matthew frost

L1426439

Plant view

Project Proposal

Contents

[Working Title 1](#_Toc473804990)

[Project Description 2](#_Toc473804991)

[Rationale for Choice 2](#_Toc473804992)

[Background Research 2](#_Toc473804993)

[Web application 2](#_Toc473804994)

[Android Application 3](#_Toc473804995)

[Integrated Development Environments (IDE) 3](#_Toc473804996)

[Areas for investigation 4](#_Toc473804997)

[Libraries 4](#_Toc473804998)

[Data storage 4](#_Toc473804999)

[Literature Review 4](#_Toc473805000)

[Books 4](#_Toc473805001)

[Articles 4](#_Toc473805002)

[Websites 4](#_Toc473805003)

[Methodology 5](#_Toc473805004)

[Research Ethics 6](#_Toc473805005)

[Project Plan 6](#_Toc473805006)

[Deliverables 7](#_Toc473805007)

[Web application 7](#_Toc473805008)

[Web service 7](#_Toc473805009)

[Android application 8](#_Toc473805010)

[References 8](#_Toc473805011)

# Working Title

Plant View

# Project Description

I will be developing an Android application intended to be used on a tablet. The application will make use of the tablets camera, GPS and Wi-Fi capabilities. It is intended to be used on an industrial site to provide the user with information on the equipment and vessels within their immediate area. The camera will be used to show the user what the tablet can see and also provide a user interface for selecting information. Once an item has been selected the application will be able to show the user relevant information about that item that they would want to know. This could include things such as current temperature and the temperature history over time. With this information the user could then decide if any immediate action needed to be taken. The camera can also be used to scan QR codes to make it easier to select an item if there are a lot in a small area.

In order to be able to mark the locations that the Android application will pick up, a separate web application will be developed to put each point on a map. It will use google maps to allow the user to see exactly where they are placing the point. Once a point has been placed it can be named and it will be stored in a database along with the latitude and longitude. The web application will also work on a tablet, allowing the user to walk around and plot new points as they see fit.

# Rationale for Choice

The project idea was provided to me by a local company called Sabisu who employed me during my year in industry. Towards the end of my placement I was given a project specification of what they wanted. Since then I have been in to speak with my manager to further flesh out the specification and talk about technical details and challenges I would face. I have also been having bi-weekly meetings with my project supervisor from Sabisu to discuss progress and next steps in the project.

Android has also been a passion of mine for a couple of years so I am looking forward to be able create a fully-fledged professional Android application, that will be used by people and be able to help make their jobs easier. Developing the app will build on my knowledge from personal projects as well as the work done during the Enterprise Project in second year. It will also allow me to explore new aspects of Android that I haven’t tried before such as reading data from a web service and using the devices compass to find the users orientation. Furthermore, Android is the only suitable choice as there are only two main competitor operating systems in mobile devices, iOS and Android. iOS is not suitable as it normally requires a machine running Apple’s OSX and the devices to run the application on aren’t as common as Android devices, especially in the chemical industry where the devices on plant need to be intrinsically safe.

Finally, as the server side code will be written in Node JS this will give me the opportunity to learn a framework that I have been wanting to try for a while. Also as Node JS is a very up and coming technology I feel that it would help with my job prospects if I can prove that I can use it effectively.

# Background Research

## Web application

When deciding what language to use for the server side code of the web application there is a large range of languages to investigate, for example Python, PHP, Java or Ruby. However, .NET and Node.JS were the two chosen to investigate. .NET was chosen as it is a framework that I am familiar with and I knew could get the job done. The reason for choosing Node.JS is that I’m familiar with using JavaScript in a client side setting and thought it would be interesting to see how it works server side.

If .NET is chosen there would be a number of advantages. Since it’s a language that I’ve learnt both at university and on placement it would be relatively simple and quick to develop. This would be further increased by the templating that Visual Studio could provide, meaning that there wouldn’t be that much code to write. Also the majority of Sabisu’s applications use .NET for their server side code so if the application was to be adopted by them it would be easy to maintain.

