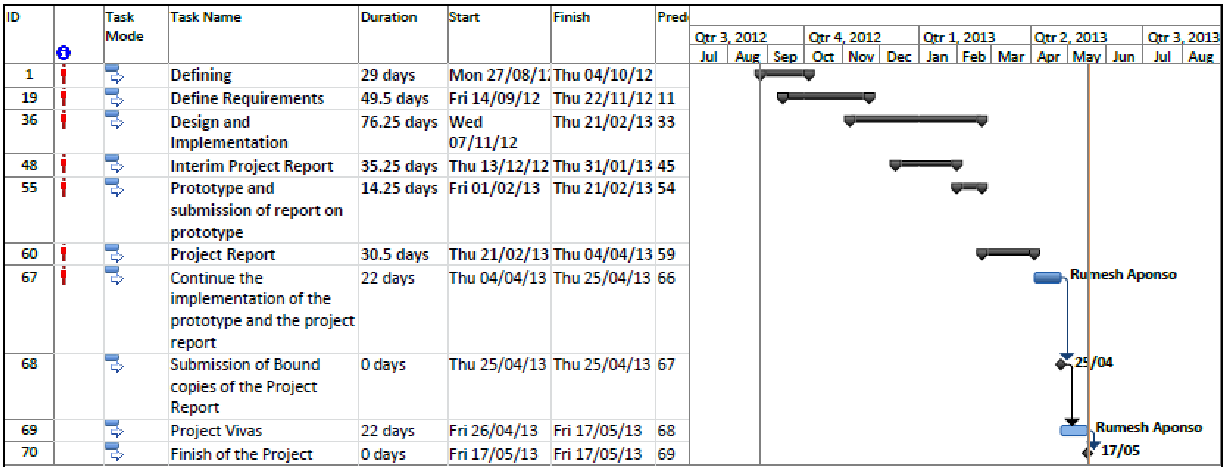
Wray, F. (2012). A Brief Future of Computing. *PlanetHPC of the University of Edinburgh* [Online]. Available at: <http://www.planethpc.eu/index.php?option=com_content&view=article&id=66:a-brief-future-of-computing&catid=1:articles&Itemid=3 [Accessed: 21 September 2012].

Zxing. (2012). [Online]. Google, Inc. Available at: http://code.google.com/p/zxing/ [Accessed: 7 October 2012].

10.Appendices

I. Project Plan

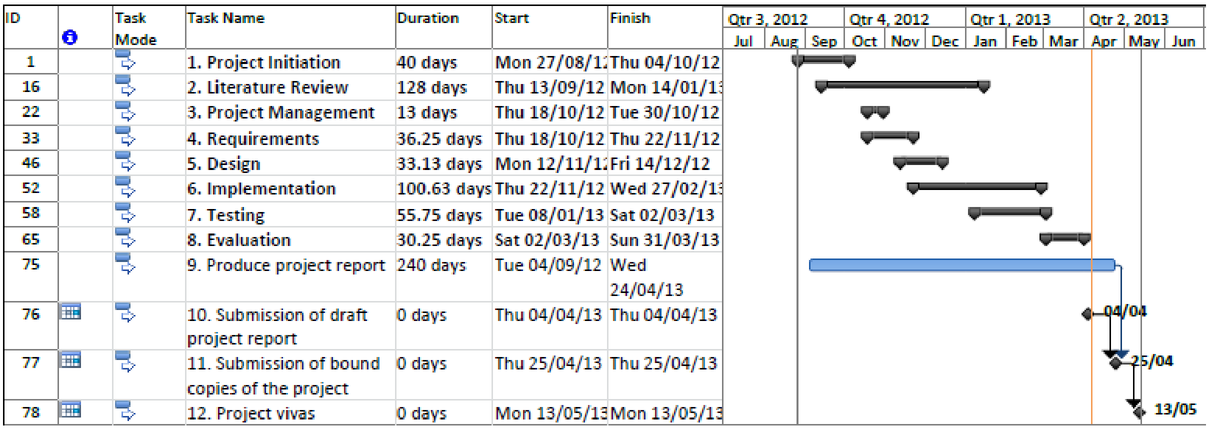
A. Tentative Project Plan



ID		Task Mode	Task Name	Duration	Start	Finish	Pred
1			Defining	29 days	Mon 27/08/1	Thu 04/10/12	
19			Define Requirements	49.5 days	Fri 14/09/12	Thu 22/11/12	11
36			Design and Implementation	76.25 days	Wed 07/11/12	Thu 21/02/13	33
48			Interim Project Report	35.25 days	Thu 13/12/12	Thu 31/01/13	45
55			Prototype and submission of report on prototype	14.25 days	Fri 01/02/13	Thu 21/02/13	54
60			Project Report	30.5 days	Thu 21/02/13	Thu 04/04/13	59
67			Continue the implementation of the prototype and the project report	22 days	Thu 04/04/13	Thu 25/04/13	66
68			Submission of Bound copies of the Project Report	0 days	Thu 25/04/13	Thu 25/04/13	67
69			Project Vivas	22 days	Fri 26/04/13	Fri 17/05/13	68
70			Finish of the Project	0 days	Fri 17/05/13	Fri 17/05/13	69

Figure 10.1 - Tentative Project Plan

B. Refined Project Plan



ID		Task Mode	Task Name	Duration	Start	Finish
1			1. Project Initiation	40 days	Mon 27/08/1	Thu 04/10/12
16			2. Literature Review	128 days	Thu 13/09/12	Mon 14/01/1
22			3. Project Management	13 days	Thu 18/10/12	Tue 30/10/12
33			4. Requirements	36.25 days	Thu 18/10/12	Thu 22/11/12
46			5. Design	33.13 days	Mon 12/11/1	Fri 14/12/12
52			6. Implementation	100.63 days	Thu 22/11/12	Wed 27/02/1
58			7. Testing	55.75 days	Tue 08/01/13	Sat 02/03/13
65			8. Evaluation	30.25 days	Sat 02/03/13	Sun 31/03/13
75			9. Produce project report	240 days	Tue 04/09/12	Wed 24/04/13
76			10. Submission of draft project report	0 days	Thu 04/04/13	Thu 04/04/13
77			11. Submission of bound copies of the project	0 days	Thu 25/04/13	Thu 25/04/13
78			12. Project vivas	0 days	Mon 13/05/1	Mon 13/05/1

Figure 10.2- Finalise Project Plan

II. Risk Log

Basic Risk Information					Risk Assessment Information			Risk Information		
Risk No	Description	Responsible	Reported Date	Last Date	Impact	Probability	Timeline Effect	Causing Action	Future Action	Project Impact
R01	Failure to manage changes. e.g. Functionality changes, technological factors	Research Student	2012-09-03	2012-12-08	Medium	Medium	Medium	Failure to manage and control change	Well manage changes	Unsuccessful project

R02	Incomplete requirements	Research Student	2012-10-15	2013-01-26	High	Medium	High	Requirements have not been well recorded while doing the project	Well recorded requirements	Project with Incomplete Requirements
R03	System breakdown e.g. Laptop broken or testing mobile phone broken	Research Student	2013-03-21	2013-04-02	High	Medium	Medium	Hardware failures or operating system failures	Backup PC with running OS	Unsuccessful project
R04	Data loss	Research Student	2013-01-21	2013-03-30	High	Medium	High	No sufficient backup data	Well managed backup data to minimize data losses	Unsuccessful project
R05	Overruns of project schedule	Research Student	2012-10-04	2012-12-20	High	High	High	Failure to meet project milestones and inaccurate project scheduling	Well monitoring and control over the project and well defined project schedule	Incomplete project
R06	Unfamiliar libraries and software	Research Student	2013-08-12	2013-03-01	Medium	Medium	Medium	Introducing new software to the project	Use suitable familiar software for the project	Unsuccessful project

Table 10.1 - Risk Log

III. The Main Parameters Used For the Survey

A. Sample Size

In a survey the number of respondents is a key factor to achieve a reliable result. A hundred respondents would have been much reliable. However due to the time constraint, a considerable size of fifty have been chosen.

