National College of Ireland
BSc in Computing
2013/2017

Kevin Clarke
13101722
x13101722@student.ncirl.ie

"*A monitoring tool designed for mental health professionals to enhance patient care*"

Table of Contents

Executive Summary

The purpose of this document is to set out the technical details for the development of an application that can be used by mental health professionals for monitoring patients. In particular, to monitor the dosages and of patients being prescribed antipsychotic & mood stabilising drugs, by comparing inputted patient data against regulatory guidelines and allowances. The key aspect of this application is that it will allow for a centralised repository of vital patient information that can be accessed by relevant mental health professionals across the Irish health system when required. This is an important aspect of the application because there is currently no central repository in existence that Irish mental health professionals can query when they quickly require data on patients suffering from mental ilness. This application benefits both the patient and health care professional by providing a high level of interactivity that until now has not existed within the Irish mental health system. The combination of a graphical user interface that provides quick access to patient records and the ability to visualise complex data with the help of various chart types improves upon the workflow that currently exists.

The intended customers for this product will be the HSE and private mental health clinics within Ireland. Futhermore, this software application solves the issue of handwritten and decentralised record keeping, and allows for the HSE to more easily carry out audits of patient care within the Irish health system by providing the ability for hospital managers to export reports detailing prescriptions that exceed agreed upon dosing guidelines.

In the “Future Evolution” section of this report I detail the improvements I feel could be made to this concept in order to deliver an enterprise level application that solves the current problem within our health system.

Introduction

1.1 Background

I chose to create this application after a conversation with a psychiatrist who currently works in the Irish health system. She described the problem to me and I knew that this issue could be solved with the correct application of technology and planning. The problem domain for this application excited me as it was solving a very worthy real world problem. This application if evolved correctly could be turned into an enterprise level application that would improve the level of patient care, and provide audit functionality for health care management.

1.2 Aims

The main aim of this application is to improve the quality of patient care received by people being treated for mental illnesses in the Irish health system. The secondary aim of the application is to provide an audit trail to more accurately manage adherence to prescribing guidelines.

1.3 Technologies

- The first technology that is being used to create the application is the bootstrap CSS framework. This framework is allowing me to create a responsive website with a clean and efficient layout that allows for smooth navigation and ease of use even if the users have no previous experience with the application.

- MySQL is being used in conjunction with XAMPP and phpMyAdmin. These 3 combined allow for my application to run on a local server, and implement MySQL databases while being developed. Eventually this application will be migrated from a local server and be hosted on the web where it can be accessed by anyone. The use of MySQL is vitally important as my application is driven by the contents of the various database tables containing patient information, and prescribing guidelines.

- I will also be making use of JavaScript libraries that will allow for patient information to be graphed within the application. In particular I will use Chart.js for this purpose. These graphs will be dynamic and change depending on the values contained within the database. This allows users to clearly spot trends and warning signs in a much more visual way.

1.4 Project Scope

The scope of the project was to develop a web application that stores data in a centralized database to allow for efficient and widespread access to relevant patient data by mental health professionals across the HSE. I have been in contact with several Psychiatrists currently working within the HSE to elicit the following requirements. I have been made aware by these professionals that the National Institute for Health and Care Excellence (NICE) and Maudsley Prescribing guidelines must be used as the monitoring benchmark within the application as they provide the guidelines that mental health professionals adhere to when administering psychotropic drugs.

In addition, based on conversations with the relevant stakeholders it has been decided to create this application with various user access levels in mind as not all mental health professionals are allowed to alter a patient's dosages or schedule medical investigations. These user levels are detailed further in the "Security" section of this report.

Several constraints have been placed upon this project, the first of which is a limited selection of stakeholders to elicit requirements from. I have only been able to elicit requirements from two practicing psychiatrists and one advanced nurse practitioner currently working in one of the countries busiest accident and emergency rooms.

Another constraint that has provided an interesting challenge was my lack of knowledge relating to security principles and best practices. Through the course of this applications lifecycle I have expanded my knowledge base on this topic through online research to the point where I believe this application demonstrates that security has been a key consideration in its development.

System

1.5 Requirements

1.5.1 Functional requirements

This application is intended to be deployed as a web application that can be accessed by authorized users within the HSE. This decision was made due to the added portability of a web application, and the wide spread use of web applications throughout modern workplaces. In addition, this application will require an active network connection in order to update the database and allow for the retrieval of the relevant information that a user may require at any given time.

Full users will have access to the full functionality of the application while other users will be restricted from certain pieces of functionality as these features relate to activities outside their scope of employment.

- The creation of individual users and allocation of their specific access rights.
- The creation of patient records by users.
- The ability for users to modify patient records as and when required dependent on their access level.
- The storage and retrieval of information inputted into the system.
- The ability to compare stored data against the guidelines in order to generate system alerts and user notifications.
- The implementation of a GUI to efficiently display desired information.

- The ability to run a report that allows for effective auditing of the system and user adherence to guidelines and patient care SLA's.

1.5.2 Use Case Diagram

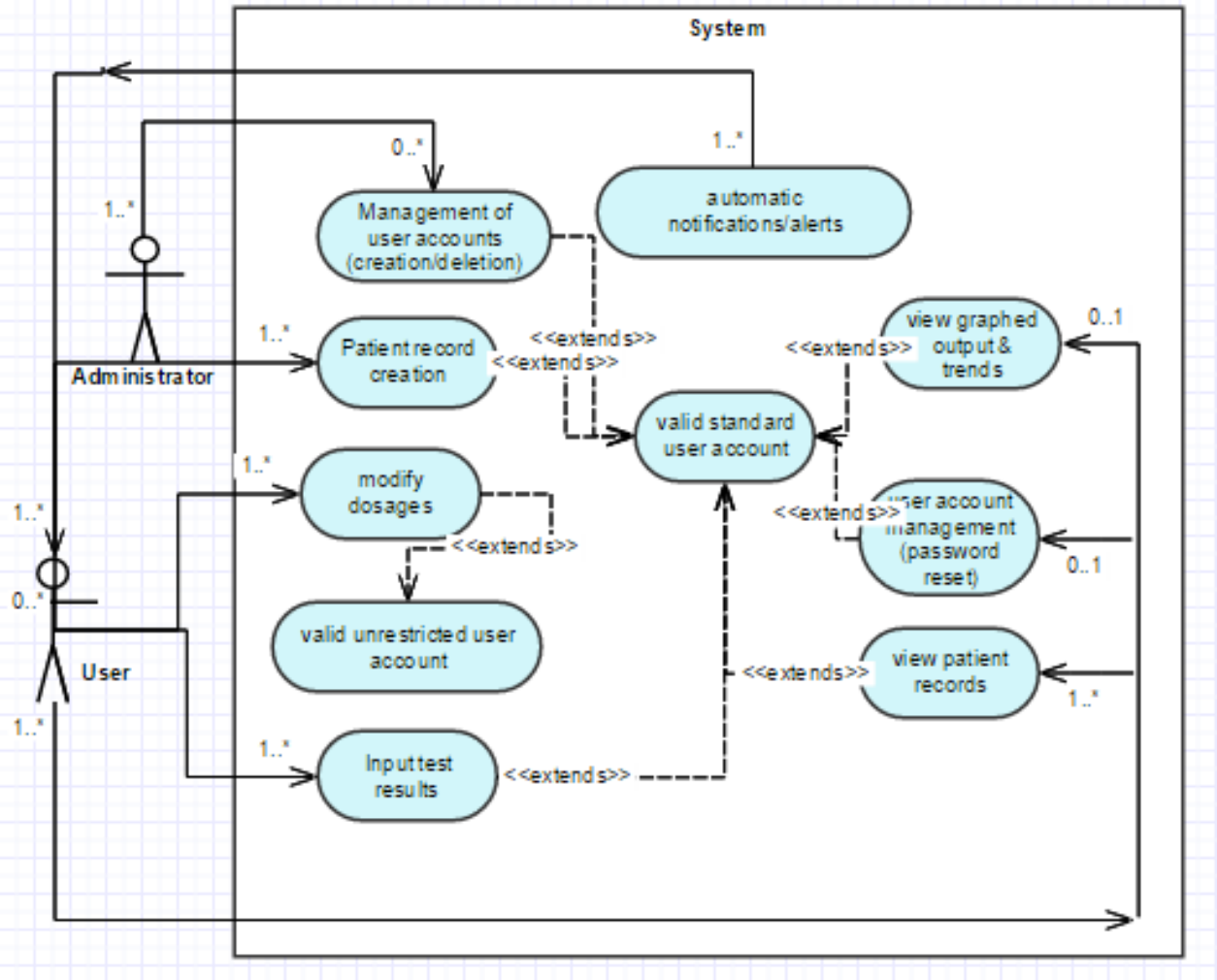


Figure 1 - *Use Case diagram*

1.5.3 Requirement 1: Patient record creation

1.5.3.1 Description & Priority

This requirement relates to the ability of a user to successfully create a new patient record within the system. This is one of the most important aspects of the application as the ability to create records is essential to the utility of the software.

1.5.3.2 (1. Patient record creation)

Scope

The scope of this use case is for a user to create a new patient record within the system.

Description

This use case describes the process of creating a new patient record and inputting any mandatory data required for the successful creation of the record.

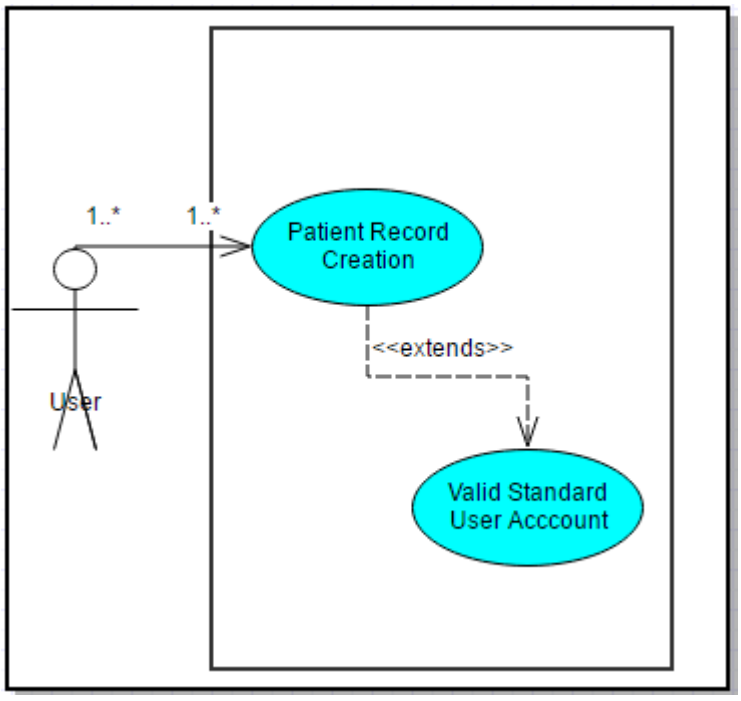


Figure 2 - *Use Case (Req. 1)*

Flow Description

Precondition

The user has successfully logged into the system and their account has been given the basic level of access.

Activation

This use case begins when the user logs into the system.

Main flow

1. The system displays the main page of the application.
2. The user selects “create new patient” button from the sidebar menu.
3. The system displays a new page containing a form for a new patient record.
4. The user completes the mandatory fields denoted on the form and any of the relevant additional fields they feel appropriate and clicks submit.
5. The system verifies that the mandatory fields have been completed and assigns the patient record with a unique record number and stores the record in the database. (See A1)
6. The system displays the created patient record to the user. (See E1)

Alternate flow

A1: Missing mandatory fields

1. The system displays a message to the user; notifying them that all mandatory fields were not completed, and returns them to the form with the missing mandatory fields highlighted.
2. The user completes the missing information and re-submits.
3. The use case continues at position 6 of the main flow.

Exceptional flow

E1: Record not stored

1. The system receives an error when attempting to store the record in the database and displays a message to the user informing them an error has occurred and the record was not created successfully.
2. The user accepts the message.
3. The use case continues at position 1 of the main flow.

Termination

The system presents the user with the home page of the application.

Post condition

The system goes into a wait state.

1.5.4 Requirement 2: Modify prescription dosages

1.5.4.1 Description & Priority

This requirement relates to the ability of a user with an unrestricted account to modify the dosages of prescriptions within the system.

1.5.4.2 (2. Modify dosages)

Scope

The scopes of this use case is for a user with unrestricted access to modify the record of a patient assigned to their care and alter the dosage amounts that the patient is currently receiving.

Description

This use case describes the process of accessing a particular patient record and modifying the prescribed dosages and drug types contained within the record.

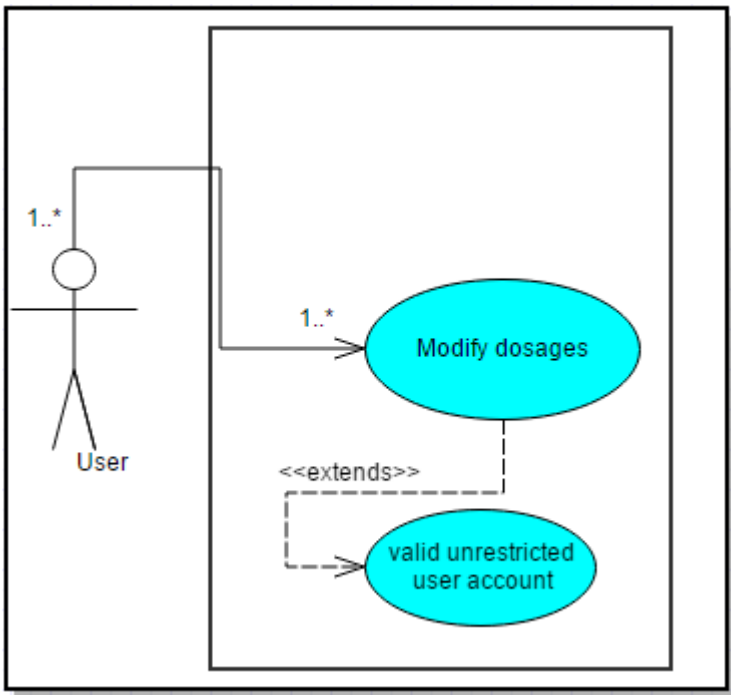


Figure 3 - *Use case (Req. 2)*

Flow Description

Precondition

The user has successfully logged in with an account that has full access to the records of patients under their care.

Activation

This use case starts when the user logs into the system.

Main flow

1. The system identifies the user's login as valid and displays the home page of the application.
2. The user selects the search function from the sidebar menu.
3. The system displays a set of search criteria that the user can choose from.
4. The user searches for patients using the applicable criteria. (See A1)
5. The system displays a list of patients who match the criteria if the patient is assigned to the user.
6. The user is able to select the modify button if the patient is assigned to their care and the user has full access. (See E1)
7. The system displays the patient record and allows for the majority of fields to be modified. (Excludes name, patient record number and other criteria that can only be altered by the administrator in exceptional circumstances)
8. The user modifies the dosages and a drug type using the input fields provided and selects update record.
9. The system updates the patient record in the database, and displays the update patient record.

Alternate flow

A1: Patient is not assigned to the user

1. The user selects that this is an exceptional circumstance when searching for a patient record.
2. The system verifies that the user has full access, and displays the patient records matching the chosen search criteria.
3. The user selects the modify button on the patient record and is allowed to modify the majority of information within the record. (Excludes name, patient record number and other criteria that can only be altered by the administrator in exceptional circumstances)
4. The system flags these modifications as exceptional and stores these as a supplement to the initial record, in addition to sending the assigned care team an email notification that these changes have been made and by whom.
5. The use case continues at position 7 of the main flow.

Exceptional flow

E1: User does not have full access

1. The system displays a message that the user does not have permission to modify the chosen records, and sends a notification to the system administrator.
2. The user accepts the message.
3. The use case continues at position 1 of the main flow

Termination

The system updates the record and displays the updated record.

Post condition

The system goes into a wait state.

1.5.5 Requirement 3: Input test results

1.5.5.1 Description & Priority

This requirement relates to the ability of a user to effectively input test results in the system for any given patient. This ability is not restricted and is available to all users. This requirement is of medium priority but is a key feature in the utility of the software.

1.5.5.2 (3. Input test results)

Scope

The scope of this use case is to input test results and update the patients record within the system to reflect these changes.

Description

This use case describes the process of how a user would enter in a set of results from various tests and diagnostic measures (e.g.; blood results, ECG results, weight, blood pressure etc.)

Use Case Diagram

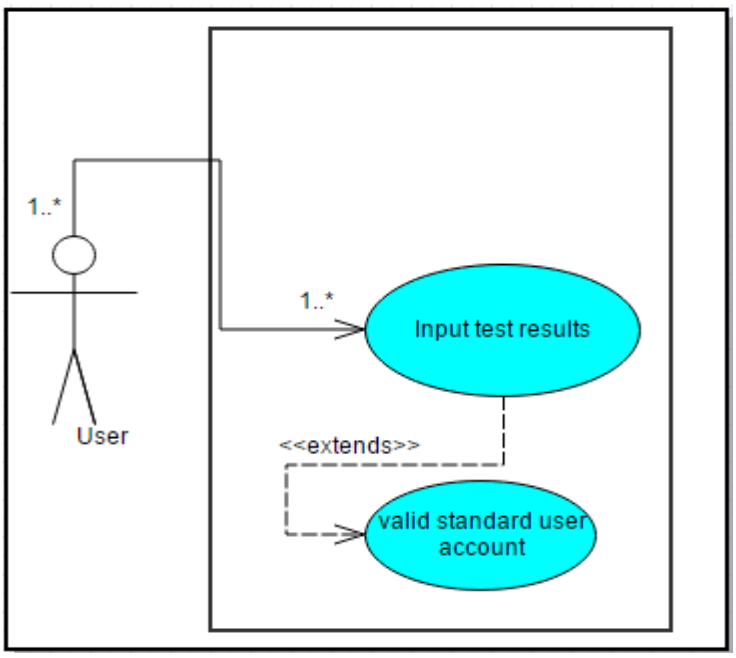


Figure 4 - *Use Case (Req. 3)*

Flow Description

Precondition

The user is successfully logged in.

