One Semester Individual Project - 40
CM3203

Online system for a doctor's surgery

Author: Jordan Wood
Student Number: c1327201
Supervisor: Irena Spasić
Moderator: Helen R Phillips
Course: BSc Business Information Systems

Abstract

This report discusses how I underwent the design, development and evaluation of a solution for an identified problem with healthcare resource management and accessibility of doctor surgeries.

The solution will allow users interacting with doctor surgeries to access an online web application. This system will allow users to access resources and carry out required tasks such as appointment scheduling, personal details modification, prescription management and much more.

Acknowledgements

I would like to my supervisor Dr. Irena Spasić for her support throughout the project, providing me with constructive feedback and useful ideas that have helped me comprise a high quality system and report.

I would like to also thank Helen R Phillips, the useful feedback provided on my initial plan helped shape my preliminary research and system requirements, leading to a unique solution to the problem at hand.

Table of Contents

1.0 Introduction

Aim

Currently doctor surgeries in Wales manage resources poorly causing inconvenience for both staff and patient. This is due to the use of outdated methods when managing day to day operations as currently:

- Patient appointments are arranged by contacting the surgery by telephone, early that morning of the day the appointment is required.
- Patients often then have to wait long periods of time in waiting room before being seen by a doctor.
- Prescriptions force patients to visit their surgery or chemist in order to request a repeat of necessary medicine.
- Receptionists work is affected due to busy phone lines, meaning more staff are required.

With the ever increasing accessibility of the internet my project aims to design and implement a suitable solution that utilises internet technologies to improve resource management in healthcare. Providing a platform suitable for those interacting with a doctor's surgery.

The scope of the project is not intended to replace all other forms of interaction with doctor surgeries but provide an alternative system that can be ran side by side with traditional methods to help provide increased accessibility and flexibility to its users. The intended purpose of this system will be to enable doctor and patient to share relevant information. Helping them to manage their details, appointments and prescriptions, increasing convenience for all and rendering early telephone calls, busy phone lines and trips to request a repeat of prescriptions a thing of the past for users.

If successful in meeting this aim, my project will consequently help to improve the current problem with doctor surgeries in Wales, whilst demonstrating the technical ability I've gained from my experiences on the university course.

Target audience & Beneficiaries

Due to the nature of solution aims, its intended audience will be those interacting with a doctor's surgery and involves just three user's types – patient, doctor and administration. Whilst all users will be able to access and modify their personal and account details each user type will involve different system requirements specific to them. With a system of this nature it is imperative to ensure user requirements are established as any mistakes could lead to drastic errors that could affect an individual's well being.

Doctors will require the system for the purpose of appointment, prescription and patient medical record features. The introduction and management of these features will help doctors carry out the daily operations they require with greater ease and accessibility.

Patients users require the system to view and arrange appointments, access their medical history and prescription features. All of these features will be obtainable from devices with internet connection providing patients with the benefit of better resource management and greater ease of access.

Administration users will require to not only manage users of the system but to reap the benefits of the newly introduced management of resources. By utilising the data stored the insertion of data analysis tools will allow administration to monitor and evaluate daily operations of their doctor surgery.

Along with system users benefiting, I consider myself as a beneficiary of the project also, as the process will help me to develop my existing knowledge of programming greatly. Gaining a greater understanding of programming technologies by furthering my skills in various topics throughout the project process, such as PHP, MySQL and JavaScript.

Approach

Originally when identifying an approach for the project, I considered the Waterfall software development model. This approach means following a sequential design process, in which progress is seen as flowing downwards like a waterfall through each phase including Requirement Analysis, Design, Implementation, Testing, Deployment and Maintenance. Although this method can be advantageous due to its simplicity and sequence, the models method means once a phase is completed it cannot be revisited, as like a waterfall the model can't go to a backwards. For example, testing would only be carried out at the end of the project, this isn't suitable as test results could show that the project requirements were not met and further implementation was required.

To overcome this models problems and retain its advantages I chose to adopt an Agile-Waterfall hybrid method (1), enabling adaption to alterations in system requirements. As with all hybrid models, both sides had to compromise some properties. Waterfall model had to give up certainty of fixed expectations for the flexibility and freedoms of the Agile model. Agile however had to compromise some of its freedom, to work against a fixed deadline, with the limited time for the project this meant that it was ideal.

With the compromise of these two methods the project scope was able to evolve overtime to fulfil newly identified requirements. For example, testing could now be carried out at each stage of implementation ensuring that any errors could be identified and removed within the development process. This would also mean it was possible to include users of the system at testing stages, helping to gain feedback that could aid implementation and optimise the solution provided to the user.

Due to time constraints I also had to adopt some form of time management for the project. As I was going to be the only person performing the work of the project I could simply produce a Gantt chart to arrange project activities in a simple linear sequence. This helped me to indicate the project sequence by creating set time periods, in which deliverables and milestones had to be met, helping to plan tasks

and forecast to ensure I didn't underestimate the time required to complete the project and fall behind.

Assumptions

To guarantee that this project is successful in achieving the aim and objectives defined above, it must comply with several assumptions.

- The system is a prototype implementation of the desired solution, focusing heavily on implementing required features and meeting user requirements rather than visual appearance due to the time constraints of project.
- The system implemented will be focused on facilitating one doctor's surgery, involving three users - doctors, administration staff and multiple patients. Creating a system that aims to facilitate more than one doctor's surgery is currently out of my project scope, again due to time limitations.
- All system users are added to the system by administration members, with login details provided to individuals by doctor surgery. Therefore, users cannot register an account by accessing the system. If there are any issues with forgotten username or passwords etc they will be advised to visit their local surgery.
- If in future the system were to be introduced into doctor surgeries, the method would run side by side with those traditional methods previously mentioned, such as telephone appointment arrangement, interaction with reception and prescription orders. This is to ensure that those interacting with the surgery without access to devices with internet connection are not rejected.
- Data currently stored in the system is test to demonstrate how the system would work in future if it were to be implemented into surgeries. When so surgeries will provide their own "real" data which could lead to different analysis outcomes.
- If the system was introduced into doctor surgeries, they would have the required technologies in place to host the web application e.g. server.
- Patients are assigned one doctor and can only have appointments with that doctor.
- Appointments selected by patient users for booking are automatically accepted and available if the time slot is free.
- Reader of the report has basic understanding of HTML, CSS, PHP and MySQL technologies.

Scope

The scope of the project is to implement a working system providing users with an alternative method of interacting with their doctor surgery, that does not replace current methods.

As of time constraints the project will aim to develop a prototype of this system, accomplishing main aims and objectives that facilitate a doctor's surgery. Anything outside of this scope will be seen as desirable and future work to be developed.

Important outcomes

The project must work to or provide a solution to tackle the identified problem by providing an alternative method for interacting with a doctor's surgery but not replace others.

- An interface for users to complete required tasks and fulfil all other requirements
- Documentation of the planning, design, implementation, testing and evaluation of the solution to form this report
- Developments made in my technical and project management knowledge from tasks performed throughout out project

2.0 Background

Although we would all like to keep interaction with our doctors to a minimum it is something that we encounter throughout our life. Technology is continuously evolving and has become part of our daily lives, it is time that we introduce it into our local doctor surgeries to help increase their accessibility and efficiency. Before I began the design and implementation of such a solution, I first had to conduct some research to determine the type of features a system should provide to be successful and how these features were to be provided.

Existing Solutions

When beginning initial background research my first step was to identify existing solutions developed for such purposes as project aim. I discovered that there are currently varied solutions including online services that had been introduced since April 2015.

These systems had limited registration, only allowing users to create an account if their registered surgery was included. These surgeries were limited to those in England meaning I was unable to view full content, however, after further research I was able to locate images of the patient area of the system (2). These images allowed me to identify what features they involved and what information they recorded, they provided patients with three services; appointments, prescriptions and access to medical records.

Additionally, I located a recent study investigating how the staff and patients at the Street Lane Practice in Leeds have benefited from the introduction of a current solution providing these online resources (3).

Below are some of the comments staff and patients had to say:

User	Comments
Ruth Whitehead (Patient)	"It really works for me, it saves time and it's just convenient, I can do it sitting in my arm chair"

Inderseet Suryanvasi (Patient)	“Online services are more flexible, I can access it as and when I want to”
John Gilfoyle (Receptionist)	“What patients usually say to me about having access online is that its fantastic”
Jonny Hobman (Doctor)	“Personally I’m a firm believer in empowering patients to look after themselves and to manage their own health and I think that the online access to records helps them to do that” “Providing online access to GP records has helped Street Lane Practice become more efficient and has benefited their patients”
John Snowden (Practice Manager)	“The benefits to the practice obviously are time, effort and we have to mention it money. Its time we save, making things more efficient, saves the practice money”

From viewing this study, we can clearly see that the introduction of the online system has been successful for both staff and patients in Leeds, bringing several benefits.

Therefore, my research indicates that the problem I am aiming to address within this project has previously been solved. However, existing solutions are currently only operational in England. My project proposes to not only introduce a solution providing these tools to Wales but to introduce authentic functionality and features also.

Existing solutions primarily focus on the patient and management of their resources, although this has been proven beneficial, I believe solutions aren’t being used to their maximum potential. These systems work to collect and share data between doctor and patient. Data that includes valuable information about appointments and prescriptions that is continually being updated, is currently being wasted. When analysed by authorised users such as administrator, it can be utilized to generate fresh informative data. I plan to provide tools to monitor trends and patterns within collected data, such as average appointment waiting time and tracking of prescription data, to provide a clear representation of a doctor surgery for improved management and efficiency.

Stakeholders Analysis

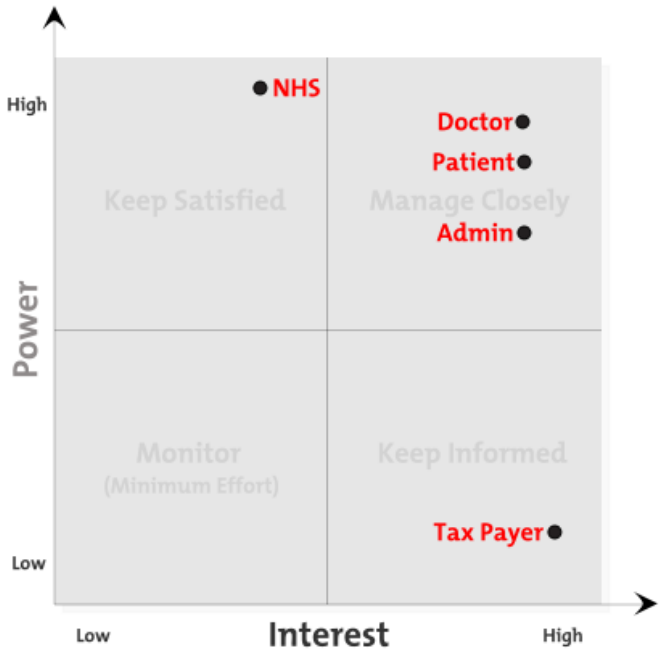
Stakeholder analysis is an important factor of any project to ensure that it succeeds. By identifying the stakeholders of the system, I could communicate with them frequently, to ensure the project aim was fully understood, by using their opinions to shape and improve it.

The first step in the stakeholder analysis was to identify who these stakeholders are (4). When considering the likely stakeholders of such a solution they were limited greatly. As the system to be implemented focuses entirely on one working environment, a local doctor’s surgery, those involved and interacting with this environment would be the projects stakeholders, including:

- Doctors – Use the system for daily operations that are crucial to the surgery. Need to be included to define requirements and improve successfulness
- Patients – Use the system for required tasks. Helpful to take steps to increase their influence to improve product quality
- Admin – Manage system operations. Should also be included to define system
- NHS – The department of health manages health care provided to legal residents of the UK
- Tax Payer – The system if successful and adopted by doctor surgeries will be funded by the tax payer. It is important that in the future stages the tax payer is kept informed

The next step is to prioritise these stakeholders as some may affect work by either providing constraints or helping to advance. Some of which may be interested in what the project aims to complete, whereas others may not.

To do this I created a power/interest grid for stakeholder prioritisation, indicating the stakeholders power from high to low and their interest also from high to low.



The next step was to understand these stakeholders, what they want from the project and how these requirements can influence the system. I will involve each stakeholder at the required stage of the project to ensure they are satisfied and that the project is a success.

Development Tools

The project requires a number of different technologies comprised as one to construct a fully operational system capable of achieving its planned functionality. The main technology I used for page content was HTML with CSS to provide styling.

PHP embedded into this HTML code was then used to create dynamic pages allowing me to connect to, request and manipulate the database created using MySQL an open source relational database management system. To handle the administration of this database the use of phpMyAdmin was made to allow me to perform tasks such as creating, modifying and deleting of tables, rows and fields.

XAMPP an open source web server, allowed me to create a local web server for the development and testing of the solution, however, when introduced it would require a client-server architecture, with the the system being stored and hosted by the doctor surgeries server. This server would act as a host for the system allowing those users interacting with the surgery with authorised accounts to access the system externally as long they have an internet connection from their device.

Web Development

Before delving into the development of the web application, due to the short time period in which I had to implement the system, rather than coding pages from scratch, I wanted to find out if there was an alternative method to speed up process. Although I hadn't used frameworks before, I had come across them and heard of their benefits. To get a greater understanding of the frameworks available to me and what they involved exactly I conducted some research.

I identified two popular methods – Bootstrap and Foundation. Both Bootstrap and Foundation work to provide all of the tools needed to quickly perform CSS-based layout and prototyping work. After trying to indicate the main differences between the two, I found that one isn't necessarily better than the other as they provide similar tools and styles. Due to my lack of knowledge about the two development tools, I chose to base my decision on which option I could find the most tutorials and information online about, in which to aid me whilst developing.

W3Schools is a web developer information website that I have the most experience with. I like their tutorials and the way they explain information by providing references and a "try it yourself" editorial section. From carrying out some research I found that W3Schools included hundreds of tutorials dedicated to Bootstrap, whereas nothing for the Foundation framework. This made my decision much easier as for me there was now only one choice, Bootstrap.

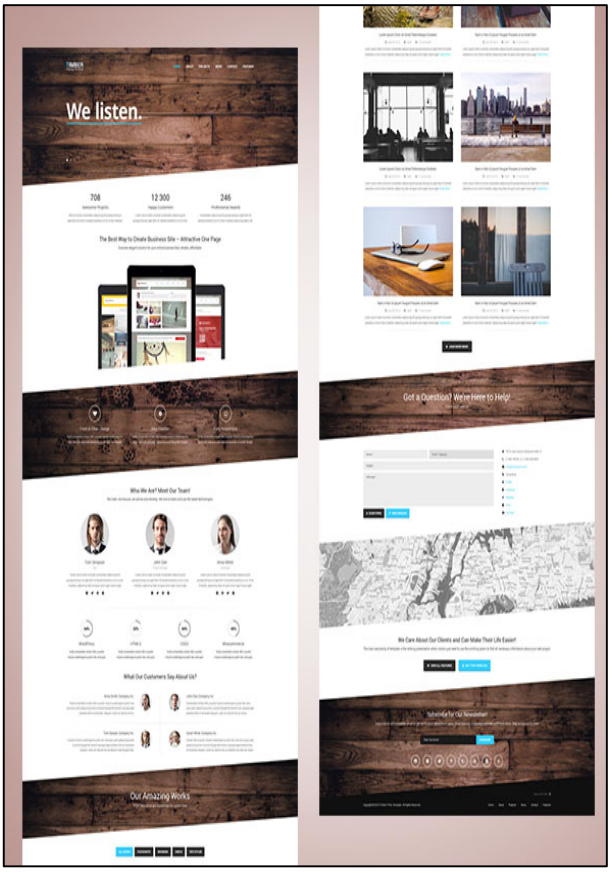
Now that I had chosen Bootstrap, I began to carry out some further research so I could document my choice. Bootstrap was a free front-end framework that could be used for faster and easier web development. Incorporating both HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and much more, as well as optional JavaScript plugins.

This meant that bootstrap was an ideal tool for me when creating the front-end of the system as by utilsing ready made blocks of code and templates the speed of development could be increased subsequently with only customisation needed to adapt pages to meet projects needs.

Bootstraps ease of use means that only basic knowledge of HTML and CSS is required and its responsive CSS features, meant it is compatible with all modern

browsers, adjusting to various smartphones, tablets and computers allowing for increased portability across devices.

Due to the short timeframe of the project, the implementation stage is a lot shorter than I would have liked. With Bootstrap helping to speed up the front-end development process it could help free up time to be focused on other areas of the project that might require it.



Timber – Free one-page bootstrap template (5)

Although bootstrap templates can bring many positives to the project, it is important to take great consideration when selecting the right one for this system. From viewing various templates available, it appears that a lot have very creative and large pages, that while looking appealing can affect the usability of a system. Many of these templates involve lots of scrolling and interactive features making it difficult to identify key aspects of the system, as seen above.

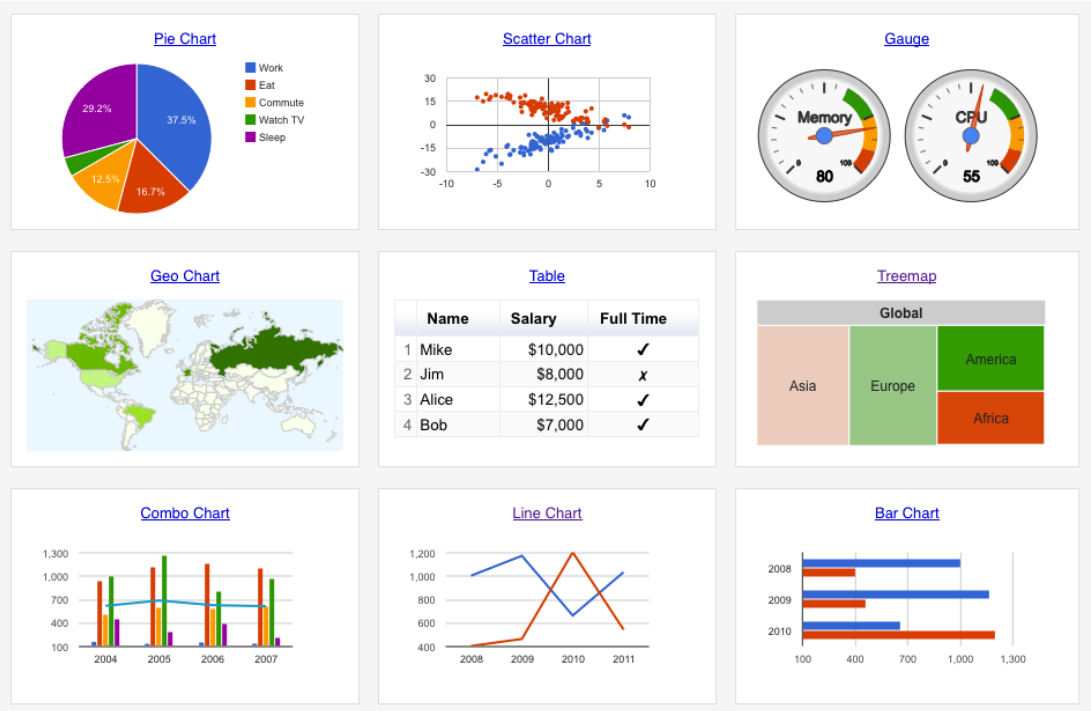
This could appeal to systems that want to focus on being stylish and modern, however, due to purpose of my project and its users, this type of template would have had a negative affect on its usability. The system needs to be accessed by the wide range of users relying on it, including those who may have disabilities and lack of computer skills. Therefore, when choosing a template, I had to ensure that it was suitable for this type of system e.g. providing a very user friendly and easily accessible interface.

Visualising Data

It was earlier identified that in order to provide a solution with unique functionality, administration users should be able to monitor and analyse trends in data stored by surgeries, using the system provided tools. Again time constraints involved in the project meant that rather than coding some form of chart/graph to display this data from scratch, I needed an alternative method to help reduce development time.

I identified three popular solutions – Google charts API, HighCharts JS and JS Charts. I previously hadn't had any experience with any of these options, however, when carrying out a former group project in university, I had used Google API's such as Google Maps. I found that when working with Googles products, due to the size of their user audience, simple tutorials were offered by Google and further help was always easy to find online. As all options appeared to offer similar tools, the fact that I was familiar with Google led me to pick Google charts API.

Google charts API would be an appropriate method in which to display this information for administration use, as it provides a wide range of chart designs to address data visualisation needs. Providing charts that could be easily adapted to display data and embed it into a web page.



Google chart gallery

Legal, Social & Ethical Issues

Due to the type of system and its users another important form of research I had to conduct was an analysis of the social, legal, and ethical issues that the system may face and identify how development will adhere to the principles involved.

Data Protection Act (1998)

The system will contain personal data about users for example name, address and medical records etc. Therefore, it essential that data is stored and used in the way governed by the Data Protection Act (1998). Breaches of data protection are often in the news. For example, NHS Surrey was fined £200,000 for selling a computer that contained patients' personal records without first destroying the data on the hard drive (6).

To ensure personal data is kept safe and secure users have different levels of authorisation access, for example doctor user type may gain access to many patient's data, whereas patient user may only access own personal data.

One of the key principles of the act is to ensure data is accurate and current. The implementation of the system will help a doctor surgery to adhere to this, as all personal data will be made viewable to users within the system. They may then update any information ensuring it is accurate and current.

Freedom of Information Act (2000)

As the system is going to be used by a public authority (NHS), the principles of the freedom of information act entitle people to the right to know about the activities of public authorities meaning that all information held must be available for public access.

For the system to comply with the act, users should be able to access the data it stores, within reason. For example, the system should and will allow patients access to data stored about themselves such as personal, medical records, prescriptions and appointments. However, due to the data protection act, data regarding other individuals should not be made accessible. For any other information the public must request in the form of writing or email to the NHS (7).

Accessibility/Disability Discrimination

It is important that the system ensures that all users can gain access and use of its provided functions to guarantee no disability discrimination. Users have a right, regardless of disability to use the online system.

When a system is badly designed and accessibility issues are not considered this can create barriers excluding people from accessing it. This is a big concern for this given project as many patients may potentially have disabilities due to the type of system. For example, an NHS study shows that in the UK there are almost 2 million people living with a visual impairment (8), these people are entitled to access the system and should not be discriminated against. To certify access for patients, disabilities have to be taken into account when developing the system, this can be achieved by providing:

- Clear use navigation
- Suitable colour combinations
- Multiple ways of interaction (mouse and keyboard or text and graphics)

- Simple and understandable documents
- Alternative descriptions for images, audio and video used

Quality Assurance

Attempting to keep to a time frame and its deadlines can often cause system programmers to rush or even skip quality assurance measures, therefore, leading to unidentified or ignored flaws in a system due to the shortage of time.

Due to the nature of this project a flaw could be the cause of major problems with the system, causing inconvenience and issues for the large population of people reliant on it. For example, if there were a fault with the prescription operations of the system, patients may not receive important medicines that they depend on. To ensure that this isn't the case, careful consideration should be taken with confirming the system is fault free. This will be achieved through investing time into the quality assurance measures of the product such testing to ensure the highest quality possible.

Professionalism

My professionalism when carrying out the project is key to having a successful business. The relationship between myself and the doctor surgery is vital. If my system included various faults and was unable to meet the expectations of the surgery this would have a negative affect on reputation and therefore future business opportunities.

In the future I would be looking to expand the deployment of the system into surgeries across Wales. In order to ensure I can accomplish this, I must strive for the trust of users by showing my professionalism of providing effective systems.

To guarantee this I must act responsibly when implementing and take great consideration of the evaluation the end system and how it meets the requirements of the problem. However, due to personal experience and lack of time in which to develop the system, quality could be affected, but in the future as development continues quality will rise.

Intellectual property

"Intellectual property (or IP) refers to creative work which can be treated as an asset or physical property. Intellectual property rights fall principally into four main areas; copyright, trademarks, design rights and patents." (9)

Developers are obligated by law to protect the integrity of intellectual property so that people can earn recognition or financial benefit from what they invent or create. In order to comply both with laws and ethics when developing my system, I cannot and must not take credit for another person's work, including if that work hasn't been explicitly protected by copyright, trademarks or patents etc.

Forecasting

When carrying out current solution research I identified that data analysis and forecasting tools are a requirement of the system if it was to be unique. These tools when implemented would allow users to analyse historic data and predict future outcomes. It is impossible to guarantee the accuracy of these forecasting techniques, therefore, it is important to inform users of the system that they are not 100% reliable. It would be unethical if I allowed users to make decisions based on the belief that forecast outcomes were always accurate, as this could have a negative effect on operations.

Risk Assessment

Before being any development it was essential to identify potential risks that may affect project and system operations and identify the management techniques to help prevent these from occurring, leading to improved system quality.