B. Reliability

This survey is mainly about mobile augmented reality application development, thus the respondent should be aware of basic domain knowledge of respective areas. Otherwise this could affect the results of the survey, thus the respondents were chosen mainly in software companies. Also, the educational level and experiences of the professionals have been considered.

C. Designing of the Questionnaire

Most of the professionals are busy with work, thus the questions of the survey aimed to extract the requirements and their opinions within a short amount of time. The questionnaire that was designed had a stable flow, where the questions were about different development techniques, problems they face with existing tools and features which they would expect from a new tool such as ARmax. Most of the questions are multiple choice questions, where some require rating.

D. Conducting the Survey

A questionnaire was designed to be distributed among professionals within a number of recognised software companies in Sri Lanka. Thus, the questionnaire was built using an online survey tool (Google Docs) and the survey link was distributed among professionals. The respondents' feedbacks were recorded in the particular survey tool.

E. Analysis of the Results

The results collected from the respondents were organised through the online survey tool. Then plotted graphs were drawn using the quantitative data that was collected from the survey tool, for easy comprehension and to provide visual representation to the reader.

IV. Results of the Survey

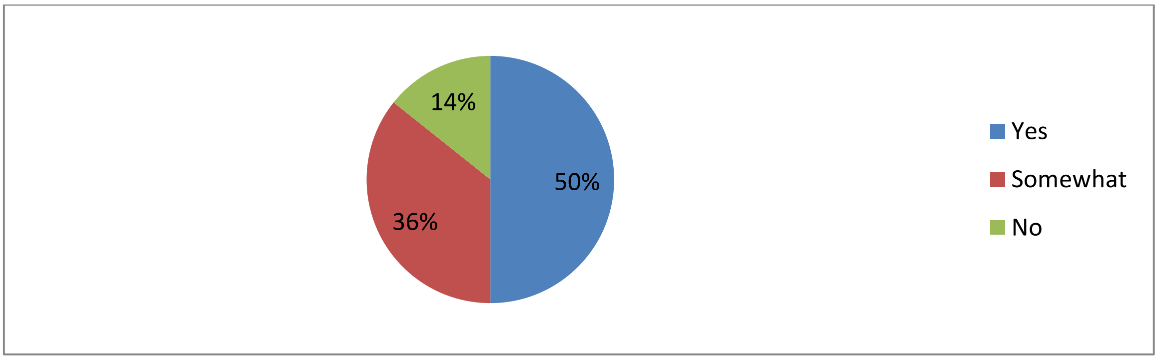


Chart 10.1 - Familiarity among Mobile Application Development

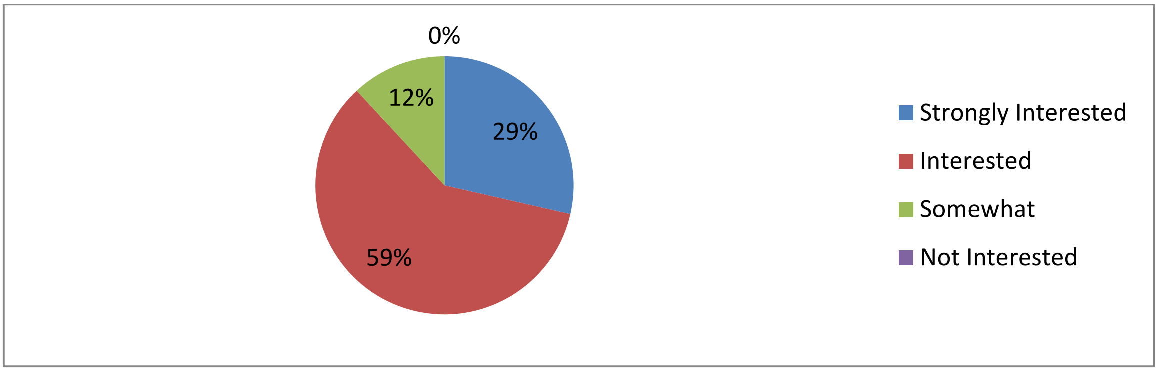


Chart 10.2 - Interest in Mobile Application Development

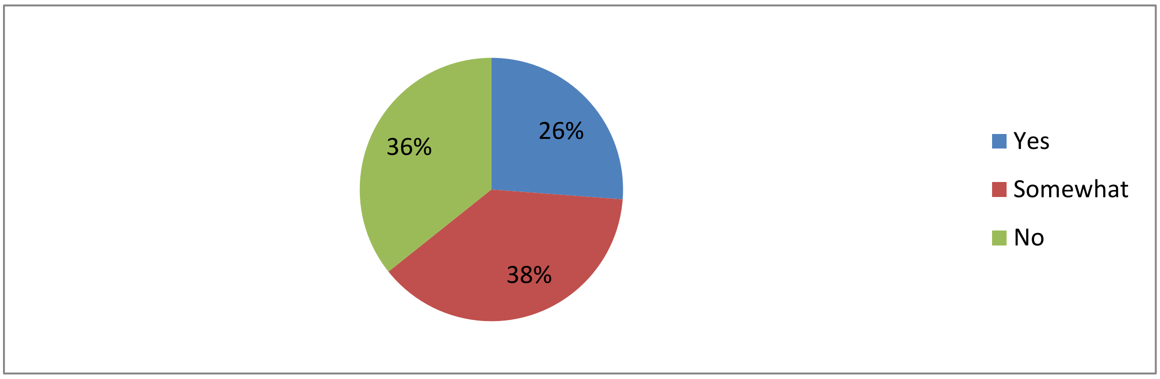


Chart 10.3 - Familiarity among Mobile Cross-Platform Application Development

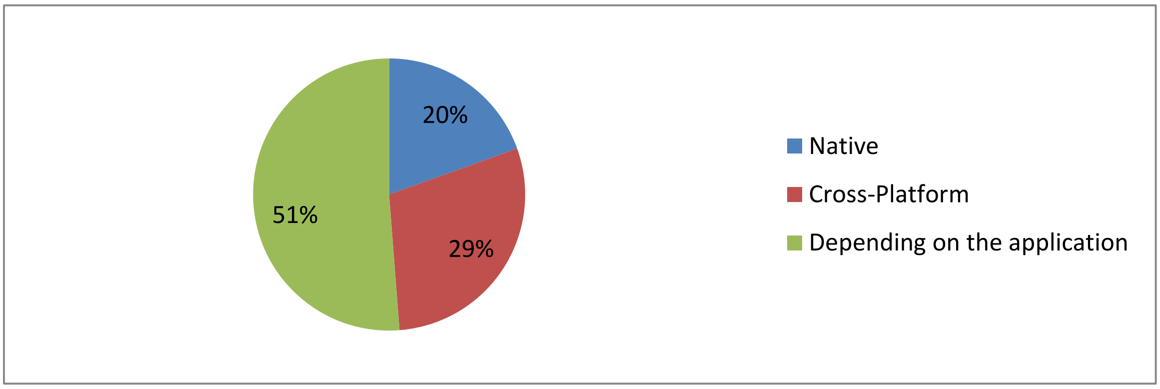


Chart 10.4 - If you were asked to develop same mobile application for different platforms, would you choose native approach or cross-platform approach or approach will be chosen depending on the application?

