NHS Wales - Injectable medicines guide Android application

Final Report for CS39440 Major Project

Author: Aidan Wynne Fewster (awf1@aber.ac.uk)
Supervisor: Dr. Andrew Starr (aos@aber.ac.uk)

May 2014
Version: 1.0 (Release)

This report was submitted as partial fulfilment of a BSc degree in Computer Science (G400)

Department of Computer Science
Aberystwyth University
Aberystwyth
Ceredigion
SY23 3DB
Wales, UK

Declaration of originality

In signing below, I confirm that:

- This submission is my own work, except where clearly indicated.
- I understand that there are severe penalties for plagiarism and other unfair practice, which can lead to loss of marks or even the withholding of a degree.
- I have read the sections on unfair practice in the Students' Examinations Handbook and the relevant sections of the current Student Handbook of the Department of Computer Science.
- I understand and agree to abide by the University's regulations governing these issues.

Signature .....

Date .....

Consent to share this work

In signing below, I hereby agree to this dissertation being made available to other students and academic staff of the Aberystwyth Computer Science Department.

Signature .....

Date .....

Acknowledgements

I am grateful to...

I’d like to thank...

Abstract

Include an abstract for your project. This should be no more than 300 words.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Background, Analysis and Process

This chapter contains evidence of the background work that was completed before the project design was created. During this stage in the process existing works were considered, an analysis of the problem was complied, methodology chosen and a plan was created.

1.1 Project Overview

The NHS [18] owns a database containing monographs [17] 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 [18] requested Aberystwyth University via the Software Alliance Wales scheme [23] 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 [26] API URLs.

The aim of this project was to fulfil that request by creating a well designed, functioning and thoroughly tested Android [4] mobile application. The completed application had to query the data to allow users 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 [8] 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 [26] 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.

1.2 Background and Analysis

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

1.2.1 Medusa website

The Medusa [9] 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 [17] for the selected drug. If calculator information is available then a button to open the calculator is also displayed.

The Medusa website [9] 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 [9] does not work effectively on mobile devices the only method of accessing the monographs [17] 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, also this is not an environmentally friendly method.

Using the Medusa website [9] allowed me to see how a monograph [17] 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 [27]. A web application [27] 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 [8], a web application was not a usable solution.

Phonegap [19] is a mobile development framework that allows developers to write mobile applications using HTML5, JavaScript and CSS3 instead of device specific languages. Phonegap [19] then compiles the application written using web technologies into a hybrid application [28] for multiple devices. A hybrid application [28] 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 [19] 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 [3]. 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 [19] 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 [4] and iOS applications would have been suitable for the project. They both have the ability to download the data over HTTP, parse the XML [26] files, execute tasks in the background and save data for offline use within local databases.

Applications for iOS [7] devices are created using the Xcode IDE and are written in Objective-C, which is an object-orientated language based on C and C++ [21]. 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 IDEs. 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 [5]. 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 [4] 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 [16] 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 [6] and begin building applications. I also read the Android design guidelines [13], 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 executed it was concluded that there is currently no other mobile application that completes all goals of this project, but there are many libraries, frameworks and tools that were used throughout the project.

The Java [16] programming language provided the application with a great amount of useful functionality allowing the applicaton to complete tasks such as downloading data using HTTP requests, parsing XML [26] into usable data and the ability to write the downloaded data onto the devices internal storage.

The Android SDK [6] allows developers to create native Android applications that have the ability to utilise all hardware on a users device. The Android SDK [6] also provides the base UI framework, allowing developers to create applications that an Android user can instinctively use.

Robospice [22] 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 [22] . The Robospice [22] library has been used 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.

Glyphicon [14] is an open source free to use iconography package. This project uses one icon from the iconography set to display an information button.

Another resource that was used whilst completing this project was StackOverflow [24]. StackOverflow is an online community where users post programming related issues and the community help solve these issues [24]. StackOverflow was used when an issues was encountered 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 [10] 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 [6] 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.

1.3 Objectives

Within this section I will outline the objectives of the project and state how background work helped reach these objectives, I will then state the final list of functional requirements used for the project.

1.3.1 Original 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 constraints for this project it would only have been possible to create one application. After completing the initial background work I decided that I would be developing a native Android [4] application, as this approach would allow the application to be used by the most

users [5].

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 [8] 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 theyre 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 [15]. I believe once the application is given to the NHS they will also execute their own tests, but providing my test data and documenting these test should greatly improve their confidence in the application.

