BTech 451A

ASIST - Asset History In Real Time

Development of a Mobile Computing Application

Phase One - Scoping and Design

Anthony Peek

ID 2628256
UPI apeee229

7th June 2015

Acknowledgements

Kodie Wixon, Team Manager, Software Development
Opus International Consultants Ltd.

Roquito Lim, Senior Financial Analyst
Opus International Consultants Ltd.

Andrew Bruce, Business Group Manager
Opus International Consultants Ltd.

Duan Zhao, Information Analyst / Asset Manager
Opus International Consultants Ltd.

Dr. Sulo Shanmuganathan, Chief Technical Officer
Opus International Consultants Ltd.

Dr. Sathiamoorthy Manoharan, Academic Supervisor
The University of Auckland

Executive Summary

Opus, an infrastructure consultancy firm have asked for ASIST, a mobile application theorised in a recent in-house innovation competition, to be developed. ASIST's, or asset history in real time's goal is to convert the significantly large amounts of data collected for their structures such as buildings and roads from convoluted and expansive, to intuitive and minimal. This has lead to an augmented reality solution by Andrew Bruce.

Augmented reality in this case utilises the mobile device's location with components such as the GPS, compass and accelerometer to derive the absolute positioning and angle of the device. When paired with the device's camera, information related to this positioning can be visualised in real time against the real world using a data layer. Information collected about assets such as condition data, past repairs, and future plans can then be displayed on this data layer in real time while working in the field. This is a far more practical solution that can be understood quickly and without prior knowledge of the domain, compared to the current data tables used which require processing and transformation to gain understanding from.

My task is to transform this idea into a functional application. The first 12 weeks, or phase one of this project was assigned to scoping and design. The key decision required in this time is the choice of platform to use. First, current related applications were researched in an attempt to aid future design decisions. The results of this research didn't overwhelming suggest a platform to use, so further research was conducted.

A web based approach was first investigated, assessing current solutions and investing time into development of required functionalities. The main advantage of a web based approach is the production of a multi-platform application with a single code base. Intel XDK was used as the testing development environment and a location based application was worked with. To further aid development, third party SDKs were also investigated and possible applicable kits noted.

Hardware requirements research was then requested. The key factor that the hardware contributes towards is the accuracy of the positioning. Hence the best GPS system possible is required. It is found to be a combination of both the American GPS system and the Russian GLONASS system that achieves the most accurate results, hence this technology is strongly recommended. Other requirements such as screen size, processor speed and access to a network connection are also recorded.

The result of research into hardware suggested an Android capable device, hence native Android development was then tested. Again, as with web based testing, the recommended tools such as Android Studio were found, and the overall structure of a native application understood. A basic camera application was then developed to test suitability of the language, which was promising.

The above research and stronger support of Java development from myself and project supervisor Kodie has lead to the decision of an Android implementation. The first task of phase two will be implementing a basic location based prototype to validate the accuracy of this approach. Once the accuracy is sufficient, the augmented data layer will be developed, attempting to display as much information as possible while using a minimal amount of screen real estate. This prototype will be one of the most important milestones of the project. Once this base application is created, time will be spent iterating on more features and an ongoing assessment of usability.

Contents

1. Introduction

The first 12 weeks of this project have been full of meetings, research, development, testing, and numerous reports. The following report examines the software project undertaken thus far, in more or less the order that the tasks have been completed. First Opus, the firm this project is in collaboration with, and ASIST, the project itself will be outlined and use-case scenarios discussed. Next, various research and testing in regard to moving forward with ASIST will be presented. Finally, recommendations and plans regarding the project's development next semester will be disclosed, supported by a risk analysis of potential problems.

2. The Firm—Opus

Opus are a multi-disciplinary infrastructure consultancy group, with operations spanning over Australia, Canada, New Zealand, the Unites States and the United Kingdom, with over half of their 3000 staff operating from the 40 offices here in New Zealand. The common domains of operation by Opus include endeavours in both the private and public sector within transport, water, environment, buildings, energy, resources, research and telecommunications [1]. Software development only plays a small role in the grand scheme of things however, resulting in a single New Zealand software team working out of Christchurch.

3. The Project—ASIST

3.1 Problem Description

ASIST is one of the product ideas to come out of the company's Big Ideas innovation competition in 2014. Designed by Andrew Bruce and further documented by Duan Zhao, ASIST, or asset history in real time aims to simplify the task of translating asset data such as the condition of roads and buildings into a comprehendible form. A vast amount of data is gathered and stored to describe these assets, and currently, this data is simply given to the user in tabular form, as below:

Table 1. Current Road Assessment and Maintenance Management (RAMM) data

Road Name	Road ID	Displacement	Start	End	Length	Start Name	Surface Date	Removed	Design Life	Offsets	Offset (LHS)	Surface Width	Full Width	Function	Surface Material	1st Chip Size
01S-0933	797	0-100m	0	100	100		26/01/2007		6	0 - 8.5	0	8.5	No	Reseal	TEXT	5
01S-0926	796	0-210m	0	210	210		26/02/2001		9	0 - 11.3	0	11.3	No	Reseal	1CHIP	3
01S-0926	796	0-290m	0	290	290		25/12/1984		10	2.5 - 11	2.5	8.5	No	Reseal	1CHIP	4
01S-0926	796	0-290m	0	290	290		5/04/1995		5	0 - 11.3	0	11.3	No	Reseal	TEXT	5
01S-0926	796	0-320m	0	320	320	KEW ROAD NORTH	21/04/2009		12	0 - 15.5	0	15.5	Yes	Reseal	2CHIP	2
01S-0926	796	0-320m	0	320	320	KEW ROAD SOUTH	17/12/2014		10	5.4 - 10.8	5.4	5.4	No	1st Coat	2CHIP	3
01S-0933	797	0-560m	0	560	560		25/12/1984		12	0 - 12	0	12	No	Reseal	1CHIP	3
01S-0933	797	0-560m	0	560	560		29/01/1996		9	0 - 12	0	12	No	Reseal	1CHIP	3
01S-0926	796	40-290m	40	290	250		1/05/1992		5	0.2 - 3	0.2	2.8	No	2nd Coat	1CHIP	5
01S-0933	797	100-560m	100	560	460		23/01/2007		11	0 - 12	0	12	No	Reseal	2CHIP	4
01S-0926	796	210-510m	210	510	300	START TURNBAY	30/06/1999		2	10 - 12.8	10	2.8	No	1st Coat	2CHIP	4
01S-0926	796	210-510m	210	510	300		26/02/2001		9	0 - 12.8	0	12.8	No	Reseal	1CHIP	3
01S-0926	796	290-730m	290	730	440		25/12/1984		10	0 - 9	0	9	No	Reseal	1CHIP	4
01S-0926	796	290-7427m	290	7427	7137		23/12/1994		1	9.5 - 10	9.5	0.5	No	1st Coat	1CHIP	4
01S-0926	796	290-7427m	290	7427	7137		24/12/1986		10	0.5 - 9.5	0.5	9	No	Reseal	1CHIP	4
01S-0926	796	290-7427m	290	7427	7137		23/12/1993		1	0 - 0.5	0	0.5	No	1st Coat	1CHIP	4
01S-0926	796	290-7427m	290	7427	7137		5/04/1995		5	0 - 10	0	10	No	Reseal	TEXT	5
01S-0926	796	320-650m	320	650	330	KEW ROAD SOUTH 2	11/12/2009		10	0 - 12	0	12	Yes	Reseal	2CHIP	3
01S-0926	796	510-6410m	510	6410	5900		26/02/2001		9	0 - 10	0	10	No	Reseal	1CHIP	3
01S-0933	797	560-1280m	560	1280	720		25/12/1985		6	0 - 8.5	0	8.5	No	Reseal	TEXT	5
01S-0933	797	560-3270m	560	3270	2710		23/01/2007		11	0 - 8.8	0	8.8	No	Reseal	2CHIP	4
01S-0933	797	560-3270m	560	3270	2710		3/11/1993		10	0 - 8.8	0	8.8	No	Reseal	1CHIP	3
01S-0926	796	650-1510m	650	1510	860	KEW ROAD	29/03/2010		1	0 - 10	0	10	Yes	1st Coat	2CHIP	3
01S-0926	796	650-1510m	650	1510	860	KEW ROAD	30/11/2010		8	0 - 10	0	10	Yes	2nd Coat	2CHIP	3

Although descriptive, this representation of data requires a significant understanding of the domain and is overall difficult and time consuming to analyse.