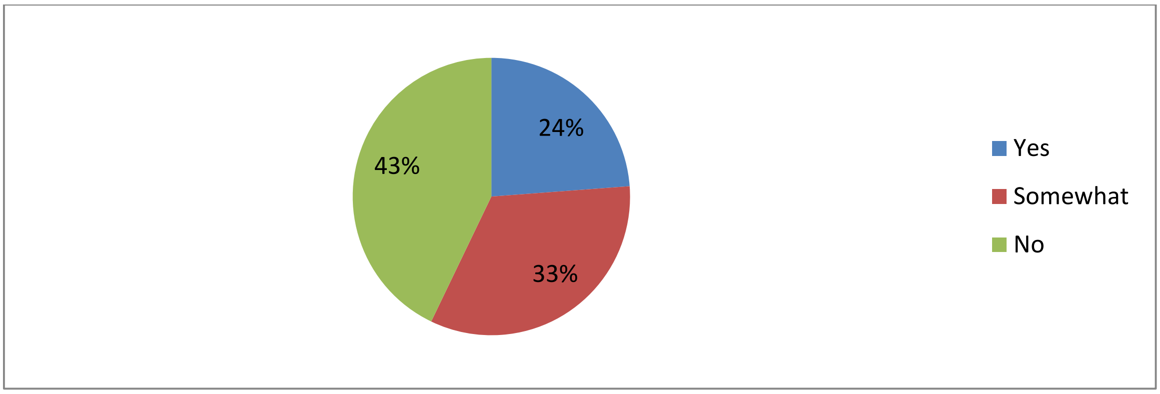


Chart 10.5 - Familiarity among Mobile Augmented Reality Applications

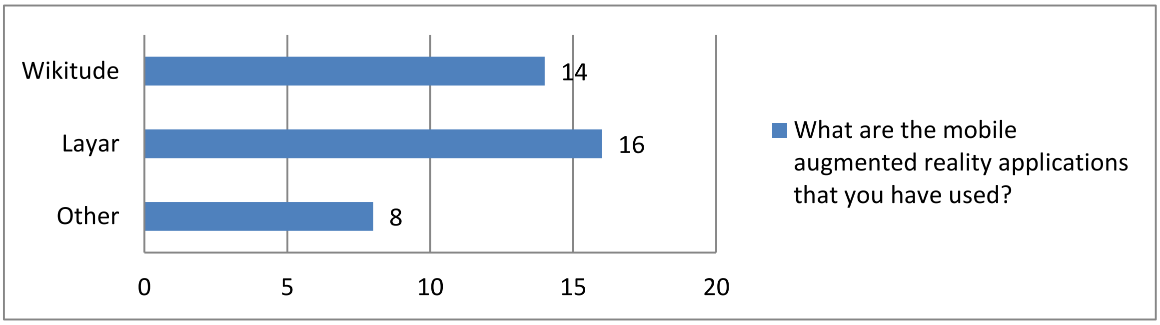


Chart 10.6 - Mobile Augmented Reality Applications which Often Uses

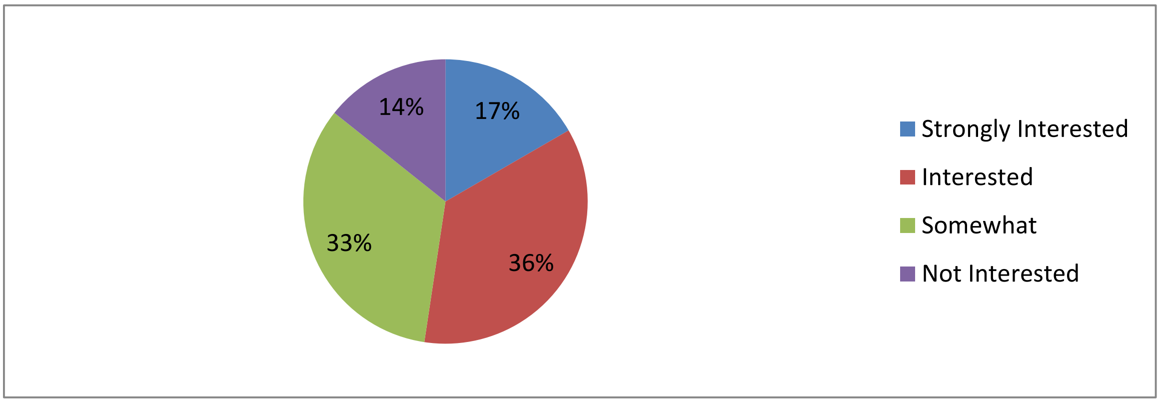


Chart 10.7 - Interest in Mobile Augmented Reality Application Development

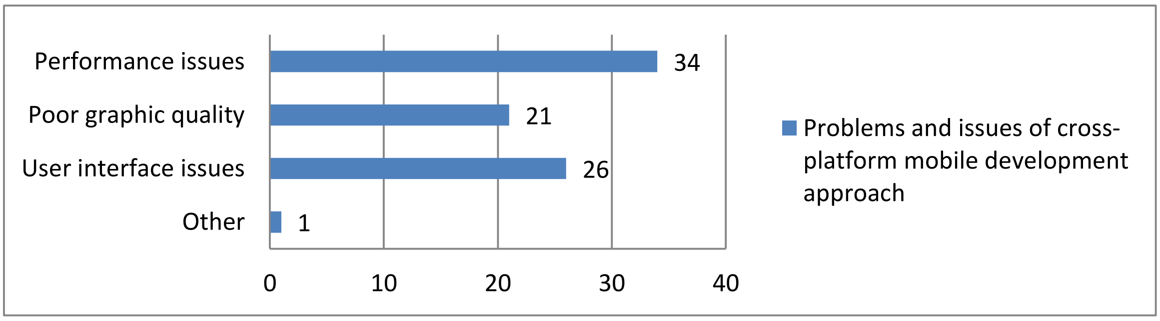


Chart 10.8 - Problems and Issues of Cross-Platform Mobile Development Approach

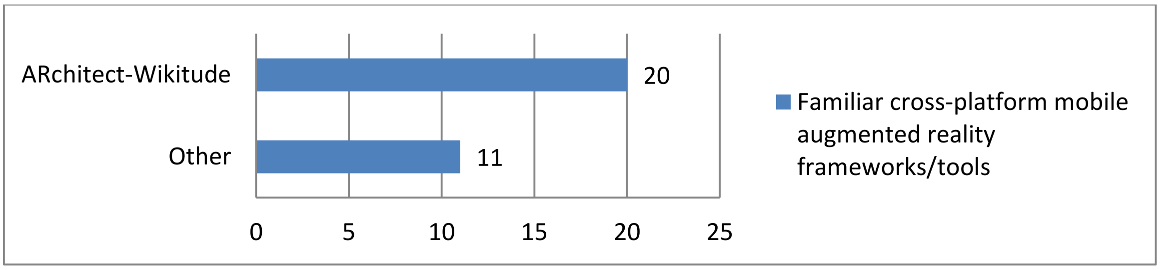


Chart 10.9 - Familiar cross-platform mobile augmented reality frameworks/tools

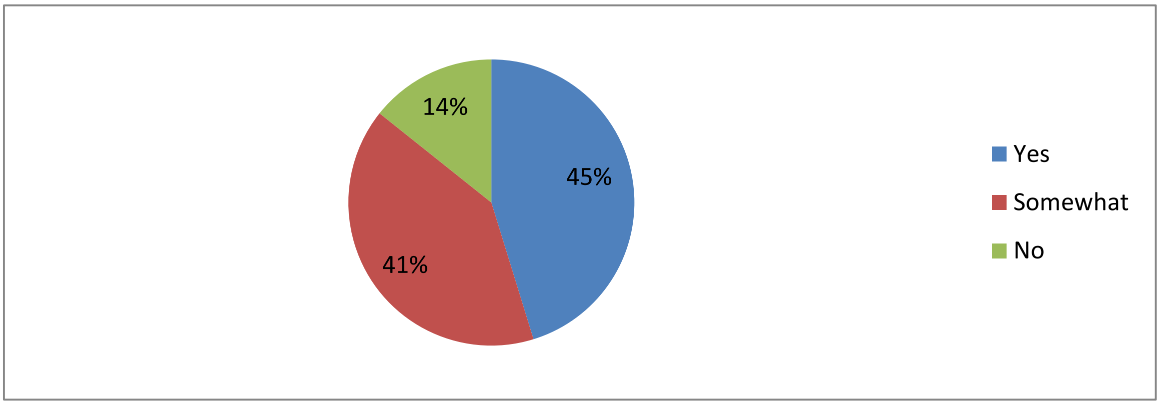


Chart 10.10 - Familiarity with NFC Technology

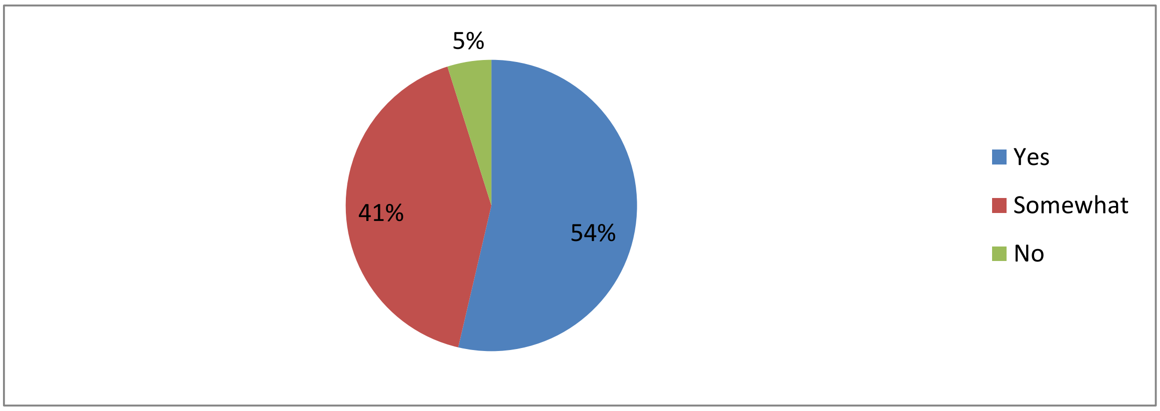


Chart 10.11 - Do you think combining NFC technology with augmented reality applications may improve the user interaction/experience?

