# Background, Analysis & Process

## Project Overview

The NHS owns a database containing monographs for injectable medicines; each monograph contains useful information such as method of administration, preparation of the drug and flushing guidelines. As well as monographs for each medicine the database also contains values needed for calculating dosage and infusion rate for the medicines.

NHS Wales requested Aberystwyth University via the Software Alliance Wales scheme to provided them with a mobile application, to utilise this data to aid their staff in administering injectable medicines. The NHS provided access to the database via multiple XML API URLs.

The aim of this project was to fulfil that request by creating a well designed, functioning and thoroughly tested Android mobile application. The completed application had to query the data to allow user to quickly and efficiently find monographs. Upon finding the wanted monograph the application has to neatly present the monograph to the user. As some medicines also contain data on calculating dosage and infusion rates, the application had to utilise that data and allow the user to enter patient information (weight and needed concentration) to calculate dosage or infusion rate for that patient.

An Internet connection may not be available in all areas of a hospital, to allow the application to be used whenever needed, the application has to be built for offline use, to achieve this a complete copy of the database has be stored on the users device, thus allowing them to utilise the data when no Internet connection is available.

As to improve the maintainability and customisability of the system the NHS also requested that the structure of the data to be outlined within XML files, thus allowing them to create multiple applications for a variety datasets using the same application code.

As the application will be used to administer potentially lethal drugs, testing had to be executed thoroughly. Therefore a major part of this project was testing the finished application, ensuring that only the correct and most accurate information is displayed to the user.

## Background

Within this section I will outline the all background research I made before starting this project. This will include a list of the available technologies I could have used and an overview of what technologies already exists that complete similar goals to this project.

## .1. Medusa website

The Medusa website is the site that NHS staff currently use to view and print monographs for injectable medicines. The website requests the member of staff to login using their NHS credentials, upon logging in a user is able to select a drug from a drop down list (suggestive search is not available). Once a drug has been selected the user is displayed with the monograph for the selected drug. If calculator information is available then a button to open the calculator is also displayed.

The Medusa website does not work on the mobile devices that I tested it on (the login functionality fails) and the site is not optimized for the screen size of most mobile devices. As the Medusa website does not work effectively on mobile devices the only method of accessing the monographs in a portable manner is to print the monographs, this not an optimal solution as the printed information may also become outdated and the user will not know; this method is also not environmentally friendly.

Using the Medusa website allowed me to see how a monograph should be displayed to the user. It also allowed me to see the shortfalls and drawbacks of the current system, such as the lack of a suggestive search functionality, which allowed me to change the projects requirements to provide a solution to these issues.

## 1.2.2. Platforms and frameworks

Due to the time constraints of this project, one of the major decisions for the project was which framework or platform should be used to allow the application to be compatible for the largest amount of users.

A solution that would have allowed the application to run on the majority of devices would have been to create a web application. A web application is essentially a website that has been optimized for a mobile device. The major issue with a web application was that there is no way to store data on the device persistently therefore a network connection would have been needed. As the application has to work in offline mode, a web application was not a possible solution.

Phonegap is a mobile development framework that allows developers to write mobile applications using HTML5, JavaScript and CSS3 instead of device specific languages. Phonegap then compiles the application written using web technologies into a hybrid application for multiple devices. A hybrid application is an application that appears to be a native application to the user but is not a native application as instead of using the devices UI framework the application uses web views to display information to the user. The framework also gives developers access to the devices local storage and sensors, thus allowing developers to create web applications with similar powers to native applications.

Phonegap would have been an excellent framework to use for this project, as it would have allowed me to create applications to be used on multiple devices, using languages I was very familiar with. The main issue with Phonegap is that it does not allow you to run processes in the background; therefore the process of downloading and updating the local database would need to run in foreground, which is not an ideal solution as the process takes several minutes and the user is likely to move out of the application while the download is in progress. Another issue with Phonegap is that as the application does not utilize the devices UI frameworks the application may lack the look and feel of a native application, meaning it may be harder for a user to use the application.

Native Android and iOS applications would have been suitable for the project; they both have the ability to download the data over HTTP, parse the XML files, execute tasks in the background and save data for offline use within local databases.