3.2 The solution

Certain characteristics of data can only be seen when data is represented graphically. Hence a more intuitive approach to interpreting this data is through visualisation [2]. This has lead to the idea of an augmented reality visualisation of the data. Assuming that intended results are like that displayed to the right, all gathered data will be intuitively recognisable, requiring little specialised knowledge of the domain, and increasing both productivity and user tolerance and acceptance of the system [3].

Fig 1. Road Surface Condition

3.3 Application Functionality

ASIST is intended to contain a handful of related features which all revolve around the idea of augmenting information of the asset in real time. This will be known as the 'live view' function throughout this document. With respect to roading, the condition, such as skid resistance and age of seal will be visualised with simple colour coding to represent the status of the road, as displayed in figures 1 and 2.

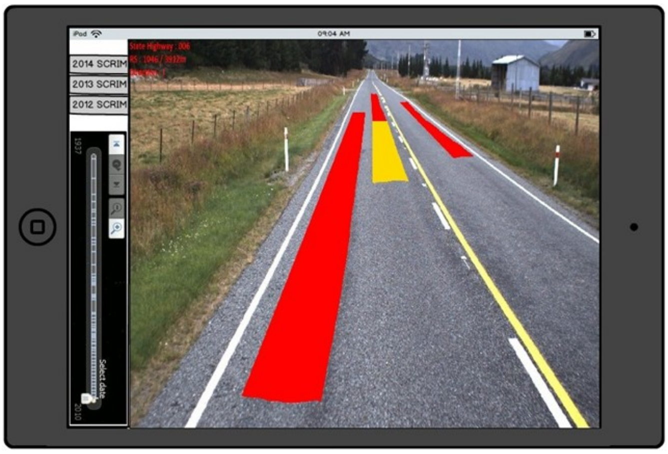


Fig 2. Road Surface Skid Resistance

Using a colour scheme that is recognisable by the user, such as green for good condition and red for bad, or the use of an intuitive legend that allows for more detailed data to be represented maximises the information gained from the application, while minimising complexity and user resistance of adoption. This data can then be used in other domains such as determining if there were any roading factors involved during a crash investigation. It is key that the data is understandable in this scenario due to outside users such as the police force needing to interpret the road condition correctly and easily.

Another augmented reality feature intended to be included in ASIST is the process of inventory validation. Assuming high location accuracy of the device being used, the application should be able to be used in the field to validate if the stored data of an object such as a culvert/drain pipe is correct in relation to the real world. Figure 3 illustrates such testing.

Fig 3. Inventory Validation—Misplaced Culvert

The last main augmented reality feature pertains to structures such as buildings and bridges. ASIST should be able to compare a current construction site or structure with either previous historic images and information as shown in figure 4, or in the case of ongoing construction, 3D rendering of the proposed finished product. Again this simplifies and speeds up the process of translating data into usable knowledge. This feature requires the same location based information as the roading component, but utilises a more elaborate technique for displaying data, so is planned to be implemented only after the roading component is complete.

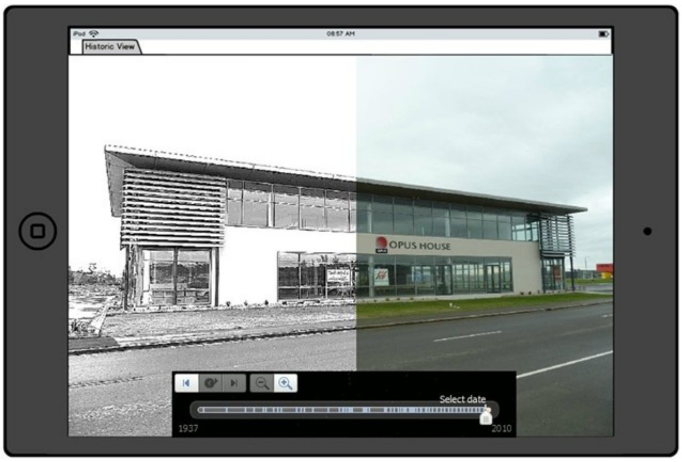


Fig 4. Asset Image Comparison

As seen in the images, all of the functions are planned to have a historic time slider to allow users to quickly compare and contrast the condition of assets over different periods of time, allowing for trends in degradation of assets to be analysed and hence decisions on future improvements to be made while on site.

3.4 Use Cases

Now that the application and it's overall functionality is understood and reasoned, a few specific use cases will be discussed and their relation to the back-end asset manager explained. The asset manager is a more traditionally laid out interface that gives users the ability to access and modify details of an asset via text and other inputs. It works alongside the aforementioned live view function and stores the data required to visualise the information in that format.

1. Management Portal—KPI Performance and Report Generation

Throughout the process of a construction project, the site manager will want to assess and document the progress of the work being done. The user will have ongoing milestones and goals which are to be achieved, and requirements which must be compared to key performance indicators (KPI). This view will provide a real time dashboard of the current project and will allow the user to generate reports regarding KPI, maintenance costs, custom reports and more. These can then be accessed from both the mobile application and desktop applications that are connected to the Opus network.

2. Asset Information Portal

Accessible from either the associated live view function, or directly through the asset manager, this view will provide a listing of nearby assets while in the field, or a list of assets at a location specified by user input. The user can then select which asset he/she would like to analyse. Information specific to that asset, similar to that observed in the earlier data table, will be displayed.

This includes;

Location data	- New Zealand State Highway Treatment Length, Offset, GPS coordinates etc.
Functionality	- Use of the asset.
Dimensions	- Length, width, height etc.
Age	- Time since installation.

3. Condition Data

Extending from the asset information portal, the user can then view and update stored condition data of a given asset. This view combines features from the previous two use cases, leveraging the performance reports written in the management portal and displaying asset information. Other related condition information of the asset is also included, such as current and previous photos and condition descriptions as seen in the live view function being accessible.

4. Maintenance Data

When a request to assess asset maintenance information is made, this portal will give access to; previous work undertaken, such as initial construction and recent repairs, current maintenance work being completed on the asset, and reports for future construction endeavours on the asset.

5. Asset Photo Library

When a user wishes to update the current photos on hand for an asset, the photo library is used. Either linked to an external camera function or preview-able from within ASIST, the user can take, annotate, and save an image. This will be automatically associated to the asset in question, derived from location and perspective data of the device

Fig 5. Asset Manager—Management Portal, Asset Condition Data, Asset Maintenance Data

4. My Role and People Involved

The above information outlining the vision and capabilities of the application has been derived from initial concept documents developed by Andrew and Duan. My role is to make it reality. Chief technical officer Sulo Shanmuganathan has passed this project to senior financial analyst Roquito Lim who has in turn given myself and software development team manager Kodie Wixon the project to complete. Kodie acts as my project supervisor, giving tasks for me to complete and ensuring that I stay on track with project requirements and deadlines.