Features of Proposed AR Development Kit (Framework)

The respondents were asked to rate the features that they would require from a cross-platform AR development kit as proposed. The results are shown in Figure 19.

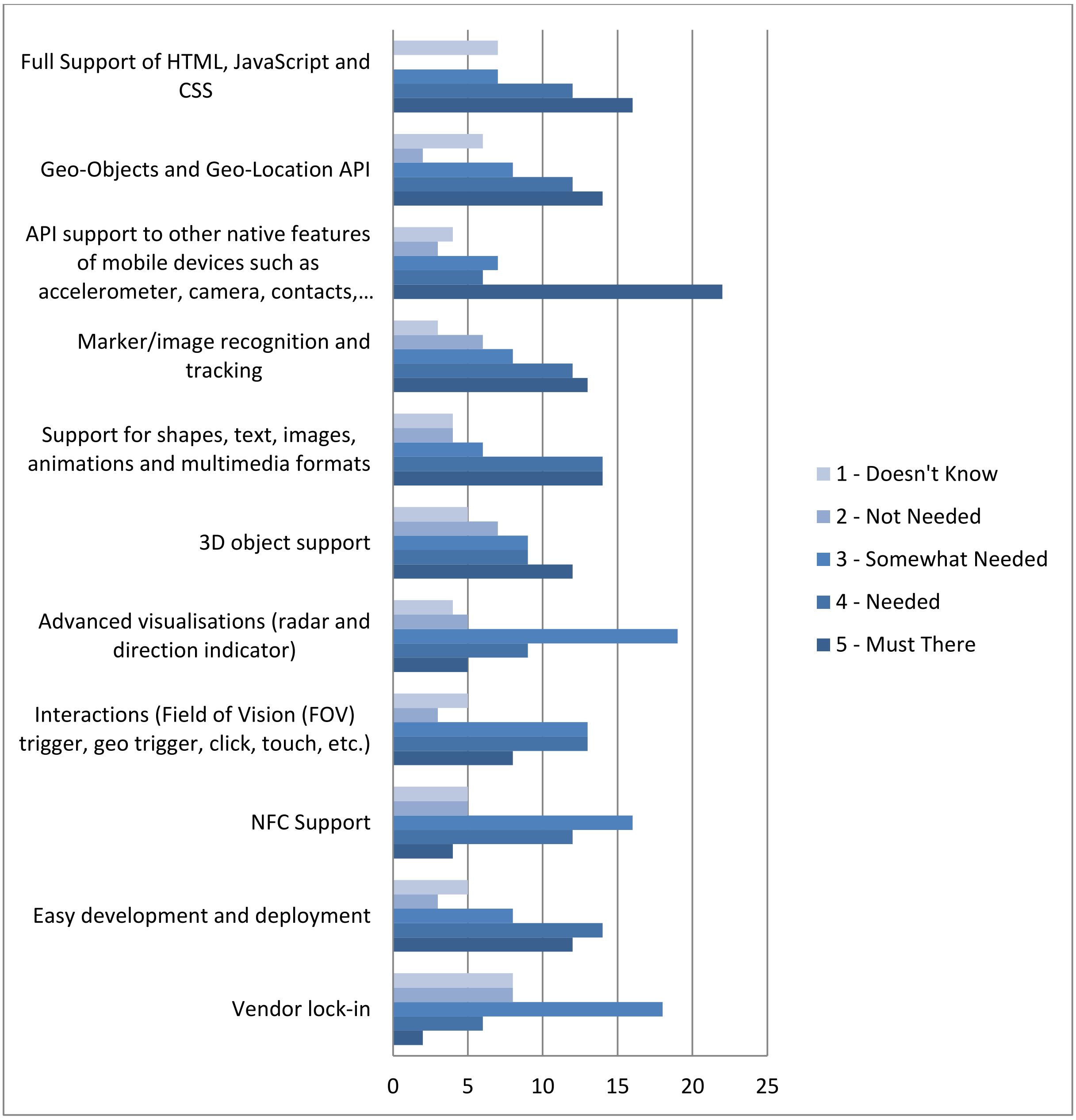


Chart 10.12 - Features of Proposed ARmax Framework

V. Use Case Specifications

Use Case ID:	UC-1	**Use Case Name:**	Test Features
Description:	This allows the user to test the features of ARmax framework.		
Actor(s):	• End User		
Preconditions:	• Mobile application should be operational.		
Flow of Events			
Basic Flow:	1. User goes to respective features view. 2. Test features.		
Alternative Flow(s):	1. In step 2, user can test compass. 2. In step 2, user can test triggers.		