Activation

This use case starts when a user navigates the home page of the application.

Main flow

1. The system displays the home page of the application.
2. The user selects the add test results function from the side bar menu.
3. The system displays a search box, and search criteria within a drop-down menu.
4. The user search for the relevant patient using the tools provided and selects the chosen patient, and the type of test results to be inputted.
5. The system displays a form containing mandatory fields that must be populated for the results to be entered correctly.
6. The user populates the fields and selects the enter button.
7. The system verifies all mandatory fields have been entered correctly and re-confirms with the user they wish to proceed. (See A1)
8. The user re-confirms the test and patient information are correct by clicking ok. (See E1)
9. The system updates the record in the system and displays the updated patient record.

Alternate flow

A1: Mandatory fields not completed correctly

1. The system displays a message indicating that all of the mandatory fields have not been entered successfully and returns the user to the form page to input the missing data.
2. The user enters the mandatory data into the fields provided a re-submits.
3. The use case continues at position 7 of the main flow.

Exceptional flow

E1: Incorrect patient assignment

1. The user notices the data is going to be added to an incorrect patient record and chooses to select cancel.
2. The system does not input the test results into the database, and applies a message that this action has been cancelled.
3. The use case continues at position 1 of the main flow.

Termination

The user has successfully entered test results into the system and assigned the to a valid user.

Post condition

The system goes into a wait state.

1.5.6 Requirement 4: View graphed patient data and trends

1.5.6.1 Description & Priority

This use case relates to the ability of a user to successfully view patient data in graphical format in order to easily spot trends and upper/lower limits that might be causes for concern.

1.5.6.2 (4. View graphed patient data and trends)

Scope

The scope of this use case is to allow for a user to easily view graphical representations of patient data stored within the system.

Description

This use case describes the process of navigating the system and using the graphing functionality available in order to view patient data such as test results of a historical time frame.

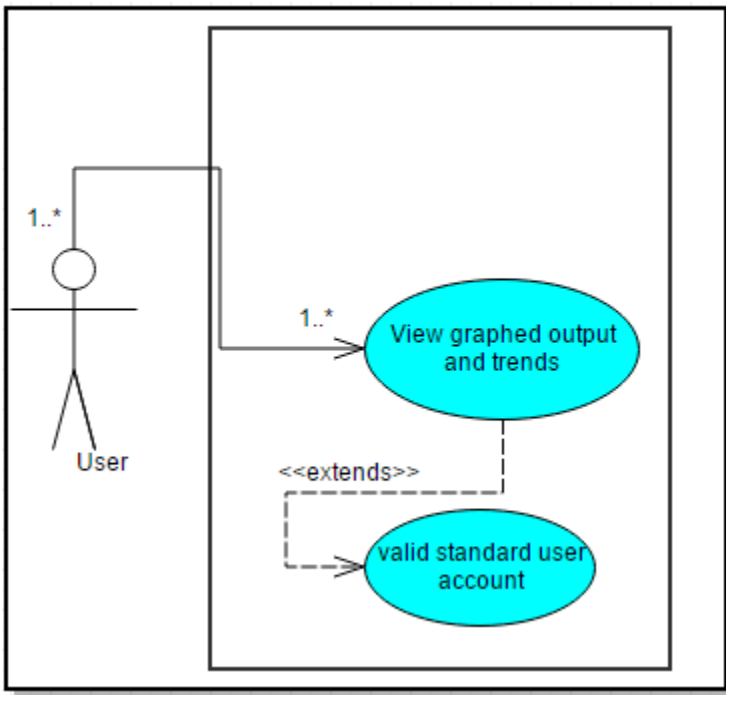


Figure 5 - *Use Case (Req. 4)*

Flow Description

Precondition

A user has successfully logged into the system.

Activation

This use case starts when the user navigates to the graphing tab of the system.

Main flow

1. The system displays a search function that can be filtered by various criteria in order to narrow down patient selection.
2. The user selects the patient they wish to represent graphically using the search tools provided.
3. The system displays several graphical representations of data that is commonly desired, and presents the user with options to view other data subsets. (See E1)
4. The user choses the results and/or information they wish to view.
5. The system updates GUI according to these user inputs.

Exceptional flow

E1: System error

1. The system displays an error due to the data not being able to be retrieve from the database, or failure of the functions governing the representation of the data on the GUI.
2. The user accepts the error message.
3. The use case continues at position 1 of the main flow.

Termination

The system presents the GUI containing patient graphs until the user navigates away from the page, or alters the search criteria.

Post condition

The system goes into a wait state

1.5.7 Requirement 6: View patient records

1.5.7.1 Description & Priority

This use case relates to the ability of a user to successfully search and view records for a particular patient/s. This feature is important because it forms an essential element of the applications functionality.

1.5.7.2 (6. View patient records)

Scope

The scope of this use case is for a user to successfully view the records of a searched for patient.

Description

This use case describes the process that a user would go through in order to search for a patient within the system and display their information.

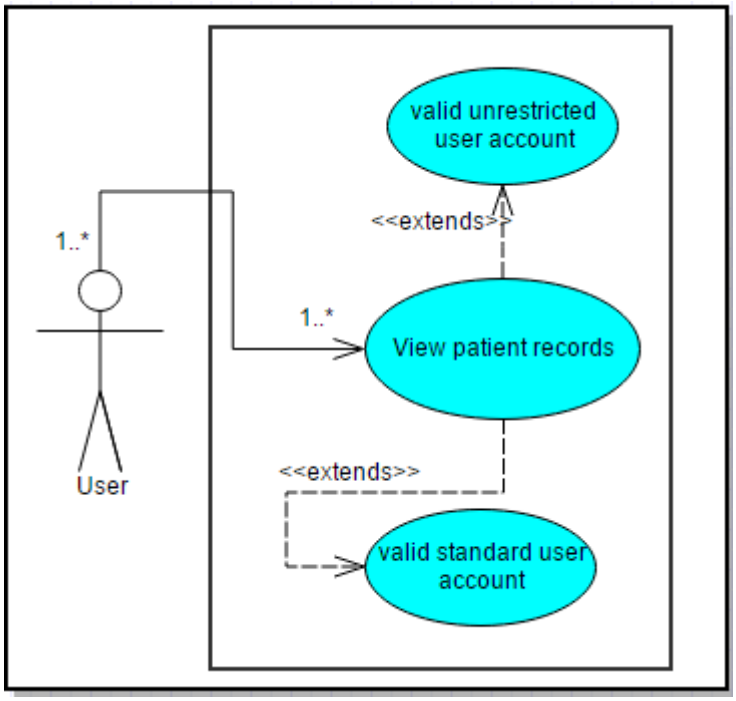


Figure 6 - *Use Case (Req. 6)*

Flow Description

Precondition

The user is logged in.

Activation

This use case starts when a user selects the search patient records button from the sidebar menu of the homepage.

Main flow

1. The system displays search functionality including dropdown menus in order to filter a patient search by certain criteria.
2. The user uses the search function to input meaningful search criteria and clicks search.
3. The system searches the database and displays patient records that match the search criteria. (See E1)

4. The user selects which record they would like to view if more than one has been returned.
5. The system displays the record selected in greater detail.

Exceptional flow

E1: No record found

1. The system can't find a record matching the search criteria and displays a message to the user that no record has been found.
2. The user accepts the message by clicking ok.
3. The use case continues at position 1 of the main flow.

Termination

The system presents the record chosen by the user until the user navigates away from the page.

Post condition

The system goes into a wait state.

1.5.8 User requirements

This section describes the set of objectives and requirements for the system from the customer's perspective. What are the clients saying they want?

I have consulted two practicing mental health professionals within the HSE in order to understand the problems that this application is attempting to solve and how best to tailor this application to their needs.

The essential elements that I have gleamed from these conversations are as follows:

- A user must be able to login to the application using a unique username and password in order to create an element of security.

- All users should have the ability to create a patient record, and update all patient information except for the modification of patient prescriptions.

- The data collected by the software should be stored in such a manner that users in other hospitals/clinics can access it in order to provide more efficient care.

- The application should have the ability to check the inputted dosage data and vital statistics against the relevant guidelines and create alerts within the system to make users aware of practices that break the agreed guidelines.

- The application must be able to graphically represent patient data and display historical trends in order to enable users to easily ingest relevant patient information.

- The user interface should be intuitive and easy to use.

- The application should also have the ability to record when alerts are created in order to form an audit trail for management in order to monitor the level of care being provided to patient's adherence to the guidelines set forth by the HSE.

- The implementation of relevant security measures has also been stressed to me by the various stakeholders due to the amount of sensitive information that will be collected by this application.

1.6 Stakeholder Matrix

This matrix outlines the interests of the key stakeholders in this project.

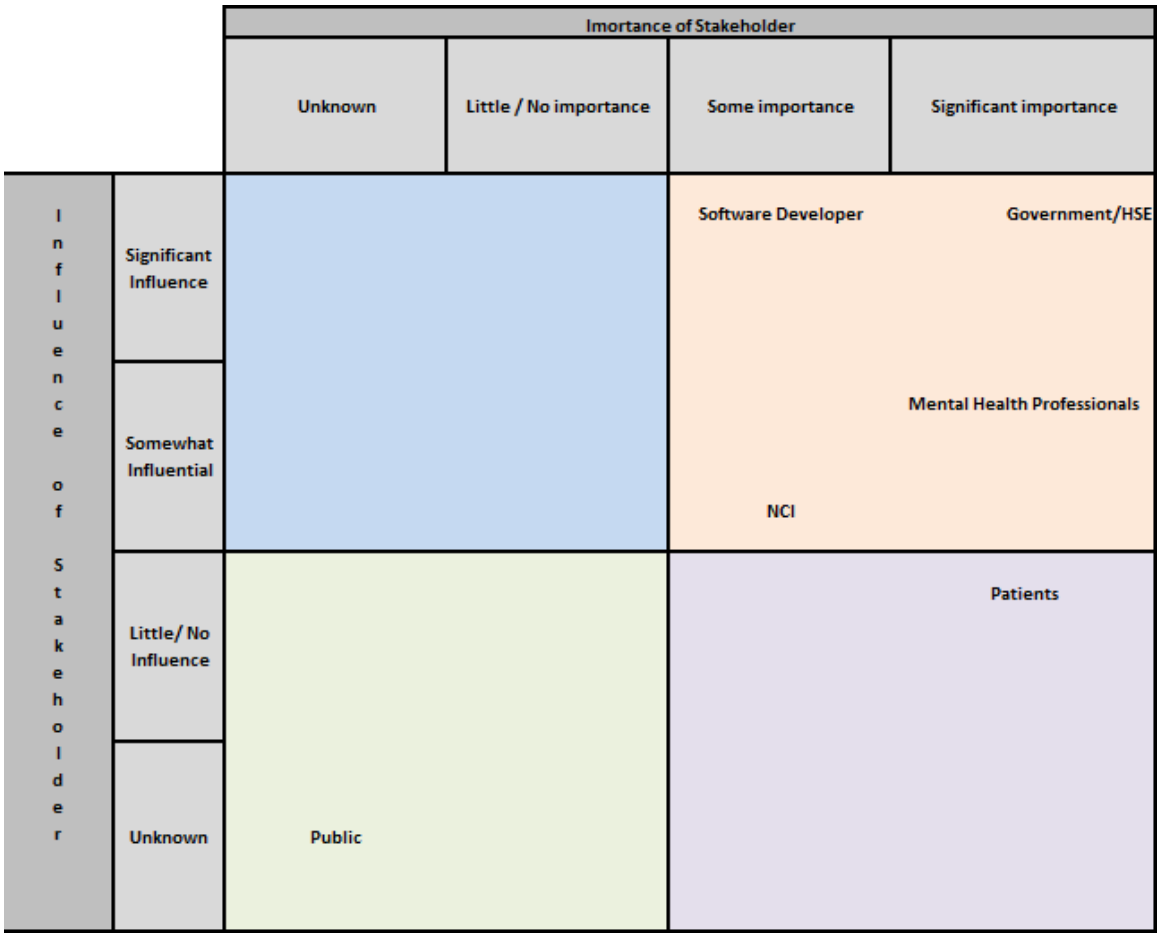


Figure 7 - *Stakeholder matrix*

1.7 Non-Functional Requirements

1.7.1 Performance/Response time requirement

- The system should display search results in less than 2 seconds to allow for easy use of the system.

- All graphical representations generated by the system should retrieve and display the requested data in less than 2 seconds from the time of request.

1.7.2 Availability requirement

- As this piece of software is intended for use in in a clinical setting it should have extremely high availability to be an effective tool. However due to the restrictions placed on where I will be hosting this application I cannot guarantee the availability that is required.

1.7.3 Security requirement

- Due to the sensitive nature of the data that will potentially be stored in this application security features will be of vital importance.
- The first line of security will be the use of a valid username and password in order to login. The passwords for these users will be stored in an encrypted format within the database.
- Also, users who enter their password incorrectly 3 times will be have their account suspended and will require reactivation by the system administrator.
- In addition, different levels of user access will be implemented to segregate what information is available.

In addition to these basic features there are more advanced techniques that can potentially be implemented into my system. Some of which will be discussed in theory because I don't have the skill-base or the required timeframe to implement all of the security features that would be required in a real-world scenario for my application. ("Web Application Security: Methods And Best Practices")

1.7.4 Reliability requirement

- The system is required to be available at all times as the clinical environment in which it would be placed does not have set hours of operation. However constant availability is not realistic and the system will likely be offline at certain times during the year due to maintenance or unforeseen errors. The goal would be to reduce this downtime as much as possible.

Interface requirements

1.8 GUI

1.8.1 Login Page – Wireframe

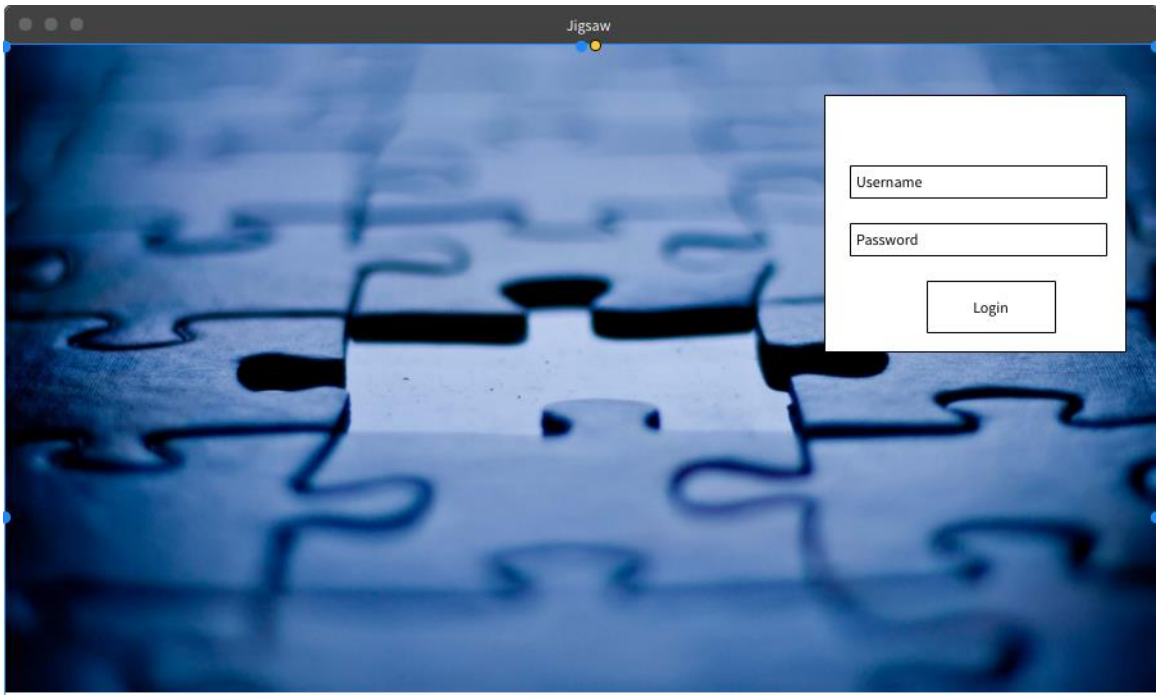


Figure 8 - *Wireframe (Login)*

- *This wireframe depicts the login page that is used to access the application using valid credentials.*

1.8.2 Search Patient Record – Wireframe

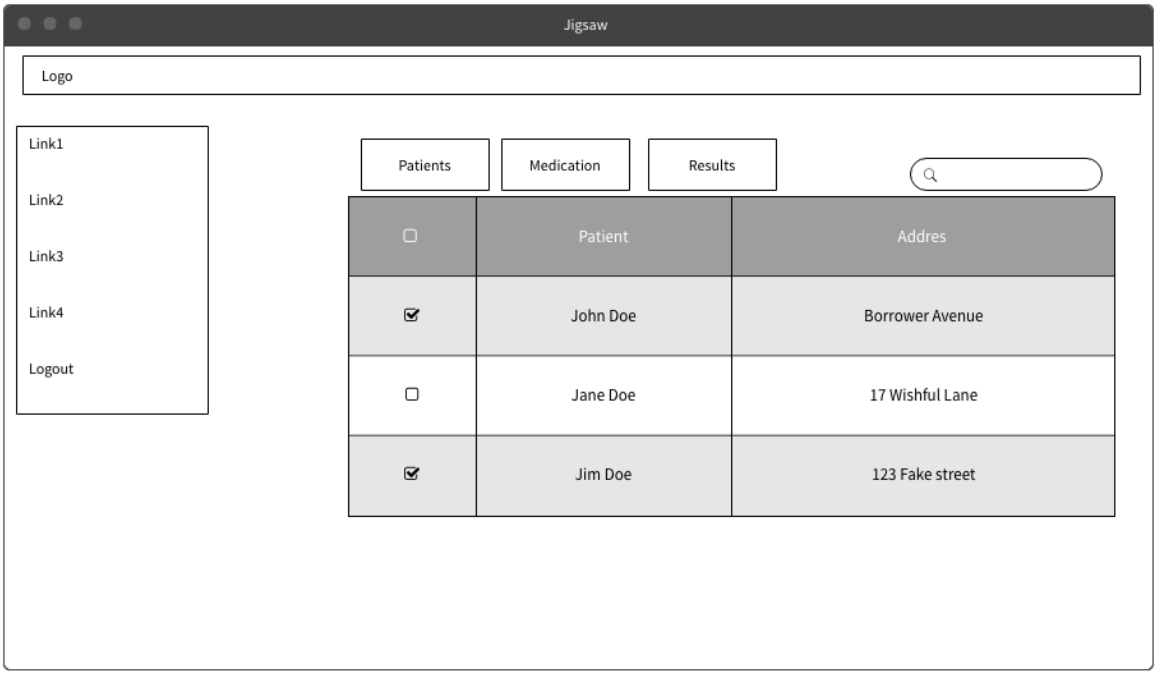


Figure 9 - *Wireframe (Search Patient record)*

- *This wireframe depicts how I envisage the interface that allows users to search for patient records that are accessible through the system.*