Category	Potential Risk	Risk-Management Technique
Timing	• Underestimating the amount of time required to complete project	• I will follow my proposed timeframe throughout the project using work plan and Gantt chart
	• Struggling to arrange supervisor meeting due to busy timetable	• Doodle Poll (web based scheduler) will be used, so both myself and supervisor can agree on a time when we are available • If schedule busy contact and document sharing will be made via email or google docs
Productivity	• Project has a long and busy schedule that might cause motivation to decrease overtime	• Setting objectives that have to be completed within a certain timeframe • By defining my vision for the end result system
Technical knowledge	• I have limited knowledge of the programming languages being used to implement the system, therefore, I may not have the technical ability to code certain functions of system	• I will ensure that I delegate more time to the implementation section of the project to ensure I can overcome any problems that may occur • I will carry out research and practice my coding skills over the course of the project to familiarise and increase my knowledge with programming languages

Technical Requirements	• The system may not be thoroughly tested, leaving errors and bugs. This could affect requirements from being met	• I will conduct a full testing of every function of the system • Test system thoroughly against acceptance criteria
	• System may not meet requirements	• Make sure all essential requirements are met before those desirable and optional
Operational Failure	• The system could have major flaws leading to operational failure, causing inconvenience and issues for the large population of people reliant on it	• I will conduct a full testing of every function of the system • Test system thoroughly against acceptance criteria
Human Resources	• Supervisor absence due to unforeseen circumstances could potentially slow down the project's progress. A domino effect could occur, delaying work that depends on another task being completed prior	• Email will be used to keep contact, allowing for notification of any absences • Google Docs will be used to share documents
	• If I personally were to be affected by unforeseen circumstances such as illness etc this could effect the projects progression	
Requirements	• The progression of the system may identify various non functional, functional requirements or features of the system that haven't yet been discovered. If implemented these could cause distribution to the time frame of the project	• Continuous review of my ideas and vision for the system will be made and exchanged with my supervisor throughout the project • Requirements will be organised to ensure appropriate time is delegated to prevent risk to other objectives
Programming environment	• If the wrong programming languages are chosen for development, it could have a negative effect on the outcome of the system	• I will undergo research to establish the correct programming languages to fit the projects specification

Professionalism	• If the implemented system doesn't meet requirements or has faults this could have a negative affect on my professionalism. Meaning other surgeries would question my ability to provide them with similar products	• Thorough consideration of requirements will have to be taken throughout the implementation and evaluation stages of the project • Provide a professional end product
Data Loss	• Project documents and work could be lost due to unforeseen circumstances. This would have a negative affect on project progress and motivation	• Throughout the project I will regularly back up work, using both Drop Box and External Hard Drive
	• System failures could lead to losses in important data that could affect not only operations of the surgery but the large amount of people that rely on it	• Back up techniques will be implemented to the system to regularly create copies of data • Disaster recovery strategies should be considered to provide solutions to return to a normal state if data loss were to occur
Security	• If system isn't secure unauthorised personnel could access data. This would be a breach of the data protection act and therefore legal action could be taken against the surgery	• Registered accounts will be required to access the system • User types will have various levels of authorisation depicting what data they may access
Usability	• Poor usability of the system could make it difficult for users to navigate and carry out tasks	• Usability will be considered greatly at the design stage for the front-end of the system • Thorough testing will be carried out, in order to maximise usability
Accessibility	• Users with disabilities may have difficulty accessing and using the systems provided functions, therefore, causing discrimination against disabilities	• Disabilities will be taken into account when developing the system

Maintenance	• The system may cause difficulty when specified users require to maintain or modify certain aspects	• Quality documentation of the system will be created • Comments will be used to provide information about code • This documentation and comments will be easily accessible
Scalability Issues	• When specified users try to introduce additional features to the system, old features are effected in a negative way	• Testing of the system to measure its capability to scale up or scale out in terms of different functionality
Portability	• System interface and features aren't consistent or compatible across devices	• Bootstrap will be used to create the front-end of the system as it incorporates responsive CSS features, increasing portability across devices

3.0 Specification & Design

The aim of the project is to create an online system allowing those interacting with a doctor's surgery to undergo required tasks and manage resources, without replacing current methods. In order for this to be accomplished the first step to be taken is to specify what the system should comprise of to be a successful solution.

Requirement Analysis

Requirement analysis is a very important stage of system development. To completely develop my project and address the problems faced successfully it was essential that system requirements were correctly identified, analysed and documented. By identifying the functional and non-functional requirements of the project I could define the features to overcome the problem and gather a greater insight as to how to develop a successful solution.

To collect these, I first had to consider the problem and the needs of the system users. Previous research into existing solutions helped me understand what the key elements of a system of this nature should include to allow its users to perform required tasks. This helped me obtain a set of requirements in which a successful and unique system should be comprised of.

These requirements were specified as either "Essential" or "Desirable" to the system and then identified as functional or non-functional. "Essential" requirements are those that the system must have in order to satisfy the desired goal of the project. "Desirable" requirements are those that the system could have, these are subject to the time constraints of the project. However, if added would be beneficial to the final solution and provided added value.

Functional			
Requirement	**Description**	**Acceptance criteria**	**Priority**
Online System	• The system must be provided to users online via web application	• Users can access the system externally as long they have an internet connection from their device	Essential
User Types	• Three user types with various authorisation should be provided to limit access to areas of the system	• System user types have different interfaces and levels of access depending on their authorisation type	Essential
User Log In	• System allows users to login with authorised user account credentials	• System allows users to input login credentials for verification. If authorised accessed is gained	Essential
User Log Out	• System allows users to log out of their user accounts	• Users can log out of their system user accounts	Essential
Exclusivity	• System must be exclusive to registered users interacting with the doctor surgery	• System can only be accessed by those with registered user account	Essential
Appointments	• Doctor and Patient users must be able to view their appointments	• System allows users to view their appointments	Essential
Appointment Booking	• Patient users must be able to book doctor appointments	• Patients can use the system to schedule an appointment with their doctor	Essential
Appointment Update	• Doctor users must be able to provide details after an appointment with patient	• System allows doctor users to input appointment update details for later use	Essential
Prescribe Medication	• System must provide functionality for doctor users to prescribe patients medication	• Doctors can use the system to prescribe patient medication	Essential

Prescriptions	• Must include functions to allow patients to request re-order of prescribed medicines	• System contains functionality to allow patient users to request prescribed medicines	Essential
Personal Details	• System should allow users to access and update their personal details	• System allows users to access and update their personal details	Essential
Account Details	• System should allow users to access and update their account details	• System allows users to access and update their account details	Essential
Medical records	• Patients should be able to view their medical records using the systems user interface	• Patient users to access their medical record	Essential
View Patient Records	• System should allow doctor users to view patient medical records	• Doctor users can view their patients medical records	Essential
Add Users	• Administration should be able to create new user accounts	• Admin users can add users to the system	Essential
Remove Users	• Administration should be able to delete user accounts	• Admin users can remove users from the system	Essential
Data Analysis	• Should incorporate data analysis tools to allow administration staff to analysis existing data	• Admin users can view data analysis tools within system	Essential
Forecasting	• Forecasting tools should be used to help users analyse and predict future	• Forecasts are provided to provide future predictions to admin users	Essential
Medicine Research	• Doctors could use the system to carry out medication research to ensure they prescribed the most effective choice to patients	• Doctors can carry out medication research	Desirable
Mobile Application	• A mobile application version of the system can be downloaded to users smartphone device	• Alternative smart device application can be access by users	Desirable
Location	• System could users with location of and directions to doctor surgery	• System provides location of doctors surgery and application uses	Desirable

		phone GPS to provide directions	
Appointment Cancelation	• Users could be able to cancel scheduled appointments	• System allows users to cancel scheduled doctors appointments	Desirable
Messaging	• System could allow patient and doctors to interact via email style messaging feature	• System allows doctor and patient users to communicate using messaging feature	Desirable

Non-Functional			
Requirement	**Description**	**Acceptance criteria**	**Priority**
Traditional Methods	• System must not replace traditional methods of interacting with doctor and surgery	• System works alongside traditional methods as an additional form of interaction	Essential
Security	• When introduced to doctor surgeries, data stored within the system must be stored securely due to its sensitivity and not compromised	• System is secure restricting data access to authorised users only	Essential
Usability	• The system must be simple and intuitive	• Users can use the system with minimal help from other users or external sources	Essential
Reliability	• The system should run error free	• The system should run error free for all staff and patients, no matter what device is used	Essential
Maintainability	• System must allow accommodate easy and convenient maintenance and modification	• Specified users can easily maintain and modify system due to quality documentation, comments and accessibility	Essential
Interface	• The user interface should be professional, clear, concise and easy to navigate	• User interface is clear with all required functions and easy to navigate layout	Essential
Portability	• The system should be able to be accessed on any device with a web	• System runs error free on users device	Essential

	browser and internet connection		
Professionalism	• Development and deployment of the system must be carried out professionally	• All of the users requirements are met to a high level	Essential
Scalability	• The system should be highly scalable allowing for additional features to be added without compromising other implemented features	• System allows for additional features to be added without effecting current features in a negative way	Essential

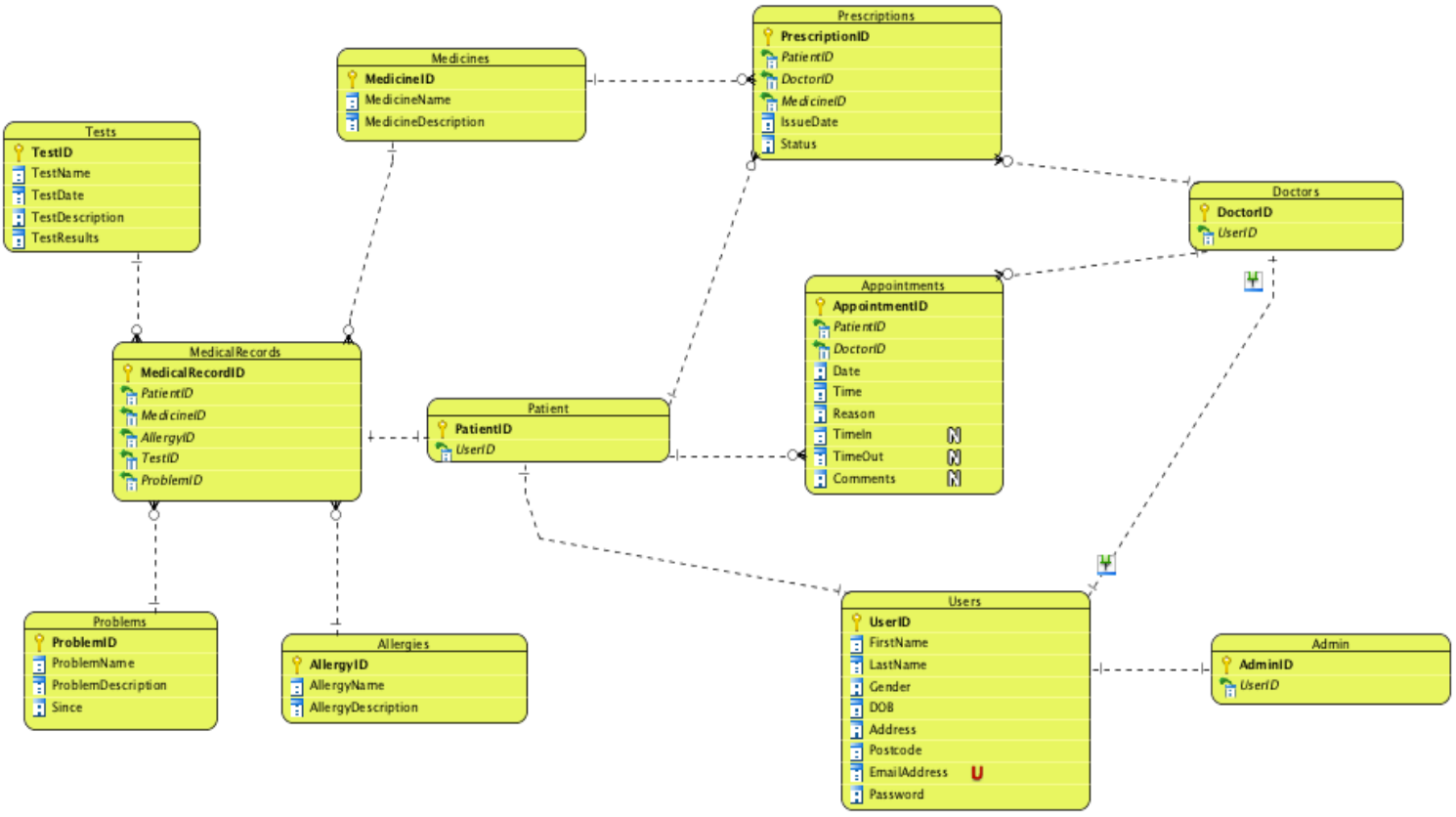
Design

Before implementing anything the first step involved in the development process is system design. The process involves defining the database and interface of the system to satisfy requirement specification.

For this process I decided to adopt the agile design approach, this meant that my overall system design would be developed over time, evolving to fulfil requirements and minimising complexity. This method involved defining some initial architectural modelling at the beginning of the project, without having to fully document design before I had began coding.

Database Design

After analysing previously collected requirements, I began to consider the organisation of data stored within the system and created an entity relationship diagram, providing a graphical representation of entities and how they would relate to each other.



This model consisted of a series of tables involving the data to be stored and used within the system. These tables contained entities, some that contained relationships in which they were reliant on others. For example, AdminID is reliant on UserID attributes.

Users

The Users class contains entities relating to individual users such as; UserID, First Name, Last Name, Gender etc and login credentials consisting of a unique email address and a password.

Patient

The Patient class contains a unique Patient ID and corresponding User ID to identify that, that user's user type is patient.

Doctor

The Doctor class contains a unique Doctor ID and corresponding User ID to identify that, that user's user type is doctor.

Admin

The Admin class contains a unique Admin ID and corresponding User ID to identify that, that user's user type is admin.

Appointments

The Appointment class contains entities relating to individual appointments, involving the Patient ID and Doctor ID of the patient and doctor the appointment was between. This would be information related a unique appointment (AppointmentID), such as scheduled time, reason and once updated after the appointment has taken place – time in, time out for data analysis purposes, as well as any comments.

Prescriptions

The Prescriptions class contains entities relating to individual prescriptions, this involves a unique identifier (PrescriptionID), the Doctor ID of the prescribing doctor and the Patient ID the prescription is for. The entity would also provide a MedicineID to define the medication, an issue date, status and a repeat prescription request.

Medical records

The Medical records class contains entities defining a patient user's unique medical record (MedicalRecordID). This involves the foreign keys relating to this information such as PatientID, MedicineID, AllergyID, TestID and ProblemID.

Tests

The Tests class contains entities relating to unique tests (TestID) that have be conducted on patients. Each one includes a test name, test date, test description, and test result.

Medicines

The Medicines class contains entities relating to unique medicines (MedicineID) doctors may prescribe. This involves their name and description.

Problems

The Problems class contains entities relating to unique patient problems (ProblemID). Each includes a problem name, description and since date.

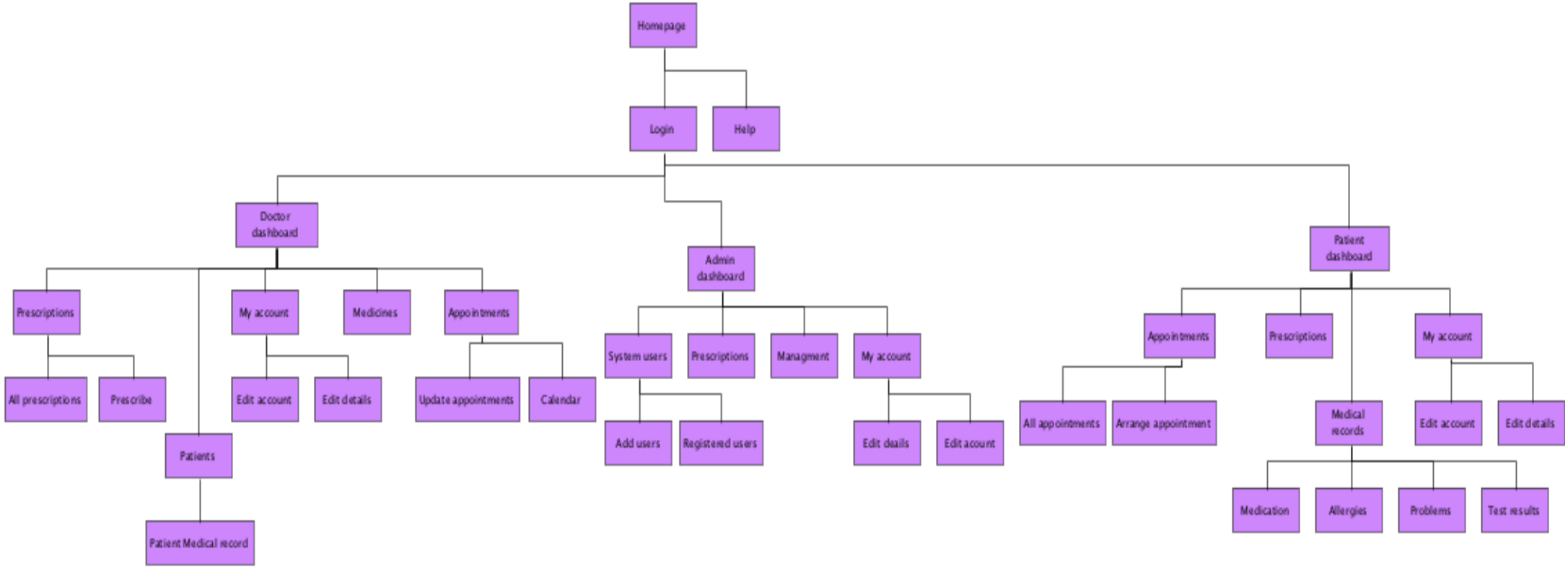
Allergies

The Allergies class contains entities relating to unique allergy types (AllergyID) and provides a description for each.

User Interface Design

Site map

After completing the structure of the database I began to create a site map to define system navigation and how each page would comprise together as one. By viewing this site map, you can see that the first point of contact is the homepage, this page allows users to login and access different areas of the system. Once authorised access has been gained the users, user type would determine what area of the system could be accessed – Doctor, Administration and Patient. Each user type would be redirected to its corresponding dashboard in which the required pages and functions could be navigated to.



Interface design

With the database and site map design of the system completed it was now time to focus on system user interface design. The proposed system was to be developed and accessed by users in web application form, it was essential that when designing the interface users would be interacting with, that the layout was clear, professional and not overwhelming due to its proposed use. Within this section of the report I will explain the measures and methods used by myself to ensure these design requirements could be met successfully and provide some examples (All can be found in appendix).

Wireframes

The first interface design method used was wireframes, these allowed me to visually represent the user interface visual design by providing a page schematic/ blue print as a visual guide for each page within the proposed system. They were created within Visual Paradigm allowing me to arrange page element layouts for example input fields, boxes and tables etc.

Activity diagrams

The next step to be conducted in the design process was activity diagrams. This involved creating graphical representations of how activities and actions flow throughout each of the designed system pages.

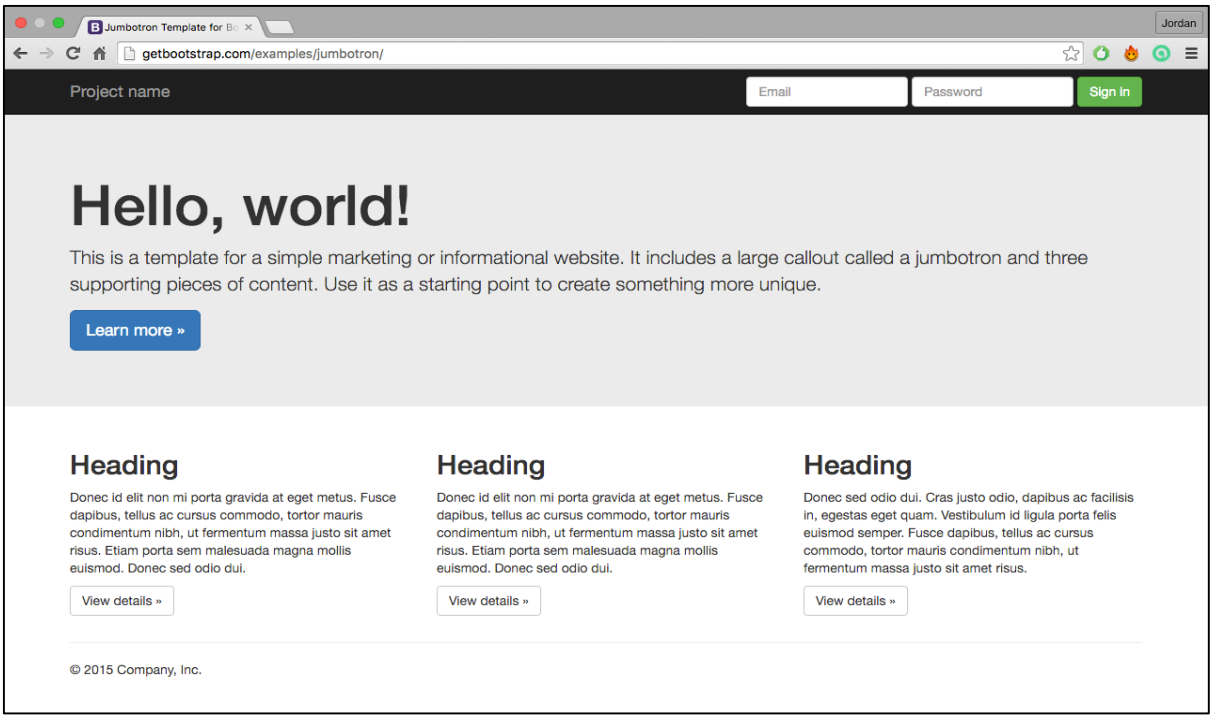
Use cases

Now that graphical representations of system activities had been created the next step was to document them. I decided that the method I would use for this process was Use Cases. A Use Case is the process of documenting a series of actions between the user and a system that allows them to achieve a goal, describing the systems behaviour as it responds to these user requests at each step.

Homepage/Login

When user's first access the website they will be presented with the homepage. This page provides an overview of what the system involves and allows registered users to login via the navigation bar by entering a valid email address and password. Users may also navigate to a login page by selecting "My Account" button and being redirected to login page.

When designing this page, I decided to make use of the Bootstrap templates made available online as previously mentioned in the report. I chose a basic template in which I would develop for required purposes (11).



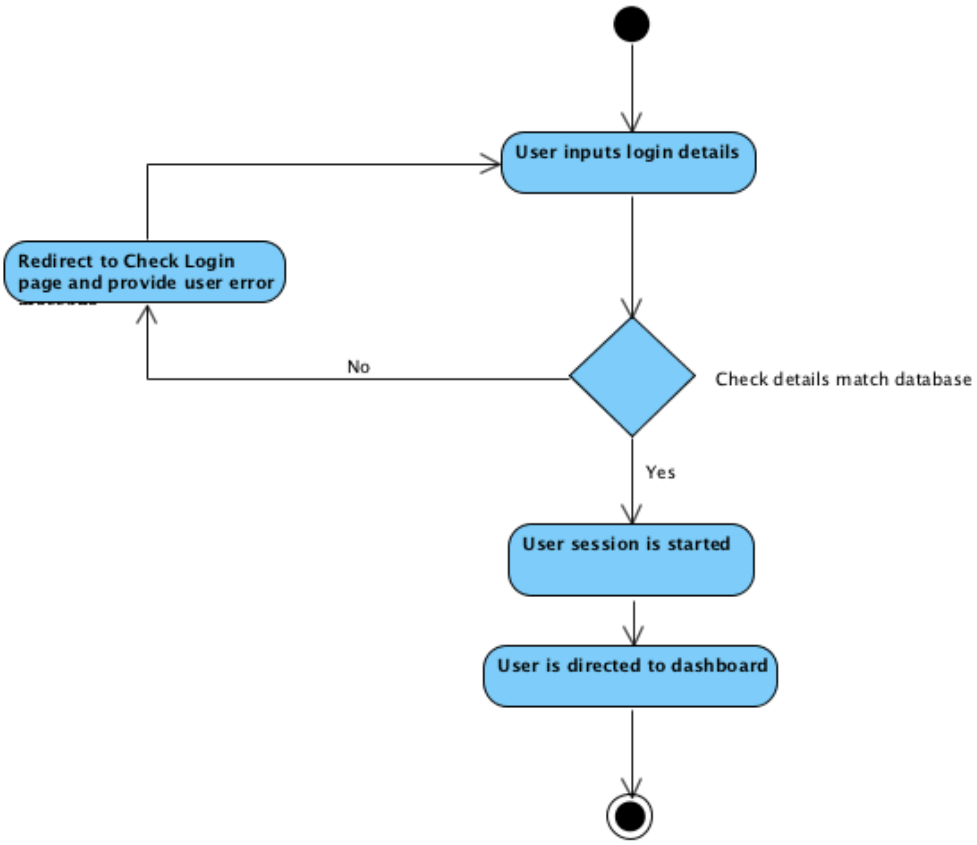
http://getbootstrap.com/examples/jumbotron/

Wireframe

After identifying the template to be used I began to create a wireframe design to design what the layout of the page would consist of. Minor changes were made on the styling of the page, with only adjustments being to the information displayed.

Activity diagram

Once the visual aspect of the page interface design was completed, it was now time to define graphical representations of system activities. The main activity this page contained was logging into the system and can be seen below. This involved the user inputting their login details and checking them against the database for a match, the results would the determine if a user session was started or if denied an error message provided.



If authorised details matched those in the database the user would be directed to their corresponding dashboard, dependant on their user type.

Use case

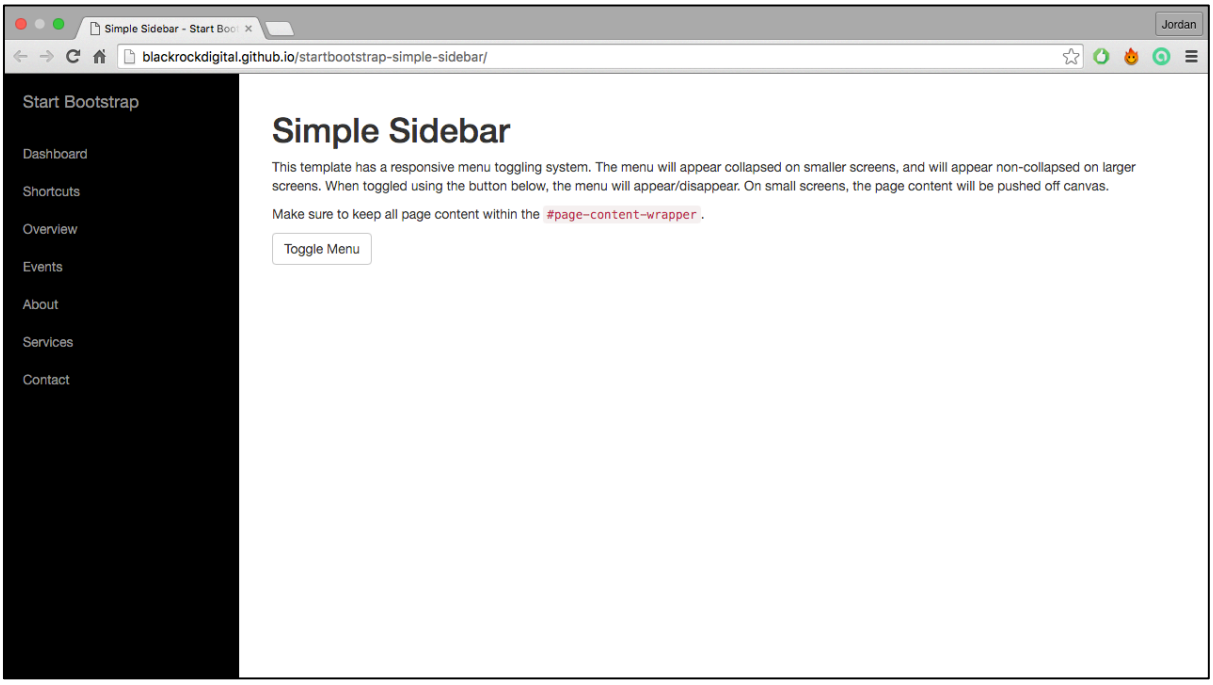
Now that graphical representations of these activities had been created the next step was to document them. As stated above the method I chose for this process was use cases.