However, if Node.JS was chosen it would be a good learning experience for me and would also build on my knowledge of JavaScript. Furthermore, it would be a more suitable language for this task. As the web app is only a small application for adding points to a database a large .NET app with all its dependencies on Entity Framework etc. it could be a bit too cumbersome. Also if for some reason the application did need to be scaled up or down it could be easily done so if it was deployed to something like Docker (Docker, 2016).

## Android Application

Again for the Android application there are a number of methods can be used for development. Most Android applications are written in Java and then run on the Java Virtual Machine (JVM) within Android, apps can also be written in C or C++ using the Android Native Development Kit (NDK). Apps written with the NDK tend to perform better as they are run directly by the OS instead of using the JVM. However, this application would not benefit from using the NDK and the unfamiliarity with C/C++ would hinder rather than help me.

Even excluding the NDK there are still multiple choices that could used. Soon after the project brief was given the head of Sabisu suggested looking into using the Unity engine (Unity Technologies, 2017) and the augmented reality libraries available for it. After some research I found the Vuforia (PTC Inc, 2017) library which uses image recognition display items on the devices screen. However, after developing a test application it was found that it wasn’t good for recognising simple shapes and requires knowledge of the image needs to recognise which would not be feasible in an industrial setting. So for this reason, Unity and Vuforia were ruled out.

After ruling out the previous technologies this left using Java and the Android Software Development Kit (SDK). Java is a language that I am very familiar with having used it on both the desktop and for creating Android applications. Also it is the main language of choice used within the Android development community so there is plenty of support online and libraries to use.

However even after deciding to use Java and the Android SDK there was one more language found to use, Kotlin. Kotlin is a language that runs on the JVM so has complete interoperability with normal Java. The advantages of Kotlin is that it removes some of the nuances of Java such as Null pointer exceptions, the large amount of boilerplate code. Due to the strong interoperability with Java it means that there is still access to existing Android libraries and can write sections of code in Java if needed.

## Integrated Development Environments (IDE)

To develop the Android application, Android Studio will be used as it is the official IDE provided and supported by Google. It includes features such as being able to create an Android emulator, pushing builds to a device and instant code run.

For everything else Visual Studio will be used as it allows compilation of web projects and the bundling of scripts using Microsoft’s Razor engine. It can also act as a text editor for Node.JS and still provide features such as intellisense.

# Areas for investigation

## Libraries

One of the features intended for the Android application is to be able to read QR codes that could be attached to pieces of equipment. This requires finding a QR code scanning library for Android and then integrating it into my app.

Also in order to be able to show historical data within the Android application a graphing library needs to be found that will clearly show data that has been retrieved.

## Data storage

Initially the application was going to read data directly from IP21 which is a data historian used by Sabic who own the plant. However, it has since been discovered access to this will not be possible, and even if it was, it would be difficult to demo the application outside of Sabic’s network. Therefore, a method of storing data needs to be found and since there is potential for large amounts of data SQL may not be the best solution. SQL is being used for storing the location of the points as there will be a relatively small amount of data stored for them even if a lot of points are added.

# Literature Review

## Books

*Professional Android 4 application development (Meier, 2012)*

Although this is a slightly older book, it provides the fundamentals for Android development.

*Android for programmers: an App-driven approach* (Deitel, Deitel, and Wald, 2015) *[4]*

This being a more modern book should provide some new features that could build on the basic fundamentals. It is based on Android 6.0 which currently has the largest market share for Android devices (Android Statistics, 2017)

## Articles

*Kotlin for Android Developers (Leiva, 2015)*

An article highlighting some of the key features of Kotlin, it includes code samples to easily get the point across and see the implementation.