1.8.3 Create Patient Record - Wireframe

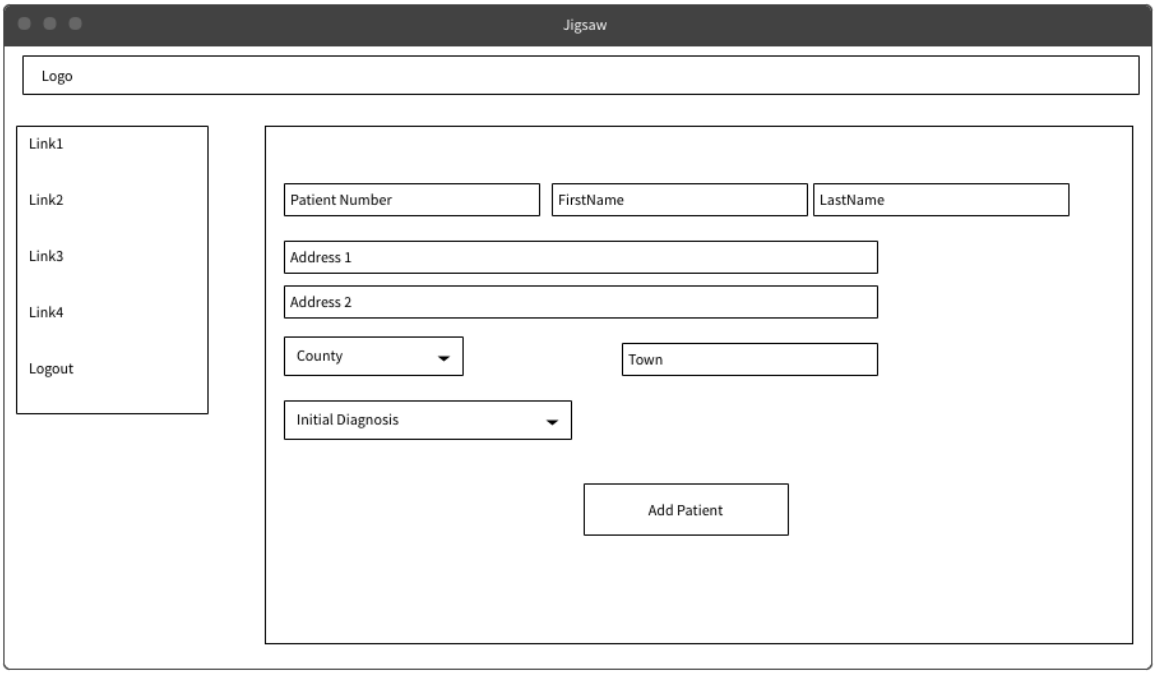


Figure 10 - *Wireframe (Create Patient record)*

- *This wireframe depicts how I hope to create the GUI that allows users to create new patient records within the system.*

1.8.4 Add Test Results - Wireframe

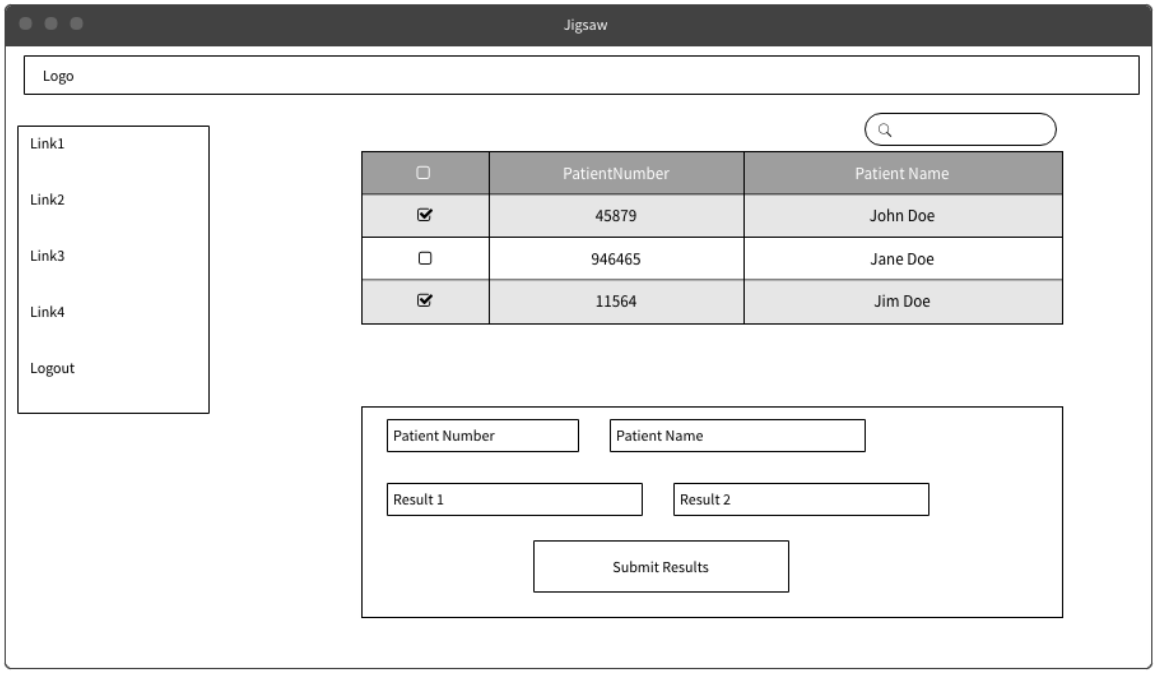


Figure 11 - *Wireframe (Add test results)*

- *This wireframe details the concept I have for allowing users to input patient test results into the application.*

1.8.5 Modify Dosage – Wireframe

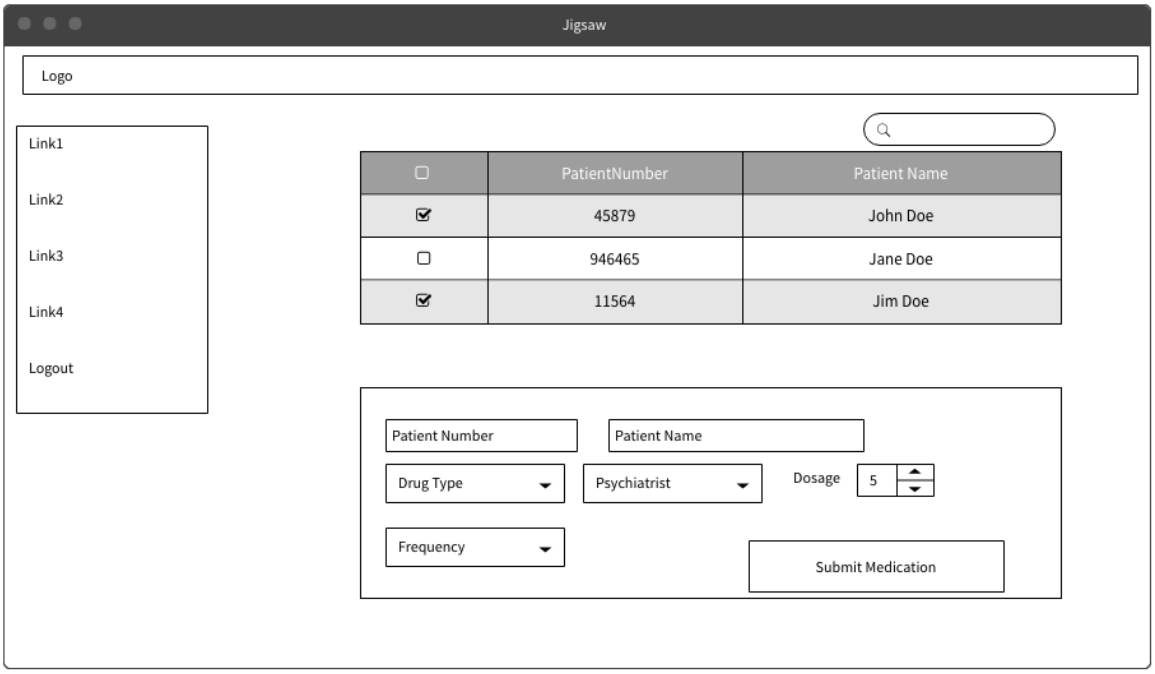


Figure 12 - *Wireframe (Modify dosage)*

- *This wireframe depicts how a user with an unrestricted access would be able to interact with the application in order to create and modify patient prescription dosages.*

1.8.6 Graphed Patient Data – Wireframe

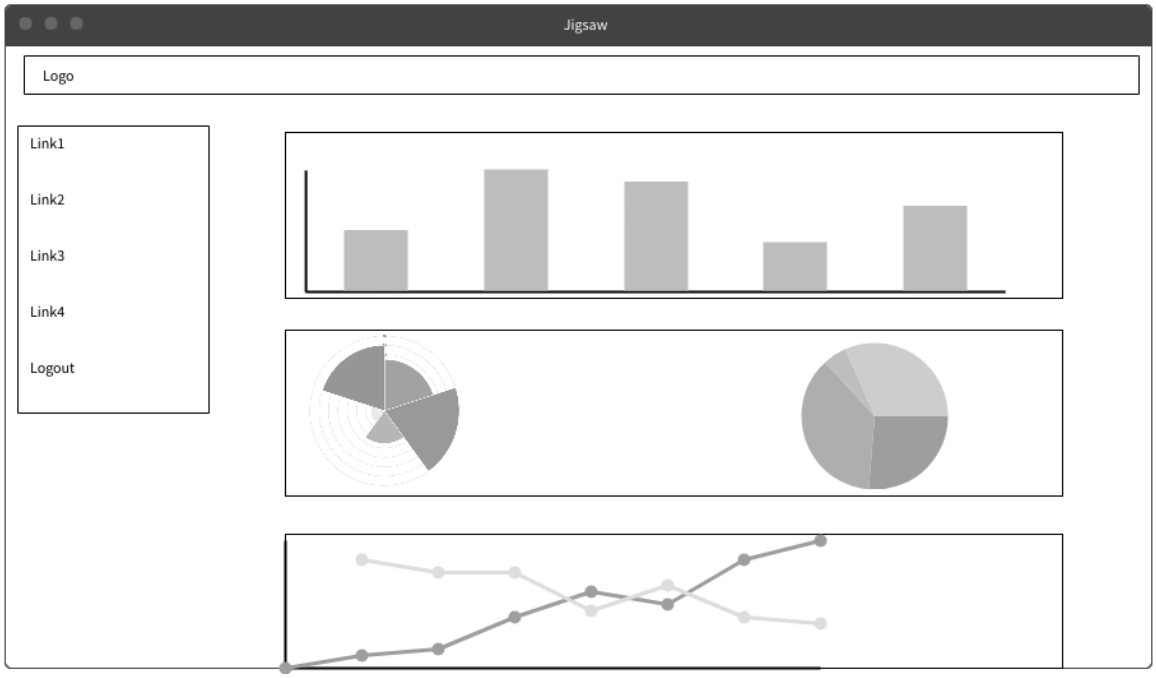


Figure 13 - *Wireframe (Graph data)*

This wireframe details how I see the final GUI potentially looking, and will allow users to view graphed patient data in order to spot trends. ("Web Application Security: Methods And Best Practices")

System Architecture

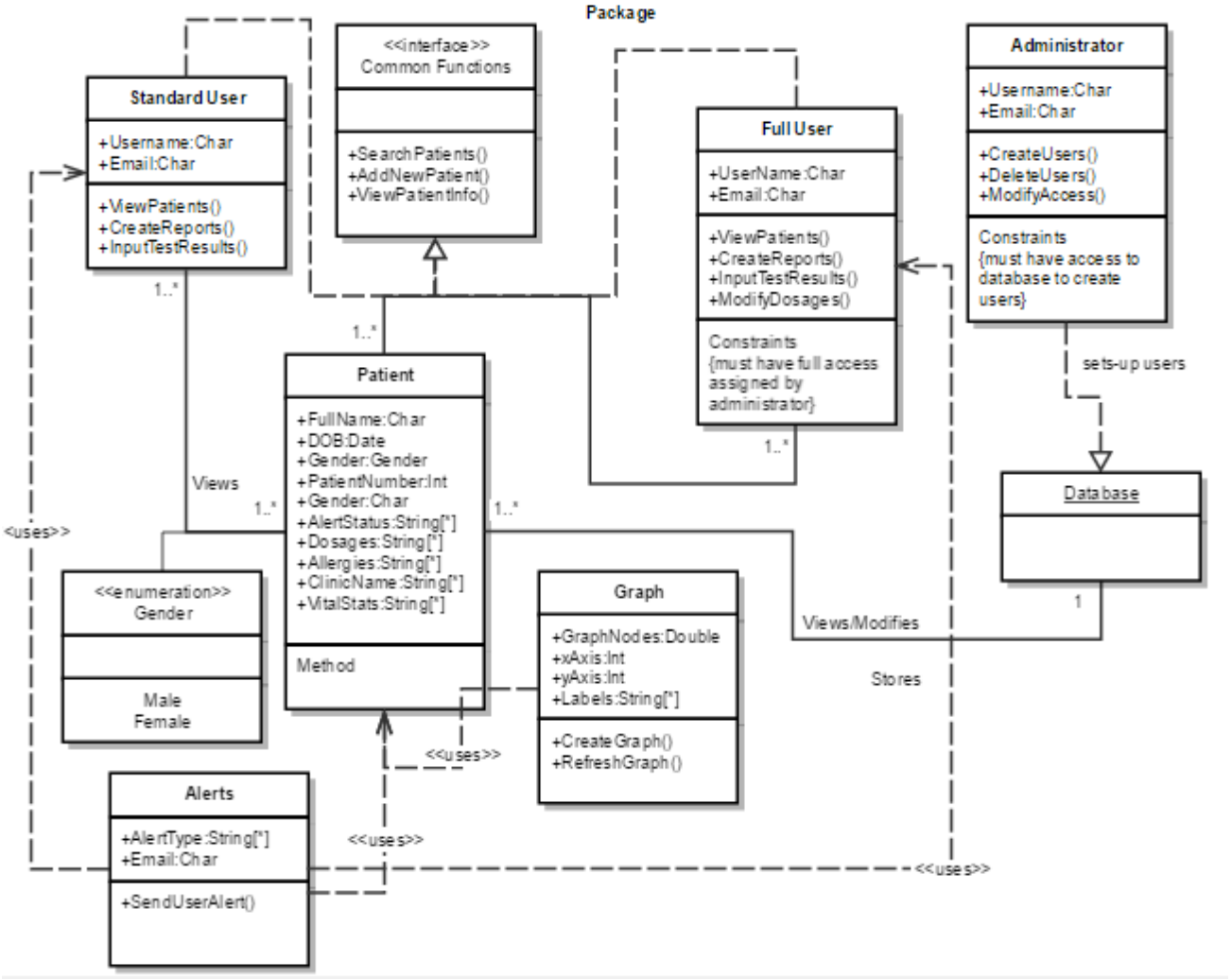


Figure 14 - *System Architecture diagram*

Implementation

1.9 Creating the charts.

Data visualization was achieved by using a combination of the Chart.js library, Ajax, PHP, and HTML. This posed an interesting challenge, as I had never worked with Ajax or a JavaScript charting library such as Chart.js previous to this project. The charts contained on the Dashboard.php page of the application have been created by creating a blank HTML canvas in any location that a chart was desired. The i.d from this canvas is then used in a JavaScript file that specifies the various attributes for the chart. These attributes control chart style, color, labels, axis points and everything related to the actually display of the data.

The data itself is passed through an Ajax call contained within the JavaScript file that creates the chart. This Ajax call requests the data retrieved from the database by the file data.php. The data.php file requests the required information from the relevant database tables and encodes this data in Json format so it can be utilized by the Ajax call. This process is repeated with different JavaScript and data files for each chart in the application.

1.10 Populating patient details by clicking a table row.

The JavaScript function that allows for the patients number, patient's first name, and patient's last name to be populated automatically with the details from the patients table once the user clicks the relevant row from the datable has proved very useful. The purpose of this function is to help prevent user error by populating the fields with the exact patient details that already exist within the table.

1.11 User access controls

User access for the application is controlled by a series of login pages that check if the user possesses appropriate credentials. The index.php page provides the first access point and checks if the user has a relevant username and password stored in the user table of the database. All users irrespective of access level are able to access the site through this portal. However the medication.php section of the site can only be accessed by a valid user who also has their password and username contained within the relevant access table for this section, and lastly the audit functionality of the site has been hidden behind a third login page that can only be accessed by users with audit privileges. All of the permissions are controlled by the system admin who creates the users and their relevant permissions.

