# Testing

Testing is extremely important to the success of any software development project. A thorough testing strategy increases robustness and customer confidence within an application. As the application is used to deliver information that will be used by medical staff, it is vital that the application is well tested, as the consequences of poor testing could be fatal. Testing is one of the fundamental steps within the water model and therefore by choosing this methodology testing was enforced. To provide details of the testing carried out during this project a test specification document was produced. This chapter will discus the approach to testing used throughout the project and also describe test strategies used in detail.

## Approach to testing

Testing was used throughout the implementation process to help create a robust and effective application. Originally it was planned to use a test-driven development throughout the entire implementation process, but this was later found to be detrimental to the progress of the application. Although test-driven development was not used throughout the implementation progress it was used when developing classes that had could potentially be fatal, such as the calculation classes.

Throughout development the application was ran on a real device and any new features were thoroughly tested, using a variety of behaviours and carefully monitoring the applications state for any anomalies. This approach to implementation allowed for a well-tested application, before any test strategy had been complete.

After the implementation phase of the project had been complete, thorough testing of the application was executed. Unit tests were written for all classes that had not already had unit tests created. User interface tests were created and executed using android activity unit tests. The application was also stress tested by using Android’s exerciser’s monkey application.

One the tests had been written, several devices of varying setups using the Genymotion emulator, the full list of tests were then executed on each individual device. This ensured that the application runs well on an array of phones and tablets.

## Test database

When the testing process began an extra parameter was added to the database constructor, this parameter would allow the application to open a separate database that was identical to the actual database, this separate database could then be used during tests. Having a separate database whilst testing meant that the data could be manipulated without effecting the main application. This was needed, as downloading a new set of data after each test would take a long amount of time.

## Unit testing

The Android SDK provides classes for unit testing your application. These classes are based off the JUnit framework, they add extra feature such as the ability to access the applications context allowing you to test the database of the application.

Unit tests were written for every class of the application, testing each public and protected method. Most tests contained multiple assertions, testing that the expected output was returned when correct information is entered and that an error is raised when the incorrect data is entered.

As the unit tests have now been written for all classes, should a developer in the future make any changes to the application, they can execute the unit tests to ensure they have not broken anything. As the units tests will be provided to the NHS with the source code, they will be able to execute the unit tests for themselves, allowing theming to verify thorough testing was carried out, thus increasing their confidence within the application.

As Android runs on a large amount of devices, unit tests are useful for quickly testing that the application runs as expect across the Android platform.

## Bug that was found due to unit testing

Test data had been gathered for testing the calculator class. This data was then implemented into unit tests. To ensure the class was well tested a wide spectrum of values were used, including both small and large values.

Within the original implementation of the calculator class I had used floats to store all values and results of the calculation. After executing the unit tests, most tests were succeeding, but few were returning values slightly off the expected value. As the dosage given to patients must be correct to good level accuracy I began investigating the cause of the problem.

It was found that the issue lay with the data types used, the NHS’s website used doubles whereas I had used floats. Due to doubles being more accurate than floats, the data types were changed to doubles. The unit test were then executed again and all tests succeeded.

## Automated user interface testing

Android’s activity tests allow you to comprehensively test the user interface of the application. Activity tests simulate the launching of the application’s activity and perform a set of unit tests on that activity. These activity unit tests can test a variety of user interface values.

Activity tests were created for every activity. Within the tests each element is checked to ensure that the element is displayed on the screen, the width and height of the element as set correctly, and if the element contains text that the text is set correctly.

Whether an element is visible to the user at that current moment, for example an element may not be visible when the element is displayed at the bottom of a scrollable view. On both the browse drugs and browse calculator activities the search box must be visible to the user at all times. Activity tests were created to ensure the search box is visible to the user.

When the user changes the calculation type on the calculator activity, the dosage and infusion rate fields are toggled, depending on the selected option. As well as this, when a drug calculator is opened for a drug that does not require time or patient weight, the appropriate fields must also be hidden. Activity test were created to test the calculator activity under a variety of situations to ensure that only the correct fields were displayed.