Implement the application using only publicly licenced libraries and resources As the NHS will use the application all libraries and resources used must not be for personal use only and therefore should be licenced under publicly free licences such as the Apache licence.

1.3.2 Functional requirements

After building the list of original objectives I contacted the representative of the NHS for this project and through this contact, the list of API URLs provided by them and the initial mock-ups sent through email I was able to compile the following list of sensible functional requirements.

FR 1	**Authentication**
FR 1.1	User must be able to authenticate themselves using their credentials
FR 1.2	User must be notified if the password they enter is incorrect
FR 1.3	User will be notified if the authentication failed due to connection issues
FR 1.4	User must be able to logout of the system, removing all data
FR 2	**Database synchronisation**
FR 2.1	After login or when the user presses update the application must truncate all database tables and begin downloading new data
FR 2.2	Download complete list of drug indexes from database
FR 2.3	Download complete list of drugs and drug informations

FR 2.3	Download all information needed for calculating doses and infusion rates.
FR 2.4	Download must still run when the application is in the background
FR 3	**Menu options**
FR 3.1	Upon pressing the Menu button on the device the user will be presented with a list of available options, which execute tasks (Logout, exit, search)
FR 4	**Main screen**
FR 4.1	Upon successful data download the user will be displayed with a screen where they can navigate to other parts of the application
FR 4.2	User will be see when an update was last performed, and perform an update from this screen.
FR 5	**Browse drugs**
FR 5.1	This screen will allow the user to view a list of all drugs
FR 5.2	There will be an input box on this screen, when the user enters text into the input box the results in the list will be filtered to only show results related to the input
FR 5.3	The user will be able to click a drug in the list to open a new screen displaying the needed information
FR 6	**View drug**
FR 6.1	When a drug has been selected the drug and all its information will be displayed in an easy to read format
FR 6.2	Where drug information headers contain help information, a help icon will be displayed next to the header.
FR 6.3	When heading help icon is clicked the helping information will be displayed
FR 6.4	If the drug had calculator information, then a button to open the calculator should be shown
FR 7	**Browse drugs with calculators**
FR 7.1	A view similar to the browse drugs view will allow the browsing of drugs that contain calculator information.
FR 8	**Calculate dose and infusion rate**
FR 8.1	The user will be able to select calculation type
FR 8.2	User will be able to enter information required for the calculation
FR 8.3	When the calculate button has been clicked the input will be thoroughly validated
FR 8.4	After validation the result of the calculation will be displayed to the user
FR 8.5	The equation and values used to calculate the answer will be neatly displayed to the user
FR 9	**XML customisability for developers**
FR 9.1	All text within the application must be changeable through XML files.

FR 9.2	The structure of the XML APIs provided must be outline within XML files, allowing easy customisation for different APIs

Table 1.1: Table of functional requirements

1.4 Compromises within the functional requirements

Creating a perfect system that met all the functional requirements that were wanted was not possible due to time constraints or API limitations, within this section I will be outlining the functional requirements I would have liked to have added but were not possible.

As mentioned earlier, creating a native application for each major platform would have been ideal for this project, but due to time constraints and the lack of a usable hybrid solution this was not possible.

Having the database synchronise with the live database, instead of deleting the entire database and downloading a new copy would have been a more efficient method of updating the database, as updates would be performed much quicker. Unfortunately synchronising the databases was not possible as the API provided only allowed for downloading of the complete dataset and not partial downloads.

Sending push notifications to the device when a change to the database occurred would have been an excellent way to alert the user that they needed to update their data, but due to lack of access to the server hosting the database this was not possible.

Allowing the user the ability to reset their login credentials should they forget them was also not possible due to API limitations.

1.5 Development process

Within this section I will be describing the methodology that I used whilst completing this project, explaining my method of planning the system and describing the prototypes that I created to help break down the project.

1.6 Methodology

Initially an eXetreme programming [1] methodology adapted for solo programming was the chosen methodology for this project. Using an eXetreme programming approach would have allowed me to create good working software faster than most other methodologies whilst preventing the project from being hacked together.

Within the initial methodology it was planned that I would engage in meetings with the NHS representative to allow us to collaboratively create a set of stories and a detailed releases schedule for the project.

After creating stories for the project the methodology stated to carry out spike work by completing prototypes for predicted difficult stories of the project thus gaining a better understanding of the complexity of the stories. These prototypes could then later be used to demonstrate work completed during the mid-project demonstration.