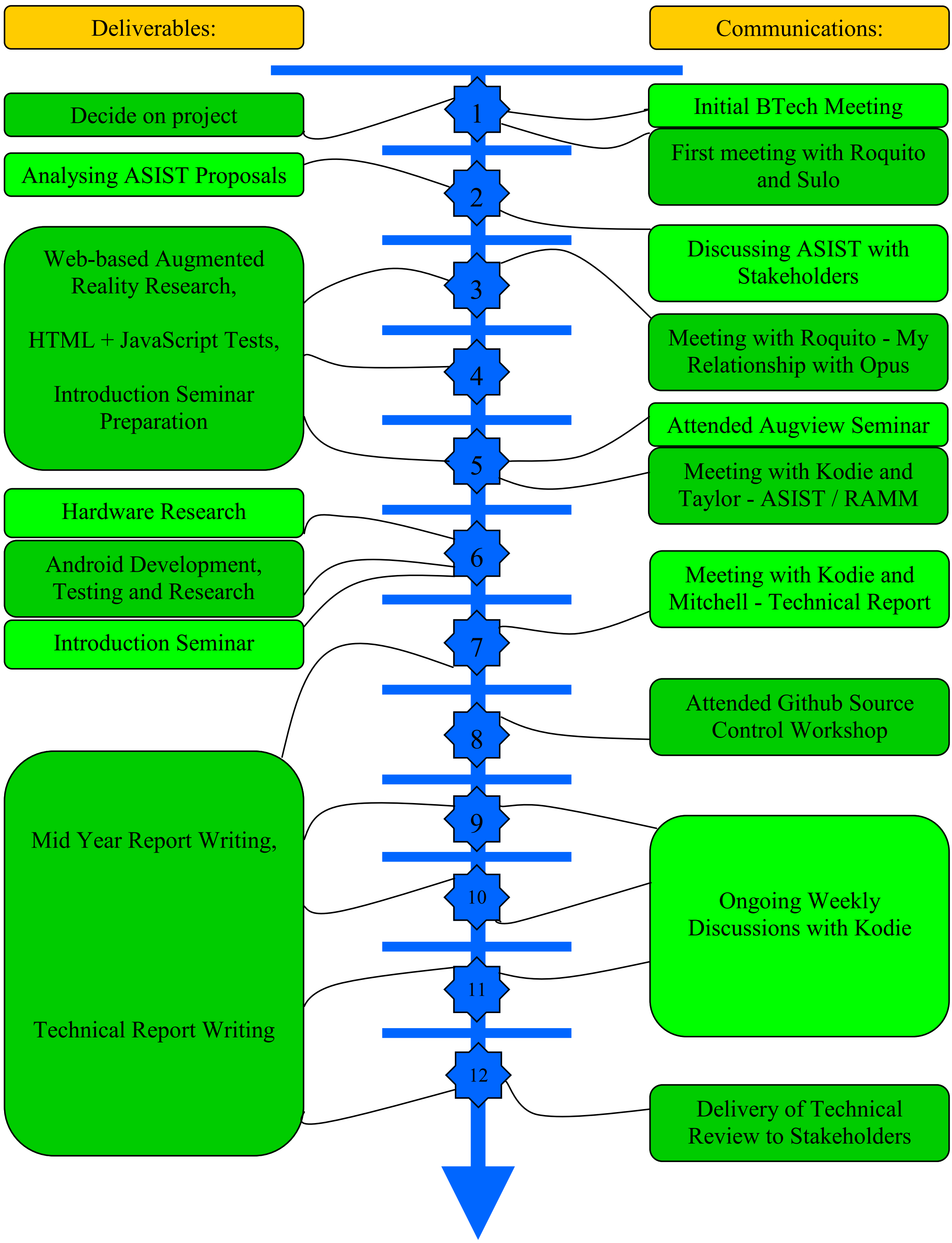
During the scoping and design phase of the past three months, I have been required to assess the potential development paths of ASIST and make the decision, with Kodie's support, on how to proceed into stage two, prototype development. Hence these are the primary topics covered in this report.

During the next stage, I will continue to play the leading role in the project, undertaking the majority of development and making most development decisions. This will still be under Kodie's supervision, with the addition of a second developer appointed by Kodie to increase productivity.

The intended outcome by the end of the year is to have, at a minimum, a working prototype with the core live view functionality. Given the amount of time assigned to this project, I expect to exceed this minimum, with most of the aforementioned use cases available and a large amount of data accessible.

5. Timeline

By week



6. Implementation Research and Reasoning

Initial meetings with, and documentation from Opus had suggested for the development of ASIST to be undertaken via either a web based application, or a Windows based native application. Executives pushed for a web-based implementation due to the cross platform operability that it would achieve, allowing all current company devices, independent of operating system, to be compatible with a single implementation of the ASIST application. However, from the development side:

> "The preferred software platform for the development of ASIST at this point in time is a Windows based platform as this is the current operating system within Opus so integration with existing infrastructure and support should be less complicated." [4]

It was also mentioned however that the platform that ASIST would operate on is not critical, and that the main factors when choosing a platform are the availability of key components within the hardware.

Before researching specific platforms and their compatibility with an augmented reality application, comparable, already completed augmented reality solutions were researched in an attempt to adapt or use them as a reference while developing ASIST, and to discover what tools and technologies were used in the development process. With augmented reality applications being available on the Apple Store since 2009, it came as no surprise to discover many applications that fulfilled a similar purpose as the plans of ASIST [5].

6.1 Augmented Reality

Augmented reality solutions can generally be divided into two categories; either location based, or image recognition based. Location based services use the device's GPS to determine the user's location, and the device's accelerometer to determine the angle the mobile device is being held on and hence what it is looking at. Image recognition services on the other hand use purely the device's camera with image processing calculations to determine what it is looking at. Simple implementations for domain specific applications can see a set of certain patterns be hard coded and then easily recognised when aligned to the camera correctly. QR code readers are a common example of this type of image recognition, with facial recognition being a more elaborate and unconstrained domain.

The second component of augmented reality is a digital data layer which is superimposed over the top of the user's perspective. In either augmented reality case, once the target of the application has been recognised, the augmented data layer is then populated with predefined information over the top of the camera's preview display.

During the early stages of this project, and at the time of writing, we have decided to implement ASIST as a location based augmented reality application as to not restrict the type of asset in which ASIST is to assess. Below are the three main levels of current location based augmented reality applications researched and how they can be applied to ASIST:

Current Augmented Reality Solutions

1) Augmented Reality Fundamentals:

Each of the three levels utilize the device functions of vision, location, and movement, alongside a data layer to convey information to the user. Even if the data displayed is not in a form as ASIST requires, the fundamental functionality behind the application can be used as a guide to help direct initial development. Applications under this category include Junaio and Wikitude Places which mark points of interest such as landmarks or restaurants on the data layer to help users navigate [6] [7]. Since this type of application contains the majority of functionality that ASIST requires, the technology used should be able to be applied to ASIST as well. Both Junaio and Wikitude Places are built using a web-based implementation. Junaio uses AREL—Augmented Reality Experience Language, which is a combination of HTML, CSS, JavaScript and PHP. Wikitude Places uses the Wikitude SDK, which again uses HTML5, CSS, and JavaScript.

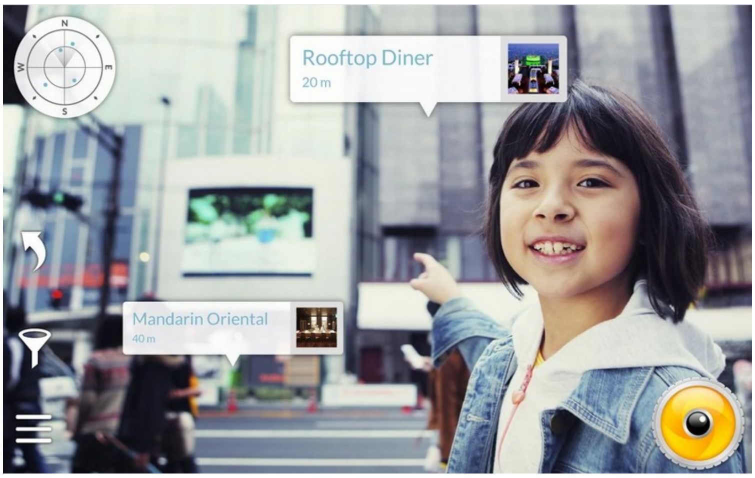


Fig 6. Wikitude Places Augmented Reality Application

2) 3D Augmented Reality Rendering:

Continuing on from the augmented reality fundamentals, applications that render both realistic and abstract 3D descriptions of assets are also widely available. Satellite AR and ESET Augmented Reality are examples of this and extend from a basic augmented reality application by applying interactive object specific graphics [8] [9]. Understanding how to use the data layers effectively as is done in these applications will be very important in contributing to the usefulness of ASIST. These techniques will be looked at further into the development lifecycle when the time comes to finalising data representation.