It was very important to run the activity tests on a variety of devices, this ensured that the user interface functions the same on all devices, without having to manual test each device’s user interface.

## Automated integration testing

Due to Android security policies you cannot test the functionality of a user interface element by using activity tests, for example when running an activity test you cannot simulate the user entering text into a text field. Robotium is an automated black-box testing framework for Android licenced under the Apache 2 licence. Robotium allows a developer to implemented black-box integration tests using code. There is also a commercially available version of Robotium, which will record user interface actions and monitor the results, then automatically generate the Robotium code.

Although automated black-box testing using Robotium would be useful for the project, it was decided that automated tests would not be needed for a project this size. The amount of time spent learning Robotinium and then implementing it, would’ve taken longer than manual executing the integration tests. If the amount of activities and the functionality within them increase, Robotinium tests should be implemented.

## Integration testing

As Robotinium was not used, a method of testing that the combined units of the application work as expected was needed. To test that the application works as expected a list of integration tests were made and then ran on two devices, one device running API version 19 and the other running API version 8. Using two devices, with varying ages increases the accuracy of the results. To further increase the accuracy of the results the tests would be executed on more devices.

A full list of the integration tests made can be found in the integration testing section of the test specification.

## Automated stress testing

Exerciser Monkey is a tool provided with the Android SDK, which is used for stress testing Android applications. The tool simulates a set amount of random events on the device, such as button presses, screen presses, volume changes and screen rotations. The tests are used to ensure that applications run well under stressful tasks and that parts of the application do not throw errors.

It was planned that the application would be installed on a device, then using Exerciser Monkey execute 5000 random events to the device. The events were executed and the application never crashed, but whilst watching the device I noticed that the exerciser monkey randomly clicked the logout button, from then on the only activity that was tested was the login activity, as it was impossible for the random events to enter a correct username and password.

It was important for all areas of the application to be tested using this stress tester. Therefore the logout functionality of the application was disabled, which was easy due to modular structure of the application. The Exerciser Monkey was then executed again, simulating 5000s events. The application handled the stress tests well and no errors or timeout warnings were displayed within the console.

## Acceptance testing

To test that the application was at an acceptable level, acceptance testing was carried out. Before sending the application to the NHS for their approval it was vital to test that the application met all of the original functional requirements. This ensured that the application created achieves all of the original goals.

The table below shows the applications results when tested against its functional requirements.

|  |  |  |
| --- | --- | --- |
|  | **Functional requirement** | **PASS/FAIL** |
| **FR 1** | **Authentication** |  |
| **FR 1.1** | User must be able to authenticate themselves using their credentials | PASS |
| **FR 1.2** | User must be notified if the password they enter is incorrect | PASS |
| **FR 1.3** | User will be notified if the authentication failed due to connection issues | PASS |
| **FR 1.4** | User must be able to logout of the system, removing all data | PASS |
|  |  |  |
| **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 | PASS |
| **FR 2.2** | Download complete list of drug indexes from database | PASS |
| **FR 2.3** | Download complete list of drugs and drug information’s | PASS |
| **FR 2.3** | Download all information needed for calculating doses and infusion rates. | PASS |
| **FR 2.4** | Download must still run when the application is in the background | PASS |
|  |  |  |
| **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…) | PASS |
|  |  |  |
| **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 | PASS |
| **FR 4.2** | User will be see when an update was last performed, and perform an update from this screen. | PASS |
|  |  |  |
| **FR 5** | **Browse drugs** |  |
| **FR 5.1** | This screen will allow the user to view a list of all drugs | PASS |
| **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 | PASS |
| **FR 5.3** | The user will be able to click a drug in the list to open a new screen displaying the needed information | PASS |
|  |  |  |
| **FR 6** | **View drug** |  |
| **FR 6.1** | When a drug has been selected the drug and all it’s information will be displayed in an easy to read format | PASS |
| **FR 6.2** | Where drug information headers contain help information, a help icon will be displayed next to the header. | PASS |
| **FR 6.3** | When heading help icon is clicked the helping information will be displayed | PASS |
| **FR 6.4** | If the drug had calculator information, then a button to open the calculator should be shown | PASS |
|  |  |  |
| **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. | PASS |
|  |  |  |
| **FR 8** | **Calculate dose and infusion rate** |  |
| **FR 8.1** | The user will be able to select calculation type | PASS |
| **FR 8.2** | User will be able to enter information required for the calculation | PASS |
| **FR 8.3** | When the calculate button has been clicked the input will be thoroughly validated | PASS |
| **FR 8.4** | After validation the result of the calculation will be displayed to the user | PASS |
| **FR 8.5** | The equation and values used to calculate the answer will be neatly displayed to the user | PASS |
|  |  |  |
| **FR 9** | **XML customisability – for developers** |  |
| **FR 9.1** | All text within the application must be changeable through XML files. | PASS |
| **FR 9.2** | The structure of the XML API’s provided must be outline within XML files, allowing easy customisation for different API’s | PASS |