Once the complexity of all stories had been estimates and spike work had been carried out it was intended that a meeting with the NHS would be organised so that stories could be ranked in order of importance, that that stories with greater importance could be implemented first, thus allowing the application to made useful to the NHS sooner.

As I started working on the project I soon released that an eXetreme programming approach to completing this project would likely lead to failure, this was mainly due to the fact that regular contact with the NHS representative was not possible. I also began to believe that using eXtreme programming [1] might not be an acceptable methodology for a medical system because maintaining a system created by another developer that is not well documented would be hard and easy to create errors.

After realising that eXetreme programming [1] was not the best methodology for this project I decided to move the project to use the waterfall model [2], whilst still using features from eXetreme programing [1] that I believed would enhance the project. Using the waterfall model [2] would allow the project to be completed with as little contact with the NHS as possible whilst enforcing me to create a list a detailed documentation to be delivered along with the project

Using then waterfall the model meant that a detailed analysis of the project would be needed, this was completed whilst carrying out background and analysis work as seen in the previous section and was compiled into the outline specification document [2].

The next stage in the waterfall model was to compile a list of requirements; the list of requirements along with a use case diagram and description was used to create the requirements specification.

Using the functional requirements, prototypes were then created for any requirements that the complexity was unknown for. This was a practice that I had taken from the eXetreme programming methodology because I believe that creating a prototype for a task that seems challenging helps to simplify that task.

Once the prototypes had been built for the major parts of the system the next step was to design the application. For this I followed the waterfall models method of design and created a design specification containing user interface design, UI mock-ups and class structure and design using UML diagrams with appropriate descriptions [2].

After the design process for the application was complete implementation began. Implementation followed the structure laid out in the design specification and used the prototypes created to build a working well-engineered Android application.

Test-driven development was used for parts of the system that outputted important information. Test-driven development is a practice that I integrated into my methodology from eXetreme programming. Using test-driven development for critical parts of the system ensured that those parts of the system were well tested.

During the implementation stage the continuous integration practice from eXetreme programming was used so that the applications was always in a state where it could be compiled and ran on a device. This motivated me to continue developing the application as I could see the application

gradually becoming more useful.

Another practice that was taken from eXetreme programming [1] was the use of coding standards. Throughout the implementation and testing process all code followed the Android developer coding standards and JavaDocs comments were created.

The final step in the waterfall model [2] is to verify and test the implemented application. During this stage the parts of the applications that remained untested were tested and further testing was carried out on the already tested parts. The application was then sent to the NHS representative for acceptance testing.

1.7 Planning

Planning is vital to the success of a project and was inherently brought to the project through the waterfall method [2]. Both the analysis and design stages of the waterfall model are planning stages therefore completing this stages ensured the project was well planned.

After the analysis stage of the project and a full list of dates for the project had been published, I produced a Gantt chart plan for the project time span. I set out the Gantt chart plan to fit with the academic calendar for the module, which ensured I had a large amount of work complete before the mid-project demonstration. I also planned the Gantt chart so that to allow for 3 weeks of spare time at the end of the project. This extra time was reserved in case I encountered any unseen problems during any stages of the project, meaning I wouldnt have to worry about time constraints should this have happened.

Throughout the development process the projects progress was compared to the Gantt chart plan, which allowed me to keep track of how far ahead or behind the project was with the timeline.

During the time when I had intended to begin implementing the application the representative for the NHS was unavailable and I had to wait until they were back in the office until I could further progress the project. Having already planned for an unexpected event happening such as this greatly reduced the impact on the project.

1.7.1 Prototypes

Before starting the design stage of the waterfall model [2] I wanted to improve my Android skills whilst still enhancing the project. To do this I decided to build multiple prototypes that complete small tasks for the final implementation. This section will describe each prototype made and what they achieved.

NHS Prototype Package A Java [16] package containing common models that would be used between prototypes was created, this ensured that the prototypes would work together flawlessly once implemented into the final application whilst also making the code written DRY.

View Drug Prototype The main purpose of this prototype was to display a drug to the user. All information for the example drug was hard coded into the prototype. The prototype displayed the example drug and all its information onto the screen dynamically. This prototype taught me how to use the Android layout inflator, which was used to display the drug infor-

mations. This prototype also taught me how to implement a view that scrolls when the view size exceeds the screen size.