Fig 7. Satellite Augmented Reality Application

3) Geographic Information System (GIS)

As part of the research process, I attended a three hour seminar by local firm Augview discussing their own augmented reality products. Their main product is an augmented reality powered geographic information system, and is by far the most comparable application to ASIST discovered when researching what was currently available. A GIS is a "system for capturing, storing, checking, and displaying data related to positions on Earth's surface" and hence relates closely to what ASIST is trying to achieve [10]. Winner of various 2014 NZ spatial excellence awards, Augview "allows users to visualize underground objects that they wouldn't usually see" [11]. After talking to Augview business development manager Melanie Langlotz, she said that the application was developed natively for both Android and iOS devices.

Fig 8. Augview GIS Augmented Reality Application

Due to the fact that the current applications researched had been developed using various techniques, the conclusion from these findings was to continue research. Knowing that each of the aforementioned platforms are capable of housing an augmented reality application, the decision on which platform(s) to pursue will depend on;

1) Findings with respect to each of the different platforms

2) Hardware limitations of the devices that are compatible with the different platforms

3) Development capability of myself and others involved with development with respect to the different platforms

Hence the following sections will discuss these areas in the order in which they were completed.

6.2 Web Application Research

Due to a web-based application being the least limiting in terms of platform from the solutions suggested, much of the initial pre-development time was spent researching and testing the capabilities of this approach with respect to ASIST. Other than cross platform functionality, one of the appealing factors of creating a mobile web application is that alongside a UI framework, the only core tools required are simply HTML5, JavaScript, and CSS. This allows anyone with experience in web design, one of the more common forms of software development, the ability to create a mobile application.

Due to ASIST's requirements of having access to the device's camera, accelerometer, and location, the first task was ensuring that these functions were accessible in JavaScript. This quickly lead to the adoption of Apache's Cordova:

> "Apache Cordova is a set of device APIs that allow a mobile app developer to access native device function such as the camera or accelerometer from JavaScript. Combined with a UI framework such as jQuery Mobile or Dojo Mobile or Sencha Touch, this allows a smart phone app to be developed with just HTML, CSS, and JavaScript." [12]

Once it was apparent that a web based implementation had access to the essential features required by ASIST, various tools and methods such as IDEs, SDKs and APIs for augmented reality web development were researched and tested.

Integrated Development Environment (IDE)

Due to being relatively new to web development, I decided that the first tool to look into would be the integrated development environment. An IDE combines all the tools required to develop for a chosen platform to help automate or simplify common tasks such as compiling and testing applications while eliminating the need to use the command line or other external applications [13]. Therefore finding the right IDE would fast track my adoption of the HTML5/JavaScript platform and reduce the number of teething issues faced when starting development with this new platform.

After researching and testing a few different environments, the most appropriate choice to continue with was Intel's Cross Development Kit (XDK) [14]. As a free download from a reputable name, along with sufficient development guides and samples, it was an easy decision to make and using the XDK sufficiently reduced the time taken to adopt the platform.

Once familiar with the system, the sample augmented reality application from Intel was tested:

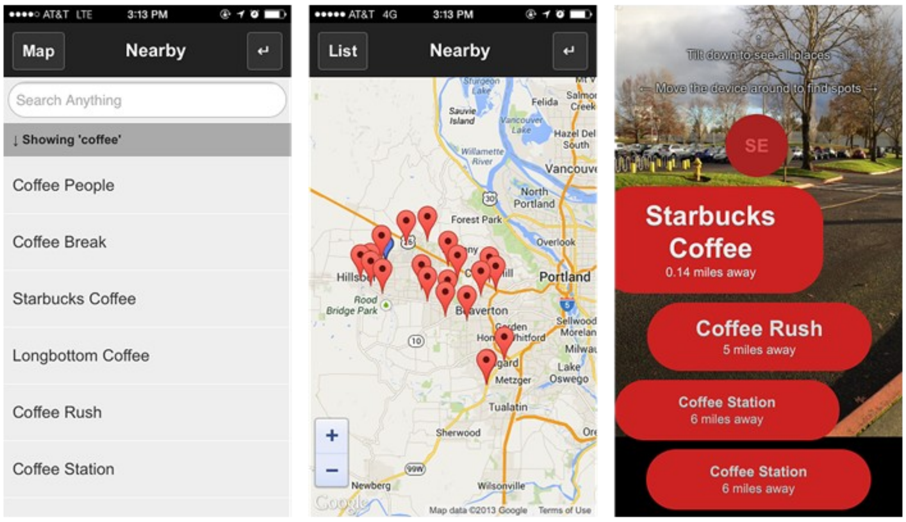


Fig 9. Location Based Augmented Reality Sample Application [15]

Much like the augmented reality fundamentals applications discussed previously, this application appeared to be a great starting block to build ASIST from, with only the visual data layers needing modification to display data correctly for ASIST. Unfortunately, as many other users had mentioned, the application failed to perform correctly, and even after hours of modifications and additions, I was still at a dead end of not having access to the camera background and hence decided to move on to further research in the form of third party software development kits (SDKs):

Software Development Kits (SDKs)

As with most domains of software development, a code base of some description has already been written to simplify development, and despite augmented reality's reasonably recent arrival, a hand full of SDKs and APIs are available for use. Three viable SDK options were analysed; Wikitude, Layar, and Metaio.

Wikitude SDK

Carrying on from the aforementioned Wikitude augmented reality application, Wikitude also offers their SDK to developers to create their own applications [16]. "Wikitude's all-in-one AR solution includes image recognition & tracking, 3D model rendering, video overlay and location based AR" - everything that ASIST needs, and more.

An example of an ASIST-like application created using the Wikitude SDK is the Hermes Virtual Tour [17]:

Fig 10. Hermes Virtual Tour Augmented Reality

The Hermes Virtual Tour recognizes via location and image recognition missing monuments, and renders them on the augmented data layer.

The SDK is available in a few options, ranging from a free, watermarked trial, to an all inclusive ~$6600 per year license. If it is decided to complete the development of ASIST with a web based implementation, then the free version of Wikitude will be tested, and funding discussed with Opus if it appears to be a viable solution.

Layar SDK

As with the Wikitude SDK, the Layar SDK also implements both vision based and location based augmented reality [18]. As well as JavaScript Cordova libraries, Layar also offers plug-ins for both native Android and iOS development, offering flexibility of platform if Layar is chosen to aid development. Unfortunately the Layar SDK comes with only a 30 day free trial before a purchase is required, so we decided against investigating this tool further.

Metaio SDK

Metaio is the last web based add-on researched [19]. It shares a lot of similarities with Wikitude, including a similar price model of a free unlimited trial followed by various pricing for different features. Metaio then extends past the requirements of ASIST with a whole host of features including Unity support and facial tracking.