As shown in the table above the application fulfilled the complete list of functional requirements. The next logical step was to send the compiled APK to the NHS representative for approval.

The representative was very pleased with the final application and only suggested one change, which was to order the calculator’s in alphabetical order. This change was implemented and an updated sent to the NHS. Once the changes had been made, the NHS representative demonstrated the application to the IMG Executives Group that consisted of pharmacists and nurse experts, whom stated that the application was intuitive to use and easy to navigate.

## Acceptance testing feedback

This section contains the feedback provided by the NHS representative.

*“For the first time this year, NHS Wales Informatics Service became involved with Aberystwyth University when we proposed the development of a mobile phone app for the NHS Injectable Medicines Guide (IMG) as a possible dissertation project. The IMG is a website that that provides information on how to prepare and administer injectable drugs and is used by over 90% of NHS Trusts across the UK by nurses, doctors and pharmacists. It is a major contribution to patient safety and is recommended by the National Patient Safety Agency. We have been under pressure to develop a mobile phone app version of the website.*

*We supplied an outline design of the app. Aidan was one of two students to select this proposal to take forward as his dissertation. I am providing this feedback as a result of working with Aidan while he has written the software.*

*My contact with Aidan has been via email and he has always responded promptly to my questions and provided full and accurate answers. Aidan has had to ask us to provide access to the various data sources and I suspect the access that we have provided has not been what he would have chosen but he has always been willing to adapt to what we have provided and work with the our timescales. He has only requested minimal information so he must have worked within his own initiative to complete his work.*

*And then he delivered the app for us to look at. I was impressed that he had considered how we would implement the software and provided a simple means for us to load the app. Initially we had problems loading the software but Aidan researched the problem and worked with us to resolve the issue which was largely due to the age of our equipment. Implementation and support is a different skill set to development and I was impressed by his problem-solving skills and his persistence in resolving the issue.*

*I was bowled over by the app that he has produced; it is well beyond our outline design. He has clearly had to adapt what was essentially an iPhone design to the Android platform and he has produced a design that is very clean and easy to use. He has clearly understood the user interface design principles and I much prefer his design of the pages and the navigation from one function to another over our design.*

*One aspect of the app is the ability to download and update the data from the central website and to manage the security of this function. Aidan has demonstrated that he understands the security issue and has devised an update that is simple to use but downloads the complex data sets that are involved and has been able to transform the datasets into the information as it appears on the screen, matching across the keys to the each file.*

*I have demonstrated the app to the IMG Executive Group which consists of Pharmacists from Imperial NHS Trust and an RCN nurse expert on injectable drugs and they were very impressed and found the app intuitive and easy to use. The view was that nurses, for whom very few apps exist and so are not expert users, would find welcome this app and find it easy to navigate.*

*Overall we are very impressed with the way that Aidan has interacted with us and with the product that he has produced. I am sure he will have an excellent career in software if this is the path that he chooses.*”

**Robin Burfield, Development Manager, NHS Wales Informatics Service.**

This feedback proves that the application completes its origiinal purpose and that the user interface is clean and easy to use.