Applications for iOS devices are created in Xcode IDE and are written in Objective-C, which is an object-orientated language based on C and C++. I have some experience in C and C++ but no experience in Objective-C or iOS development. I also do not own an iOS device therefore testing would have been primarily executed within an emulator.

Android applications are written in Java which is an object-orientated language using either Eclipse or Android Studio IDE’s. I have had a large amount of experiencing writing applications in Java, but had no experience in Android development. I also have two Android devices, which would allow me to test the application on a live device rather than an emulator.

Android currently has the largest percentage of market share, having 78% of the market share in 2013 [1]. Therefore developing the application as a native Android application will allow the application to be used by the most users. Due to this statistic and that I own Android devices resulted in me choosing to develop the application as a native Android application.

## 1.2.3. Learning Android development

As I had very little experience in Android development before starting this project during the background work I also had to teach myself Android development.

I began this process by refreshing my Java skills by reading over previous Java projects and writing basic applications. I then followed the tutorials found within the training section of the Android documentation, this helped me setup the Android SDK and begin building applications. I also read the Android design guidelines, which helped me to better design the project.

I then built small prototypes for major sections of the final application. This allowed me to spike any parts of the final application I was unsure about whilst continuing to learn Android.

## 1.2.4. Existing works

From the research I have executed there is currently no other mobile application that completes all goals of this project, but there are many libraries, frameworks and tools that I used throughout the project.

The Java programming language provided me with a great amount of useful functionality allowing me to complete tasks such as downloading data using HTTP requests, parsing XML into usable data and the ability to write the downloaded data onto the devices internal storage.

The Android SDK allowed me to create a native Android application that has the ability to utilise all hardware on the device. The Android SDK also provided me with the base UI framework, allowing me to create an application that an Android user can instinctively use.

Robospice is a library for Android that is released under the Apache licence. Robospice simplifies the process of making asynchronous network requests in the background whilst continuously notifying the UI thread of progress. I used the Robospice library to allow the complete database download to be executed in the background as an Android service, whilst still updating the user interface showing progress to the user.

Another resource I used whilst completing this project was StackOverflow. StackOverflow is an online community where users post programming related issues and the community help solve these issues. I used StackOverflow when I encountered issues that I believed would be a common issue or when best practices for a solution were unknown by myself. I never posted any issues of my own, just read other users posts.

Genymotion is an application that creates and runs emulators for a variety of devices running Android. In my experience Genymotion is quicker and less error prone than the standard Android SDK emulator. I used Genymotion as it allowed me to test my application on multiple devices without requiring me to setup each device individually as I would have had to with the standard emulator.

## Analysis

Within this section I will outline the objectives of the project and state how background work helped reach these objectives.

**Build a native Android application using the Android SDK**

Building a native application for each of the major platforms would have been the ideal solution for this project, but due to the time commitments for this project it would only be possible to create one application. After completing the initial background work I decided that I would be developing a native Android application, as this approach would allow the application to be used by the most users.

**Use the user interface framework provided by the Android SDK**

Using the user interface framework provide by Android allowed me to create an application that a user already familiar with the Android OS would be able to use with ease. This also allowed me develop the interface quickly instead of having to design each element before hand.

**Easy to use user experience**

The NHS staff that will be using the application might not be technically minded therefore the application must be easy to use for people with little technical knowledge. I achieved this by following the design guidelines within the Android documentation, which I read during the background research for the project.

**Download the database so the application can be used without an active Internet connection**

As the radio waves used to transmit data over Wi-Fi or the mobile data network can effect medical equipment it is vital the application runs perfectly in airplane mode, through background research I found that the best way to achieve this was to store all data needed for the application on the devices local storage. Using this method the user can download the database when they have an Internet connection and then continue to use the data when they’re without an Internet connection.

**Thoroughly test the application throughout all stages**

As the application is used to administer drugs to real patients it is vital that all aspects of the application are thoroughly tested. Therefore a test-driven development approach was taken towards classes that output life critical information. I believe once the application is given to the NHS they will also execute their own tests, but by providing my test data and documenting these test should greatly improve their confidence in my application

**Ensure the application only uses commercially licenced libraries**

As the application will be used by the NHS all libraries and resources used must not be for personal use only and therefore should contain licences