1.12 Data display

The patient data throughout the application is displayed using Data tables, which allow for a very clean and user-friendly display. Implementing these tables for the most part was not difficult, however the main display of patient information was trickier as I needed to implement the data tables in such a way so as to show various categories of patient data on a button click within the same table window.

This effect was eventually achieved through a series of hiding and revealing various columns within the same window and re-drawing the table each time. This method works very well and provides a comprehensive visual representation of the large volume of patient data contained within the application.

Testing

1.13 Black Box testing

Black Box testing is the process of testing the required functionality of the application without seeing the source code. The tester chosen had no previous I.T programming knowledge and was only made aware of what the application should do. ("Black Box Testing – Software Testing Fundamentals")

Black Box Test 1			
AUT Name	Log in	**Version**	1.0
Iteration ID	1.0	**Date of Test**	26/04/2017

Test ID	BlackBox1
Purpose of Test	The purpose of this test is to ensure that a user with a valid username and password can gain access to the application.
Test Environment	The test environment was as follows: 27 inch (iMac- mid 2011) Processor: 3.4 Ghz Intel Core i7 running on macOS Sierra version 10.13.4
Test Steps	Once navigated to index.php the user should be able to successfully login using the credentials provided. Username: test1 Password: test1
Expected Result	The user should be allowed access to the application and be re-directed to dashboard.php
Actual result	The user was permitted access and re-directed as expected.

Figure 15 *– BlackBox1 (log in)*

Black Box Test 2			
AUT Name	Add Patient	**Version**	1.0
Iteration ID	1.0	**Date of Test**	26/04/2017

Test ID	BlackBox2
Purpose of Test	The purpose of this test is to ensure that a user can successfully input a new patient record into the system.
Test Environment	The test environment was as follows: 27 inch (iMac- mid 2011) Processor: 3.4 Ghz Intel Core i7 running on macOS Sierra version 10.13.4
Test Steps	Once a user has navigated to patient.php by clicking on “Add Patient” from the menu, they should be able to successfully input a patient record following the form prompts while doing so.

Expected Result	The user should be able to successfully input a new patient record, and then navigate to the “View Patient Records” tab in order to ensure the record has been inputted.
Actual result	The user successfully completed the new patient form and followed the prompts that ensure data is entered correctly. The new patient appeared in the list of patients as expected.

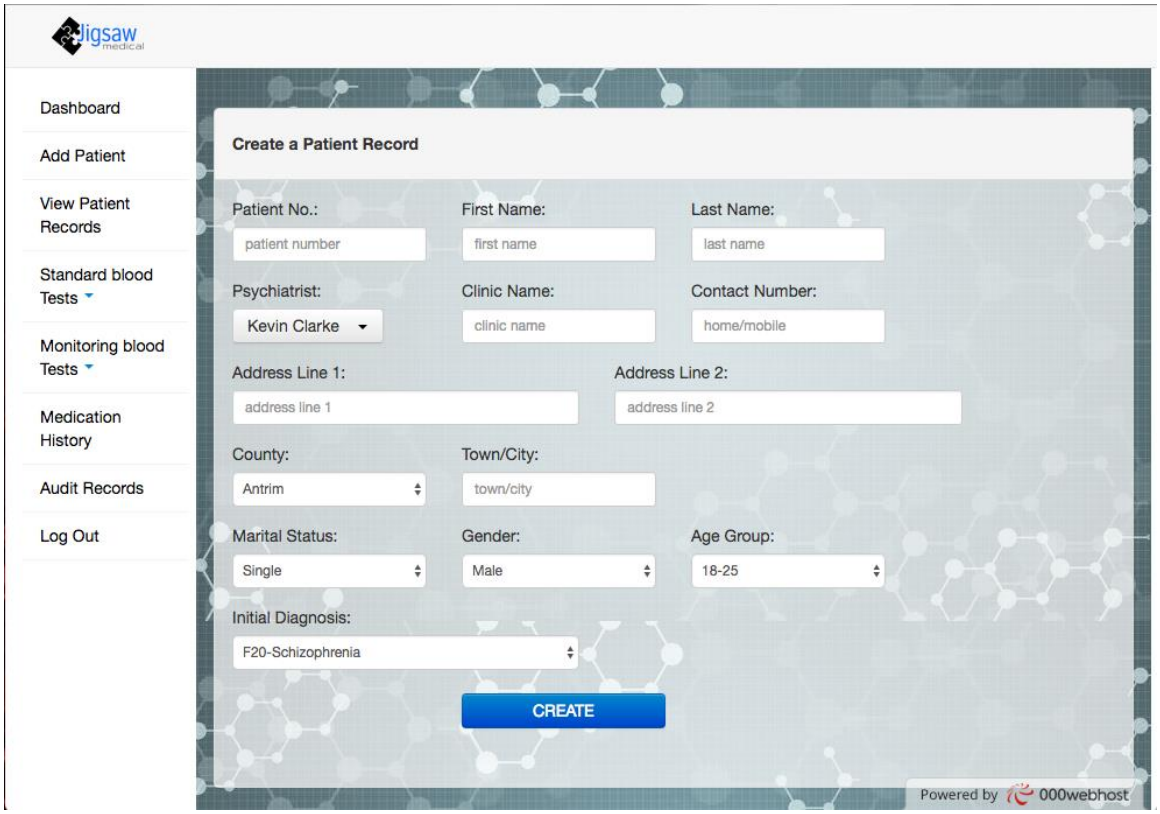


Figure 16 *- BlackBox2 (Add patient)*

Black Box Test 3			
AUT Name	View Patient Records	**Version**	1.0
Iteration ID	1.0	**Date of Test**	26/04/2017

Test ID	BlackBox3
Purpose of Test	The purpose of this test is to ensure that a user can successfully view the data related to a patient's blood test, registration data and medication data.
Test Environment	The test environment was as follows: 27 inch (iMac- mid 2011) Processor: 3.4 Ghz Intel Core i7 running on macOS Sierra version 10.13.4
Test Steps	Once a user has navigated to the appropriate page by selecting "View Patient Records" from the navigation menu they should be presented with a table that populates with records form the chosen category that the user selects using the buttons aligned to the top of the table.
Expected Result	The user should be presented with data from the chosen category and have the ability to search and filter the data as they wish.
Actual result	The user was successfully presented with the data they wished to view and were able to search and filter the data as expected.

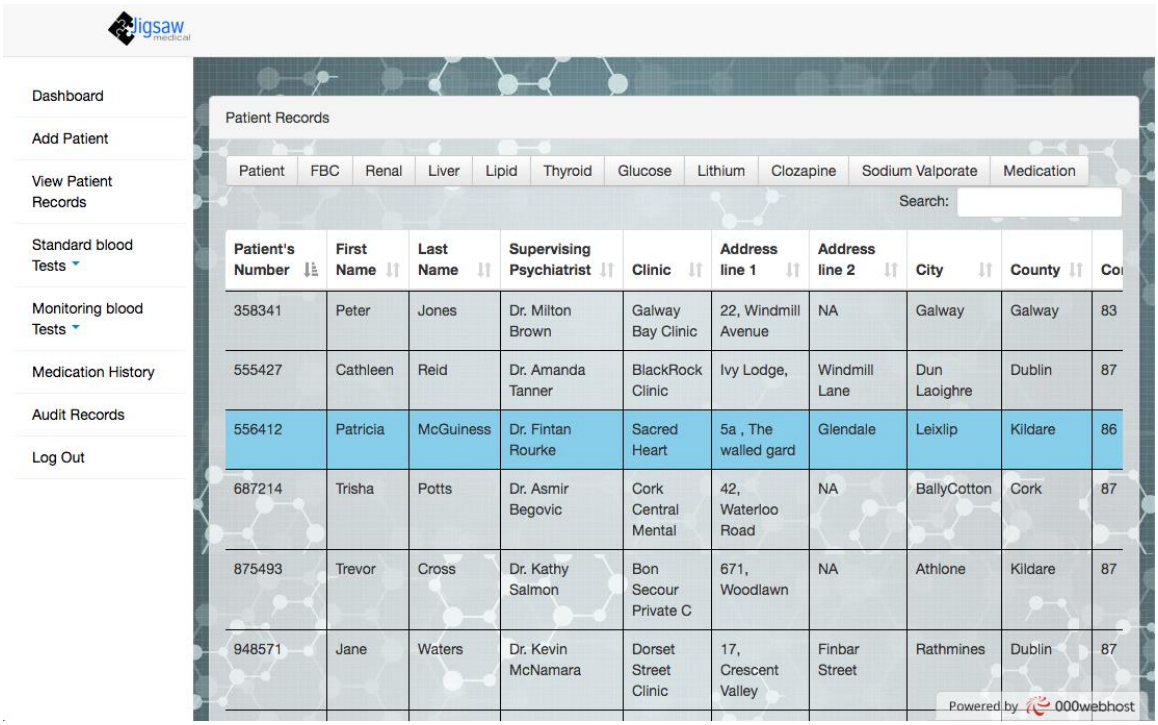


Figure 17 - *BlackBox3 (View patient records)*

Black Box Test 4			
AUT Name	Insert Full Blood Count	**Version**	1.0
Iteration ID	1.0	**Date of Test**	26/04/2017

Test ID	BlackBox4
Purpose of Test	The purpose of this test is to ensure that a user can successfully input the details of a full blood count test result into the system and assign the results to an existing user within the system.
Test Environment	The test environment was as follows: 27 inch (iMac- mid 2011) Processor: 3.4 Ghz Intel Core i7 running on macOS Sierra version 10.13.4

Test Steps	Once the user has navigated to the appropriate form by choosing “Full Blood Count” from the navigation menu, they should then choose a patient from the table that they want to assign the results to. The user should then proceed to fill out the form fields and submit the record to the database.
Expected Result	The user should be able to successfully follow the steps detailed above and submit a Full Blood Count record to the database.
Actual result	The user successfully submitted the form to the database assigned to the intended patient.

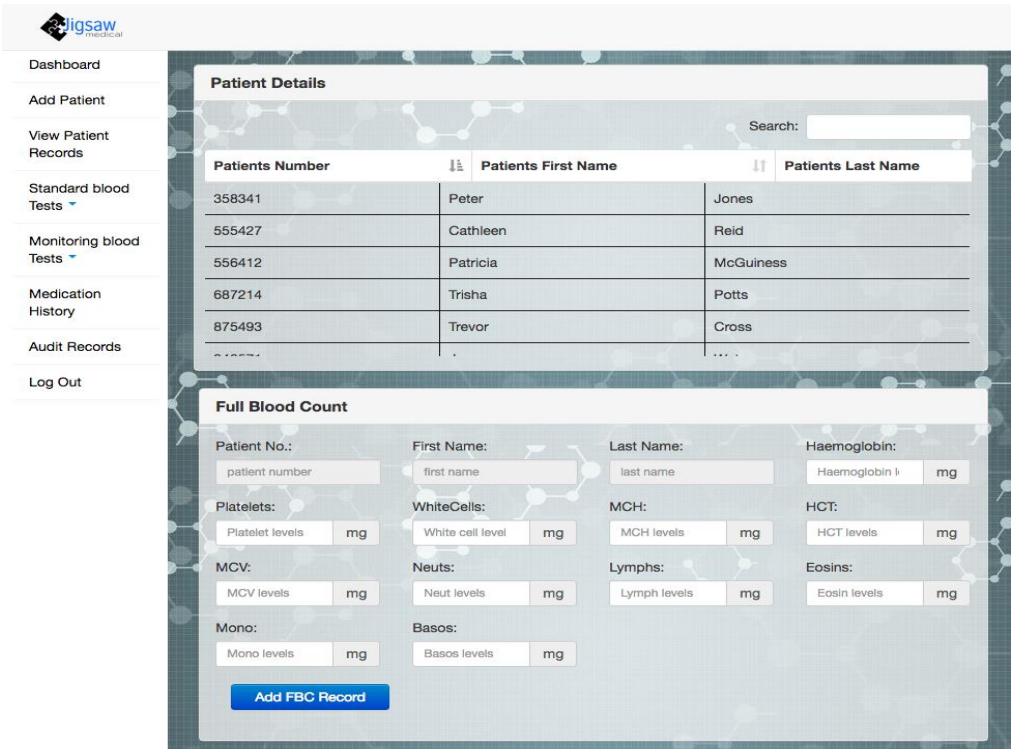


Figure 18 - *BlackBox4 (Insert Full Blood Count)*

Black Box Test 5			
AUT Name	Insert Renal profile	**Version**	1.0
Iteration ID	1.0	**Date of Test**	26/04/2017

Test ID	BlackBox5
Purpose of Test	The purpose of this test is to ensure that a user can successfully input the details of a Renal test result into the system and assign the results to an existing user within the system.
Test Environment	The test environment was as follows: 27 inch (iMac- mid 2011) Processor: 3.4 Ghz Intel Core i7 running on macOS Sierra version 10.13.4
Test Steps	Once the user has navigated to the appropriate form by choosing “Renal Profile” from the navigation menu, they should then choose a patient from the table that they want to assign the results to. The user should then proceed to fill out the form fields and submit the record to the database.
Expected Result	The user should be able to successfully follow the steps detailed above and submit a “Renal Profile” record to the database.
Actual result	The user successfully submitted the form to the database assigned to the intended patient.

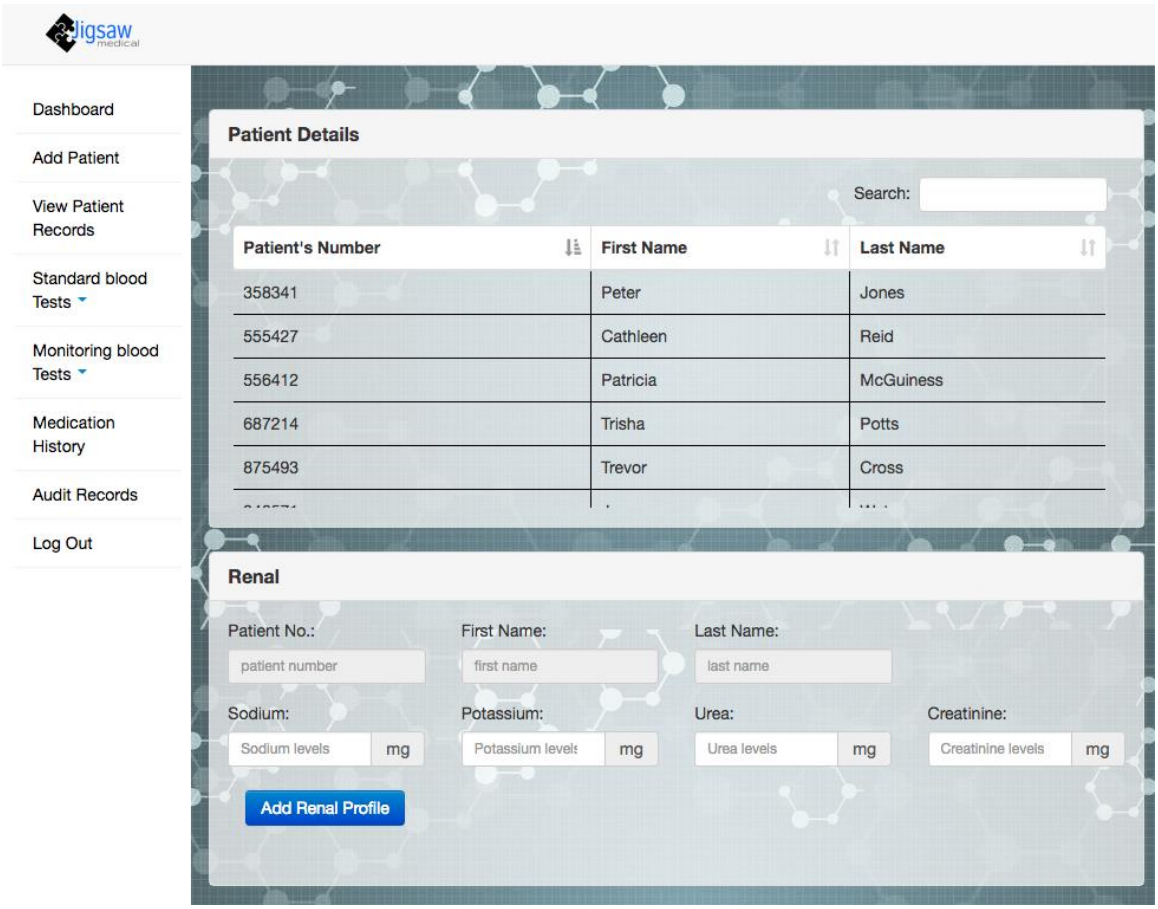


Figure 19 - *BlackBox5 (Insert Renal profile)*

Black Box Test 6			
AUT Name	Insert Liver Profile	**Version**	1.0
Iteration ID	1.0	**Date of Test**	26/04/2017

Test ID	BlackBox6
Purpose of Test	The purpose of this test is to ensure that a user can successfully input the details of a Liver test result into the system and assign the results to an existing user within the system.

Test Environment	The test environment was as follows: 27 inch (iMac- mid 2011) Processor: 3.4 Ghz Intel Core i7 running on macOS Sierra version 10.13.4
Test Steps	Once the user has navigated to the appropriate form by choosing “Liver Profile” from the navigation menu, they should then choose a patient from the table that they want to assign the results to. The user should then proceed to fill out the form fields and submit the record to the database.
Expected Result	The user should be able to successfully follow the steps detailed above and submit a “Liver Profile” record to the database.
Actual result	The user successfully submitted the form to the database assigned to the intended patient.