*Android Data Binding Library: A Blitzkrieg (Ladwa, 2017)*

Data binding in Android is a library that I have used previously and has proved to be very powerful and useful. This article shows how to set it up and integrate it with my Java classes.

## Websites

*Android Compass Example (LIutin, July 2015)* [8]

A GitHub repository containing an example of how to use the compass on Android. Getting the location of a device is easy but getting the direction being faced is a bit more complex

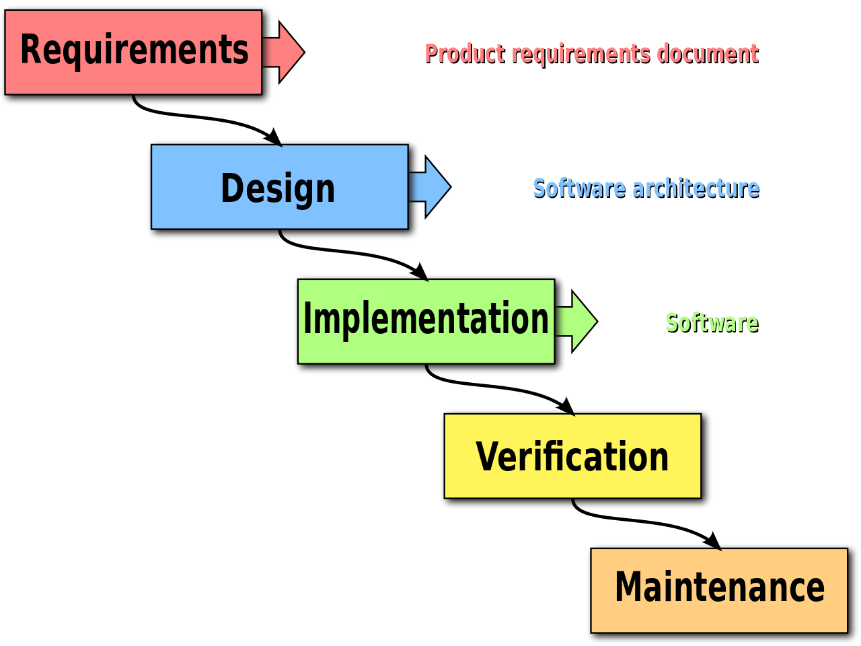
*Kotlin (Jetbrains, 2017)* [9]

The website for the Kotlin programming language, it has the official Kotlin reference guide as well as tutorials and getting started guides.

# Methodology

When planning a project there are a range of methods and techniques that can be used to most effectively plan out what needs to be done and when it should be done by. Project management methodologies can be split up into a number of groups, these include sequential, agile and change management(*Project Methodologies*, no date).

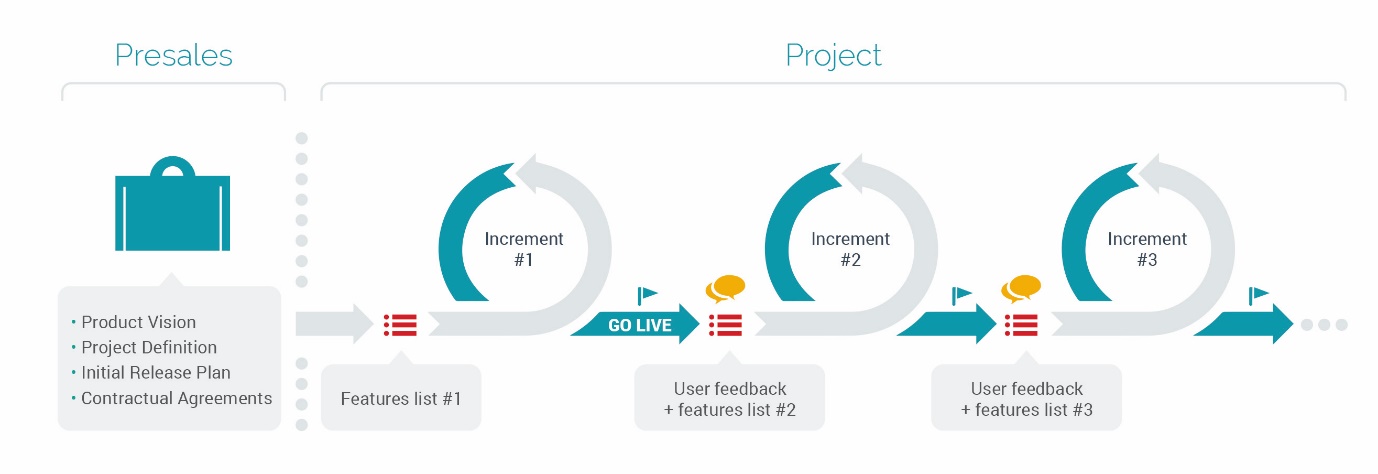
An example of a sequential methodology is waterfall. Waterfall means that each section is fully completed before moving onto the next step and stops backwards steps. This means that ideally each step will be perfect in order to move onto the next one. However, as I am working with a real client this is not always the case as requirements often change. So as the waterfall methodology doesn’t fit around revisiting previous stages I don’t think it would be a suitable strategy.