Use case name: Login
User type: Patient, Doctor, Admin
Description: The user will navigate the system to input account details and login
Pre-conditions: User has an authorised account

Basic flow:	1. User inputs login details into provided input boxes on system homepage 2. User selects login button to submit details 3. User gains access to system and is redirected to corresponding dashboard
Alternative flow 1:	1. User selects “my account” located on system homepage 2. User is redirected to login page 3. User inputs login details into provided input boxes and selects login to submit entered details 4. User gains access to system and is redirected to corresponding dashboard
Alternative flow 2:	1. User enters invalid account login details 2. User is redirected to login page 3. User is displayed an error message informing of incorrect details 4. User must re-enter account details 5. User must select submit button
Post conditions	The user is logged in and redirected to corresponding dashboard

User navigation

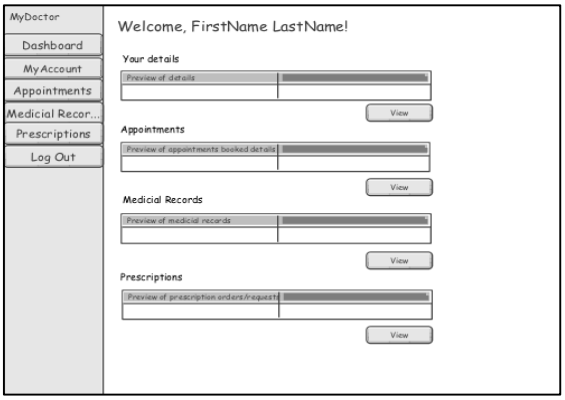
Once authorised access has been gained users require a method of system navigation. When designing how users would carry out this navigation I decided to the implementation of a Bootstrap template would be the most effective and time saving method (12).



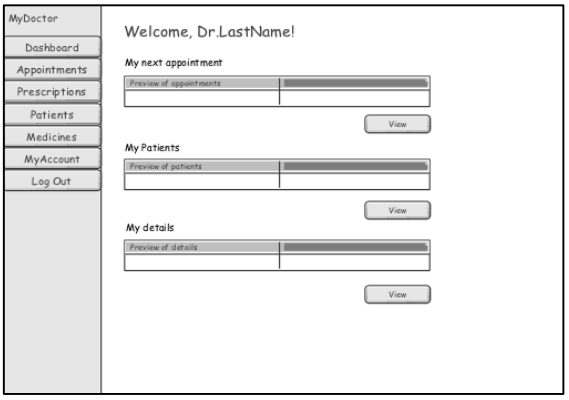
http://blackrockdigital.github.io/startbootstrap-simple-sidebar/

Wireframe

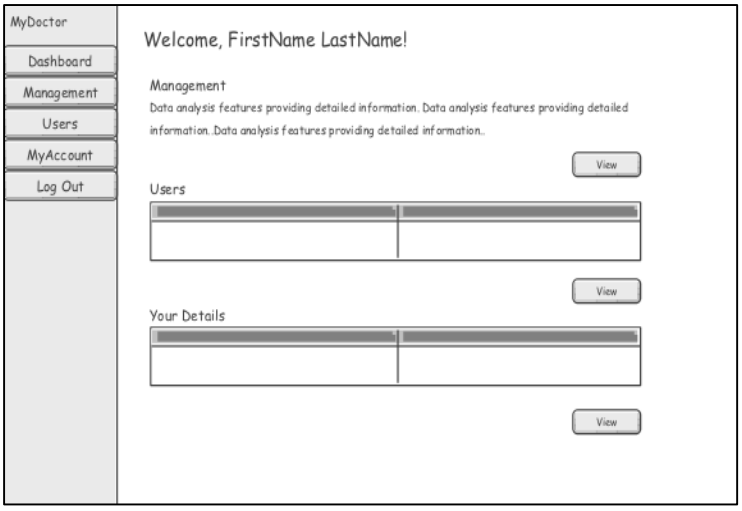
This template could provide a static navigation bar that displayed links to the various sections of the system.



User dashboard



Doctor dashboard



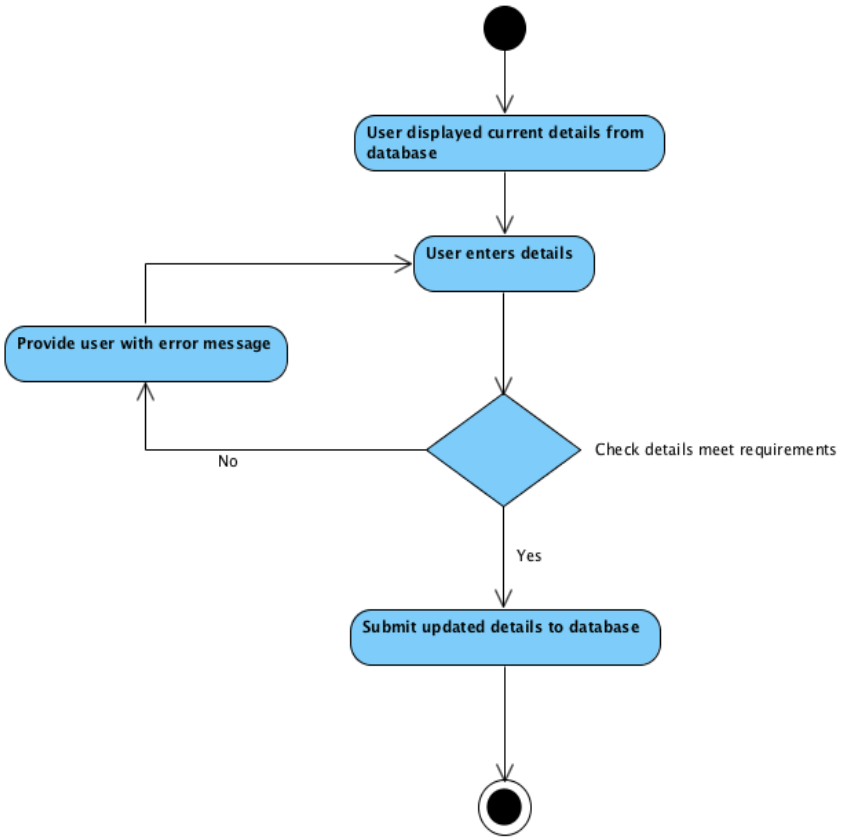
Admin dashboard

Edit personal details

Each user type within the system can view and modify their personal details within the My Account section. This is beneficial to the user as it ensures their contact details are accurate and current, but also to the doctor surgery as it aids compliance with the Data Protection Act previously mentioned.

The page comprises of input fields with the individuals first name, last name, address, postcode and telephone number as placeholders. The user can then input any changes they desire and submit this new information.

Activity diagram



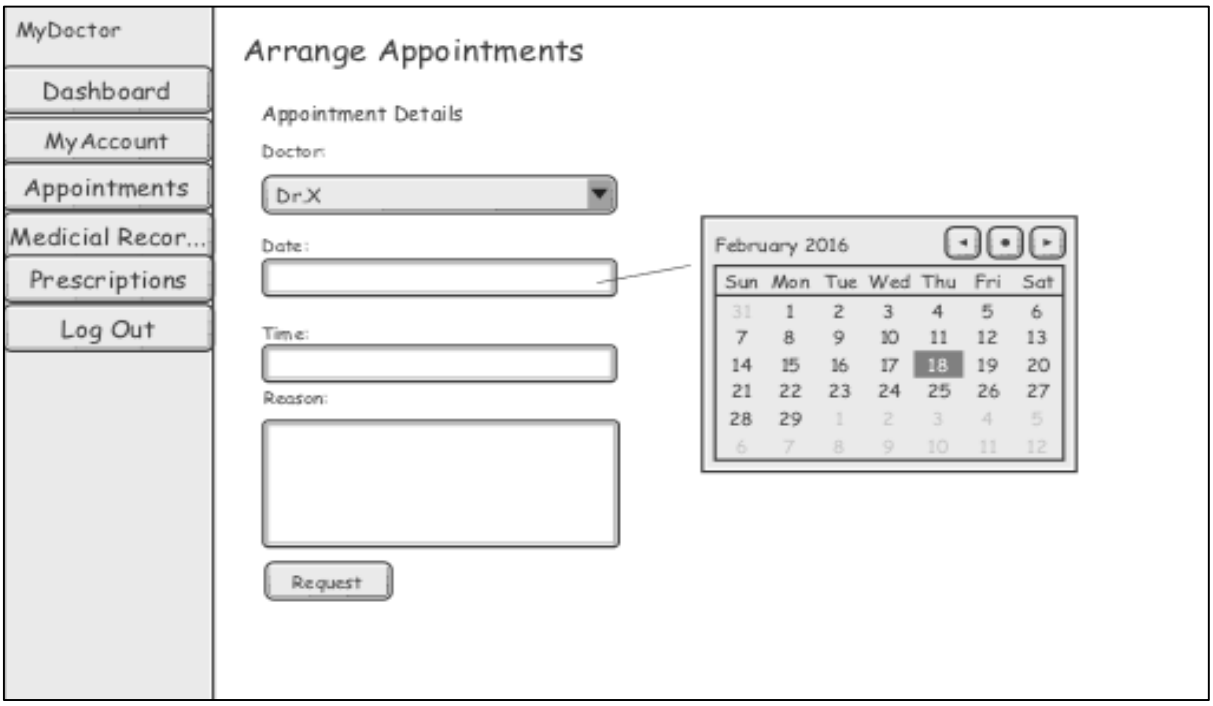
Use case

Use case name: Update personal details	
User type: Patient, Doctor, Admin	
Description: The user will navigate the system to input and update personal details	
Pre-conditions: User has logged in with an authorised account	
Basic flow:	1. User selects "my account" tab within navigation bar 2. User locates personal details and selects edit button 3. User provided input boxes displaying current personal details 4. User must input desired new details into one or more input boxes 5. User must select submit button
Alternative flow:	1. User enters invalid details into input boxes 2. User is provided with an error message to inform them 3. User must re-enter personal details ensuring they match criteria 4. User must select submit button
Post conditions	The user personal details are updated in system database

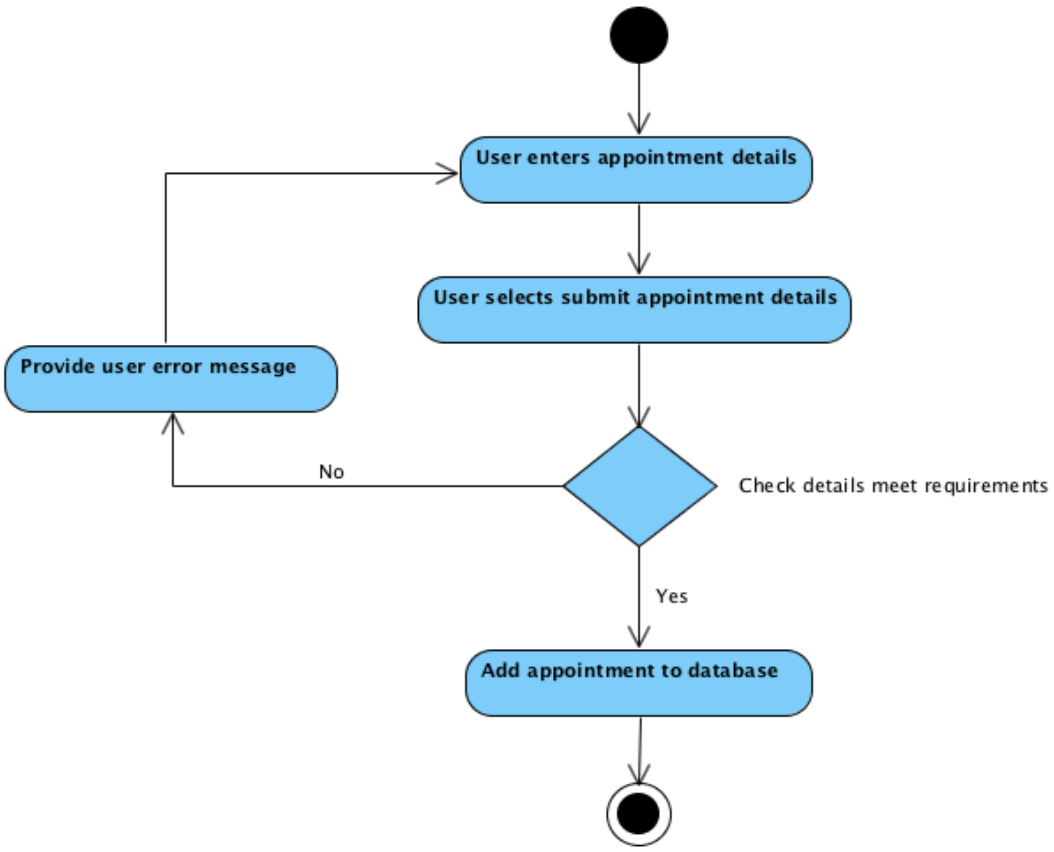
Arrange appointment (Patient)

Patient users can access arrange appointment page within the appointments section. This page does as expected, providing the user with the tools to schedule an appointment with their doctor.

Various input fields are available to input doctor, date, time and reason requirements before submission.



Activity diagram



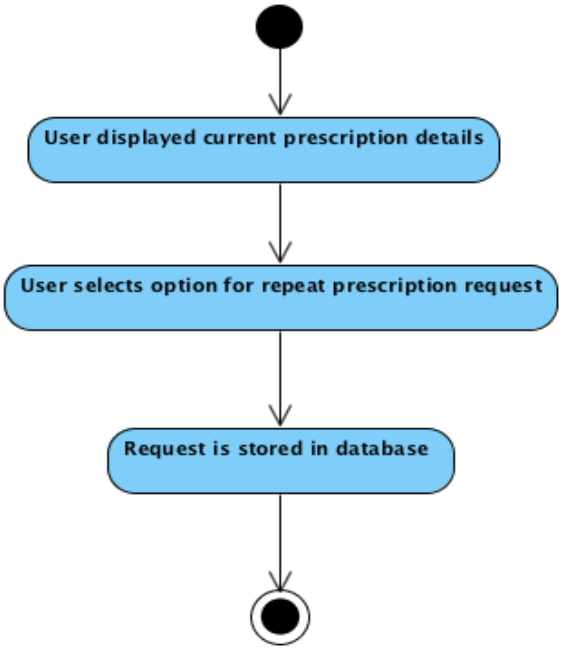
Use case

Use case name: Appointment arrangement	
User type: Patient	
Description: The user will navigate the system to arrange a doctors appointment	
Pre-conditions: User has logged in with an authorised patient account	
Basic flow:	1. User selects “appointments” tab within navigation bar 2. User locates arrange appointment section and selects request button 3. User provided input boxes 4. User must input required appointment information 5. User must select request button
Alternative flow:	1. User enters invalid or unavailable details into input boxes 2. User is provided with an error message to inform them 3. User must re-enter appointment details ensuring they match criteria 4. User must select request button
Post conditions	The user appointment is added to system database

Prescriptions (Patient)

Patient users can view their prescription history, providing medication name, issue date and status within the prescriptions page. If user requires a repeat prescription the request repeat option can be selected in updating the database of the request.

Activity diagram



Use case

Use case name: Patient requests repeat prescription	
User type: Patient	
Description: The user will navigate the system to submit a request for repeat prescription	
Pre-conditions: User has logged in with an authorised patient account and has previous prescriptions	
Basic flow:	1. User selects “prescriptions” tab within navigation bar 2. User provided table of prescription history 3. User selects Click Here within Repeat Request column
Post conditions	The user updates their prescription medicine status to requested within the systems database

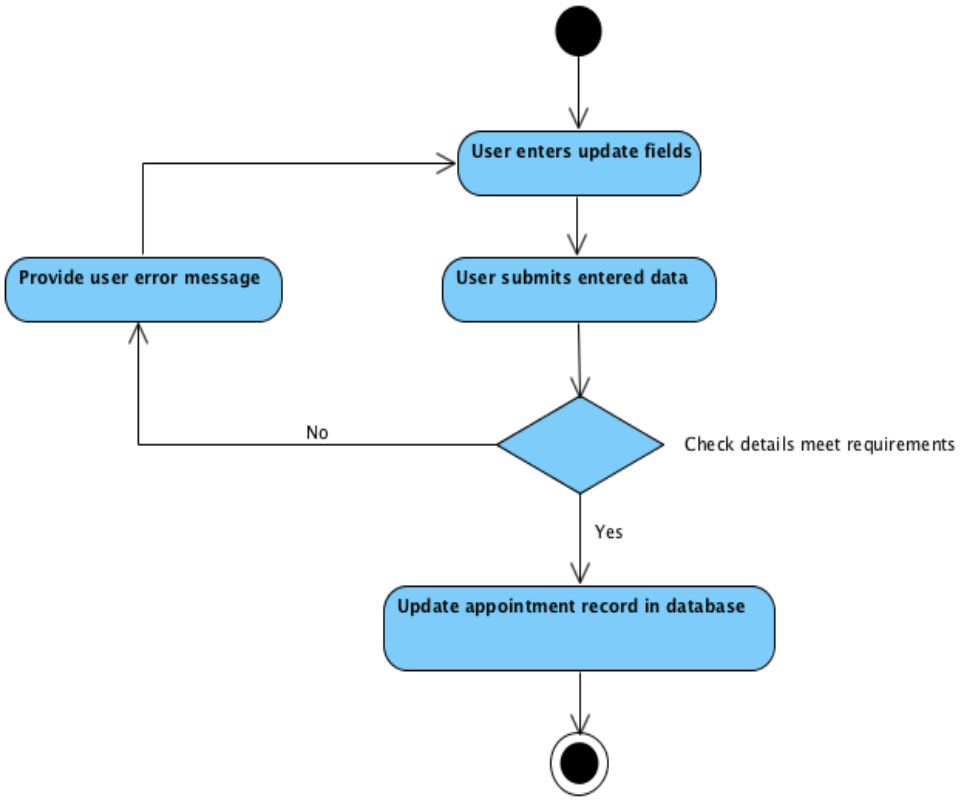
Appointment update (Doctor)

Once a patient has been seen by a doctor it is vital that the doctor updates the appointment record to document details of what occurred. This is not only for future reference but aids the data analysis tools of the system.

The Appointment update page allows users to record these details within the system by providing the required input fields – appointment id, time in, time out, and comments.

Wireframe

Activity diagram



Use case

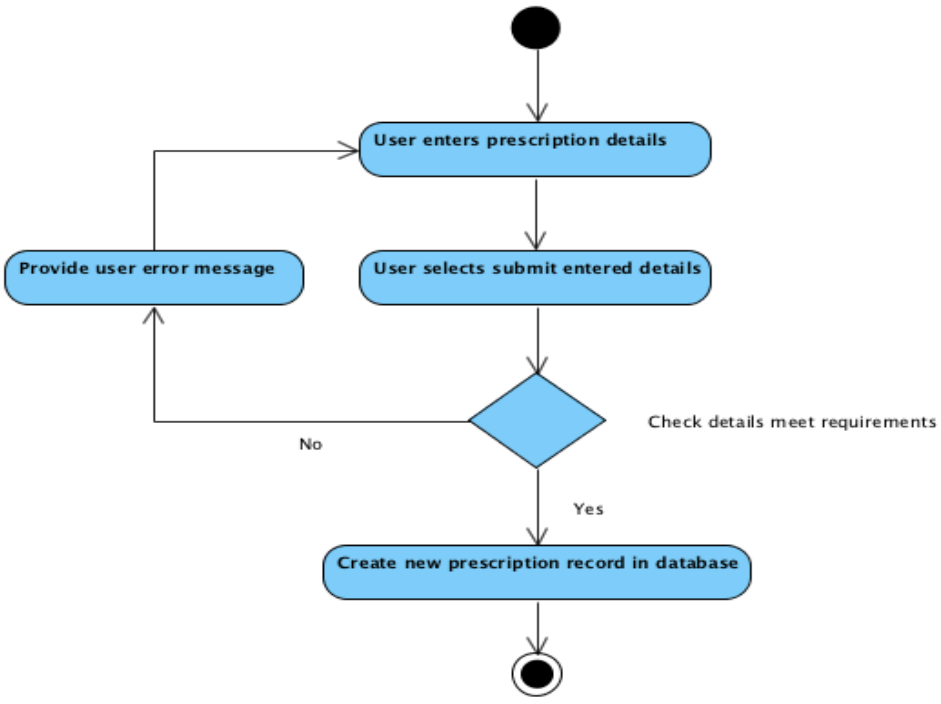
Use case name: Update appointments	
User type: Doctor	
Description: The user will navigate the system to provide an appointment update	
Pre-conditions: User has logged in with an authorised doctor account	
Basic flow:	1. User selects “appointments” tab within navigation bar 2. User locates update appointments section and selects update appointment button 3. User provided input boxes to enter appointment information 4. User inputs required appointment information 5. User must select submit button
Alternative flow:	1. User selects “appointments” tab within navigation bar 2. User locates my calendar section and selects view calendar button 3. User provided calendar of all appointments with patients 4. User locates appointment requiring updating 5. User selects that appointment event 6. User provided input boxes to enter appointment information 7. User inputs required appointment information 8. User must select submit button
Alternative flow 2:	1. User enters invalid details into input boxes 2. User is provided with an error message to inform them 3. User must ensure all input boxes are completed correctly

	4. User must select submit button
Post conditions	The user updates appointment records within the systems database

Prescribe (Doctor)

One of the main requirements identified was that doctors must be able to prescribe medication to their patients. This page provides doctor users a form to input a new prescription record to the system to prescribe individual patients medication on a one course or on going basis.

Activity diagram

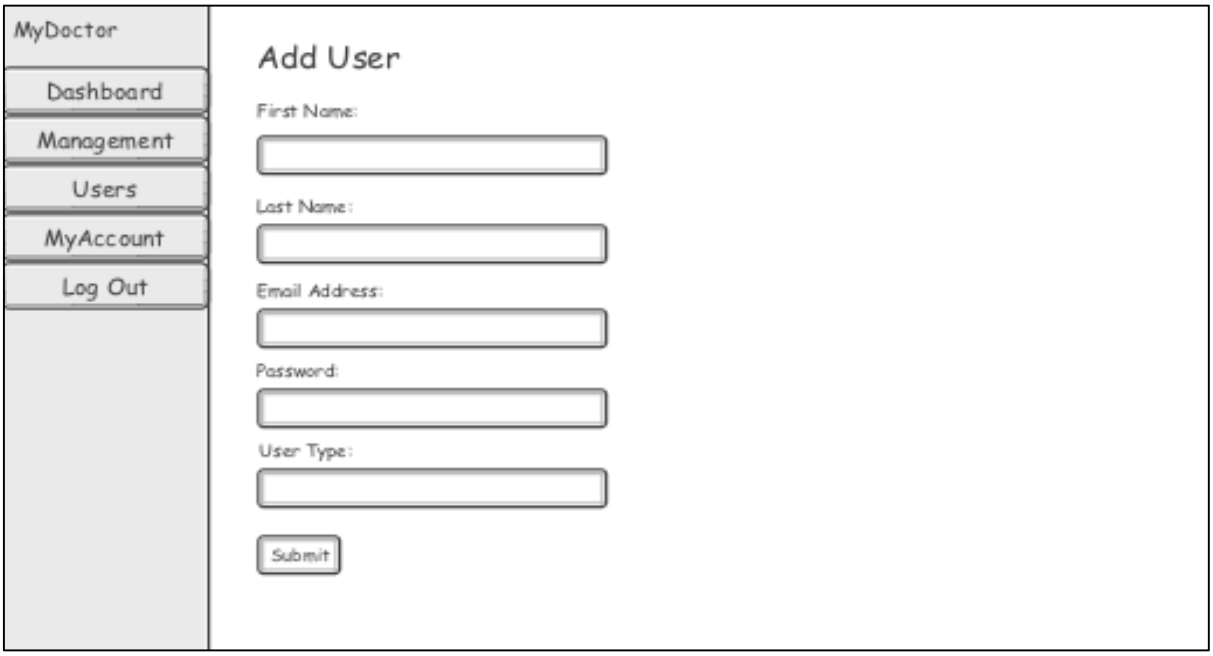


Use case

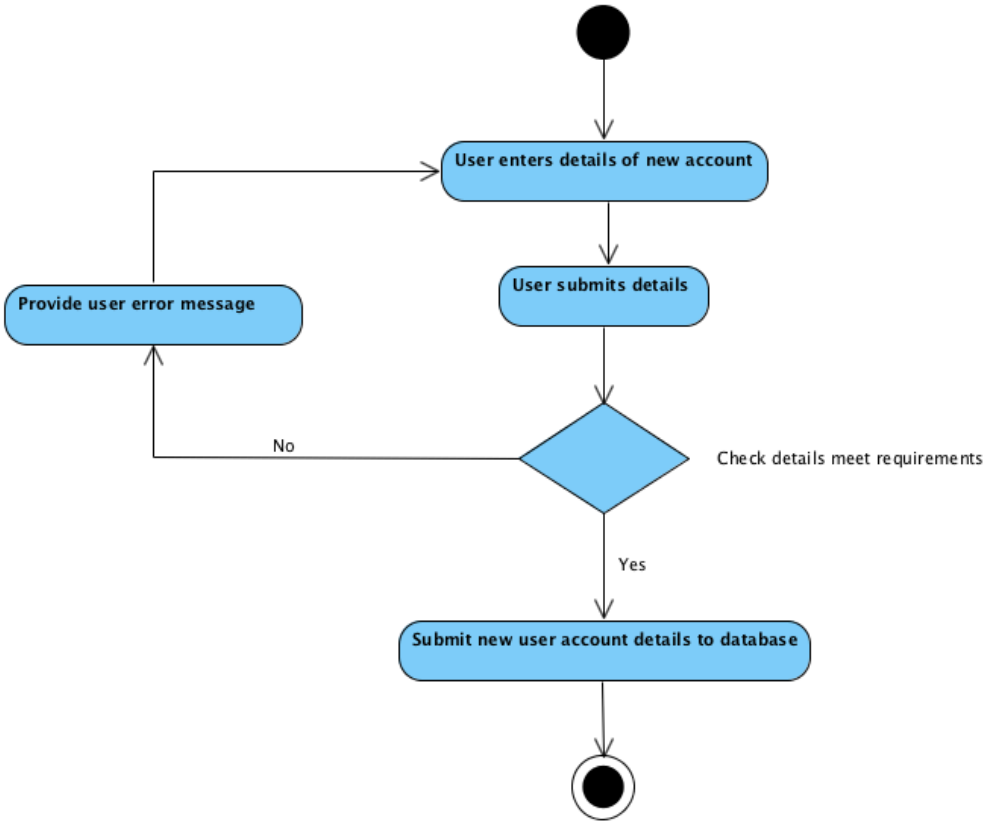
Use case name: Prescribe medication	
User type: Doctor	
Description: The user will navigate the system to provide a patient with prescription medicine	
Pre-conditions: User has logged in with an authorised doctor account	
Basic flow:	1. User selects “prescriptions” tab within navigation bar 2. User locates prescribe section and selects prescribe button 3. User provided input boxes to enter prescription information 4. User inputs required prescription information 5. User must select prescribe button
Alternative flow:	1. User enters invalid details into input boxes 2. User is provided with an error message to inform them 3. User must ensure all input boxes are completed correctly 4. User must select submit button
Post conditions	The user updates prescription records within the systems database

Add user – (Admin)

Administration users require some form of control and management of user accessibility to the system. This page provides admin users with the tools to input a new user’s details, determine their login credentials and select their user type before creating a new user record in the database.