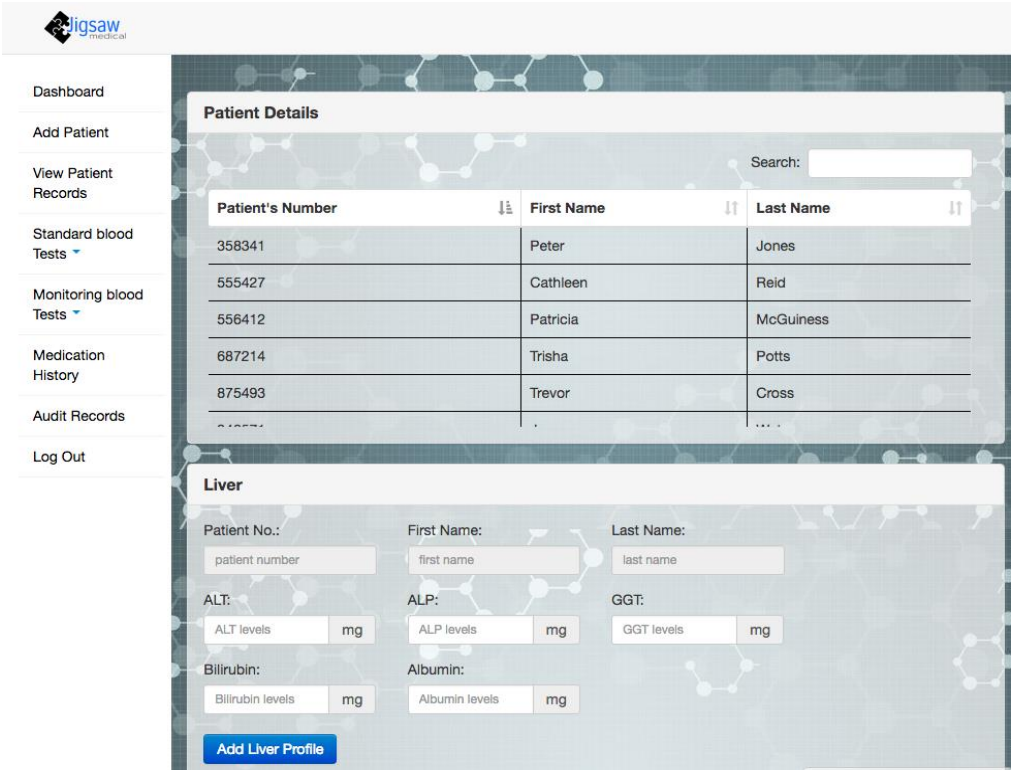


Figure 20 - *BlackBox6 (Insert Liver profile)*

Black Box Test 7			
AUT Name	Insert Lipid Profile	**Version**	1.0
Iteration ID	1.0	**Date of Test**	26/04/2017

Test ID	BlackBox7
Purpose of Test	The purpose of this test is to ensure that a user can successfully input the details of a Lipid test result into the system and assign the results to an existing user within the system.
Test Environment	The test environment was as follows: 27 inch (iMac- mid 2011) Processor: 3.4 Ghz Intel Core i7 running on macOS Sierra version 10.13.4

Test Steps	Once the user has navigated to the appropriate form by choosing “Lipid Profile” from the navigation menu, they should then choose a patient from the table that they want to assign the results to. The user should then proceed to fill out the form fields and submit the record to the database.
Expected Result	The user should be able to successfully follow the steps detailed above and submit a “Lipid Profile” record to the database.
Actual result	The user successfully submitted the form to the database assigned to the intended patient.

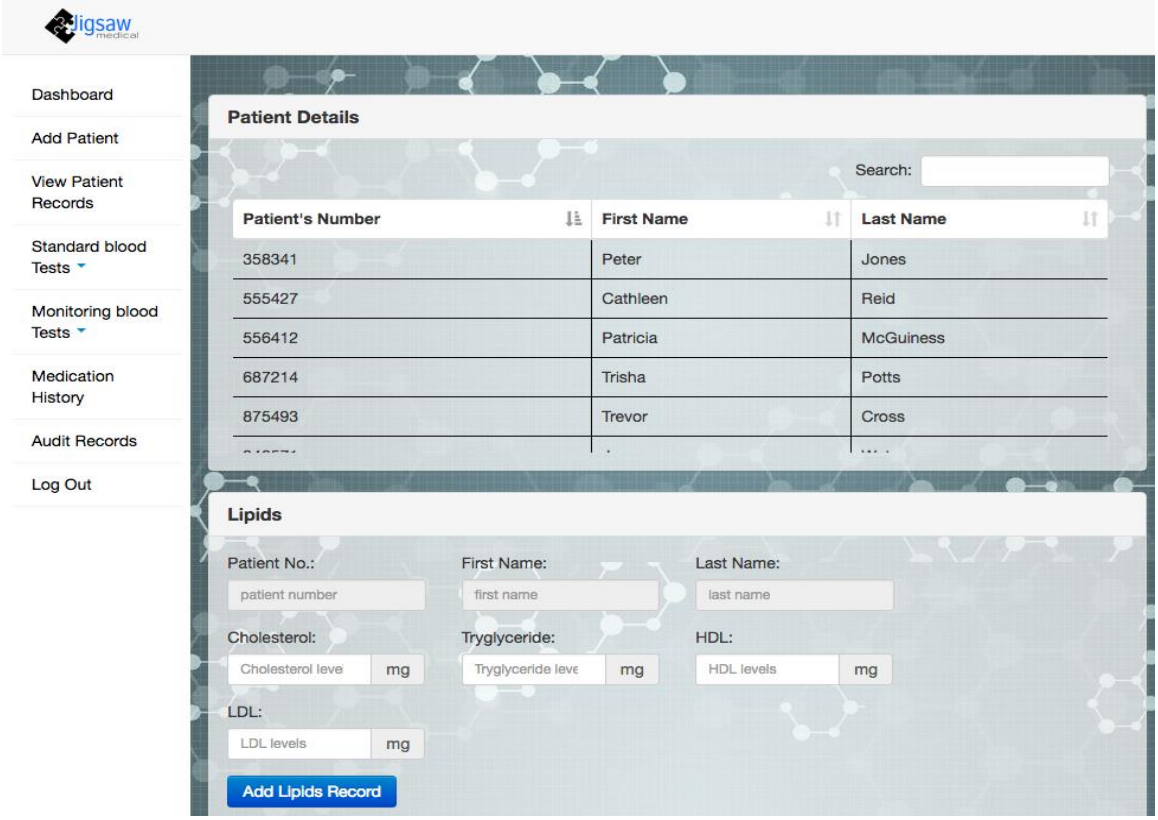


Figure 21 - *BlackBox7 (Insert Lipid profile)*

Black Box Test 8			
AUT Name	Insert Thyroid function test results	**Version**	1.0
Iteration ID	1.0	**Date of Test**	26/04/2017

Test ID	BlackBox8
Purpose of Test	The purpose of this test is to ensure that a user can successfully input the details of a Thyroid function test result into the system and assign the results to an existing user within the system.

Test Environment	The test environment was as follows: 27 inch (iMac- mid 2011) Processor: 3.4 Ghz Intel Core i7 running on macOS Sierra version 10.13.4
Test Steps	Once the user has navigated to the appropriate form by choosing “Thyroid Function” from the navigation menu, they should then choose a patient from the table that they want to assign the results to. The user should then proceed to fill out the form fields and submit the record to the database.
Expected Result	The user should be able to successfully follow the steps detailed above and submit a “Thyroid Function” record to the database.
Actual result	The user successfully submitted the form to the database assigned to the intended patient.

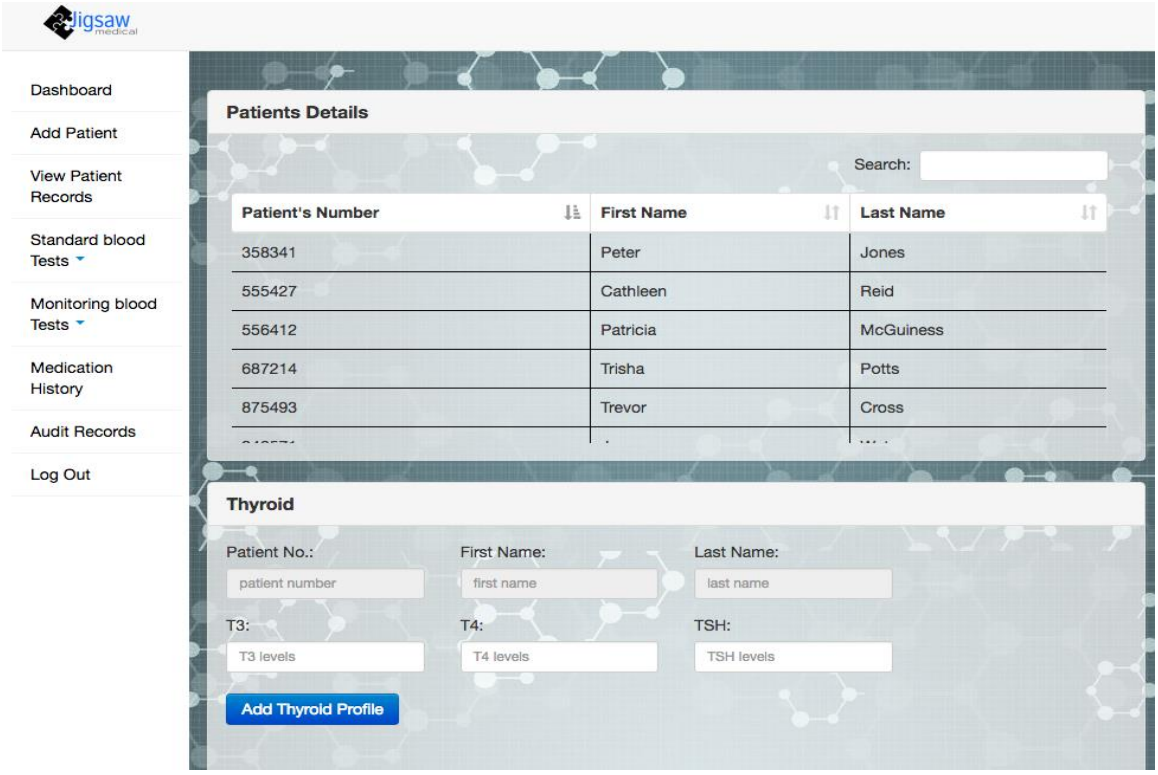


Figure 22 - *BlackBox8 (Insert Thyroid function)*

Black Box Test 9			
AUT Name	Insert Glucose results	**Version**	1.0
Iteration ID	1.0	**Date of Test**	26/04/2017

Test ID	BlackBox9
Purpose of Test	The purpose of this test is to ensure that a user can successfully input the details of a Glucose test result into the system and assign the results to an existing user within the system.
Test Environment	The test environment was as follows: 27 inch (iMac- mid 2011) Processor: 3.4 Ghz Intel Core i7 running on macOS Sierra version 10.13.4

Test Steps	Once the user has navigated to the appropriate form by choosing “Glucose” from the navigation menu, they should then choose a patient from the table that they want to assign the results to. The user should then proceed to fill out the form fields and submit the record to the database.
Expected Result	The user should be able to successfully follow the steps detailed above and submit ”Glucose” record to the database.
Actual result	The user successfully submitted the form to the database assigned to the intended patient.

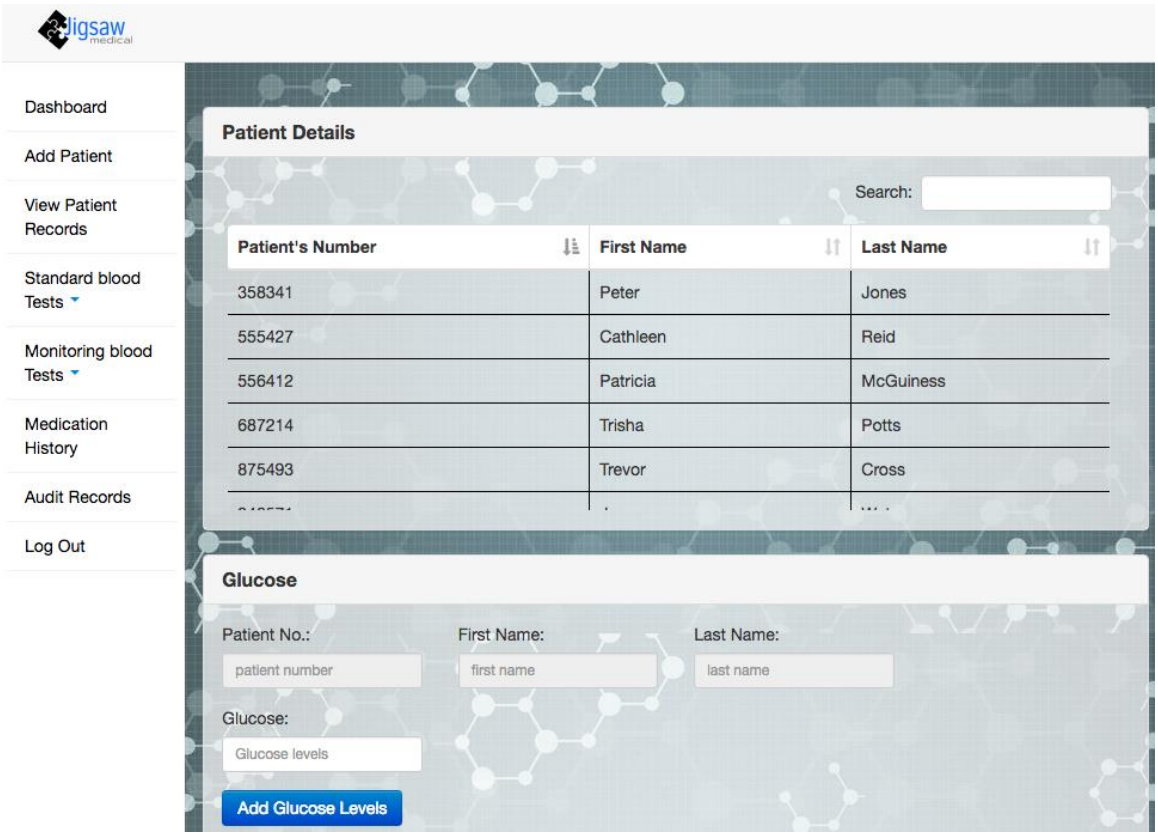


Figure 23 - *BlackBox9 (Insert Glucose)*

Black Box Test 10			
AUT Name	Insert Lithium levels	**Version**	1.0
Iteration ID	1.0	**Date of Test**	26/04/2017

Test ID	BlackBox10
Purpose of Test	The purpose of this test is to ensure that a user can successfully input the details of a Lithium levels test result into the system and assign the results to an existing user within the system.
Test Environment	The test environment was as follows:

	27 inch (iMac- mid 2011) Processor: 3.4 Ghz Intel Core i7 running on macOS Sierra version 10.13.4
Test Steps	Once the user has navigated to the appropriate form by choosing “Lithium levels” from the navigation menu, they should then choose a patient from the table that they want to assign the results to. The user should then proceed to fill out the form fields and submit the record to the database.
Expected Result	The user should be able to successfully follow the steps detailed above and submit a “Lithium levels” record to the database.
Actual result	The user successfully submitted the form to the database assigned to the intended patient.

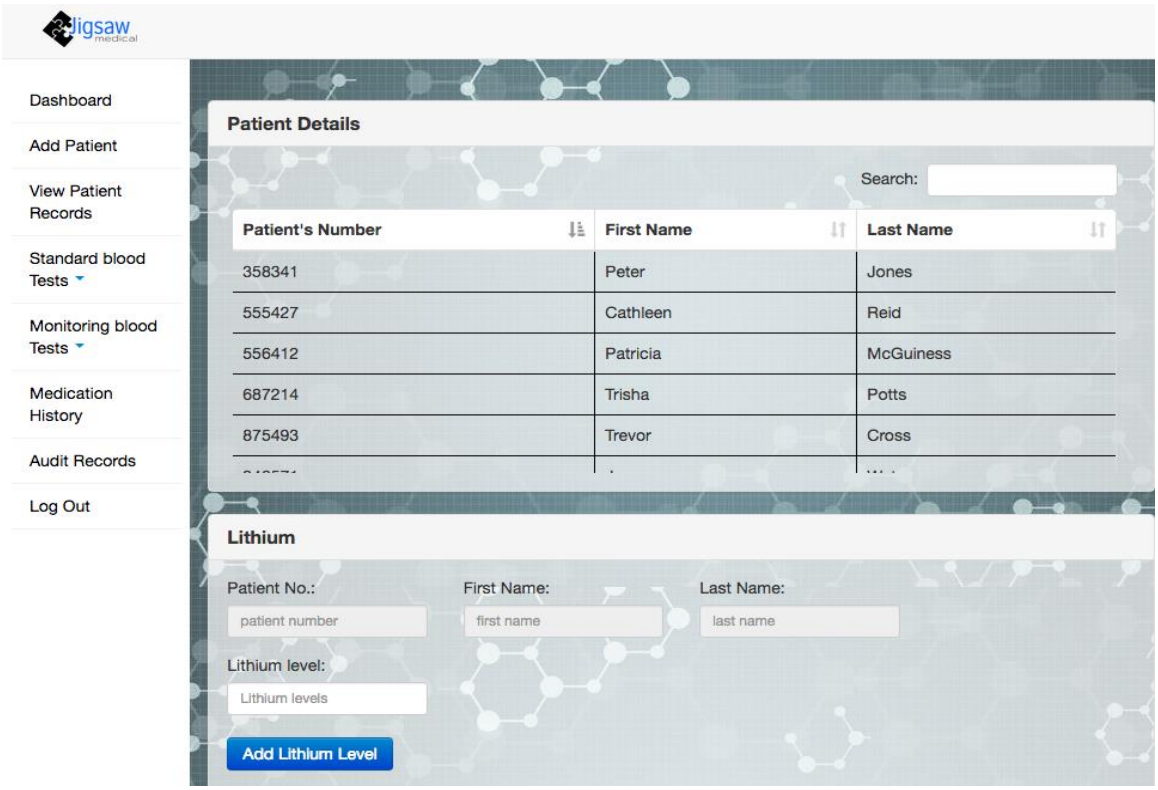


Figure 24 - *BlackBox10 (Insert Lithium levels*

Black Box Test 11			
AUT Name	Insert Sodium Valproate Levels	**Version**	1.0
Iteration ID	1.0	**Date of Test**	26/04/2017

Test ID	BlackBox11
Purpose of Test	The purpose of this test is to ensure that a user can successfully input the details of a Sodium Valproate levels test result into the system and assign the results to an existing user within the system.
Test Environment	The test environment was as follows: 27 inch (iMac- mid 2011) Processor: 3.4 Ghz Intel Core i7 running on macOS Sierra version 10.13.4
Test Steps	Once the user has navigated to the appropriate form by choosing “Sodium Valproate levels” from the navigation menu, they should then choose a patient from the table that they want to assign the results to. The user should then proceed to fill out the form fields and submit the record to the database.
Expected Result	The user should be able to successfully follow the steps detailed above and submit a “Sodium Valproate levels” record to the database.
Actual result	The user successfully submitted the form to the database assigned to the intended patient.