	3. In step 2, user can test performance.
Exception(s):	1. In step 2, if the application crashes, then the user will be notified with an appropriate error message.
Post Conditions:	• Successfully test features.

Table 10.2 - Use Case Specification of Test Features

Use Case ID:	UC-2	**Use Case Name:**	Test Compass
Description:	User is provided with the ability to interact with different advance visualisation objects such as compass, direction indicator, animations, etc. in the AR view.		
Actor(s):	• End User		
Preconditions:	• Mobile application should be operational.		
Flow of Events			
Basic Flow:	1. User goes to respective feature view. 2. Interact with different advance visualisation objects.		
Alternative Flow(s):	---		
Exception(s):	1. In step 2, if the application crashes, then the user will be notified with an appropriate error message.		
Post Conditions:	• Successfully test visualisation features.		

Table 10.3 - Use Case Specification of Test Compass

Use Case ID:	UC-3	**Use Case Name:**	Test Triggers
Description:	User is provided with the ability to interact with different triggering methods which the framework supports.		
Actor(s):	• End User		
Preconditions:	• Mobile application should be operational.		
Flow of Events			
Basic Flow:	1. User goes to respective feature view. 2. Interact with the application with different interaction methods such as touch, geo-trigger, etc.		
Alternative Flow(s):	---		
Exception(s):	1. In step 2, if the application crashes, then the user will be notified with an appropriate error message.		
Post Conditions:	• Successfully test triggering and interaction features.		

Table 10.4 - Use Case Specification of Test Triggers

Use Case ID:	UC-4	**Use Case Name:**	Test Performance
Description:	User can test the application's performance in the respective mobile device.		
Actor(s):	• End User		

Preconditions:	• Mobile application should be operational.
Flow of Events	
Basic Flow:	1. User goes to respective feature view. 2. User presses the test performance image button. 3. The application responds by notifying the user by a message, that performance test has been initiated. 4. The application responds by displaying the statistics of the performance test to the user.
Alternative Flow(s):	---
Exception(s):	1. In step 2, if the application crashes during performance test, then the user will be notified with an appropriate error message.
Post Conditions:	• Successfully test application's performance.

Table 10.5 - Use Case Specification of Test Performance

Use Case ID:	UC-6	**Use Case Name:**	Scan QR Code
Description:	User is provided with the ability to scan QR codes.		
Actor(s):	• End User		
Preconditions:	• Mobile application should be operational.		
Flow of Events			
Basic Flow:	1. User points the device's camera view to a QR code. 2. The application will capture the QR code by auto focusing. 3. The application will respond by displaying an information pop-up message to the user including the information of the QR code.		
Alternative Flow(s):	---		
Exception(s):	1. In step 2, if the application crashes during capturing QR code, then the user will be notified with an appropriate error message.		
Post Conditions:	• Successfully scans a QR code.		

Table 10.6 - Use Case Specification of Scan QR Code

Use Case ID:	UC-7	**Use Case Name:**	Scan NFC Tag
Description:	User is provided with the ability to scan NFC tags.		
Actor(s):	• End User		
Preconditions:	• Mobile application should be operational.		
Flow of Events			
Basic Flow:	1. User closes the device to NFC tag. 2. The application will read the NFC tag. 3. The application will respond by displaying an information pop-up message to the user including the information of the NFC tag.		
Alternative Flow(s):	---		

Exception(s):	1. In step 2, if the application crashes during reading NFC tag, then the user will be notified with an appropriate error message.
Post Conditions:	• Successfully reads a NFC tag.

Table 10.7 - Use Case Specification of Scan NFC Tag

Use Case ID:	UC-8	**Use Case Name:**	Navigate To Place
Description:	User can navigate to a place with the help of the guided direction indicator in as AR view. Also user has the ability to switch into map view.		
Actor(s):	• End User		
Preconditions:	• Mobile application should be operational. • Mobile device should have internet connectivity. • Mobile device's GPS should be operational.		
Flow of Events			
Basic Flow:	1. User presses a particular place in the view places nearby view (i.e. Use Case: View Places Nearby). 2. Then user presses navigate button in the place's information pop-up message. 3. The application responds by guiding the user by showing direction indicator and compass. 4. When the user reaches the destination, the application will respond by displaying information messages that the destination has been reached.		
Alternative Flow(s):	1. In step 3, user can switch to the map view.		
Exception(s):	1. In step 2, if the application crashes, then the user will be notified with an appropriate error message. 2. In step 3, if the direction indicator and compass load fails, then the user will be notified with an appropriate error message. 3. In step 3, if the user is deviating from the destination, then the user will be notified with an appropriate information message. 4. In step 4, if the application crashes, then the user will be notified with an appropriate error message.		
Post Conditions:	• User successfully reaches the destination.		

Table 10.8 - Use Case Specification of Navigate To Place

Use Case ID:	UC-9	**Use Case Name:**	Set A Place
Description:	User can store the current position of him/her in the application.		
Actor(s):	• End User		
Preconditions:	• Mobile application should be operational. • Mobile device should have internet connectivity. • Mobile device's GPS should be operational.		
Flow of Events			
Basic Flow:	1. User presses store current location image button. 2. The application will fetch the location information through device GPS and asks for the user's confirmation to store. 3. Then user presses yes and fills the location information form. 4. The application responds by validating the user entered information and stores the location information into the application.		

	5. The application will notify the user that the location's information has been successfully stored through a successful message.
Alternative Flow(s):	---
Exception(s):	1. In step 2, if the application fails to retrieve the location's information, then the user will be notified with an appropriate error message. 2. In step 4, if the form validation fails, then the user will be notified with an appropriate error message. 3. In step 4, the location information storing fails, then the user will be notified with an appropriate error message.
Post Conditions:	• User successfully stores the location information into the system.