Calculator prototype The purpose of the calculator prototype was to create an application that would accept user entered information and provide the user with an answer to a calculation. At this stage in the project I had not been provided with any information on how the NHS performed their calculations, so the equations used were made up. Although this prototype was never used in the final application, completing this prototype taught me how to accept and validate user input.

Download data index prototype The purpose of this prototype was to download the index data from the drug indexes XML [26] API and then return an array of drugs created from the index. The application begins by asking the user to enter their login credentials and then upon successful authentication the application begins the download of the indexes. The login section of this activity was used in the final application but the actually downloading functionality was not used. Whilst completing the prototype I learn how to receive data over HTTP and how to parse XML within Android. I also discovered several bugs in the XML API provided by the NHS whilst building this prototype, as this was very early in the development I was able to notify the NHS representative to have these bugs fixed.

1.8 Version control and backup

Due to the size of this project, it was vital that a version control system was used and due to the academic value of the work completed its must also be securely backed up. Although there are many backup and version control system available, I only considered two (SVN and git).

SVN is a centralised version control system where as git [11] is a distributed version control system, but as a single developer will develop this project the difference is negligible. One advantage of SVN is that the University can provide students with a free SVN repository.

I have greater experience in using git as I have used git for the majority or my academic and freelance projects. I also have a free Github students account and therefore decided to use git as the version control and to use Github [12] as a backup for the repository. Should the developers of the NHS want to see the process I used to create they application I can share the repository along will all previous versions to them via Github.

After every considerable change made within the repository the change was committed and pushed to Github. I decided to keep all contents of this project, including all documentation within the repository. This ensured that all data was securely backed up.

Chapter 2

Design

You should concentrate on the more important aspects of the design. It is essential that an overview is presented before going into detail. As well as describing the design adopted it must also explain what other designs were considered and why they were rejected.

The design should describe what you expected to do, and might also explain areas that you had to revise after some investigation.

Typically, for an object-oriented design, the discussion will focus on the choice of objects and classes and the allocation of methods to classes. The use made of reusable components should be described and their source referenced. Particularly important decisions concerning data structures usually affect the architecture of a system and so should be described here.

How much material you include on detailed design and implementation will depend very much on the nature of the project. It should not be padded out. Think about the significant aspects of your system. For example, describe the design of the user interface if it is a critical aspect of your system, or provide detail about methods and data structures that are not trivial. Do not spend time on long lists of trivial items and repetitive descriptions. If in doubt about what is appropriate, speak to your supervisor.

You should also identify any support tools that you used. You should discuss your choice of implementation tools - programming language, compilers, database management system, program development environment, etc.

Some example sub-sections may be as follows, but the specific sections are for you to define.

2.1 Overall Architecture

2.2 Some detailed design

2.2.1 Even more detail

2.3 User Interface

2.4 Other relevant sections

Chapter 3

Implementation

The implementation should look at any issues you encountered as you tried to implement your design. During the work, you might have found that elements of your design were unnecessary or overly complex; perhaps third party libraries were available that simplified some of the functions that you intended to implement. If things were easier in some areas, then how did you adapt your project to take account of your findings?

It is more likely that things were more complex than you first thought. In particular, were there any problems or difficulties that you found during implementation that you had to address? Did such problems simply delay you or were they more significant?

You can conclude this section by reviewing the end of the implementation stage against the planned requirements.

Chapter 4

Testing

Detailed descriptions of every test case are definitely not what is required here. What is important is to show that you adopted a sensible strategy that was, in principle, capable of testing the system adequately even if you did not have the time to test the system fully.

Have you tested your system on real users? For example, if your system is supposed to solve a problem for a business, then it would be appropriate to present your approach to involve the users in the testing process and to record the results that you obtained. Depending on the level of detail, it is likely that you would put any detailed results in an appendix.

The following sections indicate some areas you might include. Other sections may be more appropriate to your project.

4.1 Overall Approach to Testing

4.2 Automated Testing

4.2.1 Unit Tests

4.2.2 User Interface Testing

4.2.3 Stress Testing

4.2.4 Other types of testing

4.3 Integration Testing

4.4 User Testing

Chapter 5

Evaluation

Examiners expect to find in your dissertation a section addressing such questions as:

- Were the requirements correctly identified?
- Were the design decisions correct?
- Could a more suitable set of tools have been chosen?
- How well did the software meet the needs of those who were expecting to use it?
- How well were any other project aims achieved?
- If you were starting again, what would you do differently?