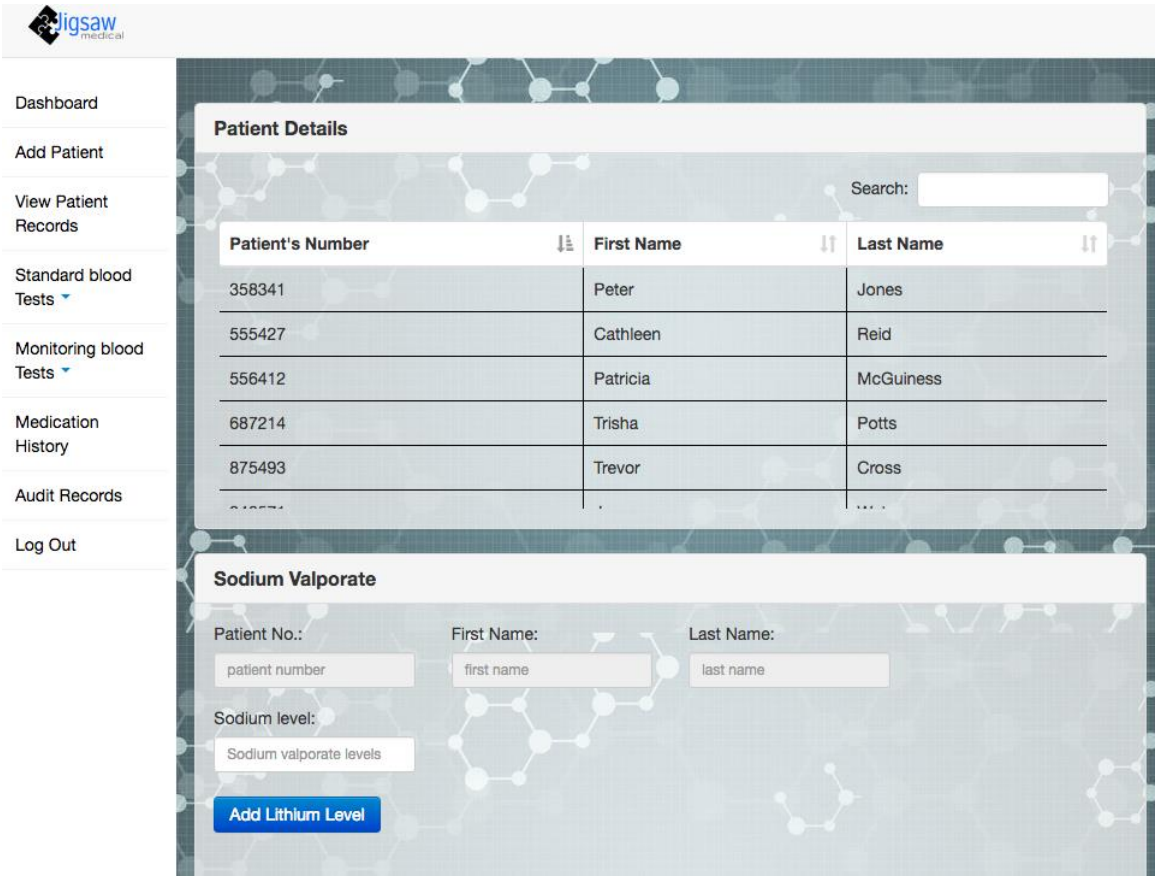


Figure 25 - *BlackBox11 (Insert Sodium Valproate levels)*

Black Box Test 12			
AUT Name	Insert Clozapine Levels	**Version**	1.0
Iteration ID	1.0	**Date of Test**	26/04/2017

Test ID	BlackBox12
Purpose of Test	The purpose of this test is to ensure that a user can successfully input the details of a Clozapine levels test result into the system and assign the results to an existing user within the system.
Test Environment	The test environment was as follows:

	27 inch (iMac- mid 2011) Processor: 3.4 Ghz Intel Core i7 running on macOS Sierra version 10.13.4
Test Steps	Once the user has navigated to the appropriate form by choosing “Clozapine levels” from the navigation menu, they should then choose a patient from the table that they want to assign the results to. The user should then proceed to fill out the form fields and submit the record to the database.
Expected Result	The user should be able to successfully follow the steps detailed above and submit a “Clozapine levels” record to the database.
Actual result	The user successfully submitted the form to the database assigned to the intended patient.

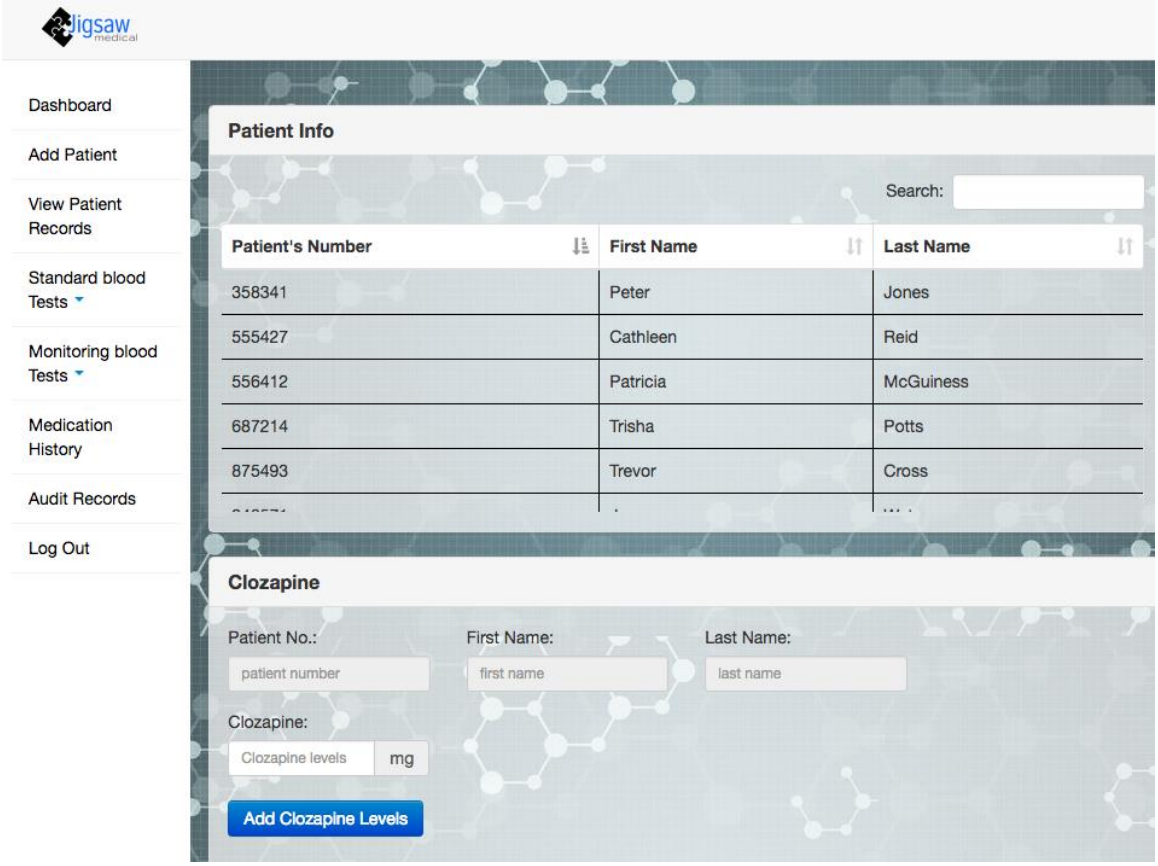


Figure 26 - *BlackBox12 (Insert Clozapine levels)*

Black Box Test 13			
AUT Name	Log into Medications section	**Version**	1.0
Iteration ID	1.0	**Date of Test**	26/04/2017

Test ID	BlackBox13
Purpose of Test	The purpose of this test is to ensure that a user with medication access rights can successfully log into the medications section of the application.

Test Environment	The test environment was as follows: 27 inch (iMac- mid 2011) Processor: 3.4 Ghz Intel Core i7 running on macOS Sierra version 10.13.4
Test Steps	Once the user has navigated to the appropriate page by clicking the “Medications” link from the navigation bar they should be presented with a login page requesting their username and password. Once they enter their credentials they should be permitted access if those credentials are associated with the appropriate access rights. (In this case the user was provided with the appropriate credentials to access this section)
Expected Result	The user should be able to successfully log into the medications.php page.
Actual result	The user successfully submitted logged into the medications.php page.

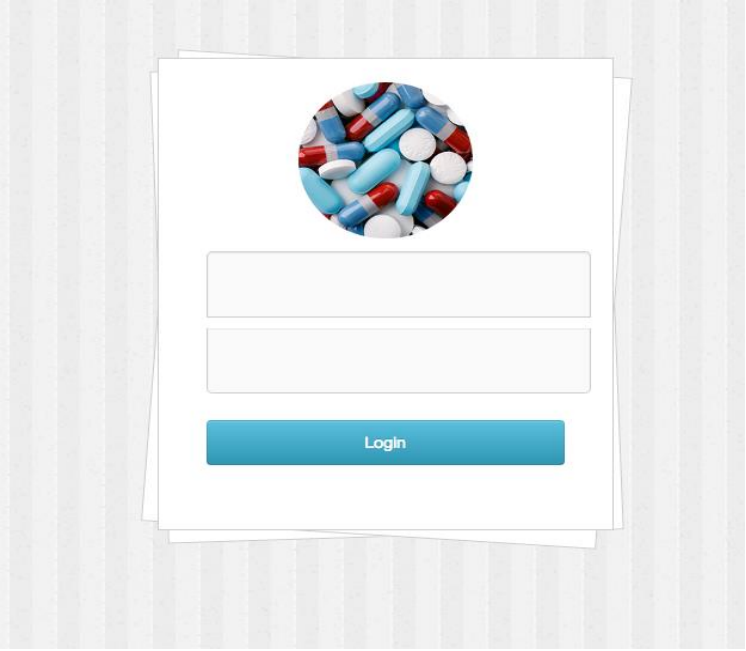


Figure 27 - *BlackBox13 (Medication Login*

Black Box Test 14			
AUT Name	Add a medication record for a patient	**Version**	1.0
Iteration ID	1.0	**Date of Test**	26/04/2017

Test ID	BlackBox14
Purpose of Test	The purpose of this test is to ensure that a user can successfully add a medication record to a patient using the form fields provided.
Test Environment	The test environment was as follows: 27 inch (iMac- mid 2011) Processor: 3.4 Ghz Intel Core i7 running on macOS Sierra version 10.13.4
Test Steps	Once the user has navigated to the appropriate form by choosing “Medications” from the navigation menu and logging in, they should then choose a patient from the table that they want to assign the medication dosage to. The user should then proceed to fill out the form fields and submit the record to the database.
Expected Result	The user should be able to successfully add a medication record for a patient chosen form the system.
Actual result	The user successfully added a medication record to the chosen patient record.

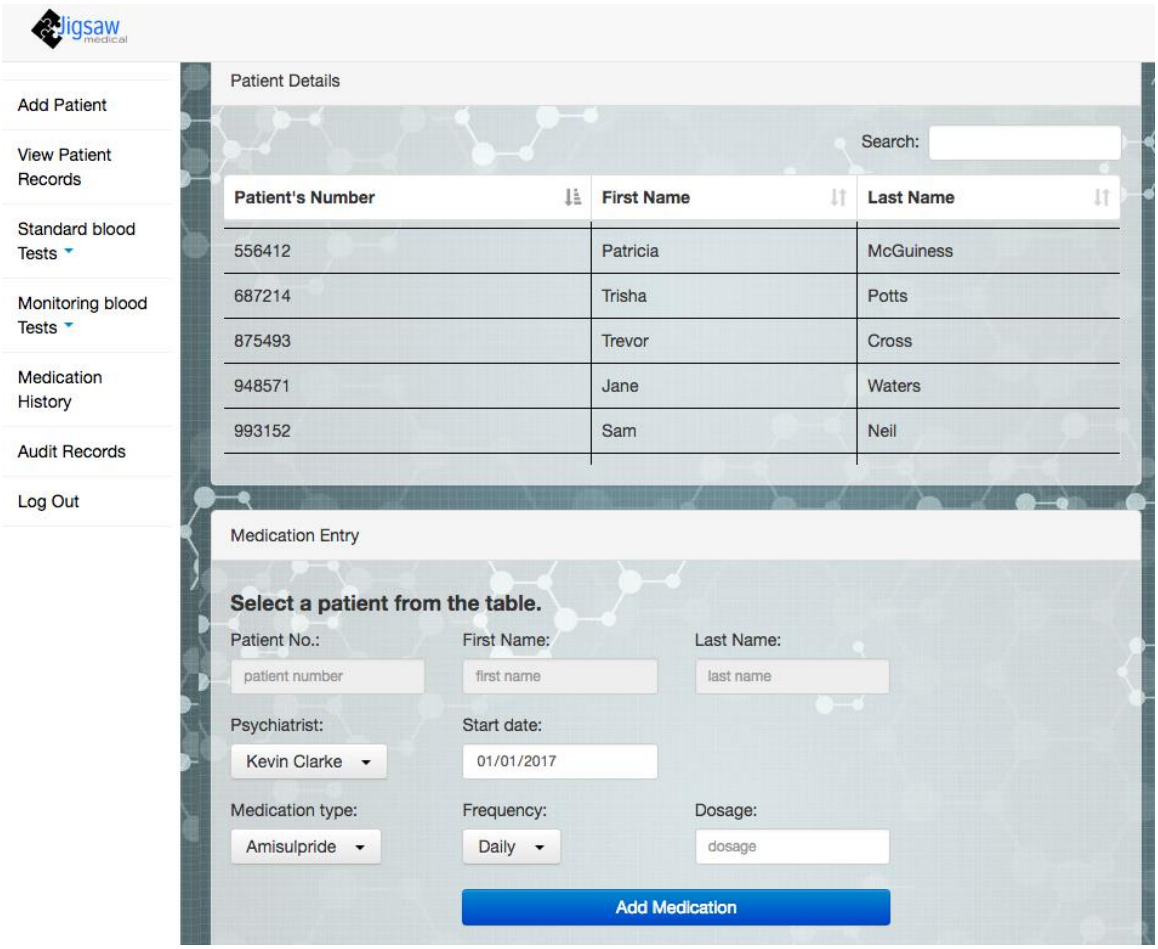


Figure 28 - *BlackBox14 (Insert Medication record)*

Black Box Test 15			
AUT Name	Log into Audit section	**Version**	1.0
Iteration ID	1.0	**Date of Test**	26/04/2017

Test ID	BlackBox15
Purpose of Test	The purpose of this test is to ensure that a user with audit access rights can successfully log into the audit section of the application.

Test Environment	The test environment was as follows: 27 inch (iMac- mid 2011) Processor: 3.4 Ghz Intel Core i7 running on macOS Sierra version 10.13.4
Test Steps	Once the user has navigated to the appropriate page by clicking the “Audit Records” link from the navigation bar they should be presented with a login page requesting their username and password. Once they enter their credentials they should be permitted access if those credentials are associated with the appropriate access rights. (In this case the user was provided with the appropriate credentials to access this section)
Expected Result	The user should be able to successfully log into the audit.php page.
Actual result	The user successfully submitted logged into the audit.php page.

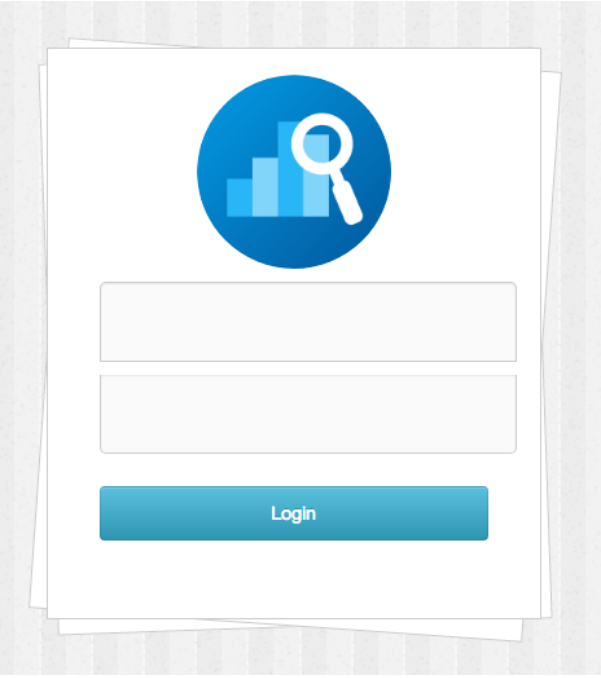


Figure 29 - *BlackBox15 (Audit Login)*

Black Box Test 16

AUT Name	View the Audit section and export a report.	**Version**	1.0
Iteration ID	1.0	**Date of Test**	26/04/2017

Test ID	BlackBox16
Purpose of Test	The purpose of this test is to ensure that a user can successfully view the audit section of the application and export a report.
Test Environment	The test environment was as follows: 27 inch (iMac- mid 2011) Processor: 3.4 Ghz Intel Core i7 running on macOS Sierra version 10.13.4
Test Steps	Once the user has navigated to the appropriate form by choosing “Audit Records” from the navigation menu and logging in, they should then be able to view the audit table and the accompanying graph that visualizes some of the key data from the table. A report of the table details can be extracted in several formats.
Expected Result	The user should be able to successfully view the audit table and export a report.
Actual result	The user successfully viewed the audit table and graph. In addition they extracted a report from the table in PDF format.