In conclusion, if a web-app were pursued, any of the above SDKs would be applicable to the ASIST project. However, all of the SDKs researched go well and above what is required by ASIST at this stage, such as image recognition and wearable technology integration. So provided that the software development team and I are confident in using the fundamental Cordova device plug-ins, an SDK may not be needed. If the decision to use an SDK is made however, that decision would be based on ease of use, documentation, and price. From the findings thus far, Wikitude is by far the front runner due to granting access to an unlimited free (watermarked) license, alongside very recent documentation discussing the steps required to integrate and use the SDK with Intel's XDK and other IDEs.

6.3 Hardware Requirements

Once research on the web platform had been completed, I was asked to put time into researching hardware that satisfies the needs of ASIST. The following was presented to Kodie and used as reasoning when suggesting the desired platform to develop with in the mid year project plan:

OPUS

Opus International Consultants Ltd
Christchurch Office
20 Moorhouse Avenue
PO Box 1482, Christchurch Mail Centre, Christchurch 8140
New Zealand

t: +64 3 363 5400
f: +64 3 365 7858
www.opus.co.nz

To	Kodie Wixon
From	Tony Peek
Date	20 April 2015
Subject	ASIST Hardware Research

1.1 Summary

GPS technology, notably GLONASS is discussed. Android and Apple devices, and Qualcomm processors are then explained. Finally other device requirements are detailed and recommended devices suggested.

1.2 GPS

When choosing a mobile device to use with the ASIST application, the most important feature to consider is the accuracy of the device's GPS. If the GPS is inaccurate, then ASIST's usability will be severely limited. There are a few different GPS technologies that can be used, each with varying degrees of accuracy.

The standard GPS in older mobile devices uses triangulation (or multilateration) between nearby cell-phone towers, input from in-range Wi-Fi, and tracking from your last known position - without the need of an aerial or view of the sky [20].

'Real' i.e. satellite GPS systems can then be combined with this type of GPS to be known as Assisted GPS/A-GPS/AGPS to enhance results further. It appears most modern mobile devices have some form of assisted GPS technology, at least. The 'real' GPS system used is run by the U.S. and provides services worldwide with 32 satellites [21].

To aid this service, other satellite systems can be used in conjunction to provide more accurate location positioning. These systems include:

- BeiDou and Compass - China
- Galileo - Europe
- IRNSS - India
- QZSS - Japan
- WAAS - North America [22]
- GLONASS - Russia

1.3 GLONASS

GLONASS is Russia's equivalent to the above American satellite GPS system, is implemented worldwide with 24 satellites, and is integrated in a number of common phones [23].

How GLONASS improves positioning

By supporting both US and Russian systems in a receiver, the number of available global satellites increases. Therefore, more geographic locations are able to receive four or more signals from satellites, which in turn means more successful position calculations and also better accuracy of the calculated positions in challenging environments.

Field tests in downtown San Francisco found that the positioning accuracy increased by as much as 50% when adding GLONASS. The tests were executed with the help of Qualcomm using two Sony Ericsson Live with Walkman™ smart phones, and 600 measurements were recorded and analyzed per device [24]:

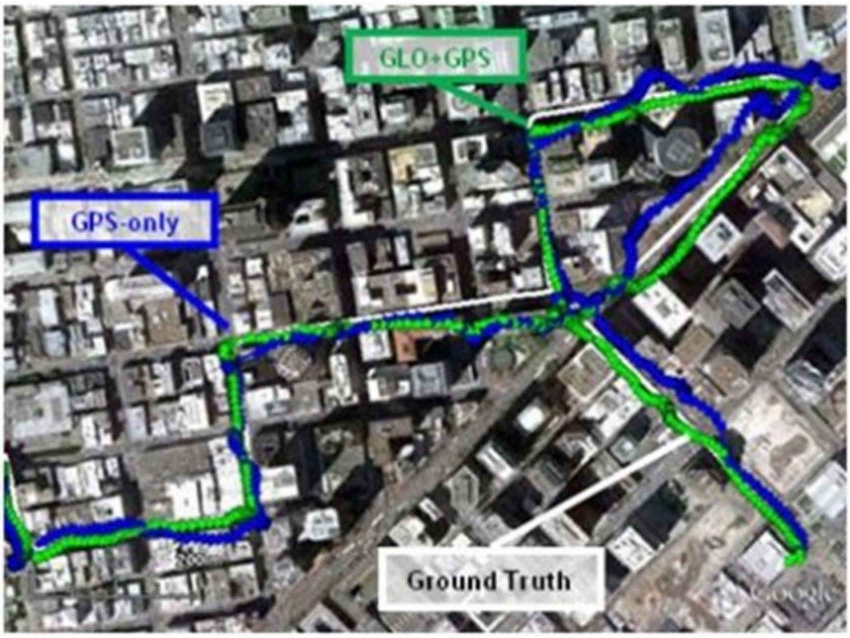


Fig 11. GPS and GLONASS Accuracy Comparison

Table 2. GPS and GLONASS Accuracy Comparison

Configuration	Number of satellites used*	CEP 68% (m)**
GPS only	5, 5	30, 4
GPS + GLONASS	10, 9	17, 7

* The number of satellites on average that were used to calculate the position. More satellites usually mean a more reliable and accurate location.
**CEP 68% means that 68% of the 600 measurements are within this distance, in meters, from the reference location. Hence lower values mean better accuracy.

Testing with my Sony Xperia ZR smart phone which uses GLONASS achieved accuracy of three metres when running a Chartcross Ltd GPS test.

From these findings, it will be in our best interest to use such a device in conjunction with ASIST. Fortunately, Qualcomm Snapdragon processors and chipsets, which are commonly found in the vast majority of Android consumer mobile devices post October 2012 appear to have GLONASS and BeiDou built in via iZat [25][26].

The Qualcomm site shows 21 Android, 1 Windows, and 3 Amazon Fire devices using this chip, so the advice from this information would be to develop for Android tablets [27]. Other chipsets offering GLONASS are available, but again, most are Android powered [28].

A quick breakdown on Apple tablets:

- Wi-Fi only iPads do not have a GPS at all [29][30].
- 3G/4G cellular models do, but exactly what technology they use differs.
- GLONASS support has been integrated since the 3rd generation iPad (March 2012 release) [31].

Due to some remote locations not having network access to calculate position information from, Wi-Fi only tablets without a 'real' GPS should not be considered.

1.4 Hardware Requirements

Required hardware specifications as given in original Opus document:

- Screen size - minimum of 7 inches
- Camera - minimum of 2 megapixels
- Storage - minimum of 16GB

PriceSpy lists 76 tablets with GLONASS support that satisfy the above attributes.

Requirements Expanded

Screen Size - The larger the screen size, the better ASIST will be able to convey information, so 10 inches and above should be considered as well. For roads and undetailed assets, 7 inches will be fine, but may become cluttered with more complicated assets.

Camera - At this stage, since the augmented reality is planned to only use positioning and accelerometer data, image recognition isn't needed so camera quality isn't too much of a concern. As long as the terrain is distinguishable on screen, the camera will be sufficient. However, if accuracy needs to be increased through some sort of image recognition, then a higher quality camera may be needed.

Storage - The application size will most likely be minimal, with cached map data requiring the majority of the storage. Depending on how much space the operating system uses, 16GB should be sufficient for ASIST and any other Opus and work related software.

Memory and Processing - Can't comment yet as to exact figures that will be needed to run an application such as ASIST. However, given that the tablet is adequate in the above categories and has a fairly recent chipset that includes GLONASS capability, then chances are that the CPU and memory will also be sufficient.

Network - It would be useful for the tablet to have 3G/4G capabilities for live data communications rather than relying on using an additional device to be a hotspot data source.

1.5 Conclusions

The original Opus report suggested a Windows device so to integrate with the Opus network easier, but at this stage, there doesn't appear to be a Windows product that supports all project requirements. Additionally, the Opus network integration is most likely manageable on any operating system and so this shouldn't be a reason to choose Windows over the other options.