Such material is regarded as an important part of the dissertation; it should demonstrate that you are capable not only of carrying out a piece of work but also of thinking critically about how you did it and how you might have done it better. This is seen as an important part of an honours degree.

There will be good things and room for improvement with any project. As you write this section, identify and discuss the parts of the work that went well and also consider ways in which the work could be improved.

Review the discussion on the Evaluation section from the lectures. A recording is available on Blackboard.

Appendices

Appendix A

Third-Party Code and Libraries

If you have made use of any third party code or software libraries, i.e. any code that you have not designed and written yourself, then you must include this appendix.

As has been said in lectures, it is acceptable and likely that you will make use of third-party code and software libraries. The key requirement is that we understand what is your original work and what work is based on that of other people.

Therefore, you need to clearly state what you have used and where the original material can be found. Also, if you have made any changes to the original versions, you must explain what you have changed.

As an example, you might include a definition such as:

Apache POI library The project has been used to read and write Microsoft Excel files (XLS) as part of the interaction with the clients existing system for processing data. Version 3.10-FINAL was used. The library is open source and it is available from the Apache Software Foundation [**?**]. The library is released using the Apache License [**?**]. This library was used without modification.

Appendix B

Code samples

2.1 Random Number Generator

The Bayes Durham Shuffle ensures that the psuedo random numbers used in the simulation are further shuffled, ensuring minimal correlation between subsequent random outputs [**?**].

```
#define IM1 2147483563
#define IM2 2147483399
#define AM (1.0/IM1)
#define IMM1 (IM1-1)
#define IA1 40014
#define IA2 40692
#define IQ1 53668
#define IQ2 52774
#define IR1 12211
#define IR2 3791
#define NTAB 32
#define NDIV (1+IMM1/NTAB)
#define EPS 1.2e-7
#define RNMX (1.0 - EPS)

double ran2(long *idum)
{
  /*-------------------------------------------------*/
  /* Minimum Standard Random Number Generator        */
  /* Taken from Numerical recipies in C              */
  /* Based on Park and Miller with Bays Durham Shuffle */
  /* Coupled Schrage methods for extra periodicity   */
  /* Always call with negative number to initialise  */
  /*-------------------------------------------------*/

  int j;
  long k;
  static long idum2=123456789;
```

```
static long iy=0;
static long iv[NTAB];
double temp;

if (*idum <=0)
{
  if (-(*idum) < 1)
  {
    *idum = 1;
  }else
  {
    *idum = -(*idum);
  }
  idum2=(*idum);
  for (j=NTAB+7;j>=0;j--)
  {
    k = (*idum)/IQ1;
    *idum = IA1 *(*idum-k*IQ1) - IR1*k;
    if (*idum < 0)
    {
      *idum += IM1;
    }
    if (j < NTAB)
    {
      iv[j] = *idum;
    }
  }
  iy = iv[0];
}
k = (*idum)/IQ1;
*idum = IA1*(*idum-k*IQ1) - IR1*k;
if (*idum < 0)
{
  *idum += IM1;
}
k = (idum2)/IQ2;
idum2 = IA2*(idum2-k*IQ2) - IR2*k;
if (idum2 < 0)
{
  idum2 += IM2;
}
j = iy/NDIV;
iy=iv[j] - idum2;
iv[j] = *idum;
if (iy < 1)
{
  iy += IMM1;
}
```

```
  if ((temp=AM*iy) > RNMX)
  {
    return RNMX;
  }else
  {
    return temp;
  }
}
```

Annotated Bibliography

[1] "Extreme Programming: A gentle introduction," http://www.extremeprogramming.org/, 2010.

Extreme programming (XP) was the methodology that I first planned to use for this project. It uses 12 practices each of which would've benefited the project.

[2] "Waterfall model," http://c2.com/cgi/wiki?WaterFall, 2011.

The waterfall model is a popular methodology used by a large amount of software projects. The waterfall model was the methodology that was used throughout this project

[3] "PhoneGap API Documentation - Local storage," http://stackoverflow.com/questions/14210832/can-phonegap-app-work-in-background, January 2013.

StackOverflow question asking how to implement background process within phonegap

[4] "Android," http://www.android.com/, 2014.

Android is the operating system that has been used to create the application.

[5] "Android and iOS Continue to Dominate the Worldwide Smartphone Market with Android Shipments Just Shy of 800 Million in 2013," http://www.apple.com/uk/ios/, February 2014.