*Waterfall methodology*

An example of an agile methodology is Scrum. Scrum involves splitting up the project into small tasks sometimes known as story points, this helps the members of the project easily see what has been done and what is left to do. Scrum can also be combined with another methodology called Kanban (DeGrandis et al., 2010), which utilises story points and puts them into groups such as “to do”, “development”, “testing” and “done”. Another key feature of Scrum is the use of sprints; sprints are a short period of time normally one or two weeks where a chunk of development work is done. After a sprint the current stage of the project is shown to the customer and reviewed, this keeps engagement with the customer and allows for any changes to be made in a future sprint. By keeping in contact with the customer this makes sure that they end up with exactly what they want and they can see how much progress is being made.

As I have a live customer I think that Scrum is the best methodology to use, sprints will be one week long as this fits in with my weekly meeting with my project supervisor at Sabisu. Since I am the only developer working on the project there will not be a potentially shippable product at the end of each sprint, but a reasonable segment or feature should have been implemented after each sprint.



*Scrum sprint cycle*

# Research Ethics

Since this project involves a real client it is important that the British Computer Society code of conduct (British Computing Society, 2017)is adhered to. The section that would apply specifically would be “professional competence and integrity” meaning that I should only undertake work that I think I am competent of and I should be willing to accept criticisms and alternative viewpoints.

As my application is intended to be used within the chemical industry it could be reporting some critical information to the user. However, as the application is only reading the information from a data source that has been inputted by another system I don’t think that my application has any ethical issues in this regard.

# Project Plan

|  |  |  |  |
| --- | --- | --- | --- |
| Week Number | Week Commencing | Task | Deadlines |
| 1 | 30/01/2017 | Finish work on web application | - |
| 2 | 06/02/2017 | Testing of web app and start on C# web service | - |
| 3 | 13/02/2017 | Completion of C# web service and testing | - |
| 4 | 20/02/2017 | Start of Android Application – basic camera and location functionality. Start of review poster | - |
| 5 | 27/02/2017 | App development – compass working. Finish poster, start report | Poster review |
| 6 | 06/03/2017 | App development – connecting with Node JS web service & C#.  Report on going | - |
| 7 | 13/03/2017 | App development – UI implementation.  Report on going | Embedded Systems ICA 1 |
| 8 | 20/03/2017 | App development – showing data on UI.  Report on going | - |
| 9 | 27/03/2017 | App development – adding graphs.  Report on going | - |
| Easter Week 1 | 03/04/2017 | App development – QR scanning.  Report on going | - |
| Easter Week 2 | 10/04/2017 | App testing, rework from testing. Report on going | - |
| Easter Week 3 | 17/04/2017 | Rework from testing | - |
| 10 | 24/04/2017 | Report | - |
| 11 | 01/05/2017 | Report and final checks on all aspects of project | - |
| 12 | 08/05/2017 | Review of report | Report and item submission |
| 13 | 15/05/2017 | Project Viva 1 | Embedded Systems ICA 2 |
| 14 | 22/05/2017 | Project Viva 2 | - |

# Deliverables

## Web application

A web application used to add points onto a google map view and store them into an SQL database. This is used by the app to get information on what points are near the user and then pull relevant information about each point.