This has left us to choose from tablets produced from either: Samsung (Android), Sony (Android), Google (Android), and Apple (iOS). Price wise, the Android solutions are generally cheaper.

If building for a single operating system while wanting to be compatible with as many products and brands as possible, Android development is the way to go from a hardware perspective. I will next review Android's capability of housing and developing this type of app compared to HTML5 and JavaScript. I have already researched into web development and found plenty of examples of working augmented reality applications, but am unsure of the difficulty of programming augmented reality web apps as opposed to Android based ones. If neither Android or HTML5 appear to be the correct solution, then Apple's Swift will be assessed as a possible development language.

1.6 Specific Tablet Recommendation

Tablets that fulfil all requirements start at around $500. The lower end of the Samsung Galaxy Tab range is what you'll be looking at. Screen sizes in this price bracket can range from 7 to 10 inches. Storage 16GB, but quite often with expandable storage up to 64GB. Low resolution camera (2-5 MP). CPUs around the 1-2GHz range, but usually dual or quad core. Then above $500, the four brands mentioned are all represented by their various tablet models with various specifications. For testing purposes at least, and most likely deployment, the lower priced models should be adequate as the main features that ASIST relies on such as the GPS are very similar or identical over all price ranges.

The cheapest suitable model would be the Samsung Galaxy Tab 4 8" 16GB 4G, $448 [32]:

- 1.2GHz Quad Core Processor
- 1.5GB RAM
- 16GB Hard Drive
- 3MP Rear Camera and 1.3MP Front Camera
- Android KitKat
- SD Card slot for up to 64GB
- 8 Inch Screen
- 4G Cellular Network Connection

1.7 Alternative Hardware Solution

Another option that wouldn't restrict development language and quite possibly achieve more accurate GPS results is the use of an external GPS.

If accuracy with the above technology isn't enough, then external GPS products that communicate with mobile devices are also an option that should be considered due to their ability to increase accuracy and being able to be positioned away from other utensils that may degrade the device's internal GPS such as magnetic cases. Products are generally in the $100-$300 range.

Examples include:

- Garmin GLO—combines GPS and GLONASS and connects to device via Bluetooth [33].
- GNS 2000 [34].
- SkyPro XGPS160 [35].

At first ASIST should be tested with an un-aided tablet, and only if accuracy of results are insufficient, then should an external device like this be trialled.

Due to this research giving preference towards an Android platform, the next step was to research and test the augmented reality capabilities of native Android development:

6.4 Native Development—Android

Again, as with mobile web development, one of the first steps in the development/testing process is to find, if possible, a satisfactory IDE. In previous years, the tool of choice would have been Eclipse with an Android plug-in. However since December 2014, Google's Android Studio has superseded Eclipse and offers a far more dedicated and elaborate development environment for mobile Android applications [36].

Initial research was again spent looking for available SDKs to simplify the development process. Augmented reality kits such as Qualcomm's Vuforia were found, but with Android Studio only being recently released, documentation to utilise the SDKs were insufficient, so research quickly turned into hands on development and testing with the standard Android APIs [37].

First I required an understanding of the overall framework structure and interaction between different files involved in an Android application. There are four main application components that make up an Android app, each of which serves as a point for the application to be accessed. They are; Activities, Services, Content Providers, and Broadcast Receivers [38]:

Activities

Activities are generally the main component used when developing applications. A single activity represents one screen of the application and any interactions available on that screen [39]. They are independent of one another and are then linked to form a multi-screen application. ASIST will most likely primarily use activities; one for the main live view function, and one for each asset management task.

Services

The opposite to an activity is a service. Services are user interface-less processes which run in the background and do not effect the user's current interaction with the application he/she is using, whether it be the application that this service belongs to, or another application [40]. ASIST may use services to periodically update cached asset and structure data when road works or construction has been conducted.

Content Providers

A content provider is the application's file system, which allows access from other authorised applications [41]. In ASIST's case, using a content provider will allow other Opus software to communicate with ASIST's data source.

Broadcast Receivers

Broadcast receivers take action based on events from; the system, the current application, or other applications [42]. An example of ASIST utilizing this functionality may include initiating a service such as inventory validation once the GPS of the system has achieved an accuracy below a certain threshold.

Once I was familiar with the composition of a native Android application, the remaining development time went towards developing a prototype for one of the core requirements of ASIST, the camera. When first deciding on implementation, the choice between two camera APIs had to be made. With the recent release of Android 5.0 Lollipop, a new camera API, camera2, has been released, deprecating the original camera API, meaning that no future capabilities will be added [43] [44]. Unfortunately due to the main tablet being used in the Opus work force not having an operating system update scheduled in the foreseeable future, I was required to use the older API. This is only a minor issue however and isn't going to be a problem for at least a few years, which then the code can be adapted for adoption by the newer platforms. Additionally by using the older API, many advantages exist, including the availability of far more documentation and examples.

The camera application was made from two java classes; one extending Activity, and the other extending the SurfaceView class, along with a hand full of xml files to set up the interface:

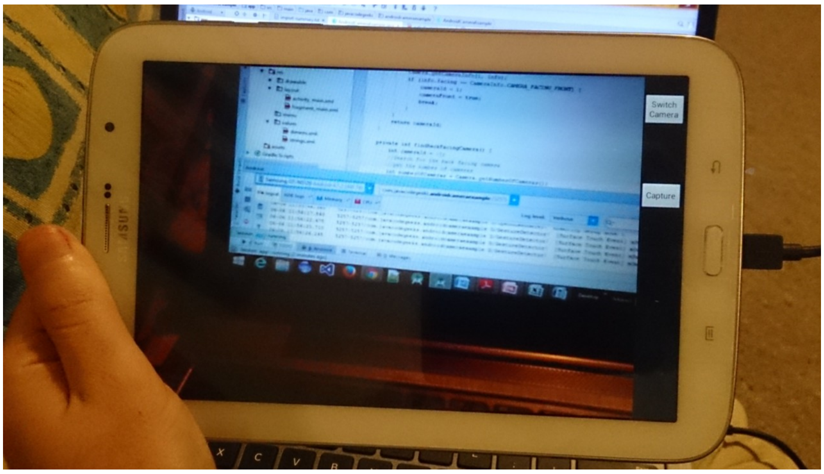


Fig 12. Self Developed Android Camera Application

This solution was achieved easier than the web implementation, as was expected due to my development capabilities, as discussed next.

7. Developer Capabilities

Over the past thee and a half years I have been exposed to a wide variety of programming languages of various levels, utilising various frameworks, and of various domains. The majority of programming work completed to date, and hence strongest and most understood language is Java. Due to native Android development being primarily Java based, this skill set lends itself to this development path. A considerable amount of time has also been spent investing into HMTL5 and JavaScript over the past six months as well to give the multiplatform approach a fair trial. This includes website development study as well as opting to complete a recent mobile application assignment with web technologies, relating directly to the ASIST project due to it's use of location and spatial data. Knowledge of SQL and database principles also exist if required to manipulate the data involved in this project.

8. Recommendations Moving Forward

It is on the back of these findings that the decision to go ahead with a native Android application has been made. The advantage of cross platform compatibility that a web based solution would bring is enticing, but is ultimately outweighed by the increased development efficiencies that will be observed moving forward with the project due to my personal experience and capabilities. In addition to this, the vast majority of tablets already deployed throughout Opus are Samsung Android devices, so a cross platform solution isn't strongly required. Discussions with Kodie and other members of the development team support this decision, with another member of the team also being confident with Java development.

The first task of development will be testing the accuracy of a purely location based implementation using the GPS, compass and accelerometer. To save time this will be conducted with self generated GPS coordinates, rather than using real data provided by Opus due to a transformation of the data into GPS coordinates being required before being usable.