Table 10.9 - Use Case Specification of Set A Place

VI. Mock UIs

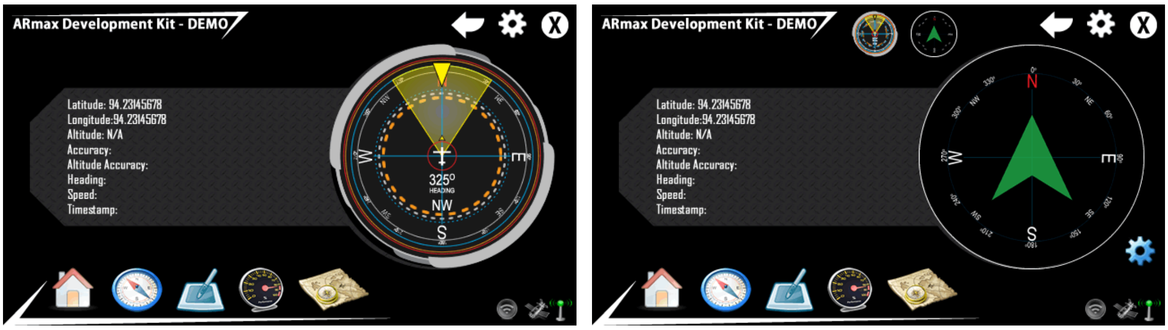


Figure 10.3 - ARmax Prototype - Test Compass and Direction Indicator

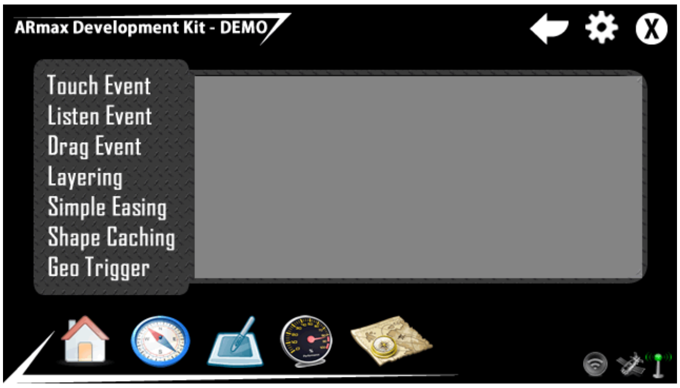


Figure 10.4 - ARmax Prototype - Triggers Screen

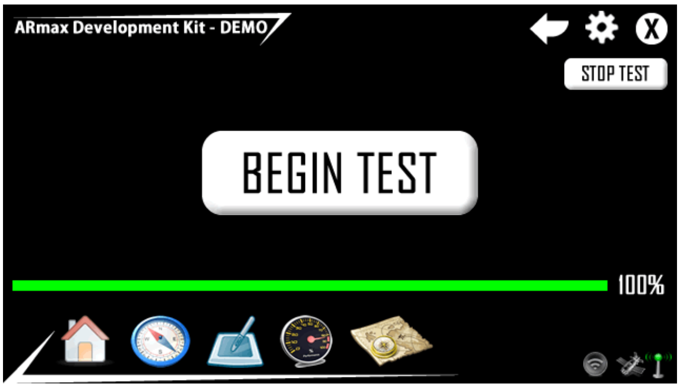


Figure 10.5 - ARmax Prototype - Performance Test Screen

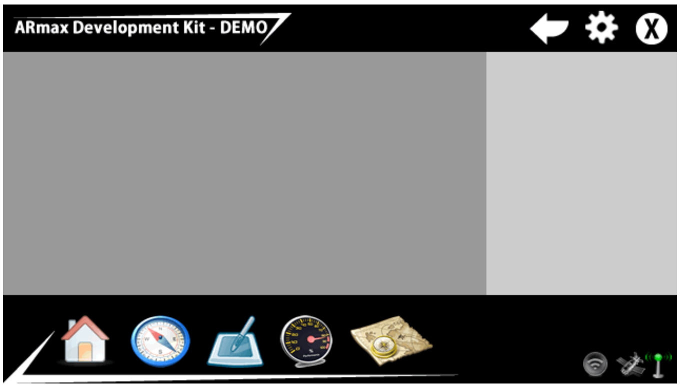


Figure 10.6 - ARmax Prototype - Map Screen

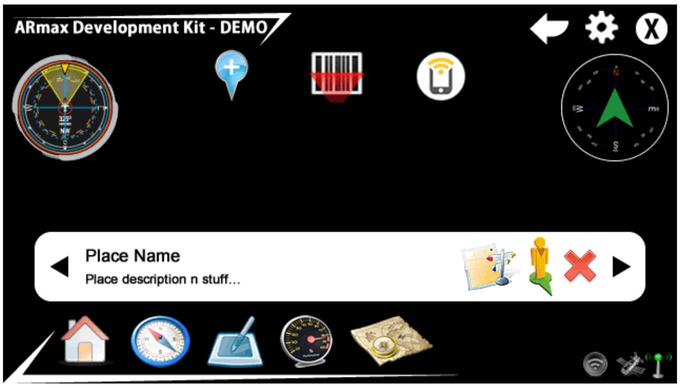


Figure 10.7 - ARmax Prototype - Places Nearby Screen

VII. Mobile Browser Support Benchmark Results

Table 10.10 show the custom benchmarking result that has been carried out in a number of mobile phones.

Browser	Tested Device	Score	Bonus
Android 4.1.2	Samsung Galaxy SIII	434	3
Android 2.3.5	HTC Desire HD	200	1
Android 2.3.6	Samsung Galaxy Y	200	1

Chrome Nightly 26.0.1410.58	LG Nexus 4	415	11
Safari	iOS Simulator - iPhone (Retina 4-inch) / iOS 6.1	386	9

Table 10.10 - Custom benchmarking results

Table 10.11 shows the detailed comparison information of HTML5 feature support on three different mobile browsers.

Feature	**Android 4.0**	**iOS 6.0**	**BlackBerry 10**
Parsing rules	10	10	10
Canvas	20	20	20
Video	21	21	30
Audio	20	20	20
Elements	28	29	30
Forms	66	102	115
User interaction	20	20	20
History and navigation	0	10	10
Microdata	0	0	15
Web applications	15	15	18
Security	10	15	20
Various	5	5	10
Related Specifications			
Location and Orientation	20	20	20
WebGL	9	10	25
Communication	13	33	35
Files	10	10	10
Storage	25	15	25
Workers	0	15	15
Local multimedia	0	0	10
Notifications	0	0	10
Other	5	8	10
Experimental			
Audio	0	5	0
Video and Animation	0	3	7

Table 10.11 - Comparison between mobile browsers by The HTML5 Test (Accessed Date: 2013-04-13)

VIII. Resource Requirements

A. Tentative Resource Requirements

Resource	Hardware Requirements	Software Requirements
PC/Laptop	− Intel Core i5/i7 Processor − Minimum 2GB RAM (4GB or recommended)	− Windows 7/8 OS − Eclipse 3.4 or higher − Android SDK − ADT Plugin − Adobe Dreamweaver − Adobe Fireworks − Browser (Google Chrome / Firefox / Internet Explorer) − Microsoft Office
Mobile Phone (Android)	− 3G Network: HSDPA 900 / 2100 − Internal Memory: 512 MB ROM, 512 MB RAM or higher − Primary Camera: 5 MP or higher with autofocus − CPU: 600 MHz ARM 11 or higher − GPU: Adreno 200 or higher − Sensors: Accelerometer, compass − GPS with A-GPS support − NFC	− OS Android OS, v2.3 (Gingerbread) or higher − Browser with HTML/HTML5 support
Mobile Phone (iOS)	− 3G Network: HSDPA 850 / 900 / 1900 / 2100 − Internal Memory: 512 MB RAM or higher − Primary Camera: 5 MP or higher with autofocus − CPU: 1 GHz Cortex-A8 or higher − GPU: PowerVR SGX535 or higher − Sensors: Accelerometer, compass − Geo-tagging, touch focus	− iOS 4 higher − Browser with HTML/HTML5 support

Table 10.12 - Tentative Resource Requirements

Note: Above requirements are subject to change.