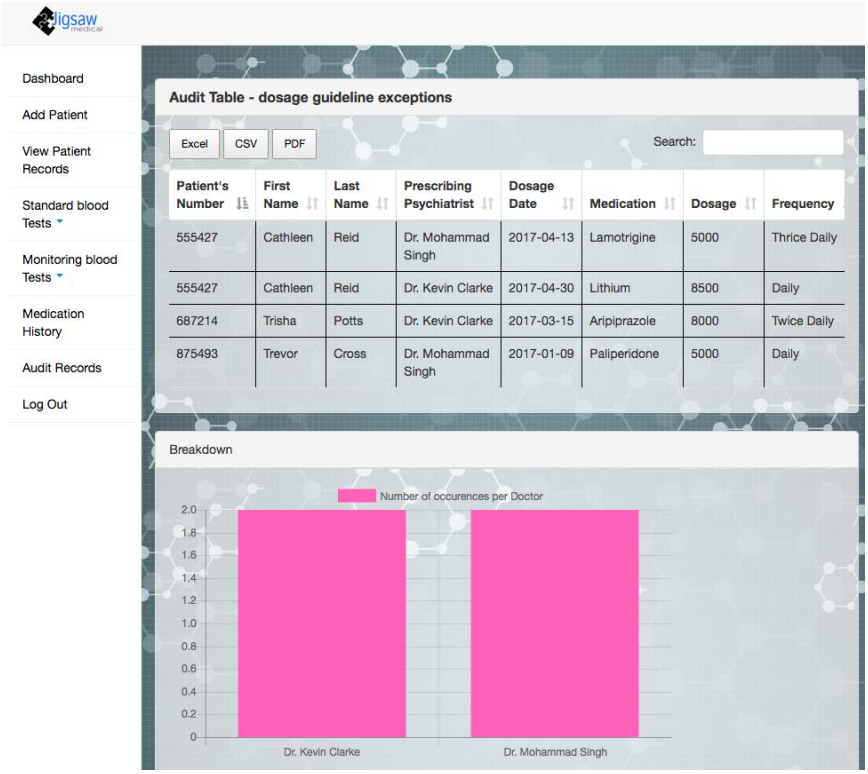


Figure 30 - *BlackBox16 (View Audit records)*

Audit Data - Excess dosages per doctor

Patient's Number	First Name	Last Name	Prescribing Psychiatrist	Dosage Date	Medication	Dosage	Frequency
555427	Cathleen	Reid	Dr. Mohammad Singh	2017-04-13	Lamotrigine	5000	Thrice Daily
555427	Cathleen	Reid	Dr. Kevin Clarke	2017-04-30	Lithium	8500	Daily
687214	Trisha	Potts	Dr. Kevin Clarke	2017-03-15	Aripiprazole	8000	Twice Daily
875493	Trevor	Cross	Dr. Mohammad Singh	2017-01-09	Paliperidone	5000	Daily

Figure 31 - *BlackBox16 (Sample Audit report)*

1.14 Usability testing

I focused on 3 different types of usability testing for the *"Jigsaw medical application"*, - five-second test, trunk test and think aloud test.

I selected 3 participants to take part in these testing exercises. All 3 of these participants are medical professionals and are all people who could realistically come in contact with the application in a real world environment.

- Ana-Maria, a 29 year old Psychiatrist currently working for the HSE in Limerick.
- Sorcha, a 40 year old Psychiatrist currently working for the HSE in Dublin.
- Cora, a 53 year old Advanced Nurse Practitioner working for the HSE in Dublin.

Each user was told that the whole process was to be informal and they couldn't ask me questions during the testing process as this would defeat the purpose of the testing. ("Usability First - Methods - Usability Testing | Usability First")

1.15 Testing techniques employed

1.15.1 Five Second Test

The first activity that I had the users complete was the "five second test". This is a method of testing usability by exposing different pages of the application to a user for only 5 seconds. Once the 5 seconds is complete the page is removed and the participants are asked for their initial impressions and what they remember seeing on the page. ("Pidoco - The Rapid Prototyping Tool")

I presented the main dashboard to the participants and posed the following questions to them.

- What does the application do?
- What did you like or dislike about the design?
- What stuck out the most on the page?
- Any other general observations?

The results that were generated from this test were interesting. Firstly 2 out of the 3 participants were able to ascertain that it was a medical application, and the 3rd participant said they were unable to see the logo or understand what they were seeing in such a short timeframe.

All 3 participants said they noticed the navigation bar running down the side of the page and that it looked straightforward.

2 of the 3 participants stated that they liked the color scheme and that it stood out more than the other features that they noticed.

Some other general observations that were received from the test showed that the page looked misaligned and a more symmetrical feel would be appreciated.

I gathered all of this feedback and endeavored to correct as many of the issues that were identified as possible.

1.15.2 Trunk Test

The next test that the 3 participants were asked to take part in was the trunk test. The purpose of the trunk test is to ascertain how well a website does in terms of navigation. ("Elements Of Effective Web Design: Trunk Test")

The 5 questions that I posed to the participants were:

1. What site is this?
2. What page am I on?
3. What are the main sections?
4. What navigations do I have?
5. Where am I overall?

The participants were placed on a random page within the application and asked to answer these questions.

The results showed that each participant was able to identify what site they were on and also what page they had been sent to. All 3 participants stated the logo and address bar were clear and identified the application as *"Jigsaw medical".*

All 3 participants were also able to determine what the main sections of the application were by reading the headings contained both within the navigation bar and also the headings on the page itself.

All participants stated the navigation bar was very clear and concise. 1 user mentioned that the navigation didn't leave any ambiguity as to how to navigate around the application.

The question that scored the worst was number 5. All 3 participants said they could not determine exactly where they were. This feedback then led me to alter the URL in order to provide clearer feedback as to what section a user was currently in.

1.15.3 Think Aloud

The last test that I gave to the participants was a 'Think Aloud Test'. The concept behind this test is to ask a user to complete a specific task and talk you through it as they are completing the task. ("Think Aloud Protocol")

I posed one main task to the users with the following steps:

1. Log in to the application.
2. Navigate to the *"Add Patient Records"* section.
3. Create a patient.
4. Navigate to the *"View Records"* section.
5. Search for record that was just created.
6. Logout of the application.

I have compiled the results of this test in a table so as to better visualise the feedback.

User	Log in	Navigate to "Add Patient Records"	Create a Patient	Navigate to "View Records"	Search for the record created.	Log out
Ana-Maria	straightfoward	easy navigation.	Helpful hints on the form fields.	simple	easy because the record was already on top of the table as most recent.	no issues.
Sorcha	easy to use	clear and easy to understand	restrictive, but straightforward	simple	This was a straightforward task.	it logged me out staight away, it should probably prompt me.
Cora	simple	nothing complicated, standard navigation.	very easy	very easy	simple and intuitive.	simple process.

Figure 32 - *Feedback table*

Overall the results were favourable. Each user was happy with the navigation and log in process. The one area were I received less than positive feedback was log-out. 1 user indicated they felt the system should prompt the user if they really mean to log-out.

1.16 Heuristic Evaluation

A heuristic evaluation is a usability inspection method for computer software that helps to identify usability problems in the user interface design (Uxness.in, 2017). Generally regarded as the most popular method for achieving such evaluation, is 'Jakob Nielsen's 10 general principles for interaction design'.

To carry out this Heuristic evaluation I asked the same 3 users to participate and applied the principles set out by Nieslen and Molich. I had each of the three group members inspect the applications interface and critiqued the application against the guidelines described below. ("10 Heuristics For User Interface Design: Article By Jakob Nielsen")

The heuristic evaluation that was carried out for the "*Jigsaw medical"* application used the most up to date version of Nielsen's list of Heuristics. Below is a detailed list of the guidelines I used.

	Heuristic	Description
H1	Visibility of System Status	The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
H2	Match Between System and the Real World	The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.
H3	User Control and Freedom	Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.
H4	Consistency and Standards	Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.
H5	Error Prevention	Even better than good error messages is a careful design which prevents a problem from occurring in the first place.
H6	Recognition Rather than Recall	Make objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.
H7	Flexibility and Efficiency of Use	Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.
H8	Aesthetic and Minimalist Design	Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.
H9	Error Recovery	Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.
H10	Help and Documentation	Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

Figure 33 - *Heuristic Evaluation guidelines*

1.16.1 Individual Evaluations

I requested that each participant carry out an independent viewing and analysis of the “Jigsaw medical” application. During this analysis each participant was tasked with comparing the site to the guidelines provided.

Below is a detailed breakdown of the results produced by each group member.

Users**:** *Ana-Maria, Sorcha, Cora*

1.16.2 H1 – Visibility of System status

Ana-Maria

I found the application to be informative and easy to use. It was always clear when the system was performing an action, and provided me with clear feedback.

Sorcha

The application information was clear and didn’t leave me confused.

Cora

The website kept me well informed.

1.16.3 H2 – Match between System and the Real World

Ana-Maria

The application was laid out in a way that made sense and didn’t confuse me.

Sorcha

All of the language I encountered was in English. In addition the site was worded so as to be familiar.

Cora

Everything on the website appeared to be in a logical order and the layout didn't leave me confused.

1.16.4 H3 – User Control and Freedom

Ana-Maria

I found it easy to navigate away from a page when I wanted to leave, however I didn't specifically see a back button on the site.

Sorcha

I didn't come across an extended dialogue when trying to leave a page and navigating away seemed straightforward.

Cora

I couldn't find a back button on the page but it was easy to go to a different page whenever I wanted.

1.16.5 H4 – Consistency and Standards

Ana-Maria

The language and labeling of buttons on the site was very unambiguous and didn't leave me in any doubt as to what function would be carried out.

Sorcha

The functionality and meaning of the different pages and forms was clear and straightforward.

Cora

I felt the site was clear and used language that was clear in its meaning.

1.16.6 H5 – Error Prevention

Ana-Maria

I came across one error while navigating the site. The audit login page allowed me access without a username and password being populated.

Sorcha

I didn’t come across any errors while navigating the site and there seemed to be a lot of effort put into preventing errors.

Cora

I didn’t find any errors when using the application.

1.16.7 H6 – Recognition Rather than Recall

Ana-Maria

The site was clear in its instructions and didn’t require me to remember any information when navigating.

Sorcha

The one thing I noticed is a lack of labels on some of the forms. I would have liked to see clearer labeling.

Cora

I thought the site was very instructive and didn't require me to remember anything from one page to another.

1.16.8 H7 – Flexibility and Efficiency of Use

Ana-Maria

The site was easy to use but I didn't notice anything in particular that improved ease of use.

Sorcha

I found the site easy to use and felt the audit graphs improved the efficiency of the site as it allowed me to clearly see which Doctors were overprescribing rather than searching the table for each Doctor.

Cora

I felt the ability to click a patient in the table and have the details populate in the forms was a great addition that really improved the usability and efficiency of the site.

1.16.9 H8 – Aesthetic and Minimalist Design

Ana-Maria

I found the design to be clean and simple. However, in places I feel more labeling is required to ensure users know exactly what to do.

Sorcha

I really enjoyed the minimalist design employed across the site. I found it to be uncluttered and intuitive.

Cora

The site is really straightforward and doesn't confuse the eye.

1.16.10 H9 – Error Recovery

Ana-Maria

I didn't encounter any error messages while using the site so I can't speak to their clarity and usefulness.

Sorcha

I can't answer this question, as I didn't see any error messages.

Cora

I didn't necessarily see any error messages but I was prompted to complete form fields if I inputted incorrect data. These messages were very clear and helpful.

1.16.11 H10 – Help and Documentation

Ana-Maria

The site doesn't contain any specific help documentation. I would suggest a FAQ section is added at the very least.

Sorcha

The site doesn't have any help documentation however it's very straightforward and in my opinion doesn't need any.

Cora

I didn't find any help documentation beyond the form input prompts when I inputted incorrect data.

1.17 Evaluation & UAT results

This site was evaluated by putting the application in the hands of some potential end users to gauge if the application satisfied their requirements. This stage was important because even though the application passed the functional test that doesn't mean that it functions in a manner that solves the users problem.

Some of the initial feedback I received from users was that while the main functionality was correct they would like to see a number of changes to the aesthetics of the site to make it more user friendly.

List of suggested changes:

- Placeholder text to be included on all login fields throughout the application.
- Chart colors to be muted and brought in line with the styling of the application as a whole.
- Modifications to the naming of certain navigation items in order to be clearer to the end user.
- The addition of variety in the charts types. All charts were bar charts and it was suggested some data would be better represented in a different style.

In addition to the more general feedback I received on improving the aesthetics, I also received feedback from a well-informed tester who suggested that a

FAQ/landing page would be the appropriate first page of the application rather than the dashboard.

With this feedback in mind I proceeded to implement as many corrections and improvements as possible into my final code.

Security

Due to the sensitive nature of the data contained within the application it was important for me to maintain the security requirements of the application in the forefront of my mind through out the development process. Implementing appropriate security features was difficult due to a lack of any formal instruction in this field. Through online research I discovered best practice techniques for website security and incorporated as many features as my experience level would allow. ("Web Application Security: Methods And Best Practices")

- The first form of security I have implemented is hosting on a platform that allows for https connections. Having a https connection is one of the first things a user will identify about a new website that will be holding personal information. Https allows for the encrypted transmission of data between severs and adds a first line of defense against anyone with malicious intent.

- When a user attempts to access the site for the first time they must be using a device that has an IP address within a selected range, otherwise they will receive an error page. This segregation of the system allows for the application to be restricted to computers within a particular domain or company and makes it more difficult for external threats to steal or corrupt the data.

- If a user is accessing the application from an appropriate IP address they will be presented with a login page and they will input their username and password as provided by the system administrator. This prevents anyone not already set-up on the system from entering the application and receiving access to patient data.

- Once a user has gained access to the application they will be restricted from entering the medications and audit sections of the site unless they have been adequately permissioned in the database by the system administrator.

- The system administrator when creating valid system users has the ability to modify the permission levels of each user by selecting one of 3 access levels *(Basic, Doctor, Manager).*

    - Basic – provides access to all parts of the application except for the medication dosage input page and the audit records page.

    - Doctor- provides access to all areas of the application except the audit records page.

    - Manager – provides access to all areas of the application except for the medication dosage input page.

  (These permission levels have been based upon job roles as per stakeholder interviews and feedback from medical professionals.)

- The final security measure that I have incorporated into the application is a strict requirement on the type of characters that can be provided for each of the form fields throughout the site. By restricting the characters allowed in each field I'm helping to prevent against a malicious user attacking the site using an SQL injection technique.

Future Evolution

If I was provided with the resources and time to improve upon the current application there are a number of integrations that I think would enhance the site.

Firstly I would like to see this system implemented directly with the dispensary in hospitals in order to also monitor medication stocks and access control to the medications themselves. This would provide a further level of audit support to the health system.

I have also envisaged a mobile app that accompanies this system and will provide real time alerts to members of management as and when dosages exceeding the recommended guidelines are prescribed.

I can also see an expansion of the current system to encompass more standard charting functions such as vital statistics and the uploading and centralization of medical images (e.g. X-ray, CT, MRI).

Lastly a more in depth graphical interface that allows for drilling down into charts to view graphed records for each patient individually rather than the more generalised graphs that the application currently displays.

Conclusion

The time and effort I have spent creating this application over the last 9 months has been fraught with ups and downs. There has been several times during the process that I've felt I can't continue. Ultimately I have prevailed and created an application that I'm very proud of. The Jigsaw medical application begins to create a solution to a real problem that exists within our mental health service.

It was my goal to improve upon the existing system for managing the care and treatment of patients across Ireland suffering from mental illness. I have exceeded my own expectations and pushed myself beyond my comfort limits in many aspects. This experience has taught me a lot and has ben very worthwhile.

Bibliography

- "10 Heuristics For User Interface Design: Article By Jakob Nielsen". *Nngroup.com*. N.p., 2017. Web. 30 Apr. 2017.
- "20 Steps To Better Wireframing - Treehouse Blog". *Treehouse Blog*. N.p., 2017. Web. 5 May 2017.
- "Black Box Testing – Software Testing Fundamentals". *Softwaretestingfundamentals.com*. N.p., 2017. Web. 28 Apr. 2017.
- "Elements Of Effective Web Design: Trunk Test". *Mathguide.com*. N.p., 2017. Web. 20 Apr. 2017.
- "Pidoco - The Rapid Prototyping Tool". *Pidoco.com*. N.p., 2017. Web. 2 May 2017.
- "Think Aloud Protocol". *En.wikipedia.org*. N.p., 2017. Web. 1 May 2017.
- "Usability First - Methods - Usability Testing | Usability First". *Usabilityfirst.com*. N.p., 2017. Web. 18 Apr. 2017.
- "Web Application Security: Methods And Best Practices". *www.alienvault.com*. N.p., 2017. Web. 17 Apr. 2017.