When accuracy is determined to be acceptable, development of data representation will begin. Current ideas as illustrated in images throughout this document suggest some basic representations to use for road surfaces. These will be built on and expanded for a wider range of assets, and to allow for more detail to be disseminated to the user. These descriptions will have to be responsive and be determined on the user's perspective, most likely requiring OpenGL rendering, or another alternative, which will be investigated when required [45].

After these live view components are complete, expansion and iteration of the application can be conducted. This will include working through the use cases in order of priority decided by Opus, and ongoing performance evaluations with involved parties to ensure correctness and usability of the application. It is more than likely that not all functions outlined in the initial documentation will be achievable in the given time frame, so decisions and trade-offs between quantity and quality of functions will need to be made throughout development.

9. Risk Analysis

Due to much of this project being undiscovered territory for both myself and others involved, there are numerous areas for this project to fall over. Here I take note of some of these, and suggest solutions or alternative methods if forced to change our intended approach.

Low accuracy

One of the first hurdles to overcome in phase two is the accuracy of the GPS and other positioning hardware of the device. To simplify development and allow for maximal scaling of the application to different assets, efforts will first be put towards a purely location based solution. There is a real chance however that this will not be sufficient. Two alternatives have been theorised:

The first is the addition of manual calibration. Given a road for example, where an offset in accuracy of a few metres can be the difference between lanes, the user can manually input the centre line of the road by drawing on the touch screen. ASIST will then shift it's projection of the asset to align with this input.

The second alternative relates to a change in data representation. If the device is unable to project the description over the asset accurately, then data should be displayed in a way that doesn't rely on absolute accuracy. This may include a rendering of the asset that is partially independent of the location information. Given that the user is in a specific location and the asset to be analysed is selected, then that object can be rendered on the screen while not specifically looking at the asset. This is a less optimal solution, however given that the user knows what asset they've selected, this approach will still convey the same information.

Project Discontinuation

On Wednesday the 10th of June, the proceeding technical review will be presented to board members to assess the project's viability. It is almost certain that the development of ASIST will continue. However in the worst case, plans for ASIST may be cancelled. Discussions with Kodie have assured me that if this is the case, that there will still be sufficient work for me to complete. This will most likely involve developing a standalone prototype of ASIST.

Asset Data

As illustrated previously, the current structure of asset data isn't directly usable as we require GPS coordinates to plot information. I am expecting to receive the data in a usable form at a later date for testing. However this isn't confirmed, so I may also receive the task of this data conversion. If this is the case, I'll need to talk with current users of this data to aid the process of creating the necessary conversion formulas.

Platform Incompatibility

Despite research thus far suggesting a Java based Android implementation, there are still unknowns to be discovered throughout development. During the process of development I'm likely to have issues creating the requested functionality. The absolute worst case will be discovering that something is just not possible, although this is unlikely. This may then require translating the project into another language. Fortunately, with another Java confident programmer in the team, it shouldn't have to come to this and any trouble I do have can be talked through with him as well as other members of the software development team.

10. Conclusions

As required by Roquito, a review of phase one and plan of action for the months ahead was completed to keep everyone involved up to date and ensure company support of the project. This review covers the majority of the topics covered throughout this document, and hence follows as a summary to this report.

Opus International Consultants Ltd
Christchurch Office
20 Moorhouse Avenue
PO Box 1482, Christchurch Mail Centre, Christchurch 8140
New Zealand

t: +64 3 363 5400
f: +64 3 365 7858
www.opus.co.nz

To	Roquito Lim
Copy	Matt Barnes, Andrew Bruce, Duan Zhao
From	Tony Peek, Kodie Wixon
Date	9 June 2015
Subject	ASIST Technical Review and Phase 1

1.1 Introduction

ASIST aims to ease the process of assessing an asset, such as a road or building, and it's condition while in the field. Through the use of augmented reality, users should be able to easily understand an asset's condition, and make well informed decisions on what actions to take.

Phase one of this project has been spent assessing the possible development paths to take with ASIST. The requirements of the application pose various challenges, which collectively, don't suggest a well defined solution. The purpose of this document is to present the decisions made on how to progress with the application. In doing so, the research and testing conducted to overcome these challenges will be discussed, and different development options compared.

1.2 Development Path

The most important decision to make before development can commence is the platform to be used, whether this be a web based application (HTML5), or a native application for Android, IOS, or Windows tablets. The two most promising options were chosen to be researched; web based, and Android.

1.3 HTML5 and JavaScript Development

Initial meetings and documentation for ASIST recommended a web based implementation. The key benefit of this approach is that the application will be deployable on all mobile devices, regardless of operating system. The first few weeks were spent pursuing the applicability of this solution, including the augmented reality and location based tools available to developers, as well as other current solutions.

One of the main issues with a web based application is gaining access to and manipulating a device's hardware components, which a native implementation has direct access to. Fortunately there are standards available which grant accessibility to the required functions of ASIST, such as the camera, GPS, compass, and accelerometer for positioning. Testing of the platform was only mildly satisfactory however. Location based services such as tracking the user's location and landmarks was achieved, but the combination of this with the camera and other functions weren't achievable during the first stages of testing.

This led to the investigation of other services. Licenses for web based augmented reality kits are available for purchase to ease development, but at the cost of the time required to learn the systems, as well as the various licensing fees.

1.4 Native Development - Android

After mixed results with HTML5, effort was put into understanding Android's capabilities. Again, we have access to all of the required functions and more. Testing was then successful for building a simple camera application. Although testing with location data wasn't completed, we're confident it will be achievable due to native development tending to be more powerful in terms of utilizing this type of functionality.

1.5 Developer Capabilities

Tony has experience with a range of programming languages potentially applicable to this project, with the strongest being Java. Due to Java being the main component of Android applications, this would suggest that a native Android application will be the best option to pursue to minimise time required to learn the platform, and to produce a more robust solution. A combination of tertiary and self study has been conducted in HTML5 and JavaScript lately for a web based solution, but there is a far wider gap between current and required knowledge to complete this project in HTML5 in comparison to Java.

In terms of mobile development in general, he has two games on the Windows 8 and Windows Phone 8 app stores as a result of participating in a Microsoft app development programme, and has recently created a location based application for a university project which doubled as preparation for ASIST. He also has SQL/database experience if required to deal with RAMM data.

The Opus Software development team has more widespread skills in the HTML5/JavaScript space but also has some capabilities programming in Java. The team has not yet developed a native Android application but certainly has the capability to do so.

1.6 Hardware Requirements

Despite asset data not necessarily being completely accurate, the most important feature of ASIST, and what will most likely be the hardest challenge to overcome in the coming months, is the application's accuracy. If a mobile device's GPS is only accurate to five metres, then the application will be useless in the case of a road or other thin/small asset. This resulted in much of the hardware research conducted being focused on GPSs.

Currently the two most common GPS systems in place are the American GPS, and the Russian GLONASS. Most mobile devices utilize the GPS's satellites, while others combine both sets, increasing the number of satellites communicated with, and hence increasing accuracy. Tests show an increase in accuracy of up to 50% when GLONASS satellites are used in conjunction with GPS, compared to GPS alone. Hence a tablet with this technology is recommended above all else.

All other requirements of ASIST; camera resolution, processor speed, memory, screen size etc. are all reasonably minimal, and hence most tablets that include GLONASS integration will also most likely satisfy these other requirements. The only requirement that falls outside these is 3G/4G cellular network compatibility. Fortunately this option is generally available alongside the Wi-Fi only versions.

At the time of writing, the cheapest tablets which satisfy these requirements are the Samsung Galaxy Tab 3 Lite and Tab 4 ranges (Android). Fortunately the current Samsung tablets spread around Opus also satisfy these requirements. Since the current tablets satisfy our requirements, and the fact that alternative platforms such as iPad and Windows tablets cost a considerable amount more than what has been suggested, this is another point of advocacy towards developing purely for Android.