B. Refined Resource Requirements

Resource	Hardware Requirements	Software Requirements	Source
PC/Laptop (Windows)	− Intel Core i5/i7 Processor − Minimum 2GB RAM (4GB or recommended) − 1366x768 minimum resolution; graphics hardware acceleration requires a DirectX 10 graphics card.	− Windows 7/8 OS − Adobe Dreamweaver CS6 − Adobe Fireworks CS6 − Browser (Google Chrome / Firefox / Internet Explorer) − Microsoft Office − PhoneGap 2.4.0 − Eclipse + ADT plugin − Android SDK Tools − Android Platform-tools − The latest Android platform − The latest Android system image for the emulator	− Adobe Dreamweaver CS6 Official Website − Adobe Fireworks CS6 Official Website − Microsoft Office Official Website − Android Developers Website − PhoneGap API Documentation

PC/Laptop (Apple)	– Intel-based Computer with Mac® OS X® Lion or greater (10.7.4+)	– Xcode 4.5 and iOS 6 SDK – Xcode Command Line Tools – Apple Developer Account	– PhoneGap API Documentation
Mobile Phone (Android)	– 3G Network: HSDPA 900 / 2100 – Internal Memory: 512 MB RAM or higher – Primary Camera: 5 MP or higher with autofocus – CPU: 600 MHz ARM 11 or higher – GPU: Adreno 200 or higher – Sensors: Accelerometer, compass – GPS with A-GPS support – NFC	– OS Android OS, v2.3.3 (Gingerbread) or higher – Browser with HTML/HTML5 support	– PhoneGap API Documentation – Samsung Galaxy S III Official Website
Mobile Phone (iOS)	– 3G Network: HSDPA 850 / 900 / 1900 / 2100 – Internal Memory: 512 MB RAM or higher – Primary Camera: 5 MP or higher with autofocus – CPU: 1 GHz Cortex-A8 or higher – GPU: PowerVR SGX535 or higher – Sensors: Accelerometer, compass	– iOS 5.x or higher – Browser with HTML/HTML5 support	– PhoneGap API Documentation – Apple iPhone 4 Official Website

Table 10.13 - Finalised resource requirements of ARmax

IX. Bug Report

ID	Bug Name	Build Number	Severity	Priority	Reported Date	Reason	Status	Environment
1	Back button didn't work the first time	ARmax v1.1	Low	Low	2013/02/25	Defect	Active	Android 4.1.2
2	Compass and direction indicator didn't appear	ARmax v1.1	Low	Low	2013/02/26	Defect	Active	Android 4.1.2
3	Application stopped unexpectedly at the start	ARmax v1.1	High	Medium	2013/03/26	Defect	New	Android 4.1.2
4	Barcode utility stopped unexpectedly at the start	ARmax v1.1	Medium	Medium	2013/03/29	Defect	New	Android 4.1.2
5	Nearby places didn't appear on the mini compass	ARmax v1.1	High	High	2013/04/2	Defect	New	Android 4.1.2
6	POIs getting duplicated, the canvas not getting refreshed properly	ARmax v1.1	High	High	2013/04/10	Defect	New	Android 4.1.2

Table 10.14 - Bug Report

X. Deliverable and Milestone Summary

Deliverable	Milestone	Planned Submission Date	Actual Submission Date
Terms of Reference	1.2.3.2. Submission of final TOR	2012/10/04	2012/10/04
Literature Review	2.5. Submission of literature review	2012/10/18	2012/10/18
Software Requirement Specification	4.5.2. Submission of SRS	2012/11/22	2012/12/05
Software Design Specification	5.5. Finish design specification	2012/12/14	2013/01/23
Interim Project Report	6.5. Submission of interim project report	2013/01/31	2013/01/31
Test Plan, Test Results, Prototype and Report on Prototype	7.6. Demonstration of prototype	2013/02/21	2013/02/21
Final Project Report	11. Submission of bound copies of the project	2013/04/25	2013/05/21

Table 10.15 - Deliverable and Milestone Summary

XI. Evaluation Questionnaires

A. Online Survey – ARmax Questionnaire

* Required

What are the mobile platforms do you often uses? *

- ☐ Android
- ☐ iOS
- ☐ BlackBerry
- ☐ Windows Phone
- ☐ Palm WebOS
- ☐ Bada
- ☐ Symbian
- ☐ Other: ______

Are you familiar with mobile application development? *

- ○ Yes
- ○ Somewhat
- ○ No

Are you interested in mobile application development? *

- Strongly Interested
- Interested
- Somewhat
- Not Interested

If so, to which mobile platform(s) do you often develop applications?

- Android
- iOS
- BlackBerry
- Windows Phone
- Palm WebOS
- Bada
- Symbian
- Other:

Are you familiar with mobile cross-platform application development? *

- Yes
- Somewhat
- No

If so, what are the mobile cross-platform framework do you use often?

- PhoneGap (Cordova)
- Adobe Air
- Appcelerator Titanium
- RhoMobile
- WidgetPad
- OpenPlug
- WorkLight
- Other:

If you were asked to develop same mobile application for different platforms, would you choose native approach or cross-platform approach or approach will be chosen depending on the application?

- Native

- Cross-Platform
- Depending on the application

Please rate following criteria with your experience in NATIVE mobile application development. (Vendor lock-in: In economics, vendor lock-in, also known as proprietary lock-in or customer lock-in, makes a customer dependent on a vendor for products and services, unable to use another vendor without substantial switching costs. Lock-in costs which create barriers to market entry may result in antitrust action against a monopoly.)

	Strongly Agree	Agree	Sort of Agree	Disagree	Strongly Disagree
Code is reusable					
Plugins					
Easy for web developers					
Reduced development costs					
Easy deployment					
Device/OS feature support					
High performance					
High-end graphics and 3D support					
Vendor lock-in					

Are you familiar with mobile augmented reality applications? *

- Yes
- Somewhat
- No

What are the mobile augmented reality applications that you have used?

- Wikitude
- Layar
- Other:

Are you interested in mobile augmented reality application development? *

- Strongly Interested
- Interested
- Somewhat
- Not Interested

What approach do you often choose to develop mobile augmented reality applications?

- Native
- Cross-Platform

Do you think cross-platform approach would be better to develop mobile augmented reality applications?

- Yes
- Somewhat
- No

Please rate following criteria with your experience in CROSS-PLATFORM mobile application development. (Vendor lock-in: In economics, vendor lock-in, also known as proprietary lock-in or customer lock-in, makes a customer dependent on a vendor for products and services, unable to use another vendor without substantial switching costs. Lock-in costs which create barriers to market entry may result in antitrust action against a monopoly.)

	Strongly Agree	Agree	Sort of Agree	Disagree	Strongly Disagree
Code is reusable					
Plugins					
Easy for web developers					
Reduced development costs					
Easy deployment					
Device/OS feature support					
High performance					
High-end graphics and 3D support					
Vendor lock-in					

What kind of problems do you think might occur or experienced when developing mobile augmented reality applications using cross-platform approach?

- ☐ Performance issues
- ☐ Poor graphic quality
- ☐ User interface issues
- ☐ Other:

What are the cross-platform mobile augmented reality frameworks/tools you familiar with?

- ☐ ARchitect-Wikitude
- ☐ Other:

Please rate features that you would expect from a cross-platform mobile augmented reality framework. * (1 - Doesn't Know, 2 - Not Needed, 3 - Somewhat Needed, 4 - Needed, 5 - Must There)

	1	2	3	4	5
Full Support of HTML, JavaScript and CSS	○	○	○	○	○
Geo-Objects and Geo-Location API	○	○	○	○	○
API support to other native features of mobile devices such as accelerometer, camera, contacts, media, network, notification, etc.	○	○	○	○	○
Marker/image recognition and tracking	○	○	○	○	○
Support for shapes, text, images, animations and multimedia formats	○	○	○	○	○
3D object support	○	○	○	○	○
Advanced visualisations (radar and direction indicator)	○	○	○	○	○
Interactions (Field of Vision (FOV) trigger, geo trigger, click, touch, etc.)	○	○	○	○	○

	1	2	3	4	5
NFC Support					
Easy development and deployment					
Vendor lock-in					

Please specify any other features that you expect?