Activity Diagram



Use case

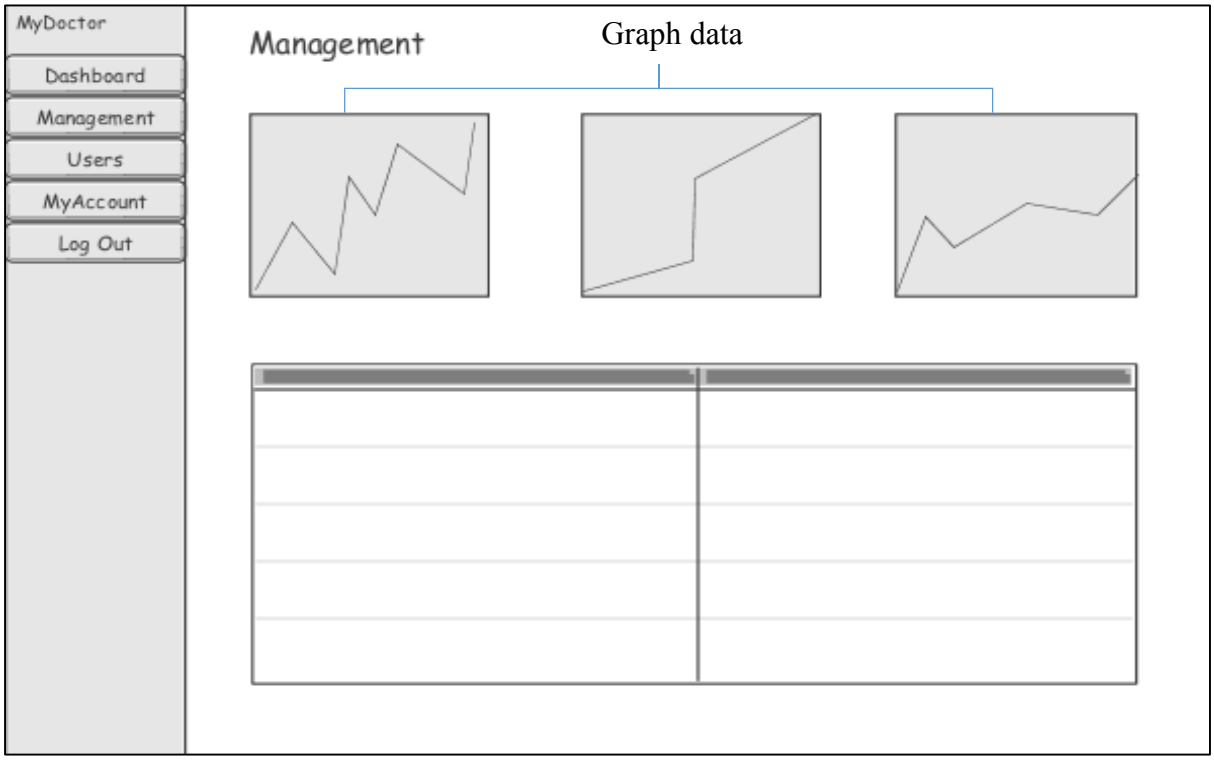
Use case name: Add user	
User type: Admin	
Description: The user will navigate the system and add a new user	
Pre-conditions: User has logged in with an authorised admin account	
Basic flow:	1. User selects “users” tab within navigation bar 2. User locates add users and selects add button 3. User provided input boxes to enter new user information 4. User must select submit button
Alternative flow:	1. User enters invalid or incomplete details 2. User is provided with an error message to inform them 3. User must re-enter account details ensuring they match criteria 4. User must select submit button
Post conditions	The user adds a new user account to the system database

Management – Data analysis tools (Admin)

Admin users will have access to a management page of the system providing up to date analysis of the information stored within the database. As previously discussed when conducting my background research, the data embedded in this page will mostly be displayed in Google Chart API.

This data involves; average appointment waiting time and length, number of patients, forecasting, prescription monitoring and more. This will allow for controlled management of the practice and aid decision making.

Wireframe



Calculations

The calculations carried out to display data on this page are mainly completed through average and count sums for example average appointment waiting time (*Average wait time sum ÷ count of appointments),* however, for more advanced calculations such as forecasting I had to use my experience gained whilst completing the Business Problem Solving and Decision Making module of the course.

The forecasting calculation Simple Moving Average could be used to calculate the number of appointments the doctor surgery should expect the following week. This method involves a moving average taken from a fixed subset size. In terms of forecasting a week's appointments this would be obtained by taking a series of data – appointment count for each day of the week for the past three weeks and then finding the average of this sum to provide our answer, an example of this process can be seen below.

// Monday one count

Count all from appointments where date is Monday

// Monday two count

Count all from appointments where date is Monday two

// Monday three count

Count all from appointments where date is Monday two

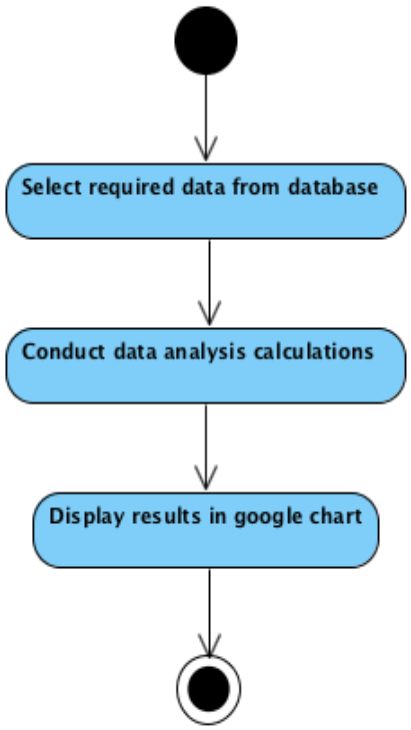
// Next monday forecast

Monday Forecast = (Monday one count + Monday two count + Monday three count) / 3

Forecast = answer

This would then be carried out for each day of the week and results displayed within a Google chart embedded in the page.

Activity diagram



Use case

Use case name: Management	
User type: Admin	
Description: The user will navigate the system and access the data analysis section	
Pre-conditions: User has logged in with an authorised admin account	
Basic flow:	1. User selects “management” tab within navigation bar 2. User provided with administration data
Post conditions	User provided analysis of system data to aid management

4.0 Implementation

Tools and Framework

Choosing the right tools and technologies to carry out the implementation of the project was highly important. Careful selection was required to ensure that the implementation process was carried out as efficiently as possible, helping to minimise the chance of errors and complete the solution within time restrictions.

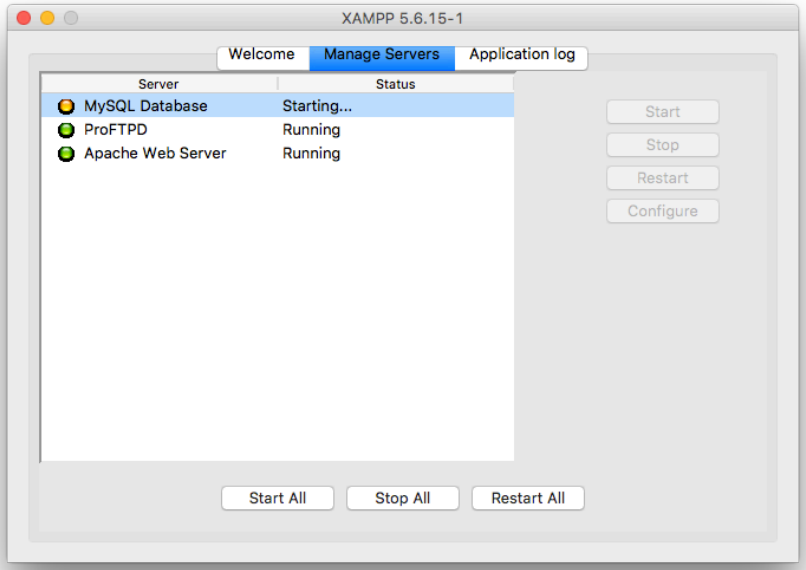
Web Development

Within this section I will outline the technologies used to implement the web application, focusing on programming and software choices.

As previously stated the main technology I used to create web application page content was HTML with CSS and Bootstrap framework to provide styling. Bootstrap helped me speed up the development of web content, by reusing blocks of code and its responsive templates. However, these pages required adaption and PHP embedded into them to create dynamic pages that allowed me to connect to, request and manipulate the database. This was done within Sublime Text 2 a free text editor. I chose to use Sublime as I had gained previous experience with it throughout my time at university. It also provided useful features to aid the coding process, for example, syntax highlighting to detect programming language being used and color code keywords etc, as seen below.

Sublime Text 2 – Syntax highlighting

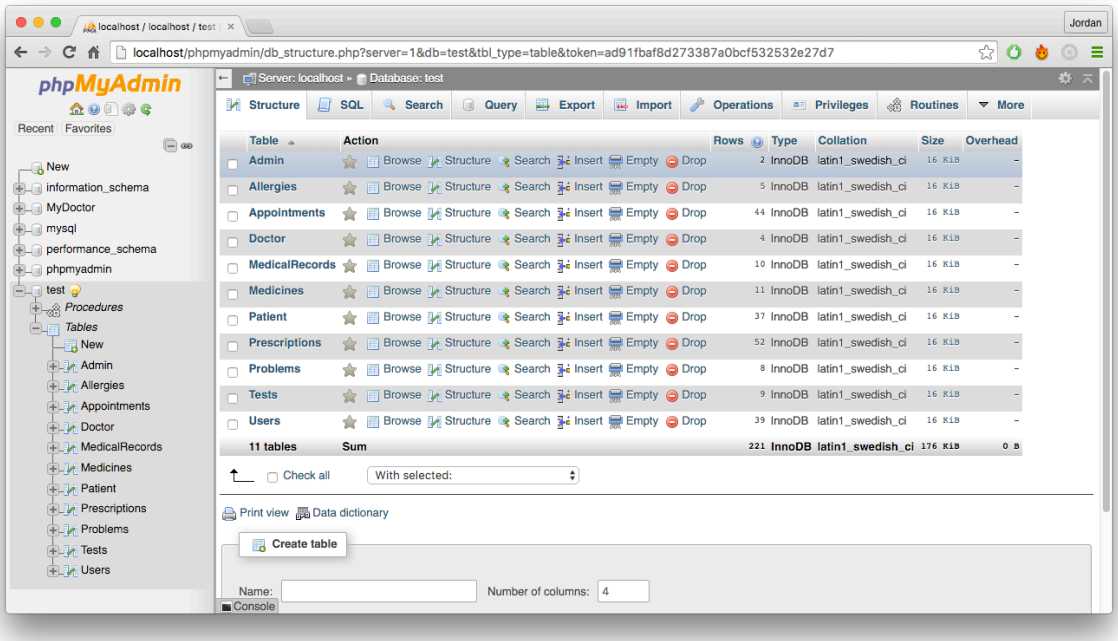
In order to develop the web application and test it within a browser environment, I adopted the use of XAMPP an open source web server developed by Apache Friends. This allowed me to create a local web server to host the web application and system database on my local device. Of course in the future this wouldn't be a viable option and a client-server architecture hosted by the doctor surgery server would be required to allow external access to users.



XAMPP software – manage servers

Database Development

As previously mentioned in the report I had chosen to use MySQL an open source relational database management system. In order to handle the administration of this database successfully, I chose to adopt the use of phpMyAdmin. This was a free software tool written in PHP made to handle the administration of MySQL databases. It allowed me to perform various tasks such as creating, modifying and deleting of tables, rows and fields within my web browser.

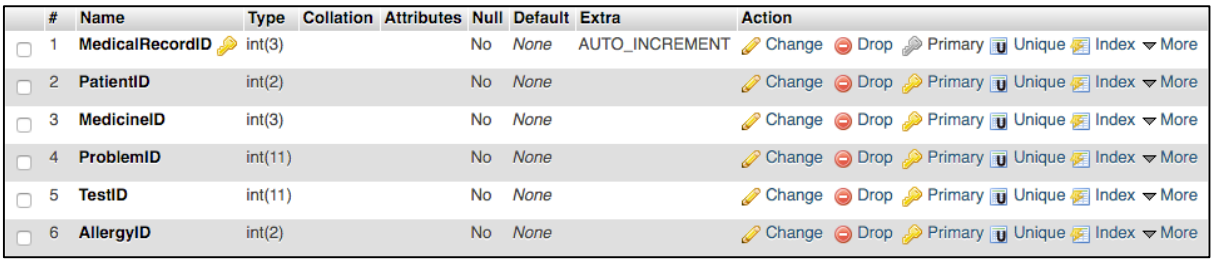


phpMyAdmin – database administration

Database Implementation

Before developing the web application, the first step I underwent was to implement the database, to manage the data stored within the system, as without this database structure the web application could not meet predefined requirements.

From analysing my previously created database design I could now use phpMyAdmin to create the database. Following my design, I created 11 tables, defining each field involved, their relationships, and data types etc.



#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	MedicalRecordID	int(3)			No	*None*	AUTO_INCREMENT	Change Drop Primary Unique Index More
2	PatientID	int(2)			No	*None*		Change Drop Primary Unique Index More
3	MedicineID	int(3)			No	*None*		Change Drop Primary Unique Index More
4	ProblemID	int(11)			No	*None*		Change Drop Primary Unique Index More
5	TestID	int(11)			No	*None*		Change Drop Primary Unique Index More
6	AllergyID	int(2)			No	*None*		Change Drop Primary Unique Index More

phpMyAdmin - Medical Records table structure

Once the MySQL database had been created, before I could access data within it, code had to be written in sublime to establish a connection between the system and server for communication purposes. Once the system was then implemented, this connection would allow for retrieval and manipulation of records within the database.

```php
<?php

$host="localhost"; // Host name
$username="root"; // Mysql username
$password=""; // Mysql password
$db_name="test"; // Database name
$tbl_name="Users"; // Table name

// Connect to server and select databse.
mysql_connect("$host", "$username", "$password")or die("cannot connect");
mysql_select_db("$db_name")or die("cannot select DB");

?>
```

Database connection code

The code seen above is for connection purposes and establishes properties that assign values to host name, MySQL username and password, as well as the name of the database to work with.

```php
<?php
session_start();
include_once 'connect.php';

```

Include connect.php code

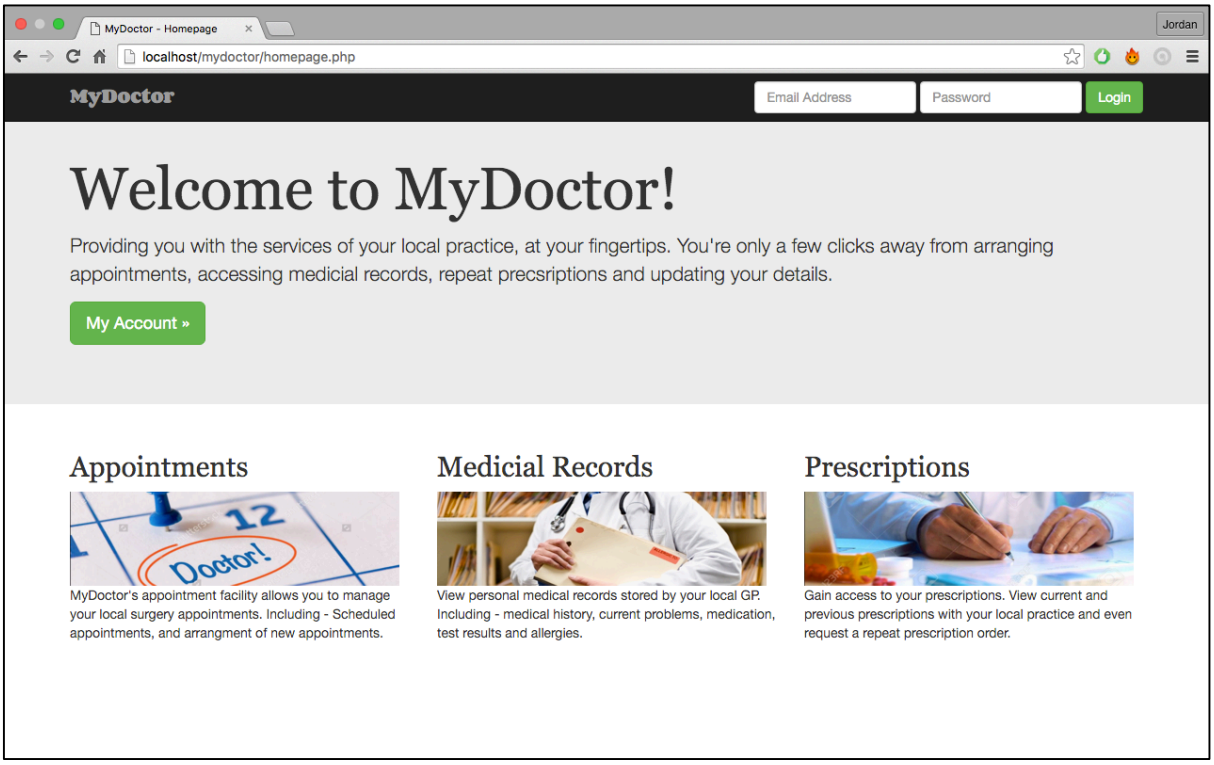
Once these have been established and saved in a file (connect.php), at the start of each page requiring database connection, "**include_once 'connect.php'**" is required. This statement includes the file connect.php within the page code without

the need of repeating the database connection code, helping to save time and reduce document size.

Web Application Implementation

Once the database had been implemented and the connection techniques established, it was time to begin the web application implementation by following my user interface design. In the following section of the report I will provide an example of these implemented system pages, explaining the code behind their functionality. All remaining pages that were designed were also implemented and can be found in the appendix.

Homepage/login



As previously designed, this was the first point of contact for the user, providing overview of system and login. Using the login input the user can enter and submit their login details. On submission these details are then submitted to be processed at 'checklogin.php' page, using the **POST** method.

```
<div id="navbar" class="navbar-collapse collapse">
  <form name="form" method="post" action="checklogin.php" class="navbar-form navbar-right">

    <div class="form-group">
      <input name="myusername" type="text" placeholder="Email Address" class="form-control" id="myusername">
    </div>

    <div class="form-group">
      <input name="mypassword" type="password" placeholder="Password" class="form-control" id="mypassword">
    </div>

    <input name="submit" type="submit" href="#" class="btn btn-success" value="Login">
  </form>
```

homepage.php - Navigation bar login code

This page retrieves the username and password sent submitted from ‘homepage.php’ page and checks if these details are stored within the database. If results show that credentials match database, then a user session is started and they’re redirected to ‘login_success.php’ page.

```php
// username and password sent from form
$myusername=$_POST['myusername'];
$mypassword=$_POST['mypassword'];

// To protect MySQL injection (more detail about MySQL injection)
$myusername = stripslashes($myusername);
$mypassword = stripslashes($mypassword);
$myusername = mysql_real_escape_string($myusername);
$mypassword = mysql_real_escape_string($mypassword);
$sql="SELECT * FROM $tbl_name WHERE username ='$myusername' and password='$mypassword'";
$result=mysql_query($sql);

// Mysql_num_row is counting table row
$count=mysql_num_rows($result);

// If result matched $myusername and $mypassword, table row must be 1 row
if($count==1){

// Register $myusername, $mypassword and redirect to file "login_success.php"
$_SESSION['myusername']= $myusername;

header("location:login_success.php");
}
else { ?>
```

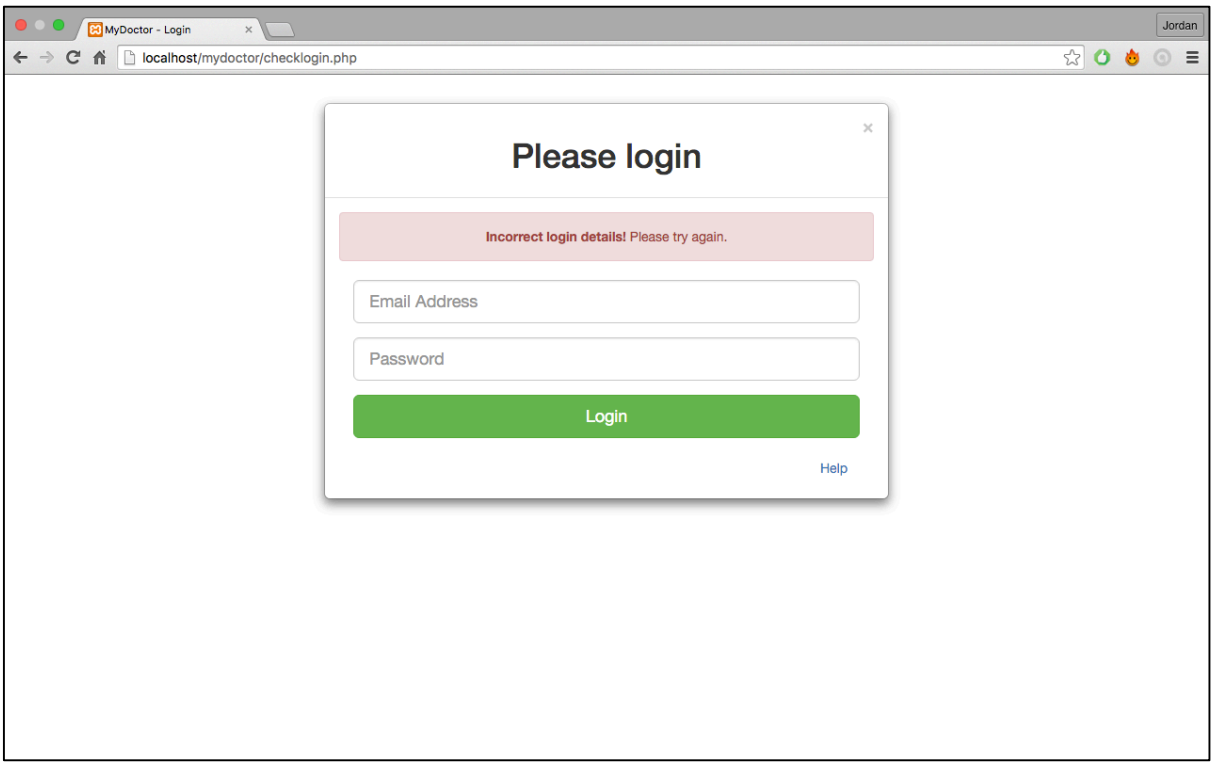
checklogin.php

If redirected to ‘login_success.php’ the user session created is used to identify the users, user id and user type. Depending on user type the user is redirected to their corresponding dashboard, an example of the code for this process can be seen below.

```php
$usersession = $_SESSION['myusername'];

// Gets users id from session
$userid = mysql_fetch_array(mysql_query("SELECT UserID FROM Users WHERE Username = '$usersession'"));

// Check if they are doctor
$doctor = mysql_query("SELECT * FROM Doctor WHERE UserID = '$userid[0]'");

// Check if they are a patient
$patient = mysql_query("SELECT * FROM Patient WHERE UserID = '$userid[0]'");

// Check if they are a patient
$admin = mysql_query("SELECT * FROM Admin WHERE UserID = '$userid[0]'");

// Check if user is in doctor table
if (mysql_num_rows($doctor) == 1) {

    // Redirect to doctor dashboard
    header ('Location: doctor.php');

}

// Check if user is in patient table
if (mysql_num_rows($patient) == 1) {

    // Redirect to patient dashboard
    header ('Location: dashboard.php');
```

login_success.php

However, if ‘checklogin.php’ retrieves login credentials that do not match the database they are displayed the ‘checklogin.php’ page seen below.