1.7 Existing Products

Regardless of implementation method, other existing applications can be drawn upon for development inspiration. Many forms of augmented reality exist, examples include; waypoints to landmarks in the user's field of view, an overlay of satellites and stars in the sky, and many more. The most comparable project that was come across while researching is Augview. Augview is an augmented reality mobile asset management application that allows users to visualize underground objects such as pipes and cables. Much like ASIST, Augview simply takes the GPS coordinates of an asset, and then displays it on the augmented data layer. A seminar on their product was attended earlier in the year and challenges faced, mainly related to accuracy, were discussed.

Fig 13. Augview AR Application

1.8 Recommendations on how to Proceed (Technology)

Due to the overwhelming experience in favour of Java compared to HTML5/JavaScript of Tony, as well as the assumed needlessness of a cross platform application, it is recommended that development of a native Android application takes place. At this stage, no additional hardware is required to begin prototyping.

A native android application will offer more stability through providing better access to the device hardware and not having to rely on certain browsers and having the correct settings as these can be set specific to the application.

1.9 Next Phase

A simple android application will be developed with the focus on using a simple data set to test accuracy and performance of the application.

The first prototyping conducted will be to assess accuracy of a purely location based implementation. This can begin with self generated dummy data, moving towards real RAMM data.

Once accuracy of the application is sufficient, efforts will be put into constructing the augmented data layer further. How to represent different structures and their condition will be discussed and trialled, with feedback on its feasibility collected from potential users.

This phase will be undertaken with more support from the software development, whereas the initial research and trialling phase was undertaken by Tony for the majority of it.

Once the core functionality has been completed, the process of expanding the application's functions, as well as embedding data sources will begin.

1.10 Student Project Plan

On the assumption that Tony still plays the major role, it will be his responsibility to keep the project on track. He has been assigned 190-240 hours of work towards working on ASIST (and related academic requirements) between now and the end of October. By that stage, a minimum of a working prototype is expected to be completed, and ideally all of the above functionality, as well as that of the use cases outlined in the original ASIST specifications.

References

[1] Opus International Consultants. Retrieved from: http://www.opus.co.nz/about-us/

[2] Schnell, K., Shetterley, N. (2013). Accenture. Understanding Data Visualisation. Retrieved from: http://www.accenture.com/SiteCollectionDocuments/PDF/Accenture-Tech-Labs-Data-Visualization-Full-Paper.pdf

[3] Heim, S. (2008). The Resonant Interface.

[4] Bruce, A. (2014). Big Ideas 2014. ASIST - ASset hISTory in real time. A mobile computing application. pp 10.

[5] Kirkpatrick, M. (2009). Retrieved from: http://readwrite.com/2009/08/27/yelp_brings_first_us_augmented_reality_to_iphone_s

[6] Junaio. Retrieved from: http://www.junaio.com/

[7] Wikitude GmbH. (2013). Wikitude Places. Retrieved from: https://play.google.com/store/apps/details?id=com.wikitude.places&hl=en

[8] Analytical Graphics, Inc. (2011). Satellite AR. Retrieved from: https://play.google.com/store/apps/ details?id=com.agi.android.augmentedreality&hl=en

[9] ESET. (2014). ESET Augmented Reality BETA. Retrieved from: https://play.google.com/store/apps/details?id=com.eset.ar&hl=en

[10] National Geographic. GIS (geographic information system). Retrieved from: http://education.nationalgeographic.com/education/encyclopedia/geographic-information-system-gis/?ar_a=1

[11] Augview. Retrieved from: http://www.augview.net/

[12] Apache Cordova. Retrieved from: https://cordova.apache.org/#about

[13] Rouse, M., Furey, A., Pottjewijd, A. Integrated Development Environment. Retrieved from: http:// searchsoftwarequality.techtarget.com/definition/integrated-development-environment

[14] Intel XDK. Retrieved from: https://software.intel.com/en-us/html5/tools

[15] Krishnappa, R. (2014). Augmented Reality Mobile App with HTML5. Retrieved from: https://software.intel.com/en-us/html5/articles/augmented-reality-mobile-app-with-html5

[16] Wikitude SDK. Retrieved from: https://www.wikitude.com/products/wikitude-sdk/

[17] aCrm Net. Hermes Virtual Tour. Retrieved from: https://www.wikitude.com/showcase/hermes-virtual-tour-surfing-in-the-past-through-augmented-reality/

[18] Layar SDK. Retrieved from: https://www.layar.com/documentation/layar-sdk/

[19] Metaio SDK. Retrieved from: http://www.metaio.com/products/sdk/features/

[20] Schofield, J. (2014). What's the best cheap tablet for GPS use? Retrieved from: http://www.theguardian.com/technology/askjack/2014/nov/13/whats-the-best-cheap-tablet-for-gps-use

[21] Bolduc, T. (2015). GLONASS - The GPS Alternative You Never New Existed. Retrieved from: http://www.makeuseof.com/tag/glonass-gps-alternative-never-knew-existed/

[22] Garmin. What is WASS? Retrieved from: http://www8.garmin.com/aboutGPS/waas.html

[23] List of smartphones supporting GLONASS navigation. Retrieved from: http://en.wikipedia.org/wiki/List_of_smartphones_supporting_GLONASS_navigation

[24] Padre, J. (2012). GLONASS support in our latest Xperis phones. Retrieved from: http://developer.sonymobile.com/2012/01/19/glonass-support-in-our-latest-xperia-phones/

[25] Qualcomm. Retrieved from: https://www.qualcomm.com/products/snapdragon/navigation

[26] Qualcomm. Retrieved from: https://www.qualcomm.com/products/izat

[27] Qualcomm. Retrieved from: https://www.qualcomm.com/products/snapdragon/devices/all? type=Tablet

[28] Comparison of tablet computers. Retrieved from: http://en.wikipedia.org/wiki/Comparison_of_tablet_computers

[29] Apple. iPad Air. Retrieved from: https://www.apple.com/nz/ipad-air/specs/

[30] iPad Pilot News. (2013). Do I really need an external GPS? Retrieved from: http://ipadpilotnews.com/2013/02/do-i-really-need-an-external-gps-2/

[31] Burkert, C. (2013). iPad Air - GPS / GLONASS. Retrieved from: https://discussions.apple.com/thread/5550207

[32] Harvey Norman. Retrieved from: http://www.harveynorman.co.nz/computers/tablet-and-accessories/tablets/samsung-galaxy-tab-4-8.0-wi-fi-lte-16gb.html

[33] Garmin. Retrieved from: https://buy.garmin.com/en-US/US/prod109827.html

[34] Amazon. Retrieved from: http://www.amazon.com/gp/product/B00COLB9FS/ref=oh_details_o04_s00_i00?ie=UTF8&psc=1

[35] Amazon. Retrieved from: http://www.amazon.com/Dual-Electronics-XGPS-160-Receiver/dp/B00E65TNYE

[36] Android Studio. Retrieved from: http://developer.android.com/tools/studio/index.html

[37] Vuforia. Retrieved from: https://www.qualcomm.com/products/vuforia

[38] Android Application Fundamentals. Retrieved from: http://developer.android.com/guide/components/fundamentals.html

[39] Activities, Android. Retrieved from: http://developer.android.com/guide/components/activities.html

[40] Services, Android. Retrieved from: http://developer.android.com/guide/components/services.html

[41] Content Providers, Android. Retrieved from: http://developer.android.com/guide/topics/providers/content-providers.html

[42] Broadcast Receiver, Android. Retrieved from: http://developer.android.com/reference/android/content/BroadcastReceiver.html

[43] Camera2, Android. Retrieved from: http://developer.android.com/reference/android/hardware/camera2/package-summary.html

[44] Camera, Android. Retrieved from: http://developer.android.com/reference/android/hardware/Camera.html

[45] OpenGL. Retrieved from: https://www.opengl.org/