Do you familiar with NFC technology? *

- Yes
- Somewhat
- No

Do you think combining NFC technology with augmented reality applications may improve the user interaction/experience?

- Yes
- Somewhat
- No

B. Evaluation Questionnaire

* Required

Name

Primary occupation or designation? *

What is your opinion on the concept of the solution? *

What is your opinion on using web technologies such as HTML5, JavaScript and CSS3 for development of an augmented reality application? *

Do you think that combining mobile cross-platform development solutions such as Adobe PhoneGap with web technologies enables the development of augmented reality applications easily compared to native development solutions? *

What are your comments on the ARmax prototype? *

Please watch http://youtu.be/Oe5R_teGFpI for an online video demonstration of the ARmax Framework.

In what application areas do you think that this solution is most useful in?

- ☐ Media
- ☐ Travel
- ☐ Music
- ☐ Advertising
- ☐ Safety
- ☐ Sports
- ☐ Other:

In your opinion what are the main benefits of the framework regarding the development of as augmented reality application for multiple mobile platforms such as Android, iOS, Blackberry, etc.? *

Do you think that proposed solution could be used to develop augmented reality applications at present?

In your opinion what are the drawbacks of the solution?

What do you suggest as future enhancements for the ARmax framework? *

XII. Test Results

A. Graphic Module Test Results

No	Purpose	Expected Output	Actual Output	Pass Rate	Status
EC1 Shapes					
1.1	Draw shapes				
1.1.1	Rectangle	Rectangle drawn	Rectangle drawn	20/20	100%
1.1.2	Circles	Circles drawn	Circles drawn	20/20	100%
1.1.3	Image	Image drawn	Image drawn	20/20	100%
1.1.4	Text	Text drawn	Text drawn	20/20	100%
1.1.5	Path	Path drawn	Path drawn	20/20	100%
1.1.6	Star	Star drawn	Star drawn	20/20	100%
EC2 Node Nesting and Layering					
2.1	Add shape(s) to layer				
2.1.1	Add one shape	Canvas should render with added shape	Shape drawn in the canvas	20/20	100%
2.1.2	Add two shapes	Canvas should render with all the added shapes	All shapes drawn in the canvas	20/20	100%
2.1.3	Add many shapes	Canvas should render with all the added shapes	All shapes drawn in the canvas	20/20	100%
2.2	Add layer(s) and shape(s) to stage				
2.2.1	Add one layer and one shape	Canvas should render with added layer	Shape drawn in the canvas	20/20	100%
2.2.2	Add two layers and two shapes	Canvas should render with all the added layers	All shapes drawn in the canvas	20/20	100%
2.2.3	Add many layers with many shapes	Canvas should render with all the added layers	All shapes drawn in the canvas	20/20	100%
EC3 Transitions and Animations					
3.1	Draw a transition	Transition should work	Transition worked	20/20	100%
3.2	Draw an animation	Animation should work	Animation worked	20/20	100%
EC4 Events					
4.1	Add a tap event to a shape	Tap event should work	Tap event worked	20/20	100%
4.2	Add listener event to a shape	Listener should work	Listener worked	20/20	100%

4.3	Perform drag and drop on a shape	Shape should be able to drag and drop	Drag and drop on shape worked	20/20	100%

Test Plan 10.1 - Test Plan of Graphic Module

B. Sensor Module Test Results

No	Purpose	Expected Output	Actual Output	Pass Rate	Status
EC1 Retrieve sensors data					
1.1	Compass	Heading data should receive (i.e. current pointing direction of the device)	Heading received	20/20	Pass
1.2	GeoLoation	Position data should receive (i.e. longitude and latitude)	Position received	20/20	Pass
1.3	Accelerometer	Acceleration should receive (i.e. x, y, z values)	Acceleration received	20/20	Pass
EC2 Retrieve location details on different internet connection types					
2.1	No connection	No connection error	No internet connectivity	20/20	Pass
2.2	3G (data package)	Position data should receive	Position received	20/20	Pass
2.3	Wi-Fi	Position data should receive	Position received	20/20	Pass
EC3 Different mobile devices					
3.1	Samsung Galaxy SIII	Sensor data should receive	Sensor data received	20/20	Pass
3.2	Apple iPhone 4	Sensor data should receive	Sensor data received	20/20	Pass
3.3	HTC One x	Sensor data should receive	Sensor data received	20/20	Pass
3.4	HTC Wildfire S	Sensor data should receive	Sensor data received	20/20	Pass
3.5	Sony Xperia P	Sensor data should receive	Sensor data received	20/20	Pass

Test Plan 10.2 - Test Plan of Sensor Module

C. Map Module Test Results

No	Purpose	Expected Output	Actual Output	Pass Rate	Status
EC1 Markers					
1.1	Add one marker	Map should show the marker	Marker showed	20/20	Pass
1.2	Add more than one marker	Map should show the markers	Markers showed	20/20	Pass
EC2 Directions					
2.1	Get directions to a location	The path and the direction should show in the map	Path and the directions updated in the map	20/20	Pass
EC3 StreetView					
3.1	Access streetview of location in USA	StreetView should show	StreetView showed	20/20	Pass
3.2	Access street view of location in Sri Lanka	Not supported error	StreetView is not available	20/20	Pass

Test Plan 10.3 - Test Plan of Map Module

D. QR Code/ Barcode Recognition Module

No	Purpose	Expected Output	Actual Output	Pass Rate	Status
EC1 Different QR code information types					
1.1	URL	Recognise the QR code	QR code recognised	20/20	Pass
1.2	Text	Recognise the QR code	QR code recognised	20/20	Pass
1.3	Phone number	Recognise the QR code	QR code recognised	20/20	Pass
1.4	SMS	Recognise the QR code	QR code recognised	20/20	Pass
1.5	Contact	Recognise the QR code	QR code recognised	20/20	Pass
EC2 Different QR code sizes					
2.1	Small	Recognise the QR code	QR code recognised	20/20	Pass
2.2	Normal	Recognise the QR code	QR code recognised	20/20	Pass
2.3	Large	Recognise the QR code	QR code recognised	20/20	Pass
2.4	Very Large	Recognise the QR code	QR code recognised	20/20	Pass
EC3 Other Barcode Types					
3.1	Code 128	Recognise success	Code recognised	20/20	Pass
3.2	UPC-A	Recognise success	Code recognised	20/20	Pass
3.3	Data Matrix	Recognise success	Code recognised	20/20	Pass
3.4	EAN_8	Recognise success	Code recognised	20/20	Pass
3.5	Code 39	Recognise success	Code recognised	20/20	Pass
3.6	Maxicode	Recognise fail	Code not recognised	20/20	Pass
3.7	Code 16K	Recognise fail	Code not recognised	20/20	Pass
EC4 Different lighting conditions					
4.1	Low light environment	Recognise the QR code	QR code recognised	20/20	Pass
4.2	Normal light environment	Recognise the QR code	QR code recognised	20/20	Pass
EC5 Different colour or custom QR codes					
5.1	Custom QR code 1 (red)	Recognise the QR code	QR code recognised	20/20	Pass
5.2	Custom QR code 2 (black)	Recognise the QR code	QR code recognised	20/20	Pass
5.3	Custom QR code 3 (green)	Recognise the QR code	QR code recognised	20/20	Pass

Test Plan 10.4 - Test Plan of QR Code/Barcode Recognition Module