checklogin.php – Error message

User session

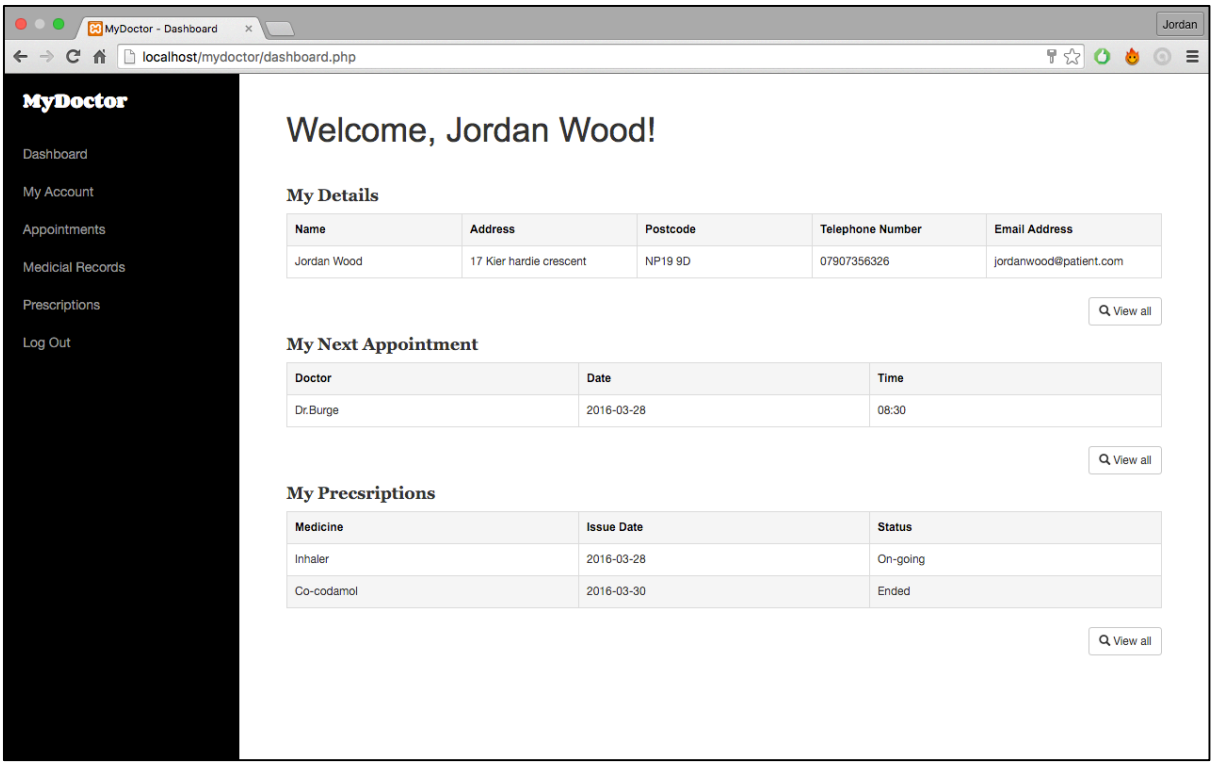
Once authorised access has been gained, in order for the system to know who the user is and store their information to be used across multiple pages, the **‘session_start()’** variable is required. This variable is placed at the top of every page within the system, to start the session set within the ‘login_success.php’ page, using **$_SESSION** global variable.

```
1  <?php
2  session_start();
3  include_once 'connect.php';
4
```

session_start() – Located at the start of every page

Dashboard

Once a user has gained authorised access and their user type determined, as previously mentioned they’re redirected to their corresponding dashboard, for example – Patient (dashboard.php), Doctor (doctor.php) and Admin (admin.php). These dashboards provide them with data and options relating to their user type and individual selves, as previously designed.



dashboard.php – Patient (Jordan Wood) dashboard

As you can see from the screen shot above, I have kept the same user interface design and previously defined, using the Bootstrap template for side navigation bar. This page depending on user type provides different navigation options and data displayed.

```php
$usersession = $_SESSION['myusername'];

// Gets users id from session
$user = mysql_query("SELECT * FROM Users WHERE Username = '$usersession'");
$userrow = mysql_fetch_array($user);

//Gets users details
$firstname = $userrow['FirstName'];
$lastname = $userrow['LastName'];
$address = $userrow['Address'];
$postcode = $userrow['Postcode'];
$telephone = $userrow['Telephone'];
$username = $userrow['Username'];

// Gets users id from session
$userid = mysql_fetch_array(mysql_query("SELECT UserID FROM Users WHERE Username = '$usersession'"));
```

dashboard.php – Retrieving user data

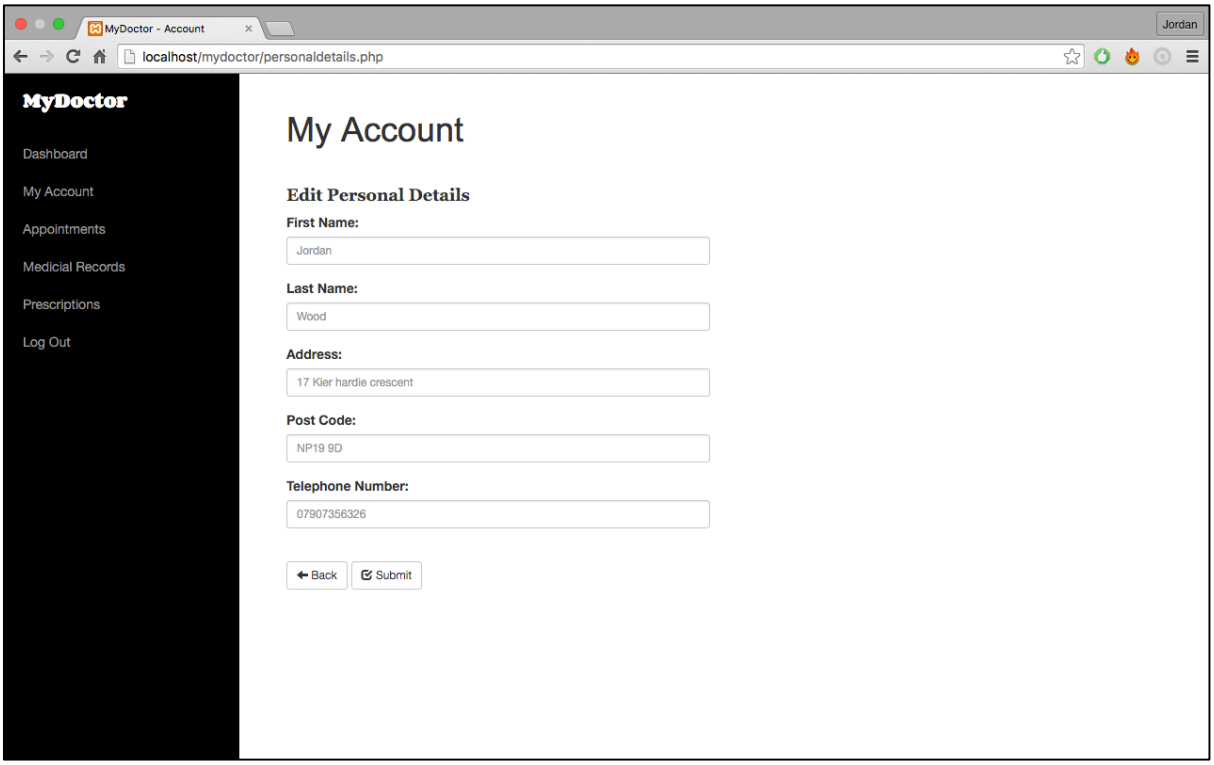
The patient dashboard retrieves information stored in the system database relating to the user. This is done using various MySQL queries with PHP, as seen above. Results from these queries are then displayed within this page to provide the user information, such as their details, next appointment and prescriptions. This process is adapted and used for all different user type dashboards, however, varying information is required and displayed to user.

```
<table class="table table-bordered table-striped table-hover">
                <tbody style="font-size:12px;">
                    <tr class="tablerow">
                        <td class="col-md-2"><b>Name<b></td>
                        <td class="col-md-2"><b>Address<b></td>
                        <td class="col-md-2"><b>Postcode<b></td>
                        <td class="col-md-2"><b>Telephone Number<b></td>
                        <td class="col-md-2"><b>Email Address<b></td>
                        </tr>
                        <tr>
                            <td><?php echo "$firstname $lastname";?></td>
                            <td><?php echo "$address";?></td>
                            <td><?php echo "$postcode";?></td>
                            <td><?php echo "$telephone";?></td>
                            <td><?php echo "$username";?></td>
                        </tr>
                    </tbody>
                </table>
```

dashboard.php – Displaying data related to user

Edit personal details

As specified in design, each user type within the system can modify their personal details. The page compromises of the required features and no changes were made from the original design.



personaldetails.php – Edit personal details

The user's current details are used as a placeholder for these input boxes, so the user is aware of their current details before modifying.

```
<form role="form" method="post">
    <div class="form-group">
      <label for="name">First Name:</label>
      <input name="FirstName" type="text" class="form-control input-sm" id="name" placeholder="<?php echo "$firstname";?>">
    </div>
    <div class="form-group">
      <label for="pwd">Last Name:</label>
      <input name="LastName" type="text" class="form-control input-sm" id="pwd" placeholder="<?php echo "$lastname";?>">
    </div>
      <div class="form-group">
      <label for="pwd">Address:</label>
      <input name="Address" type="text"  class="form-control input-sm" id="pwd" placeholder="<?php echo "$address";?>">
    </div>
      <div class="form-group">
      <label for="pwd">Post Code:</label>
      <input name="Postcode" type="text"  class="form-control input-sm"id="pwd" placeholder="<?php echo "$postcode";?>">
    </div>
      <div class="form-group">
      <label for="pwd">Telephone Number:</label>
      <input name="Telephone" type="text" class="form-control input-sm" id="pwd" placeholder="<?php echo "$telephone";?>">
    </div></br>

    <div style="float:left;">
      <a href="myaccount.php" type="button" class="btn btn-default btn-sm">
        <span  class="glyphicon glyphicon-arrow-left"></span> Back
      </a>

    <button name="update" type="submit" class="btn btn-default btn-sm">
       <span class="glyphicon glyphicon-check"></span> Submit
    </button>
  </form>
```

personaldetails.php – Updating personal details input form

On clicking of the submit button, entered details are then submitted to be processed using the **POST** method. The following if statements are used to identify if the submission criteria has been met.

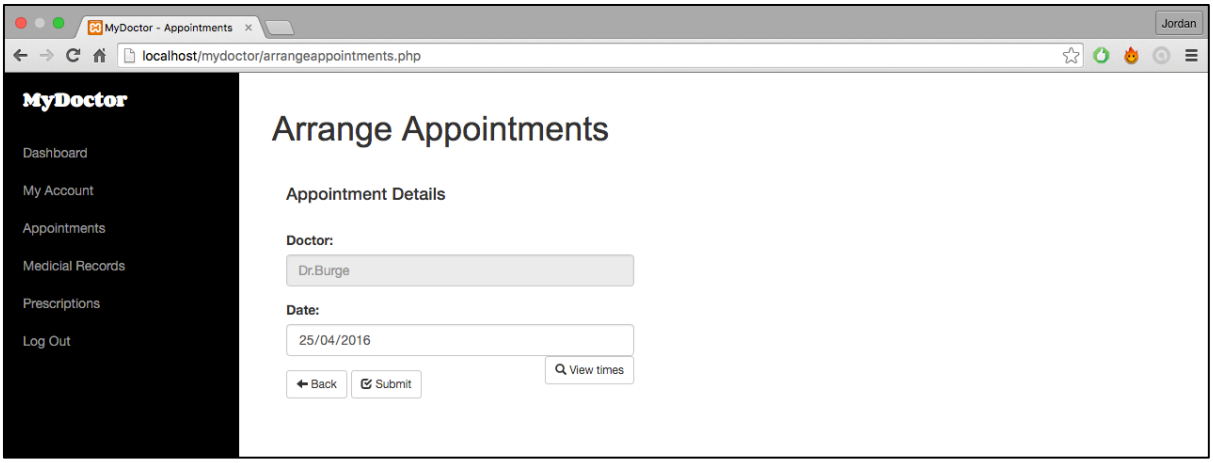
```
<?php  if(isset($_POST['update'])) {

  $newfirstname = $_POST['FirstName'];
  $newlastname = $_POST['LastName'];
  $newaddress = $_POST['Address'];
  $newpostcode = $_POST['Postcode'];
  $newtelephone = $_POST['Telephone'];


  if(empty($newfirstname or $newlastname or $newaddress or $newpostcode or $newtelephone)) {
    echo "<div class='alert alert-danger' role='alert'> <strong>Sorry!</strong> Please fill required field before selecting submit.</div>";
    } else {

      if(!empty($newfirstname)) {
      mysql_query("UPDATE Users SET FirstName = '$newfirstname' WHERE UserID = '$userid'");
        }

      if(!empty($newlastname)) {
        mysql_query("UPDATE Users SET LastName = '$newlastname' WHERE UserID = '$userid'");
                          }

       if(!empty($newaddress)) {
        mysql_query("UPDATE Users SET Address = '$newaddress' WHERE UserID  = '$userid'");
                          }

        if(!empty($newpostcode)) {
        mysql_query("UPDATE Users SET Postcode = '$newpostcode' WHERE UserID  = '$userid'");
                          }

        if(!empty($newtelephone)) {
        mysql_query("UPDATE Users SET Telephone = '$newtelephone' WHERE UserID  = '$userid'");
                          }

        echo "<div class='alert alert-success' role='alert'> <strong>Success!</strong> Your personal details have been updated!</div>";
```

personaldetails.php – Updating user record

If criteria has been met, update queries are executed, setting the corresponding fields within the User table to the newly inputted data, and displaying a success message. However, if criteria hasn't been met and input fields are empty on submission, then an error message is displayed.

Arrange appointment

One of the key features the system required to be a successful solution, was to allow patient users to arrange appointments with their doctors. This page as seen below provides the user the ability to carry out this requirement by searching for available appointments with their doctor.



arrangeappointments.php – Selecting a date and viewing times

For appointment arrangements to be managed effectively, it was important that unavailable times for doctor appointments weren't an available selection for patient users. For example, as an appointment date was entered into input field, unavailable times would be removed from drop down selection box as previously designed. However, as I hadn't previously had experience implementing a feature using this method it proved a problem.

To overcome this, I decided to create a **'view times'** button. This button would be clicked by patient user, after selecting a date. Using the POST method, once the button was clicked the following code was executed.

```
// Execute this code once the view times button is clicked
if(isset($_POST['gettimes'])) {

  // Get date user inputted
  $appdate = $_POST['Date'];

  // Convert date to MySQL format
  $newappdate = date('Y-m-d', strtotime($appdate));

  ?>
```

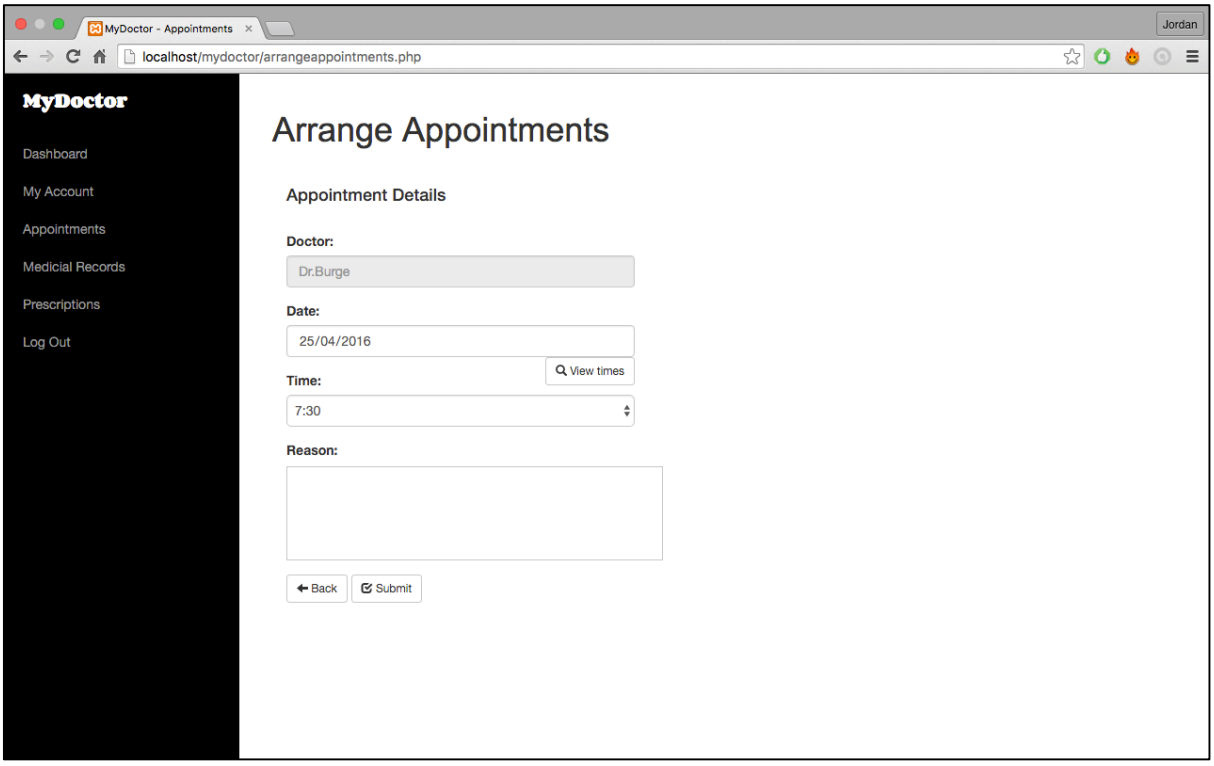
arrangeappointments.php – Date conversion

Once executed this code retrieves the date entered by the user and converts it into MySQL format, so it can be compared with dates stored within the database.

An array is created containing all possible appointments times **($apptimes)** and an array containing all taken appointment times **($bookedapps).**

```php
// Create array
$bookedapps = array();

// Get appointment times for selected date
$apptimes = mysql_query("SELECT * FROM Appointments WHERE DoctorID = '$doctorid[0]' AND date = '$newappdate'") or die(mysql_error());

while ($approw = mysql_fetch_array($apptimes)) {
    $time = $approw['Time'];

  // Insert every taken appointment time into the booked array
  array_push($bookedapps, $time);

}

// Array with all the possible times
$times = array("7:30:00", "7:45:00", "8:00:00", "8:15:00", "8:30:00", "8:45:00", "9:00:00", "9:15:00", "9:30:00", "9:45:00",
               "10:00:00", "10:15:00", "10:30:00", "10:45:00", "11:00:00", "11:15:00", "11:30:00", "11:45:00", "12:00:00");

// Compare the two arrays and remove duplicates and insert into new array
$available = array_diff($times, $bookedapps);

// Count how many values are now in the new array
$arrlength = count($available);

    echo"<div class='form-group'>
    <label for='pwd'>Time:</label>
    <select name='Time' type='time' class='form-control' id='pwd'>";

      $x = 0;
      // Insert every available time into the drop down list
      while ($x <= $arrlength) {
          $timecut = substr($available[$x], 0, -3);
          echo"<option value='$timecut'>$timecut</option>";
          $x++;
    }
?>
```

These arrays are then compared to remove duplicates and insert available times into a new array **($avaliable)**.

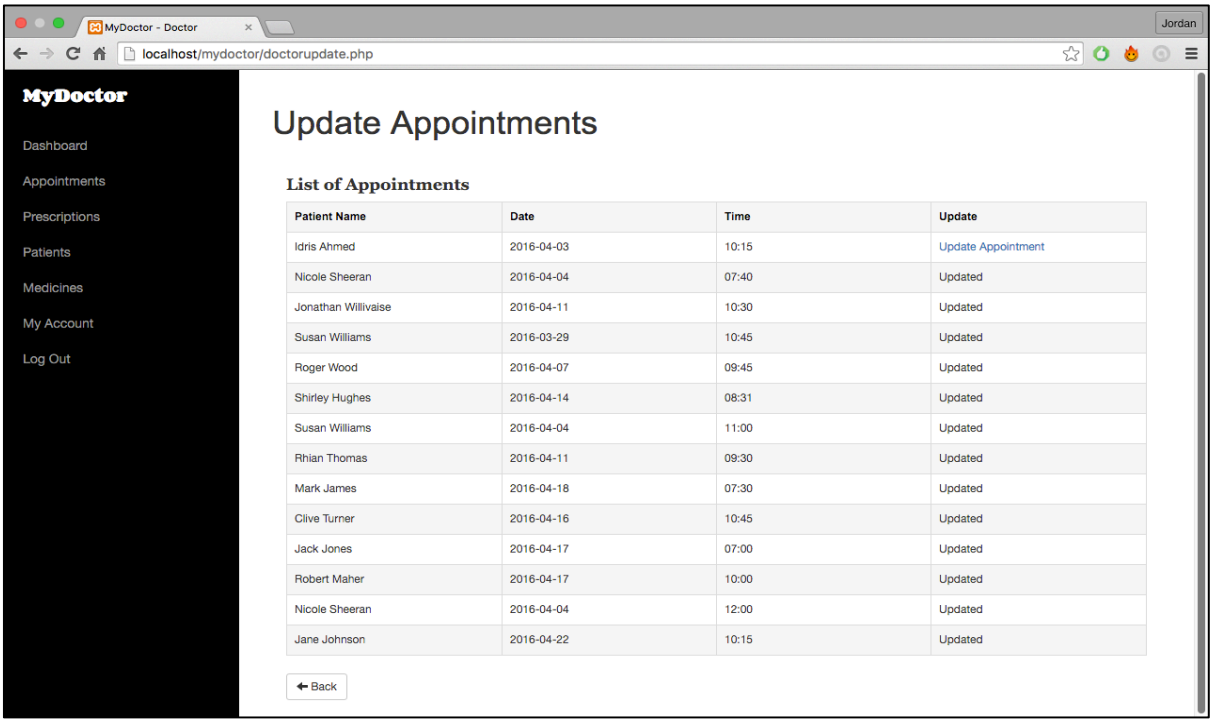


arrangeappointments.php – Avaliable times and Reason input displayed

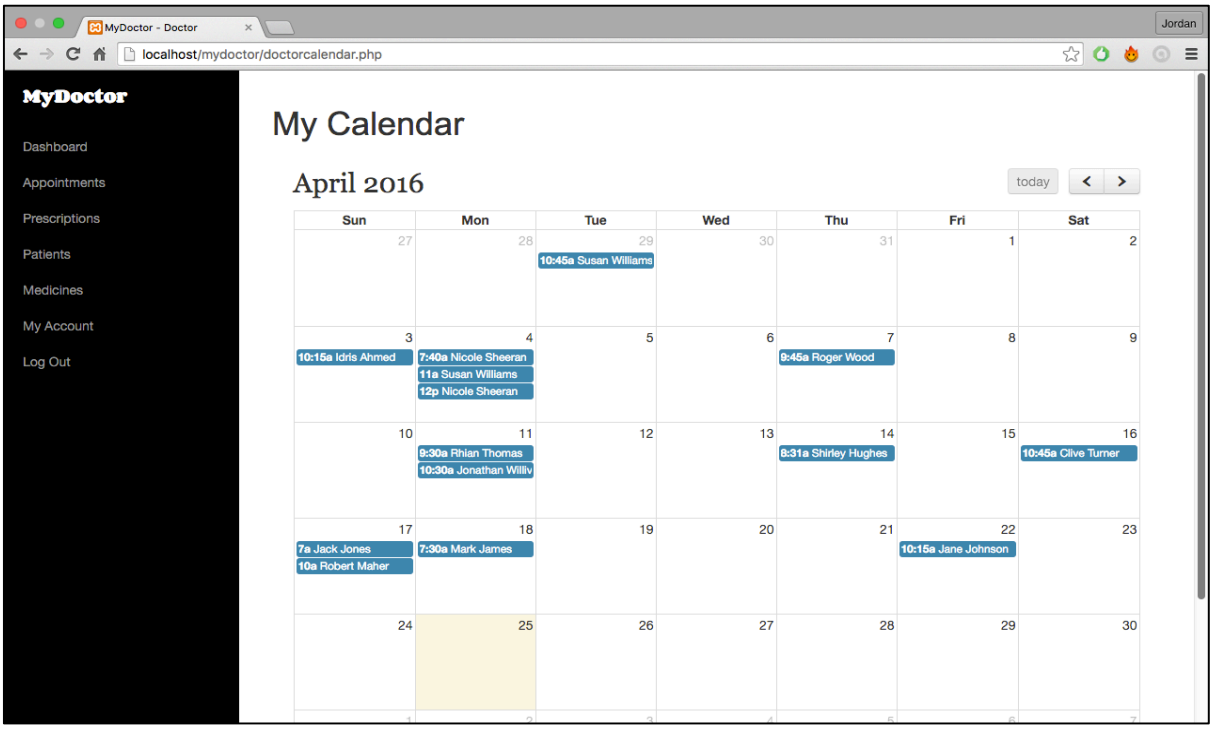
This array is then displayed into a drop down list to be selected by the patient user and a reason input box displayed also.