This press release states the 2013 market share for a variety of devices and operating systems. This was used when determining the most suitable platform.

[6] "Android SDK," http://developer.android.com/sdk/, 2014.

A brief introduction to the Android SDK, providing links to other useful resources.

[7] "Apple (United Kingdom) - iOS 7," http://www.apple.com/uk/ios/, 2014.

Apple's iOS is the operating system ran on the popular iPhone device, as well as many other devices such as the iPad and iPod.

[8] "Electromagnetic interference : MHRA," http://www.mhra.gov.uk/Safetyinformation/Generalsafetyinformationandadvice/Technicalinformation/Electromagneticinterference/, 2014.

A document outling the possiblities of mobile interference on medical equipment. This is the reason why mobile offline access most be implemented into the final application.

[9] "Electromagnetic interference : MHRA," http://www.mhra.gov.uk/Safetyinformation/Generalsafetyinformationandadvice/Technicalinformation/Electromagneticinterference/, 2014.

Medua is the desktop service that the NHS currently use to acces the database. Medua what was expected of the final application. The Medua website was also used to test the calculator page.

[10] "Genymotion," http://www.genymotion.com/, 2014.

Genymotion is an Android emulator. Genymotion is quicker than the standard Android emulator and allows you to easily create emulators of the most populat devices.

[11] "Git," http://git-scm.com/, 2014.

Git is a distributed version control system. Git was used throughout the project to keep a log of all changes. If there was ever a problem with the project, the project could be rolled back to an earlier version

[12] "Github," http://github.com/, 2014.

Github is service used for storing git repositories. Github provide a free account to students.

[13] "GLYPHICONS - library of precisely prepared monochromatic icons and symbols." https://developer.android.com/design/index.html, 2014.

Android design guidelines were used throughout the design and implementation stages, to ensure the user interface followed Android guidelines.

[14] "GLYPHICONS - library of precisely prepared monochromatic icons and symbols." http://glyphicons.com/, 2014.

Glyphicon is a set of icons released under the Apache 2 licence. This library wasy used for some iconogprahy within the application

[15] "Introduction to Test Driven Development (TDD)," http://www.agiledata.org/essays/tdd.html, 2014.

Test driven development is the procress of creating software, were the tests for the software are written before the code is implemented. Test driven development was used in parts of this project.

[16] "Java Programming Language," http://www.java.com/, 2014.

The Java programming language is the base language used throughout Android development.

[17] "Monograph format," http://www.empr.com/drug-monograph-format/section/793/, 2014.

This page shows an example of what a drug monograph should look like, this was used to allow me to research what would be created, before taking on the project

[18] "NHS Choices - Your health, your choices," http://www.nhs.uk/Pages/HomePage.aspx, 2014.

This is the homepage of the National Health Service, whom this project was created for.

[19] "Phonegap — Home," http://phonegap.com/, 2014.

Phonegap is a multi-platform hybrid application builder. Phonegap allows you to create native applications for multiple platforms using HTML, CSS and Javascript.

[20] "PhoneGap API Documentation - Local storage," http://docs.phonegap.com/en/3.0.0/cordova_storage_storage.md.html, 2014.

Documentation for Phonegaps local storage functionality. Local storage allows the hybrid application to store information on the devices local disc.

[21] "Programming with Objective-C," https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/Introduction/Introduction.html, February 2014.

A brief overview of the objective-C language and how it relates to other languages like C and C++.

[22] "Robospice," https://github.com/stephanenicolas/robospice, 2014.

Robospice is a library created for Android development, that allows developers to easily create asynchronous running services.

[23] "Software Alliance Wales," http://softwarealliancewales.com/, 2014.

The Software Alliance is who provided this project to the Abersywtyth University.

[24] "StackOverflow," http://stackoverflow.com/, 2014.

StackOverflow was used throughout the project to check on best practices and to solve common problems.

[25] "SVN - Subversion," http://subversion.tigris.org/, 2014.

SVN is a version control system that was codcidered during the background stage of the project.

[26] "W3 XML Specification," http://www.w3.org/XML/, 2014.

This is the W3 specification for XML. This was used as the data is provded in XML

[27] "Web applications (WEBAPPS) ," http://www.w3.org/2008/webapps/, 2014.

This is the W3 specification group on web applications.

[28] AppBuilder, “What is a Hybrid Mobile App?” http://phonegap.com/, June 2012.

Blog post explaining what a hybrid mobile application is.