Appendix

1.18 Project Proposal

Project Proposal

Antipsychotics & Mood Stabilizers Monitoring Tool

Kevin Clarke, 13101722, 13101722@student.ncirl.ie

BSHCE 4

Software Development

18/10/2016

Objectives

The main objective of my project is to create a desktop application that can be utilized by mental health professionals in a clinical setting to help monitor the dosages being administered to patients and identify instances where a patient may be receiving a dosage or a combination of drugs that exceeds the guidelines set out. Antipsychotic and mood stabilizer medication also require close monitoring of various parameters such as weight, involuntary movements, blood tests, ECGs etc. The monitoring depends on the type of medication the patients are on and should be individualized. The application will have the ability to create a patient record and record key information such as vital statistics, blood results, drug products being administered and the relevant dosages and frequencies. The smaller objectives that will ultimately lead to the accomplishment of my main objective are as follows:

1.1

Create the software in such a manner that it will be considered a desktop application accessible through an icon placed on the desktop.

1.2

The successful integration of a database that will hold various pieces of information relating to each patient such as vital statistics, dosages and historical patient data. This database will allow for the centralized collection of data so it can be accessed from any hospital or outpatient clinic.

1.3

The completion of an efficient and user-friendly interface that will allow mental health professional users uninhibited access to the information they require. This interface will allow for the creation and modification of records and tests results by authorized users.

1.4

The implementation of security measures such as password, and admin accounts to protect the confidential information held within the application.

1.5

I aim to graph various elements of the data and have these graphs/charts update dynamically when new information is added to a patient record.

1.6

The system will have the ability to alert the appropriate staff member when logging into the system if a patient will soon be required to undertake follow up testing such as bloods, ECG.

Background

I wanted to develop a piece of software for my 4th year project that would not only succeed in meeting the criteria set forth by the college but one that would also have a practical and meaningful application.

I came up with the idea for this project after conversations with my wife and several of her colleagues who are mental health professionals working for the HSE.

After speaking with them it became clear that there is no centralized monitoring tool within the HSE for monitoring the drugs and dosages being administered to patients with mental illness in Ireland. At first I, didn't fully understand the ramifications of this, but after further research I began to understand this lack of a centralized repository, and also the inability to check dosages and drug reactions against the latest guidelines opened both patients and staff to unnecessary risk; especially considering that this category of medications, although effective, can cause long term side effects that require regular monitoring in order to prevent serious complications.

For instance, if a patient who is being treated for a mental illness in a clinic in Dublin appears in the Emergency Department of any of Irelands major hospitals the doctors and nurses currently do not have a resource that can tell them what antipsychotics and mood stabilizers this person may be on. This information is kept in paper records in the local clinic where this patient would usually attend. Now this is not such a problem during the day as the clinic can be called for information, but outside of normal working hours there is no access to this information.

In this case, it should be noted that doctors cannot take the word of the patient on what they have been prescribed or dosages etc., because it's possible the patient has been off their medication or is incapable of providing accurate information.

In relation to the long-term monitoring of possible side effects, patients are required to have certain investigations when the treatment is initiated and regularly thereafter. The frequency and the type of investigations differ depending on the medication they are being prescribed and currently the nurses and doctors need to check the charts manually or to keep a paper record in order to identify what tests are due and when. This might result in delay or certain investigations might be missed.

Once I fully understood the problem I thought it was both a challenging project to undertake but also one that has great scope for implementation in the real world as it solves an existing problem.

Technical Approach

3.1

The first step I will undertake is further research on the frameworks and technologies that I wish to use. This task will be undertaken with the intention of spotting potential bottlenecks and incompatible technologies early in the process so as to avoid delays in the coding phase.

3.2

The next step that will be required of me since this project is solving a real-world problem is to gather user requirements from mental health professionals so as to fully understand what are the key pieces of information that need to be collected and also to ensure that I'm interpreting the *National Institute for Health and Care Excellence and Maudsley Prescribing guidelines* correctly. These guidelines are what will dictate if a dosage is too high or if the patient is currently on any medication that would mix poorly with other medications, as well as the required investigations depending on the prescribed drug.

3.3

I will then begin to compile my requirements specifications so as to have a detailed picture of each piece of functionality I wish to incorporate in order to satisfy the users requirements.

3.4

As a real-world solution, this piece of software would need to be installed on a number of desktop devices and have a database that could theoretically be accessed by several users at once.

3.5

A detailed test plan will be developed in order to identify bugs before the final implementation of the software, as part of this test plan I will be making use of user

testing and putting it into the hands of mental health professionals to test the functionality and usability of the software.

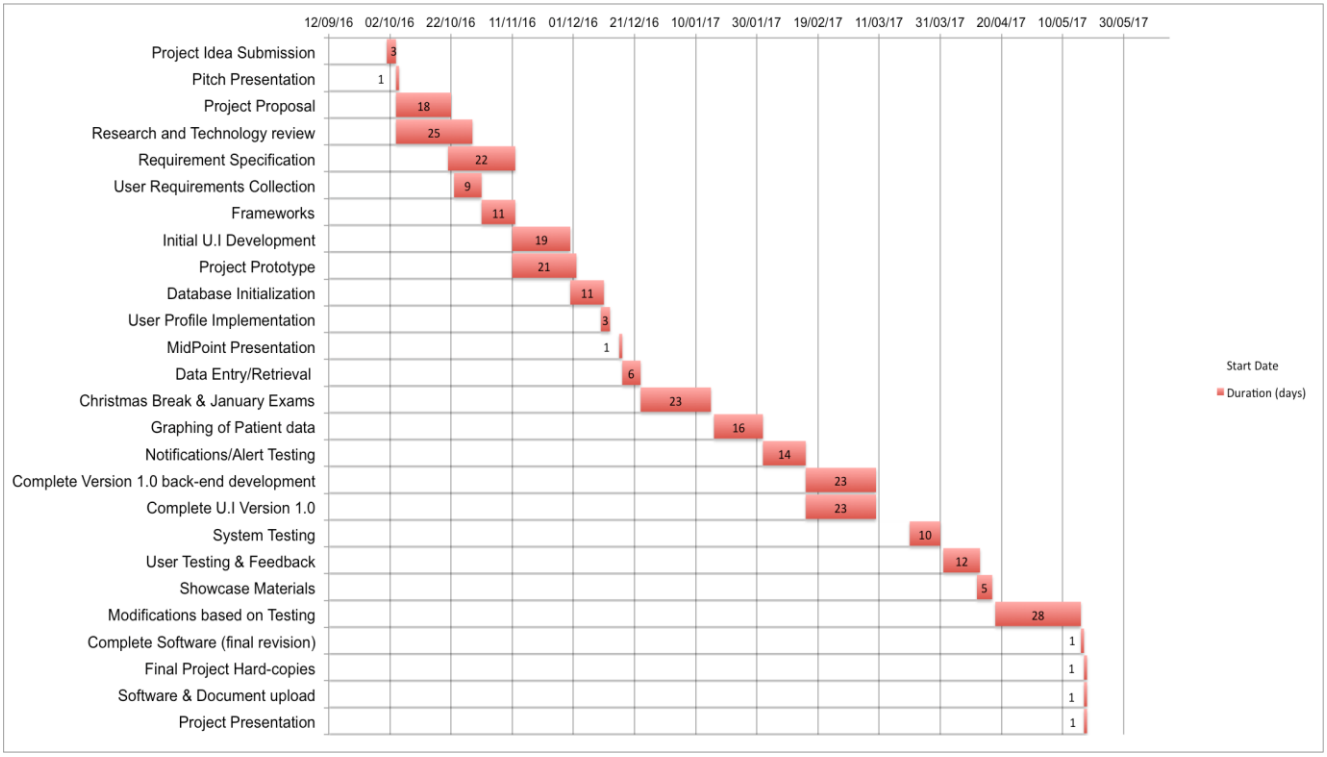
3.6

Once testing has been completed and the errors within the system are corrected the software will be ready for final implementation.

Special resources required

The only special resources I can foresee this project needing are a copy of the relevant guidelines (National Institute for Health and Care Excellence and the Maudsley Prescribing guidelines) that mental health professionals in Ireland are supposed to adhere to when administering antipsychotic and mood stabilizing drugs.

Project Plan



Completed Project Deliverables

5.1

Project idea submission: This was the deadline to finalize my concept and submit for ratification.

5.2

Pitch presentation: This project deliverable involved pitching my project idea to a panel of lecturers.

5.3

Project proposal: This is the document that is currently being read, and was intended as a means to put structure around my idea so as to create a solid foundation that can be built upon

Upcoming Project Deliverables

5.4

Research & technology review: This deliverable will include researching the technologies; frameworks and guidelines which I will make use of to fully understand their place in my project development cycle.

5.5

Requirement specification: This document will set out my requirements from a technical perspective and detail the functionality that will be present.

5.6

User Requirements Collection: This process will run alongside that of the requirements specification document because the requirements and features gathered from the potential end user will dictate some of the requirements in the document.

5.7

Frameworks: This will involve further analysis and learning around the frameworks I will be implementing. In particular, the electron framework.

5.8

Initial U.I development: This will be a rough U.I with limited functionality implemented on the back end.

5.9

Project prototype: This deliverable will be a combination of the initial U.I in conjunction with limited functionality, and wireframes of the possible final state of the software.

5.10

Database initialization: This step involves setting up the structure of the database/s that will be used to hold patient records/guideline information etc. The correct structure of the databases and their connection to the software will be critical.

5.11

User profile implementation: This step is setting up the functionality to allow a valid user to login and manage patient records.

5.12

Midpoint presentation: This will occur at the end of semester one and will act as a checkpoint to ensure the project is on schedule and progressing in such a manner that it will meet the criteria set out for success.

5.13

Data entry retrieval: This deliverable will focus on the ability of a user to retrieve patients records from the database.

5.14

Graphing of patient data: Focuses on a key feature of the final project which is the graphical representation of patient data in graph and chart form to more easily identify trends and upper/lower indicator limits.

5.15

Notifications & alert testing: The introduction of alerts to staff when a patient requires follow up testing to be carried out.

5.16

Complete version 1.0 back-end development & U.I: This is one of the final milestones of the project and concludes the first “fully functioning” version of the code.

5.17

System Testing: This testing phase will be carried out by myself and will involve stepping through the functionality of the software as a user would and testing individual requirements to identify errors.

5.18

User testing: This step of the testing process will entail putting the software into the hands of potential end users and allowing them to freely interact with the software to identify errors.

5.19

Modification based on testing feedback: All errors and feedback will be taken onboard and changes will be implemented to the software accordingly before final release.

5.20

Completed project submission/hardcopies: This step will be a great relief and will signal the hopefully successful completion of this project and its submission for grading.

Technical Details

In the development of this software I will utilize a number of technologies to the best of my ability. Ultimately it is my intention to incorporate as many familiar technologies as I can to accomplish my goals, but I know that my boundaries will be expanded in the use of technologies such as Electron, JSON, PHP and other as of yet unknown technologies.

6.1

The Electron framework is a framework that allows a developer to create a desktop application using web-based technologies. I have chosen to use this framework because a large portion of our degree has focused on developing for the web and I feel it's both an interesting technology choice for creating a desktop and app and also a tool that will play to my strengths. This framework utilizes JSON to create a package around the application and has built-in functionality such as the ability to package the application for either the windows OS or the Mac OS.

6.2

SQLite is lightweight and relatively easy to use tool for the creation and management of databases. I' am however concerned if SQLite will be the appropriate choice for integrating with the electron framework and also if concurrent user access will be an issue.

6.3

JavaScript will also be used in the development this application as well as HTML, CSS, and PHP all of which I intend to utilize in various contexts, and potentially frameworks such as bootstrap will be incorporated where appropriate throughout the project.

Evaluation

My plan for the evaluation of my software application is multifaceted.

Firstly, as part of my project plan, I have factored in the gathering of user requirements from several mental health professionals. One method of evaluation will be to allow these users to test the software and its features to determine if it satisfies their needs and what changes if any would be required to improve the functionality and usability.

Secondly, I will evaluate each requirement/feature of the software as its developed and implemented in order to ensure it carries out the required task and does not interfere with the stability of the current build.

The next aspect of my evaluation will involve a rigorous cycle of testing against my requirement specifications, and carrying out modifications to the system based on my test plan. This process will be repeated several times in order to minimize possible errors at the time of final release. In addition, this testing cycle will involve placing dummy data into the system and creating instances where key features such as alerts should take effect.

Upon completion, the final software application will be evaluated against all of the criteria set forth in my requirements document, and any and all standards required by the NCI.

1.19 Monthly Journals

September Entry

This month was interesting because we have started back to structured lessons and have been given details on our Final Year project. This project has been something I have dreaded since the course began 3 years ago, but I know that with a consistent effort throughout the year I will achieve the personal targets I have set myself in relation to this Project.

We were advised to start coming up with the concept for our project, as we would need to repent them in front of a panel on **04/04/2016** in order to receive the go ahead for our concepts and receive feedback on elements, which may need tweaking.

I initially came up with several ideas which ranged from a university timetabling application, and a billing analytics platform for corporate mobile phone users, but every time I delved deeper, I discovered glaring weaknesses in the projects that made them feel like too much of a struggle for my knowledge base and the 9-month time frame we are working to.

In the 11th hour I did what I should have done form the start and asked my wife for help ☺; as usual she was right and helped me come up with the idea of an anti-psychotic and mood stabilizer monitoring tool for mental health professionals. I knew this had the potential to be complex enough to satisfy the requirements but also practical in a real-world sense that would help me stay motivated throughout the year.

I pitched the idea in front of Ralf, Lisa and Cristina. The whole process was somewhat intimidating but in hindsight went very well as the panel did not have any suggestion of improvements with my idea nor did they find any issues with the idea.

My next challenge is to begin shaping this idea into a more rigid structure that can be expressed in my project proposal at the end of October.

October Entry

Achievements

This month I managed to complete my project proposal document. This required me to flesh out my project idea in greater detail. In particular, the creation of my Gantt chart helped me to get into a more realistic timeframe about the workload I will be required to manage over the coming weeks.

In addition to my project proposal I also began working on my requirements specification document, which is due for submission on Nov 11th. To date I have completed my Use case diagram as well as my Class diagram. This exercise challenged me to understand the system I'm attempting to build, and has raised several queries that require further investigation.

Reflection

Overall I felt this month was very productive however the workload from other subjects feels oppressive when paired with a fulltime career and it seems unrealistic to sustain this pace for 2 semesters.

I have also begun to consider the best way in which to create and manage the database for this software. I have found an interesting application called Pouch DB, which may suit my needs, but I'm still unclear if it will store the data in such a way that it will be easily accessible, and I'm also concerned with the various solutions for graphing data from the database as part of my UI.

Intended Changes

Next month I will attempt to be more productive and get answers to the questions raised by my requirements specification, as I need a solid base to begin my prototype creation.

November Entry

Achievements

This month has been a very hectic but productive month. I completed my requirement specification document and submitted it successfully. This has also now been implemented into my technical report, which I am currently compiling in anticipation of the mid-point presentation submission deadline that is fast approaching. In addition, these tasks I have also begun working on my prototype, it is still early in the process for me, and I foresee a very busy next 10 days.

Reflection

This month has not gone particularly smoothly. In fact, it has been the craziest and most chaotic semester of the degree to date. However, I am beginning to see some light at the end of a tunnel. Once I started creating my proto-type I realized that the Electron framework I had intend to use in order to develop a desktop application as my final implementation was too restrictive in terms of other languages that could be used. Mainly the fact I can't use PHP and would have had to learn unfamiliar languages. I have had to Segway into deploying my application as a website, but I feel ultimately this is a good decision, as my familiarity will allow for a more secure platform.

Intended Changes

My intended changes for next month are to continue with the development of my application and use any free time after the end of semester 1 to make progress in case semester 2 is as crazy as semester 1.

December Entry

Action Items:

I had a brief meeting with Joe this month before I submitted my final requirements specification document, in order to clarify a few points and gather feedback. Overall Joe felt my document was very ready for submission.

Reflection

December has been an excellent month for my project. The midpoint presentation went very well and I received some very useful feedback from the presentation of my project prototype. I also learned that I have achieved a grade of 77% for my project work in semester one which is brilliant and has really encouraged me to continue putting in as much effort as possible.

My Reflection

I was very nervous while presenting my prototype but in the end it received positive feedback. Some of the take away points that I received were as follows:

- Learn about the Moscow prioritization matrix and apply it to my project in order to prioritize the user requirements.
- Research first normal form and attempt to ensure my database architecture is designed so that redundant information is kept to a minimum (avoid unnecessary duplication)

Intended Changes

My intended changes for next month are to continue with the development of my application an use any free time after the end of semester 1 to make progress in case semester 2 is a s crazy as semester 1. I will have a small window of time between my last exam and the beginning of semester 2 which I hope to use efficiently. I would like to implement the audit rules successfully and maybe create a plan of attack user notifications.

January Entry

This month I was unable to progress my project or meet with Joe due to my wife being hospitalized and requiring my support.

Fortunately she is feeling much better and I will be able to resume work on my application and meet with my supervisor going forward.

February Entry

Achievements

This month has been incredibly challenging and there have been several weekends were little to no progress was made on my project because I cam up against several stumbling blocks; in particular I have had trouble effectively solving the problem of displaying such a variety of data in a pleasant and efficient manner for the end users.

Reflection

Going forward I will attempt to prioritize my remaining time in order to complete the all of the main functional elements within the next 4 weeks. This month has ultimately been slow progress but I feel it has ben rewarding the sense that I've learned a lot and ultimately solved my issues.

Intended Changes

My intended changes for the coming month are to complete the majority of the coding, and expand my final documentation in order to begin preparing it for the final submission. My aim is to leave myself with very little to do in the last 4 weeks before the final submission

March Entry

Achievements

This month has been really productive and I have completed the large majority of my project. Key functionality has been implemented and nothing incredibly challenging remains on my schedule.

Reflection

I managed my time well and managed to achieve several key milestones. Everything really came together nicely in the last 2 weeks of the month.

Intended Changes

Going forward for the next few weeks until the submission date I need to tidy up my code, add form validation, and implement my final dashboard view. In addition I’m intending to host the site on 000webhosting.com, which is free but requires some implementation. Lastly I need to complete my document and carry out a variety of testing.