View appointments

Another essential requirement was that Doctor and Patient users must be able to view their appointments. It was previously designed that this page would consist of a list of appointments, in table format, however, after further consideration it was decided that this format could make identifying appointments difficult. To overcome this both list format and calendar formats were implemented into the system.



doctorupdate.php – Appointments in table format



doctorcalendar.php – Appointments in calendar format

When considering how to implement a calendar to display appointments, I firstly considered Google Calendar API as, as previously mentioned I had some experience with Google API's. However, I had great difficulty trying to implement the API into the system, and couldn't locate any straight forward tutorials or helpful resources online. I

To overcome this obstacle, I decided to find an alternative solution. After some further research, I located an open source JavaScript event calendar called **FullCalendar** (12), that also provided a straight forward tutorial. The first step was to include the FullCalendar stylesheet as well as the FullCalendar, jQuery, and Moment JavaScript files within the head of 'doctorcalendar.php'.

```
<link href='fullcalendar/fullcalendar.css' rel='stylesheet' />
<link href='fullcalendar/fullcalendar.print.css' rel='stylesheet' media='print' />
<script src='fullcalendar/lib/moment.min.js'></script>
<script src='fullcalendar/lib/jquery.min.js'></script>
<script src='fullcalendar/fullcalendar.min.js'></script>
```

doctorcalendar.php – Header

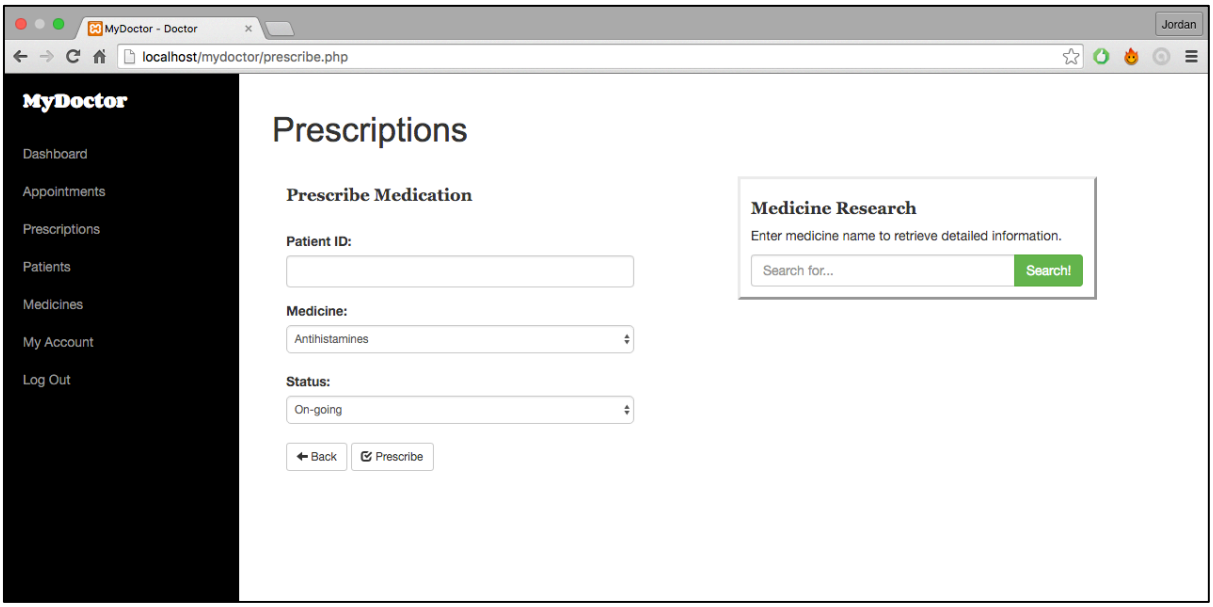
Once I had created the dependencies, the next step was to write the JavaScript code that initialises the calendar. Now I could adapt the code and create a while loop to retrieve the user's appointment data to be displayed in the calendar.

```
<script>

  $(document).ready(function() {

    $('#calendar').fullCalendar({
      defaultDate: '<?php echo $today; ?>',
      editable: true,
      eventLimit: true, // allow "more" link when too many events
      events: [
<?php

      // Get appointment details
      $allappointments = mysql_query("SELECT * FROM Appointments WHERE DoctorID = '$doctorid[0]'");

      while ($allappointmentsrow = mysql_fetch_array($allappointments)) {

        $patientsid = $allappointmentsrow['PatientID'];
        $appointmentid = $allappointmentsrow['AppointmentID'];

      $patientuserid = mysql_fetch_array(mysql_query("SELECT UserID FROM Patient WHERE PatientID = '$patientsid'"));
      $userdata = mysql_fetch_array(mysql_query("SELECT * FROM Users WHERE UserID = '$patientuserid[0]'"));

        $patientFirstName = $userdata['FirstName'];
        $patientLastName = $userdata['LastName'];
        $appointmentdate = $allappointmentsrow['Date'];
        $appointmenttime = $allappointmentsrow['Time'];
        $Timecut = substr($appointmenttime, 0, -3);
        $comments = $allappointmentsrow['Comments'];


echo "{
        title: '$patientFirstName $patientLastName',
        start: '$appointmentdate"; echo "T$Timecut', ";

        if (empty($comments)) {

        echo "url: 'doctorupdateappointments.php?id=$appointmentid'";

        }

        echo "},";

      }
?>
      ]
    });
```

doctorcalendar.php – Calendar code

Prescribe/Research

It was earlier identified that doctor users must be able to prescribe medication to patients. The page implemented shown below allows doctor users to input patient details **(patient id)**, prescribe a choice of medication and its status.



prescribe.php – Doctor users may prescribe and research medication

On implementation the interface was modified slightly from the predefined wireframe design. It became apparent that the desirable requirement of a medicine research feature could be implemented into the prescribe page with little effort and without affecting the project scope. The original wireframe design for the page can be seen below.

Original design for prescribe.php page

The implemented page uses the **POST** method similar to that previously used for appointment arrangement. Once the details inputted into the form are submitted (by selecting prescribe), they are processed using the **POST** method seen below.

```
<form role="form" method="post" class="prescriptionform">
    <div class="form-group">

        <label for="pwd">Patient ID:</label>
            <input name="PatientID" class="form-control" id="pwd" type="number">
        </div>

        <label for="name">Medicine:</label>
            <select name="MedicineName"  class="form-control input-sm">

<?php
        // Get Medicines
        $medicines = mysql_query("SELECT * FROM Medicines WHERE MedicineID != '11' ORDER BY MedicineName ASC");

        while ($medicinesrow = mysql_fetch_array($medicines)) {

            $Medicine = $medicinesrow['MedicineName'];

            echo "  <option>$Medicine</option>";
        }
?>
            </select> </br>

        <label for="name">Status:</label>

            <select name="Status"  class="form-control input-sm">
                <option>On-going</option>
                <option>One course</option>
            </select>
        </br>

    <div style="float:left;">
        <a href="doctorprescriptions.php" type="button" class="btn btn-default btn-sm">
            <span  class="glyphicon glyphicon-arrow-left"></span> Back
        </a>

        <button name="prescribe" type="submit" class="btn btn-default btn-sm">
          <span class="glyphicon glyphicon-check"></span> Prescribe
        </button>

</form>
```

prescribe.php – prescription input form

This process if meets the criteria inserts the information into the Prescriptions table creating a new record and then displaying a success message. If this criteria has not been met a new record will not be created and an error message will be displayed to user.

Inputted data

Empty field error

Insert data into new record

```
<?php    if(isset($_POST['prescribe']))  {

        $patientid = $_POST['PatientID'];
        $medicine = $_POST['MedicineName'];
        $status = $_POST['Status'];

        $medicineid = mysql_fetch_array(mysql_query("SELECT MedicineID FROM Medicines WHERE MedicineName = '$medicine'"));

        if(empty($patientid)){

        echo "<div class='alert alert-danger' role='alert'> <strong>Sorry!</strong> Please fill all fields before selecting prescribe.</div>";
        } else {

        mysql_query("INSERT INTO Prescriptions (PrescriptionID, PatientID, DoctorID, MedicineID, IssueDate, Status, RepeatRequest) VALUES
                  ('', '$patientid', '$doctorid[0]', '$medicineid[0]', '$today', '$status', '0')");

        echo "<div class='alert alert-success' role='alert'> <strong>Success!</strong> Prescription has be recorded.</div>";
    }
```

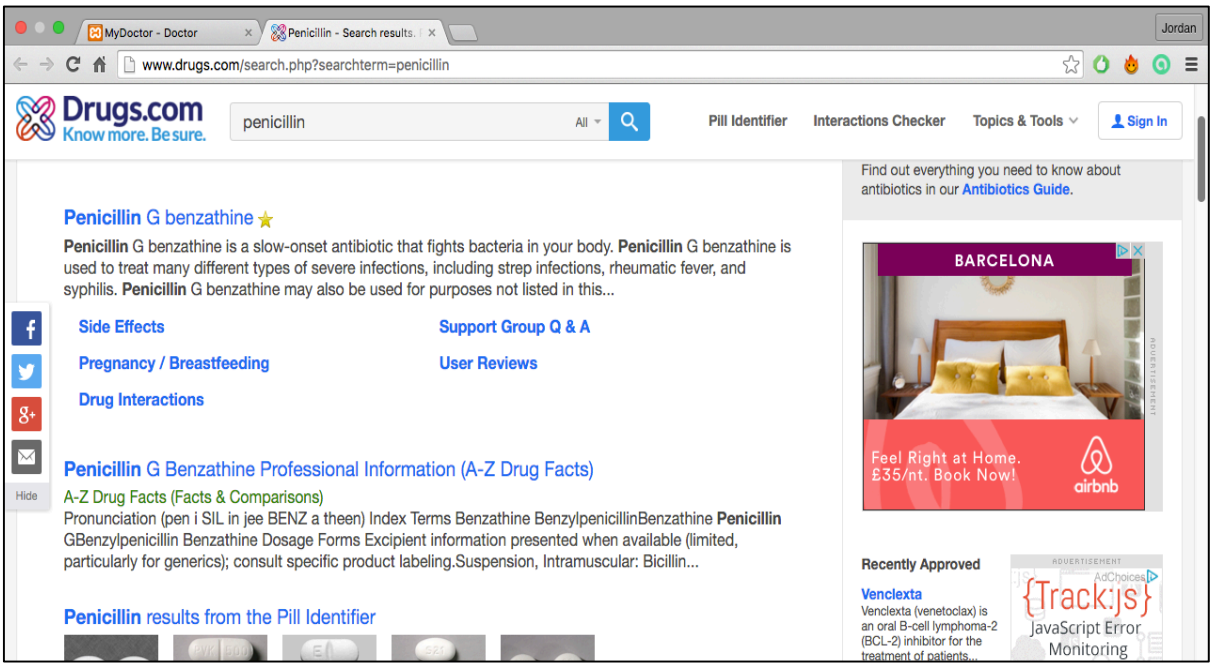
prescribe.php – Creating new prescription record with inputted details

If doctors want to ensure that they are prescribing the most effective medication to a patient, a medicine research box to retrieve this information was implemented in the page also to aid their decision.

```
<h4 class="header3">Medicine Research</h4>

    <p>Enter medicine name to retrieve detailed information.</p>

<form action="http://drugs.com/search.php?searchterm=" target="_blank">
        <div class="input-group">
            <input name="searchterm" class="form-control" placeholder="Search for...">
        <span class="input-group-btn">
            <button class="btn btn-success" type="submit">Search!</a></button>
        </span>
</form>
```

prescribe.php – Medicine search

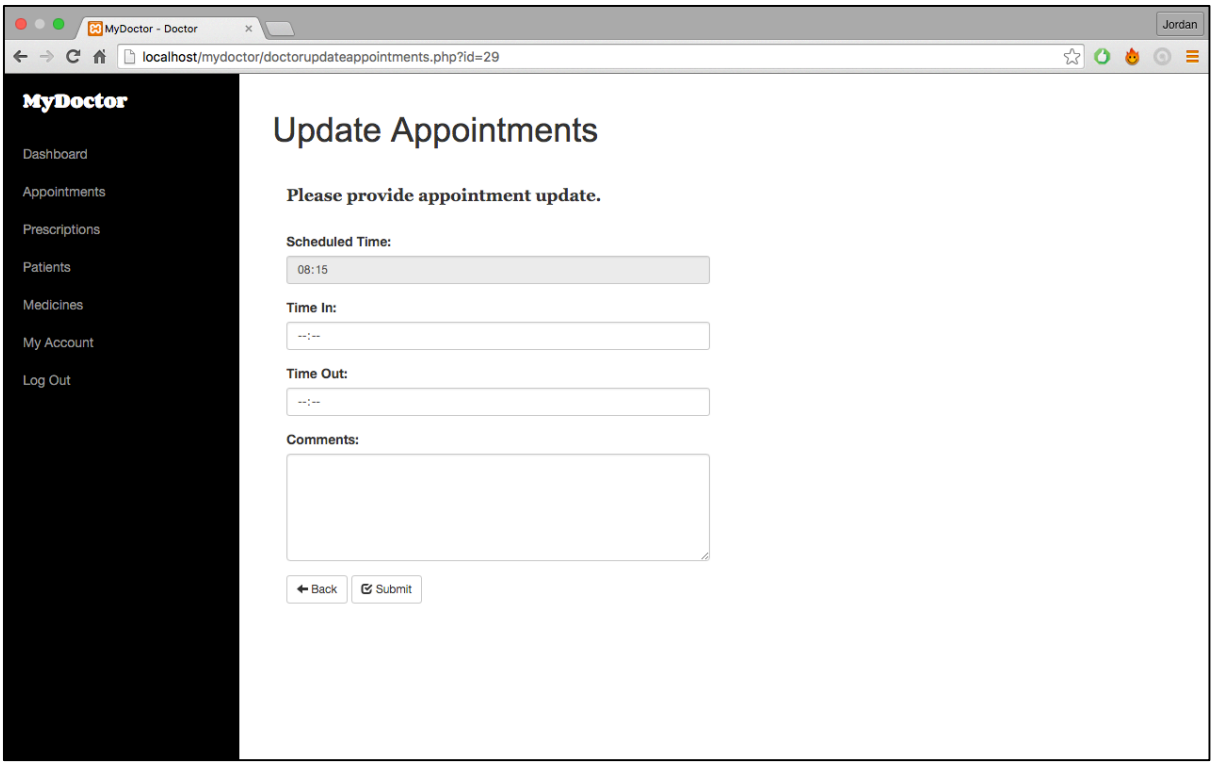
This section of code provides the search input field. Once a search term is entered and submitted by selecting the **"Search!"** button, the search term is placed on the end of the **"http://drugs.com/search.php?searchterm="** url and opened within a new tab, to display results. An example for the search "penicillin" can be seen below.



New tab for medicine search results

Appointment update

The updating of appointment records is a key requirement for doctors, this information can be stored for future reference and aid decisions etc. However, it also plays a big part in data analysis, as is related to appointment waiting time and length analysis etc.



doctorupdateappointments.php – Input form

Once appointments are selected for update the user is redirected to update appointments page, providing the scheduled time of the appointment and required input fields to complete updating.

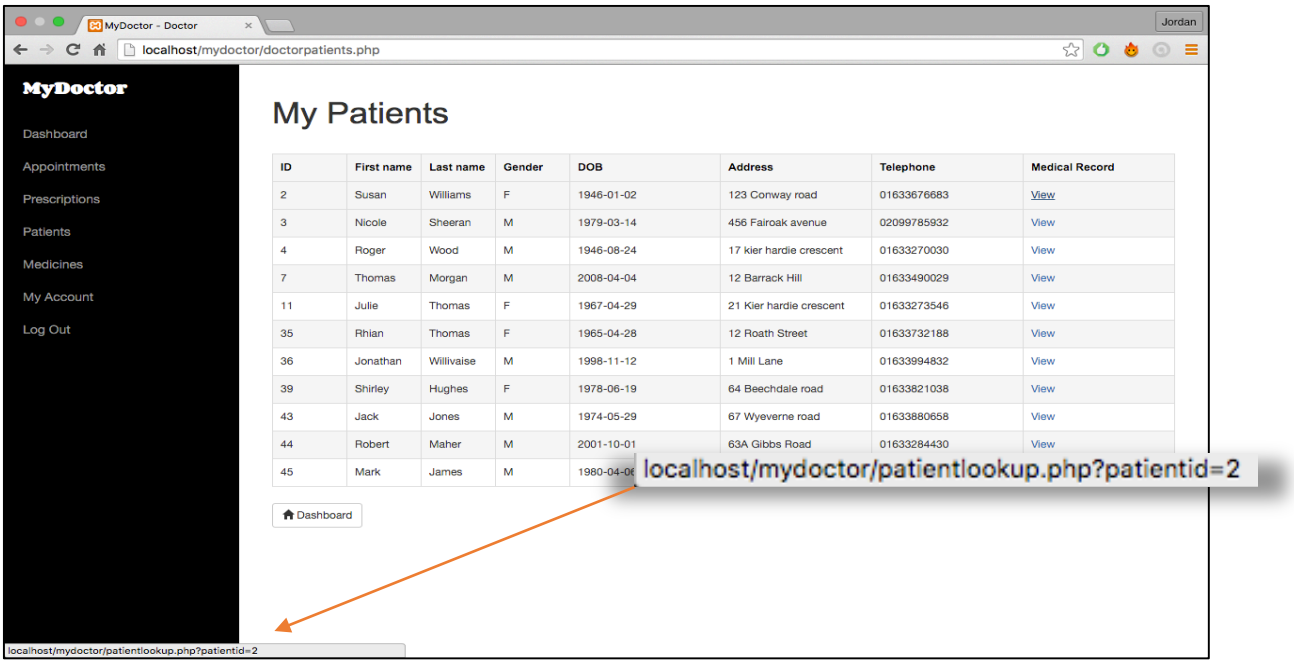
```php
<?php  if(isset($_POST['update'])) {

  $newtimein = $_POST['TimeIn'];
  $newtimeout = $_POST['TimeOut'];
  $newcomments = $_POST['Comments'];

   if(empty($newtimein and $newtimeout and $newcomments)) {
    echo "<div class='alert alert-danger' role='alert'> <strong>Sorry!</strong> Please fill all fields before selecting submit.</div>";
    } else {

      mysql_query("UPDATE Appointments SET TimeIn = '$newtimein', TimeOut = '$newtimeout', Comments ='$newcomments'
                  WHERE AppointmentID = '$appointmentid'");
      echo "<div class='alert alert-success' role='alert'> <strong>Success!</strong> Appointment record has been updated.</div>";

            $_SESSION['myusername'] = $usersession;

                                    }
  }
```

doctorupdateappointment.php – POST Method to update existing appointment record

Inputted details are processed using the **POST** method, if these meet criteria they're added to existing appointment records using **UPDATE**. If not, again like previous input forms, error message is displayed to user.

View patients & Medical records

Derived from system requirements, it was essential that doctor users could view their patients and their medical records. By navigating to the implemented patients page, doctor users can select the view link in the medical records row of the corresponding patient they would like to view.



doctorpatients.php – Link to patientlookup.php

This is done by linking to the patientlookup.php page, and using the **'$id'** variable as seen below to retrieve individual records.

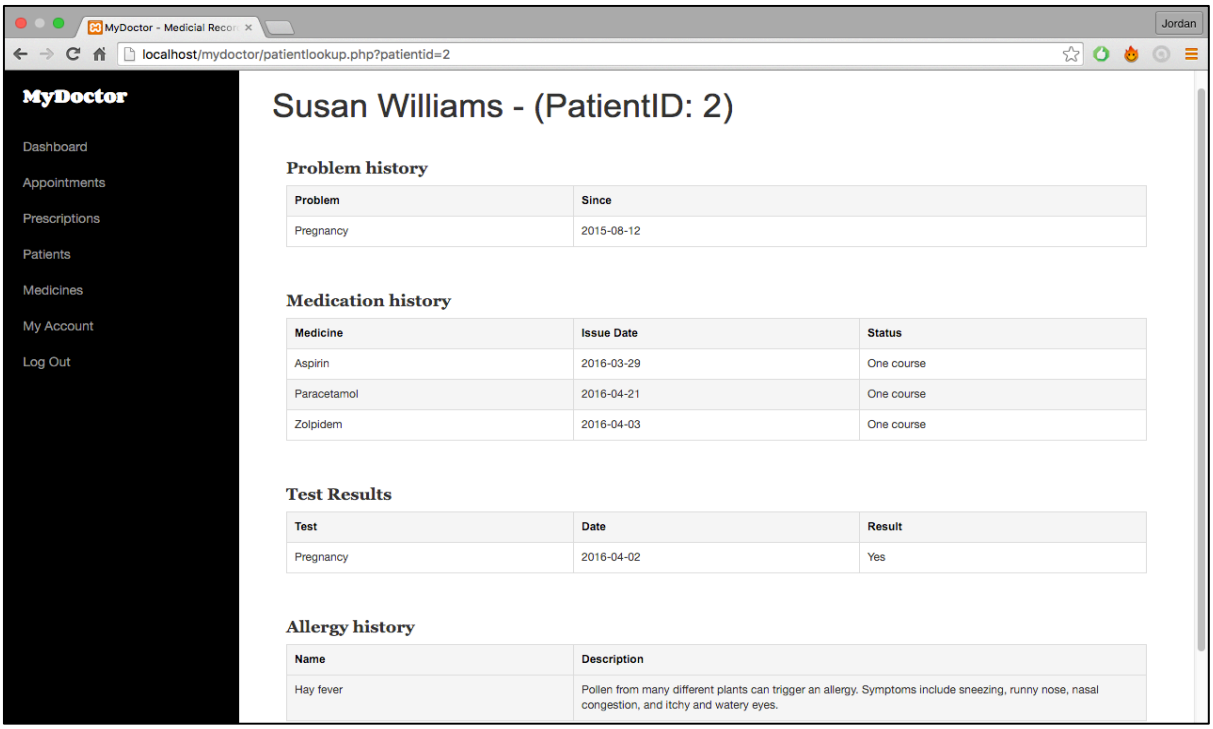
```php
<?php

        // Get patients details
        $patients = mysql_query("SELECT * FROM Patient WHERE DoctorID = '$doctorid[0]' ORDER BY PatientID");

        while ($patientdetailsrow = mysql_fetch_array($patients)) {

        $patientsid = $patientdetailsrow['UserID'];
        $id = $patientdetailsrow['PatientID'];

        $patientdata = mysql_fetch_array(mysql_query("SELECT * FROM Users WHERE UserID = '$patientsid'"));

        $FirstName = $patientdata['FirstName'];
        $LastName = $patientdata['LastName'];
        $Gender = $patientdata['Gender'];
        $DOB = $patientdata['DOB'];
        $Address = $patientdata['Address'];
        $Telephone = $patientdata['Telephone'];


echo    "<tr>
        <td>$id</td>
        <td>$FirstName</td>
        <td>$LastName</td>
        <td>$Gender</td>
        <td>$DOB</td>
        <td>$Address</td>
        <td>$Telephone</td>
        <td><a href='patientlookup.php?patientid=$id'>View</a></td>
      </tr>";

 }
```

$id variable - PatientID

Page link with variable

doctorpatients.php – Individual patient medical record retrieval

Once link has been selected the doctor is then redirected to the corresponding medical record, displaying the required information from database.



patientlookup.php?patient=id2 – Example of patient medical record (Doctor view)

This process is done by using the **GET** method. This method requests data from a specified resource and in this case it was used to retrieve the '**patientid**' from the previous page 'doctorpatients.php'. This variable is then used to fetch records from the database to be displayed to a doctor.

GET METHOD

etrieval of
ata

```php
$patientid = $_GET['patientid'];
$userid = mysql_fetch_array(mysql_query("SELECT UserID FROM Patient WHERE PatientID = '$patientid'"));
$userdetails = mysql_fetch_array(mysql_query("SELECT * FROM Users WHERE UserID = '$userid[0]'"));
$firstname = $userdetails['FirstName'];
$lastname = $userdetails['LastName'];

$MedicalRecords = mysql_fetch_array(mysql_query("SELECT ProblemID FROM MedicalRecords WHERE PatientID = '$patientid'"));

// Look up problems
$Problems = mysql_query("SELECT * FROM Problems WHERE ProblemID = '$MedicalRecords[0]'");

while ($Problemsrow = mysql_fetch_array($Problems)) {

$ProblemName = $Problemsrow['ProblemName'];
$Since = $Problemsrow['Since'];

}

// Look up medical records
$MedicalRecords = mysql_fetch_array(mysql_query("SELECT TestID FROM MedicalRecords WHERE PatientID = '$patientid[0]'"));

// Look up tests
$tests = mysql_query("SELECT * FROM Tests WHERE TestID = '$MedicalRecords[0]'");

while ($testsrow = mysql_fetch_array($tests)) {

$TestName = $testsrow['TestName'];
$TestDate = $testsrow['TestDate'];
$TestDescription = $testsrow['TestDescription'];
$TestResults = $testsrow['TestResults'];

}
```

patientlookup.php – GET method used to retrieve data

The process of displaying getting and displaying this data within the page can be seen below. It was also important that if information wasn't stored about this individual patient in the database, the doctor was aware of this. This was done by

using the **'count()'** function, this function returns the number of elements in an array. For example, as seen below the function is used to count the number of records within the MedicalRecords table where the PatientID is the individual's patient id. An If statement is then used, IF this count equals 0 then the error message "Patient has no problem history is displayed".