#### Minimum Requirements

* Should be able to add points to the map and pass them to the web service
* Should be able to read the points back from the web service and put them onto the map
* Should be able to delete points from the map
* Should be able to edit the name of points

## Web service

In order for both the web and Android applications to retrieve data from the data sources a web service will need to be developed. Depending on the architecture used this might result in separate web services, each one deployed separately to improve scalability.

#### Minimum Requirements

* Should pull location points from the SQL database
* Should be able to save location points to the SQL database
* Should be able to Edit location points in the SQL database, this includes renaming and marked as deleted
* Should be able to pull data from at least one data source to return to the Android application

## Android application

This is the main focus of the project; the app will communicate with the web services to retrieve information on key areas in the immediate vicinity of the user. These areas can then be further investigated within the app in order to see current data as well as historical data to provide context. The purpose of the app is to allow plant workers to sport any anomalies while they are doing their checks, they can then radio back to someone in the control room or ideally flag it up in the app so that someone at a computer can investigate it further.

#### Minimum Requirements

* Should be able to show the camera view on the screen
* Should be able to get the user’s current location and direction being faced
* Should be able to get the points closest to the user
* Should be able to display current data of a location as well as previous data

# References

1. *Project Methodologies* (no date) Available at: http://1.https://www.wrike.com/project-management-guide/methodologies/ (Accessed: 31 January 2017)
2. DeGrandis, D., Schumann, M.W., Marschall, M., Frankowski, H., Victor, Hut, P., Vandewiele, T., Naseer, A., Kianrad, M. and Yordanov, L. (2010) *Scrum vs Kanban*. Available at: http://www.agileweboperations.com/scrum-vs-kanban (Accessed: 31 January 2017)
3. Meier, R. (2012) *Professional Android 4 application development*. 4th edn. Indianapolis: John Wiley
4. Deitel, P., Deitel, H. and Wald, A. (2015) *Android 6 for programmers: An App-Driven approach*. United States: Prentice Hall.
5. *Android Statistics* (2017) Available at: https://developer.android.com/about/dashboards/index.html (Accessed: 31 January 2017).
6. Leiva, A. (2015) *Kotlin for Android developers - JVM advent*. Available at: http://www.javaadvent.com/2015/12/kotlin-android.html (Accessed: 31 January 2017).
7. Ladwa, A. (2017) *Android data binding library: A blitzkrieg*. Available at: https://medium.com/@ladwa.aditya/android-data-binding-library-a-blitzkrieg-504fc4462352#.6cqrwdpqx (Accessed: 31 January 2017).
8. Lutin, V. (2015) *Compass*. Available at: https://github.com/iutinvg/compass (Accessed: 31 January 2017).
9. JetBrains (2017) *Kotlin programming language*. Available at: https://kotlinlang.org/ (Accessed: 31 January 2017).
10. Docker (2016) *Docker*. Available at: https://www.docker.com/ (Accessed: 1 February 2017).
11. PTC Inc (2017) *Vuforia highlights*. Available at: https://www.vuforia.com/ (Accessed: 1 February 2017).
12. Unity Technologies (2017) *Unity - game engine*. Available at: https://unity3d.com/ (Accessed: 1 February 2017).
13. British Computing Society (2017) *BCS Code of Conduct* Available at: http://www.bcs.org/category/6030 (Accessed: 1 February 2017).