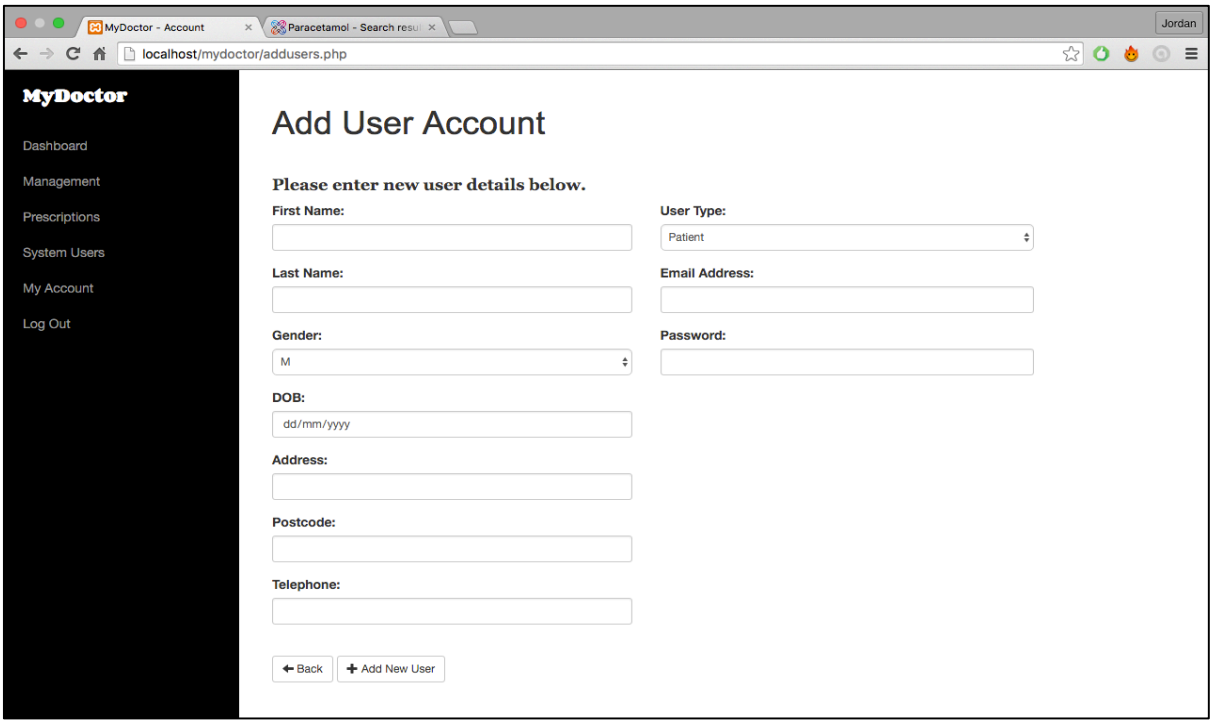
However, patients may have a medical record that shows that they do not have any problems. This is indicated in this example as having the problem id '8'. A While loop is used to execute the following block of code if the specified condition is true. The executed code **SELECTS** all from Problems table where the PatientID is the individual's id and then checks if the ProblemID is 8 using an IF statement. If problem id equals 8 the doctor would be shown no problem history message, **'else'** the corresponding details would be displayed in the problem history table.

```
    <h4 class="header3">Problem history</h4>
      <table class="table table-bordered table-striped table-hover">
          <tbody style="font-size:12px;">
              <tr class="tablerow">
                  <td class="col-md-1"><b>Problem<b></td>
                  <td class="col-md-2"><b>Since<b></td>
                  </tr>


<?php

    // Get problem details
    $problem = mysql_query("SELECT * FROM MedicalRecords WHERE PatientID = '$patientid'");
    $problemcount = mysql_fetch_array(mysql_query("SELECT count(*) FROM MedicalRecords WHERE PatientID = '$patientid'"));

        if ($problemcount[0] == 0) {
            echo "<tr><td colspan='3'>Patient has no problem history.</td></tr>";
        }

        while ($problemsrow = mysql_fetch_array($problem)) {

            $problemid = $problemsrow['ProblemID'];

            $problemsdata = mysql_fetch_array(mysql_query("SELECT * FROM Problems WHERE ProblemID = '$problemid'"));

            $ProblemName = $problemsdata['ProblemName'];
            $Since = $problemsdata['Since'];

            if ($problemid == '8') {

            echo "<tr><td colspan='3'>Patient has no problem history.</td></tr>";

        } else {
        echo "<tr>
                    <td>$ProblemName</td>
                    <td>$Since</td>
                    </tr>";

    }
}
?>
                    </tbody>
                  </table></br>
```

patientlookup.php – Displaying patient medical record details

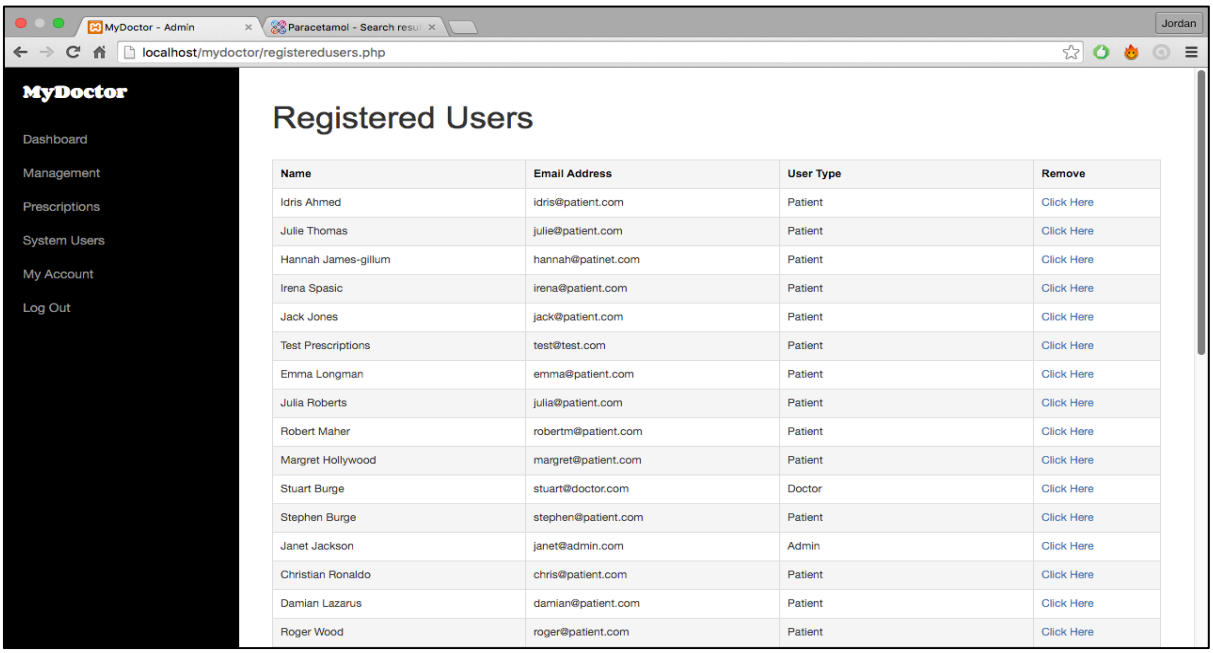
View/add/remove users

One of the key tasks for administration users is to manage users of the system. This can be done by viewing users and being able to add or remove them from the system.



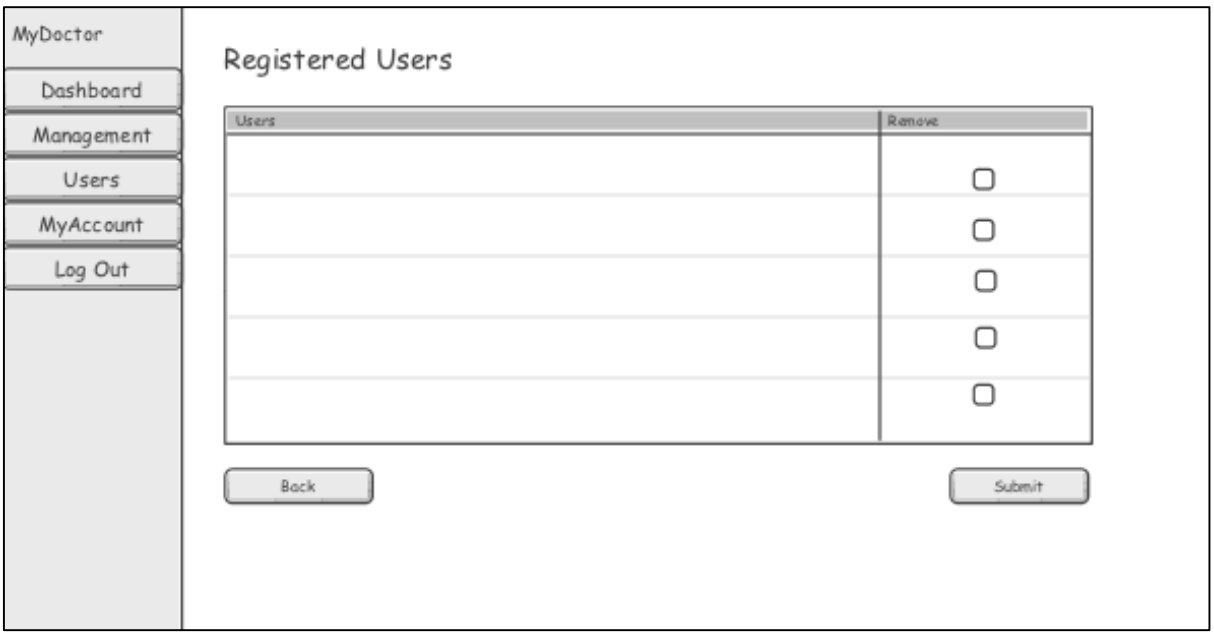
addusers.php – Input form for new user information

New users can be added to the system via the 'addusers.php' page, available to admin users only. On submission a new user record would be created and their user type defined.



registeredusers.php – Admin users can view registered system users

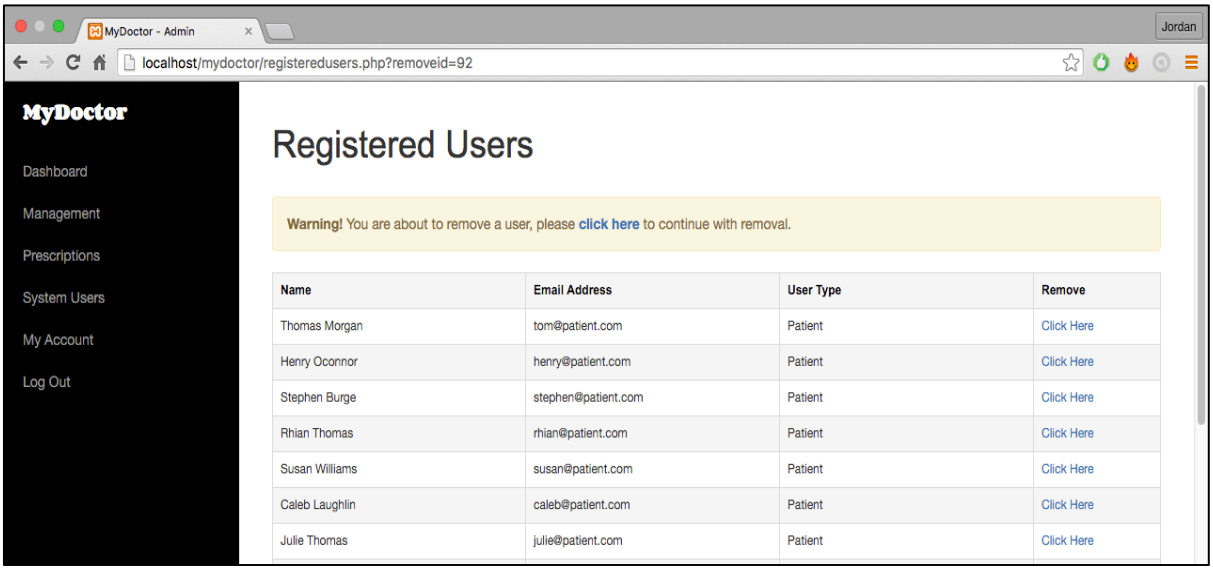
Admin users can view registered system users via the ‘registereduser.php’ page. This page also allows the user to remove users from the system. However, the implemented page does have slight modifications from the original wireframe design.



Registered users page wireframe design

It was proposed that the page would contain checkboxes for removal, however, on implementation it became apparent that due to the amount of users this page would display, scrolling to the bottom to submit the check box removal could prove time consuming and difficult to identify for new users.

To overcome this, checkboxes were changed to remove ‘Click Here’ links and the same **GET** method used for individual patient medical records was adopted. Once the link was selected for a specific user, warning message is displayed before continuing to remove.



registeredusers.phpremoveid=92 – Removing specific user (warning message)

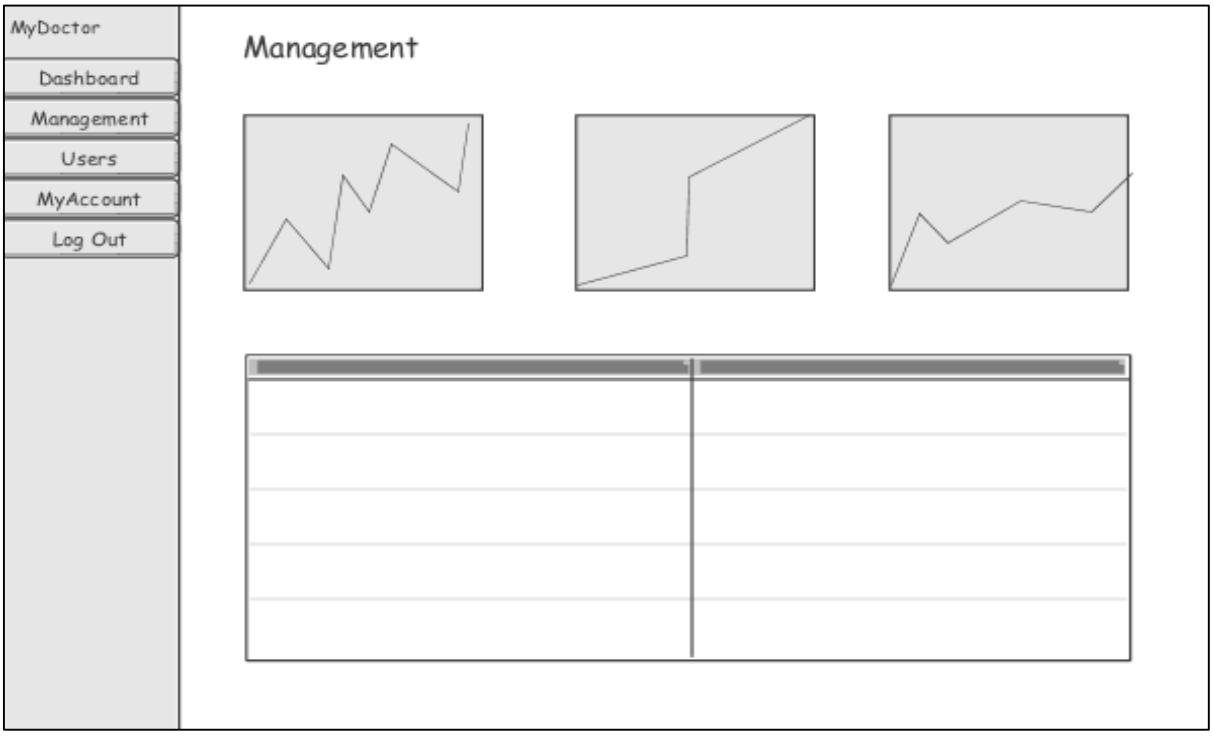
Once clicked to continue removal, user is redirected to **'remove.php?id='** page. The **GET** method then requests data to retrieve the user id of the user to be removed, from the previous page 'registeredusers.php. All of the users' records are then deleted from the database and the Admin user is returned to the registered users page.

```php
<?php
session_start();
include_once 'connect.php';

$userid = $_GET['id'];

mysql_query("DELETE FROM Users WHERE UserID = '$userid'");

header ('Location: registeredusers.php');

?>
```

remove.php – GET Method

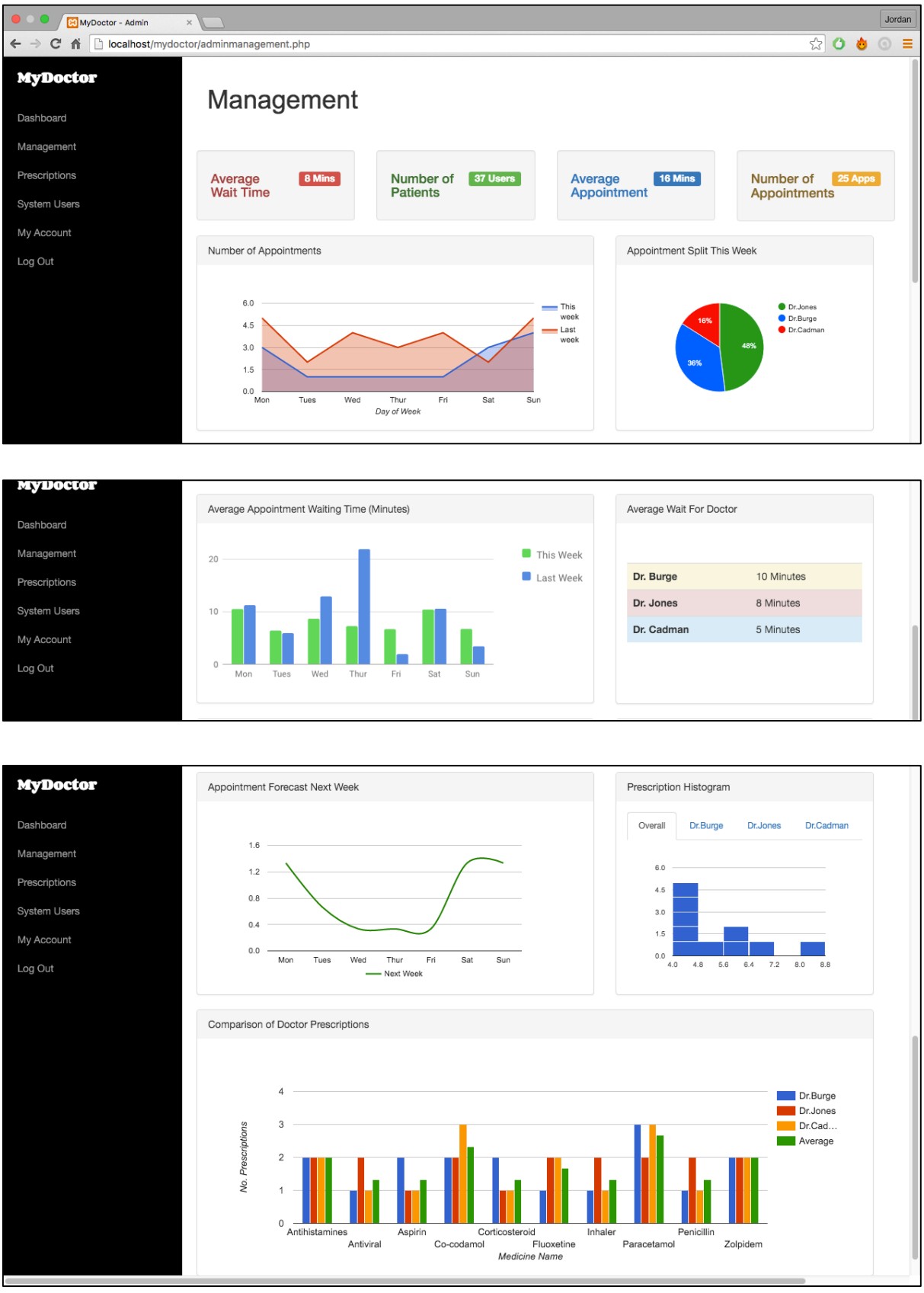
Management

When designing the data analysis page for Admin users, I wasn't aware of the exact methods I was going to implement and how they would be embedded in the page. This meant the interface of the page was constructed as it was developed.



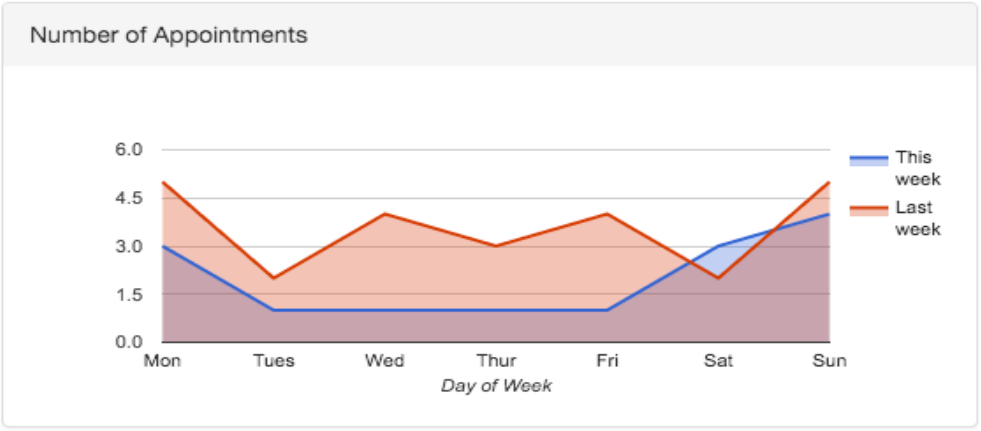
Management page – Wireframe design

Screen shots of the final user interface implemented can be seen below. The page contains a range of charts and displays used to visualise data for analysis.



adminmanagement.php – Data analysis and forecasting features

The implemented Management page contains various data analysis calculations, these were then visualised for admin purposes, using Google Charts API. Within this section of implementation, I will now identify some examples of these charts, there purpose and the calculations behind them.



adminmanagement.php – Number of Appointments

This chart displays the total number of doctor appointments across the current and previous weeks. To create the data first the dates for each of these days had to be defined.

```php
// Calculate this weeks dates
$week = date('W');
$date = strtotime("1 January 2016");
$day = date('w', $date);
$date += ((7*$week)+1-$day)*24*3600;
$monday = date('Y-m-d', $date);
$date += 6*24*3600;
$sunday = date('Y-m-d', $date);
$tue = strtotime($monday) + 86400;
$tuesday = date('Y-m-d', $tue);
$wed = strtotime($tuesday) +86400;
$wednesday = date('Y-m-d', $wed);
$thurs = strtotime($wednesday) +86400;
$thursday = date('Y-m-d', $thurs);
$fri = strtotime($thursday) +86400;
$friday = date('Y-m-d', $fri);
$sat = strtotime($friday) +86400;
$saturday = date('Y-m-d', $sat);
```

This weeks dates

```php
// Calculate last weeks dates
$lmon = strtotime($monday) - 604800;
$lmonday = date('Y-m-d', $lmon);
$ltue = strtotime($lmonday) + 86400;
$ltuesday = date('Y-m-d', $ltue);
$lwed = strtotime($ltuesday) + 86400;
$lwednesday = date('Y-m-d', $lwed);
$lthurs = strtotime($lwednesday) + 86400;
$lthursday = date('Y-m-d', $lthurs);
$lfri = strtotime($lthursday) + 86400;
$lfriday = date('Y-m-d', $lfri);
$lsat = strtotime($lfriday) + 86400;
$lsaturday = date('Y-m-d', $lsat);
$lsun = strtotime($lsaturday) + 86400;
$lsunday = date('Y-m-d', $lsun);
```

Last weeks dates

Once dates were identified the **'count()'** function was used to count how many appointments were within the database for these dates.

```php
$countmon = mysql_fetch_array(mysql_query("SELECT COUNT(*) FROM Appointments WHERE Date = '$monday'"));
$counttues = mysql_fetch_array(mysql_query("SELECT COUNT(*) FROM Appointments WHERE Date = '$tuesday'"));
$countwed = mysql_fetch_array(mysql_query("SELECT COUNT(*) FROM Appointments WHERE Date = '$wednesday'"));
$countthur = mysql_fetch_array(mysql_query("SELECT COUNT(*) FROM Appointments WHERE Date ='$thursday'"));
$countfri = mysql_fetch_array(mysql_query("SELECT COUNT(*) FROM Appointments WHERE Date ='$friday'"));
$countsat = mysql_fetch_array(mysql_query("SELECT COUNT(*) FROM Appointments WHERE Date ='$saturday'"));
$countsun = mysql_fetch_array(mysql_query("SELECT COUNT(*) FROM Appointments WHERE Date ='$sunday'"));
```

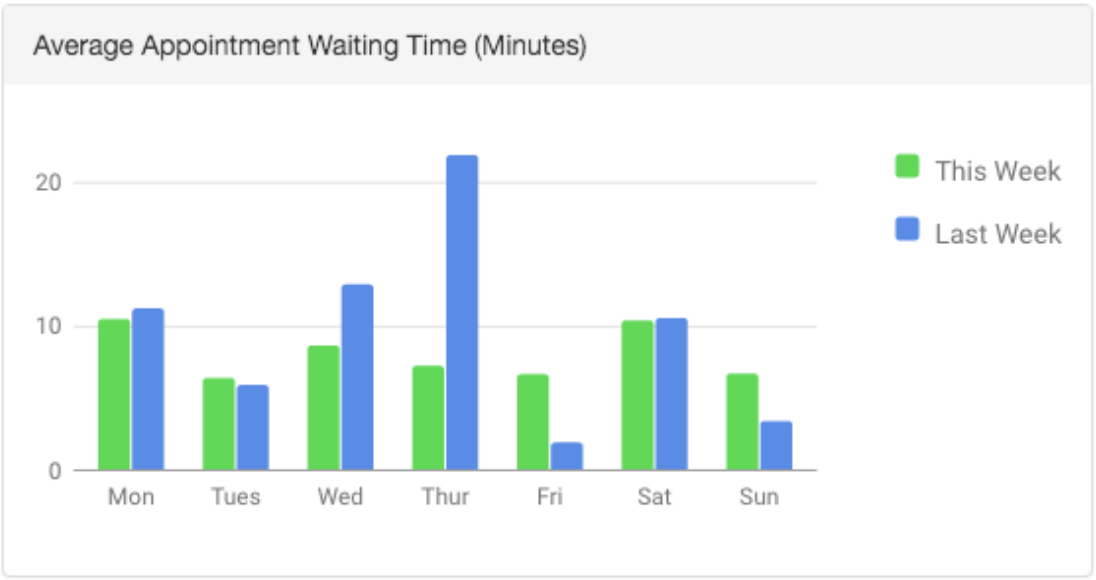
Appointment count

A Google Chart API was then used to display the results for each of these days within an Area chart.

```
<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
<script type="text/javascript">
  google.charts.setOnLoadCallback(drawChart);
  function drawChart() {
    var data = google.visualization.arrayToDataTable([
      ['Day', 'This week', 'Last week'],
      ['Mon',   <?echo "$lmonapp[0]"?>,   <?echo "$countmon[0]"?>],
      ['Tues',  <?echo "$ltuesapp[0]"?>,  <?echo "$counttues[0]"?>],
      ['Wed',   <?echo "$lwedapp[0]"?>,   <?echo "$countwed[0]"?>],
      ['Thur',  <?echo "$lthurapp[0]"?>,  <?echo "$countthur[0]"?>],
      ['Fri',   <?echo "$lfriapp[0]"?>,   <?echo "$countfri[0]"?>],
      ['Sat',   <?echo "$lsatapp[0]"?>,   <?echo "$countsat[0]"?>],
      ['Sun',   <?echo "$lsunapp[0]"?>,   <?echo "$countsun[0]"?>]
    ]);

    var options = {
      hAxis: {title: 'Day of Week',  titleTextStyle: {color: '#333'}},
      vAxis: {minValue: 0}
    };

    var chart = new google.visualization.AreaChart(document.getElementById('chartt_div'));
    chart.draw(data, options);
  }
</script>
```

adminmanagement.php – Number of Appointments chart

A column chart was used to compare the average appointment waiting time for the current week and the previous week.



adminmanagement.php – Average Appointment Waiting Time column chart

Now that the dates had previously been defined, the next step was to calculate the average appointment wait on each of these days. This was done by selecting all appointments from the Appointment table where the date was the the required day. While there were appointments with this date in the database the following code was executed.

```
// Monday
$mon = mysql_query("SELECT * FROM Appointments WHERE Date = '$monday'");
$avgtimem = 0;

while ($monrow = mysql_fetch_array($mon)) {
  $waittimem = strtotime($monrow['TimeIn']) - strtotime($monrow['Time']);
  $newwaittimem = $waittimem / 60;
  $avgtimem = $avgtimem + $newwaittimem;
}

$avgwaittimem = $avgtimem / $countmon[0];
```

adminmanagement.php – Average Appointment Wait Time calcuations

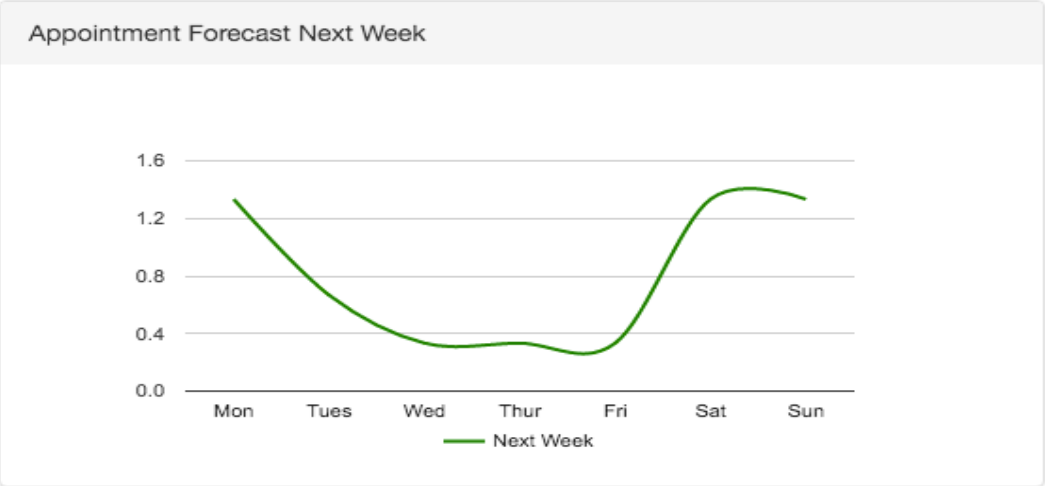
This code calculates the difference between the scheduled time of the appointment and the actual time the patient was seen. This is done by taking the scheduled time **(Time)** away from the actual time seen **(TimeIn)**. The result is then divided by 60 to convert into minutes and added to the total for all appointment waiting times within the loop.

The calculated average time **($avgtimem)** is then divided the count of appointments on that date to generate the overall average. Google Chart API was then used to display the results for each day within the chart.

```
    <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
   <script type="text/javascript">

                google.charts.load('current', {packages: ['corechart', 'bar', 'line']});
                google.charts.setOnLoadCallback(drawMaterial);

                function drawMaterial() {
                  var data = google.visualization.arrayToDataTable([
                    [' ', 'This Week', 'Last Week'],
                    ['Mon', <? echo "$avgwaittimem";?>, <? echo "$lavgwaittimem";?>],
                    ['Tues', <? echo "$avgwaittimet";?>, <? echo "$lavgwaittimet";?>],
                    ['Wed', <? echo "$avgwaittimew";?>, <? echo "$lavgwaittimew";?>],
                    ['Thur', <? echo "$avgwaittimeth";?>, <? echo "$lavgwaittimeth";?>],
                    ['Fri', <? echo "$avgwaittimef";?>, <? echo "$lavgwaittimef";?>],
                    ['Sat', <? echo "$avgwaittimes";?>, <? echo "$lavgwaittimes";?>],
                    ['Sun', <? echo "$avgwaittimesu";?>, <? echo "$lavgwaittimesu";?>]
                  ]);

var options = {
                    chart: {
                      title: ''
                    },
                    colors: ['#53D769', '#598CDF'],
                    hAxis: {
                      title: 'Minutes',
                      minValue: 0,
                    },
                    vAxis: {
                      title: 'Minutes'
                    },
chartArea: {  width: "50%" },
                    bars: 'vertical'
                  };
                  var material = new google.charts.Bar(document.getElementById('chart_div'));
                  material.draw(data, options);
                }
              </script>
```

adminmanagement.php – Average Appointment Wait column chart

Forecasting was identified as an essential requirement within the requirement specification earlier in the report. The chart implemented below provided a forecast of appointment numbers for the next week.



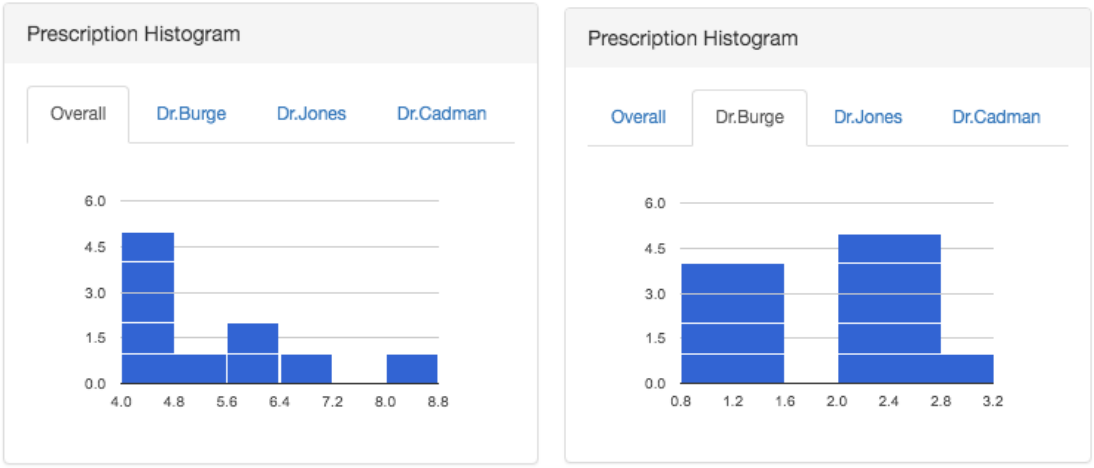
Appointment Forecast Chart

This forecast was calculated using the previously stated **Simple Moving Average** calculation. This method involved setting a moving average taken from a fixed subset size (last three weeks) and calculating the average for each day. Results are then plotted on the above chart for admin use.

```
// ************ FORECAST FOR NUMBER APPOINTMENTS NEXT WEEK  ********************


// Monday
$countforM = ($countmonfor[0] + $lmonapp[0] + $pcountmon[0]) / 3;

// Tuesday
$countforT = ($counttuesfor[0] + $ltuesapp[0] + $pcounttue[0]) / 3;

//Wednesday
$countforW = ($countwedfor[0] + $lwedapp[0] + $pcountwed[0]) / 3;

//Thursday
$countforTH = ($countthurfor[0] + $lthurapp[0] + $pcountthur[0]) / 3;

//Friday
$countforF = ($countfrifor[0] + $lfriapp[0] + $pcountfri[0]) / 3;

//Saturday
$countforS = ($countsatfor[0] + $lsatapp[0] + $pcountsat[0]) / 3;
```

adminmanagement.php – Simple Moving Average forecast calculations

Histograms were implemented to allow administration users to keep a track of prescriptions. Multiple tabs allow the user to view overall prescriptions and compare individual doctors, to ensure prescription privileges aren't being abused.



adminmanagement.php – Prescription Histograms

This was done by calculating the count of each medication prescribed by each doctor and then adding the results to Google chart API.

```
// ********************* PRESCRIPTION COUNT FOR DOCTOR 1 ************************

$medcount1 = mysql_fetch_array(mysql_query("SELECT COUNT(*) FROM Prescriptions WHERE MedicineID ='1' AND DoctorID ='1'"));

$medcount2 = mysql_fetch_array(mysql_query("SELECT COUNT(*) FROM Prescriptions WHERE MedicineID ='2' AND DoctorID ='1'"));

$medcount3 = mysql_fetch_array(mysql_query("SELECT COUNT(*) FROM Prescriptions WHERE MedicineID ='3' AND DoctorID ='1'"));

$medcount4 = mysql_fetch_array(mysql_query("SELECT COUNT(*) FROM Prescriptions WHERE MedicineID ='4' AND DoctorID ='1'"));
```

Calculations for prescription medicine count

```
<!-- Histogram prescriptions doctor 1 -->
      <script type="text/javascript">

      google.charts.setOnLoadCallback(drawChart);
      function drawChart() {
        var data = google.visualization.arrayToDataTable([
          ['Medicine', 'Prescribed'],
          ['Antihistamines', <? echo "$medcount7[0]";?>],
          ['Antiviral', <? echo "$medcount8[0]";?>],
          ['Aspirin', <? echo "$medcount1[0]";?>],
          ['Co-codamol', <? echo "$medcount3[0]";?>],
          ['Corticosteroid ', <? echo "$medcount5[0]";?>],
          ['Fluoxetine', <? echo "$medcount6[0]";?>],
          ['Inhaler', <? echo "$medcount4[0]";?>],
          ['Paracetamol', <? echo "$medcount2[0]";?>],
          ['Penicillin', <? echo "$medcount10[0]";?>],
          ['Zolpidem', <? echo "$medcount9[0]";?>]]);

        var options = {
          title: '',
          width: 340,
          height: 200,
          legend: { position: 'none' },
        };

        var chart = new google.visualization.Histogram(document.getElementById('ChartA'));
        chart.draw(data, options);
      }
    </script>
```

adminmanagement.php – Prescription Histogram

Conclusion

At the current stage of implementation, a web application has been developed allowing Patient, Doctor and Admin user types to access the online system. Depending on authorisation of user type, the user is then redirected to their required area of the system. The user can then undergo all required tasks earlier identified within the requirements specification.

5.0 Results & Evaluation

Testing

Testing is one of the most important stages of the project as I need to ensure that the system implemented works as intended. Before undergoing testing I had to first identify the most effective methods available to me. After carrying out some research, I decided to dynamically test the system by applying unit testing. This involved splitting the system and testing each component - Patient, Doctor and Admin sections. These components were to be examined using three individual methods.

Test cases

The first method of testing to be undertaken was test cases. Involving a set of conditions and variables predetermined by myself to evaluate if the system and each of its components worked to a satisfactory level. The process involved developing test cases to identify if these requirements are met. By doing so I could detect if any improvements were required to ensure that the project aim was met effectively.

Requirements were derived by examining my previously defined functionality requirements and involved testing each unit and feature of the system ranging from logging in to administration data analysis. All tests resulted in a pass and some examples can be seen below along with a test summary. Test cases for remaining tests carried out are located in appendix.

All users - Login to system

Test ID: 1	**Purpose:** User can login to the system		
Environment:	Web application		
Pre-conditions:	User has a registered user account		
Step Number:	**Predicted outcome**	**Actual outcome**	**Result**
1: Enter a valid email address into email address field	Email address is shown in the email address input field	The email address is shown in the email address input field	PASS
2: Enter a valid password into password field	The entered password is shown encrypted for security purposes	The password is shown encrypted in the password input field	PASS
3: Click “Login” button	The user is redirected to their corresponding dashboard	The user is redirected to their dashboard	PASS
Comments:	Screenshots in appendix (test case 1)		
Author: Jordan Wood		**Date:** 18/04/2016	

Patient – Arrange appointment

Test ID: 3.1	**Purpose:** User can arrange a new appointment with their doctor		
Environment:	Web application		
Pre-conditions:	User is logged into the system with a registered user patient account and has navigated to Appointment section		
Step Number:	**Predicted outcome**	**Actual outcome**	**Result**
1: Select "Arrange" button on appointments page	User is redirected to arrange appointments page and displayed input form and their corresponding doctor name	User redirected to arrange appointments page, displaying arrangement input form and patients doctor	PASS
2: Enter a date manually or via calendar drop down	Chosen date is shown in the date input field	Chosen date is shown in the date input field	PASS
3: Select "View Times" button	Time and reason input fields are displayed	Time and reason input fields are accessible to user	PASS
4: Select an available time from drop down menu	Chosen time is shown in time input field	Chosen time is shown in time input field	PASS
5: Enter text to Reason input field	Reason text is shown in input field	Reason text is shown in input field	PASS
6: Click "Submit" button	Appointment arrange details are submitted to the system database and success message shows "Success! Your appointment request details have been processed"	Inputted appointment details were stored in the system database and and success message shows "Success! Your appointment request details have been processed"	PASS
Comments:	Screenshots in appendix (test case 3.1)		
Author: Jordan Wood		**Date:** 18/04/2016	

Doctor - Prescribe medication

Test ID: 4.6	**Purpose:** User can prescribe patients medication		
Environment:	Web application		
Pre-conditions:	User is logged in with a registered doctor user account and has navigated to the prescriptions section		
Step Number:	**Predicted outcome**	**Actual outcome**	**Result**
1: Select "Prescribe" button	User is redirected to patient Prescribe Medication page and displayed input form and medication search form	User is redirected to Patient Medication page and is displayed input form for entering prescription details and a medication search form	PASS

2: Enter patient id into Patient ID input field	Entered patient id number is displayed Patient ID input field	Entered patient id number is displayed Patient ID input field	PASS
3: Enter medicine using Medicine drop down menu	Drop down menu used to enter required medicine and selection is displayed in Medicine input field	Drop down menu used to enter required medicine and selection is displayed in Medicine input field	PASS
4: Enter status using Status drop down menu	Drop down menu used to enter required status and selection is displayed in Status input field	Drop down menu used to enter required status and selection is displayed in Status input field	PASS
5: Select "Prescribe" button	User is shown success message on submission "Success! Prescription has been recorded"	User is shown success message on submission "Success! Prescription has been recorded"	PASS
Comments:	Screenshots in appendix (test case 4.6)		
Author: Jordan Wood		**Date:** 18/04/2016	

Admin – Add user to system

Test ID: 5.3	**Purpose:** User can add users to the system		
Environment:	Web application		
Pre-conditions:	User is logged in with a registered admin user account and has navigated to the system users page		
Step Number:	**Predicted outcome**	**Actual outcome**	**Result**
1: Locate the Add New User section and select "Add" button	User is redirected to Add User Account page and displayed an input form	User is redirected to correct page and displayed an input form requiring new user details	PASS
2: Enter text into First Name input field	Text is displayed in First Name input field	Text is displayed in First Name input field	PASS
3: Enter text into Last Name input field	Text is displayed in Last Name input field	Text is displayed in Last Name input field	PASS
4: Select gender using drop down menu to choose from "M" or "F"	Selection is displayed in Gender input field	Selection is displayed in Gender input field	PASS
5: Enter DOB into DOB input field	Entered Date of Birth is displayed in DOB input field	Entered Date of Birth is displayed in DOB input field	PASS
6: Enter text into Address input field	Entered text is displayed in Address input field	Entered text is displayed in Address input field	PASS

7: Enter postcode into Postcode input field	Entered post code is displayed in Post Code input field	Entered post code is displayed in Post Code input field	PASS
8: Enter telephone number into Telephone input field	Entered telephone number is displayed in Telephone input field	Entered telephone number is displayed in Telephone input field	PASS
9: Select user type using drop down menu to choose from “Patient”, “Doctor” or “Admin”	Selection is displayed in User Type input field	Selection is displayed in User Type input field	PASS
10: Enter email address into Email Address input field	Entered email address is displayed in Email Address input field	Entered email address is displayed in Email Address input field	PASS
11: Enter password into Password input box	The entered password is shown encrypted for security purposes	The password is shown encrypted in the password input field	PASS
12: Select “Add New User” button	A new user account record is added to the database and a success message shows “Success! A new account has been added!”	Database record was created and success message was shown	PASS
Comments:	Screenshots in appendix (test case 5.3)		
Author: Jordan Wood		**Date:** 18/04/2016	

Test Case Summary		
Test ID	**Purpose**	**Result**
1	User can login to the system	PASS
1.1	Check unauthorised user details are denied access	PASS
1.2	User can log their account out of the system	PASS
2	User can modify their user account details	PASS
2.1	User can’t submit blank email address and password data when updating their account	PASS
2.2	User can modify their personal details	PASS
2.3	User can’t submit blank input fields when updating their personal details	PASS
3	Patient can view their appointment history	PASS
3.1	Patient can arrange a new appointment with their doctor	PASS
3.2	Patient cannot select a date earlier than todays	PASS
3.3	Patient cannot select taken appointment times	PASS
3.4	Patient can view their problem history	PASS
3.5	Patient can view their medication history	PASS
3.6	Patient can view their test result history	PASS

3.7	Patient can view their allergy history	PASS
3.8	Patient can view their prescription history	PASS
3.9	Patient can request a repeat prescription of medication	PASS
4	Doctor can view their appointments	PASS
4.1	Doctor can update appointments	PASS
4.2	Doctor can view their patients	PASS
4.3	Doctor can view individual patient medical record	PASS
4.4	Doctor cannot submit appointment update without completing all fields	PASS
4.5	Doctor can view their patient prescription history	PASS
4.6	Doctor can prescribe patients medication	PASS
4.7	Doctor cannot submit incomplete prescription	PASS
4.8	Doctor can research medicines	PASS
5	Admin can view all prescriptions	PASS
5.1	Admin can view all registered system users	PASS
5.2	Admin can remove users from the system	PASS
5.3	Admin can add users to the system	PASS
5.4	Admin can access Management data	PASS

Evaluation

Having conducted my test case evaluation, I could now see how well the system was able to meet its requirements. Results show that all features of the developed system behave correctly, satisfying requirements and passing each test successfully.

Usability Testing

Usability testing is a technique focusing on user interaction. Testing how various individual users undergo system tasks, helping to identify flaws and possible problems users may experience, that myself the developer may not be able to. It helped provide direct input as to how real users use the system, in contrast to if I were to try replicate their mind set. Where I may unconsciously make decisions from my experience of developing the system, a user will have an untainted view of how the system works having no previous experience with it, therefore their viewpoint can help identify aspects I couldn't.

As the system involves three user types, usability testing was carried out from participant's perspective acting as Patient, Doctor and Admin members of a local doctor's surgery. The method of usability testing I adopted was think-aloud testing, this involved asking the test participant to carry out various system tasks whilst continuously thinking out loud. That is simply by verbalising whatever they were thinking, doing or feeling at each moment as they move through the user interface. Helping to determine the user's expectations and identify what aspects they found were confusing or felt could be improved. These tasks were also derived by examining my predefined functional and non-functional requirements of the system, however were focused entirely on usability of the system user interface.

17 think-aloud tests were conducted in total with all three test participants capable of completing the tasks set out with relative ease. A selection of these tests can been seen below, with the rest located in my appendix.

Test ID 1 – Logging into the system

Preconditions: User is on the homepage of the system			
Step	**Think Aloud**	**Problem Areas**	**Good Usability**
1	I have located the login details input fields, I am now entering my email address and password details.	No problems found.	The login inputs are recognisable and easy to locate.
2	I have selected the login button and been redirected to a new page welcoming me.	No problems found.	Login process was easy and I didn't experience any problems.

Test ID 2.1 – Arrange an appointment

Preconditions: User has logged into the system with a patient user account and navigated to the appointments page			
Step	**Think Aloud**	**Problem Areas**	**Good Usability**
1	I have located the arrange appointments section and selected arrange button.	No problems found.	Informative paragraph about the section.
2	I have selected the arrange button and been taken an arrange appointments page.	No problems found.	
3	I have chosen a date and selected view times button. A time input and reason box have appeared.	No problems found.	Drop down menu for available appointment times is good.
4	I have entered my appointment details and selected submit. I was notified of my successful submission of details.	No problems found.	Success message is informative.

Test ID 3 – View all patients & individual patient medical record (Doctor)

Preconditions: User is logged into the system with a doctor account.			
Step	**Think Aloud**	**Problem Areas**	**Good Usability**
1	I have located a patients button in the menu bar. I am selecting it. I have been taken to a page of my patients.	User found no problems.	No difficulty navigating to patients section.

2	I can currently see all of my patients displayed in a table. I can see a column titled medical records that contains a view link, I assume this is for each patient. I must first locate the patient I require.	I believe that in the future the volume of patients I may have would could lead to difficulties when locating individuals in this format. I would suggest including a search bar to search for patients.	The layout is clear to understand once my patient has been located.
3	I have found the patient I require and am now pressing view button. I have been taken to the medical record page for the resulting patient.	User found no problems.	No problems experienced when navigating and all information seems clear.

Test ID 3.4 – View prescriptions (Doctor)

Preconditions: User is logged into the system with a doctor account.			
Step	**Think Aloud**	**Problem Areas**	**Good Usability**
1	I am my patient prescriptions. To do so, I clicked on the prescription tab on the toolbar. I was correctly redirected.	No issues.	Very simple and quick.
2	I am clicking on the view all button for prescriptions	No problems, however to improve the functionality of finding prescriptions I would recommend adding filters on date, patient id etc. to easily locate my prescription.	Good usability – for improvements see box to the left.
3	The prescription has been correctly applied to the customer.	No issues detected.	Worked effectively.

Test ID 4 – View registered users (Admin)

Preconditions: User is logged into the system with a admin account.			
Step	**Think Aloud**	**Problem Areas**	**Good Usability**
1	System users is in the side bar, I am clicking that. I am now on users page and can see some registered users	None	No problem finding page

2	Clicking view all under registered users. I can now see all user registered users	There is a big list of users. If I wanted to find a specific one, it may be difficult. I would suggest adding a search or sort option	

Test ID 4.2 – Add user (Admin)

Preconditions: User is logged into the system with a admin account.			
Step	**Think Aloud**	**Problem Areas**	**Good Usability**
1	I'm clicking system users. I have been taken to users page and can see add new user section	None	Easy to find
2	I am now on add new user page and am putting the new users details in to the form	None	Simple
3	Now that I have filled the form I am going to click add new user button. A message has appeared telling me a new user account has been added	None	Message is helpful

Evaluation

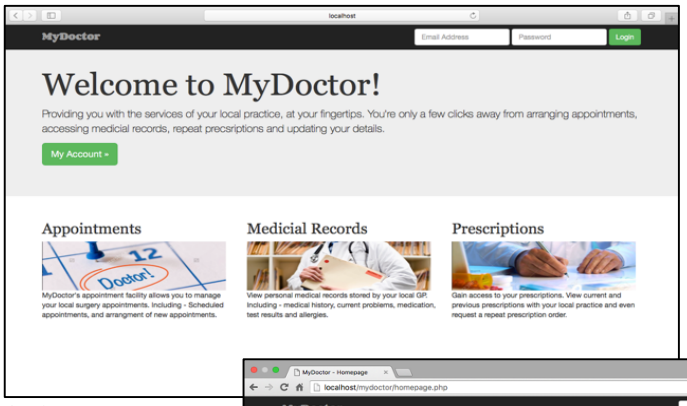
Before testing I like many other developers believed that I had taken the right steps to maximise system usability, however, I could now gain an insight to features I hadn't previously considered.

Looking over the test results I could see from participant feedback that future work on the system could be used to carry out improvements on the systems usability. For example, I noticed a trend in feedback, both doctor and admin users identified that once implemented into a real life situation the volume of patients and system users would be much larger than it currently is, leading to difficulties locating individuals in the current format. These users then suggested that they would like to see a search bar used to identify individuals or some method of sorting data, which is something I hadn't taken into consideration before and is now something I believe to be a crucial part of future development.

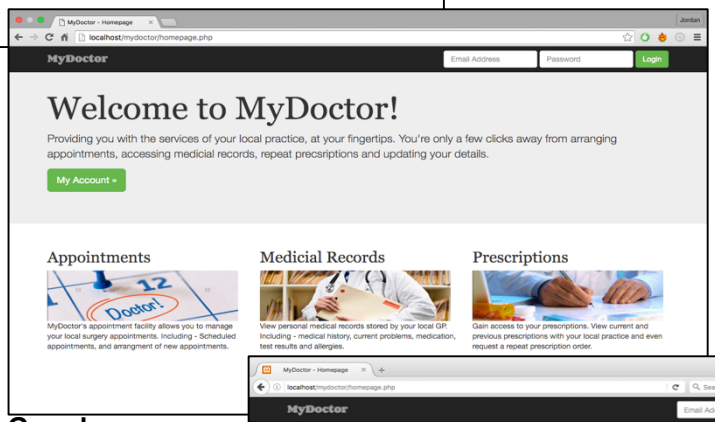
Cross browser

Cross browser testing is the process of testing web applications across multiple browsers. It is a highly important form of testing for this project, as the system can only be accessed via web browser. Due to the range of browser environments available to users nowadays it is fundamental that system adaption and performance across these is evaluated. Test results would allow me to determine if the system required manipulation to ensure its adaptability.

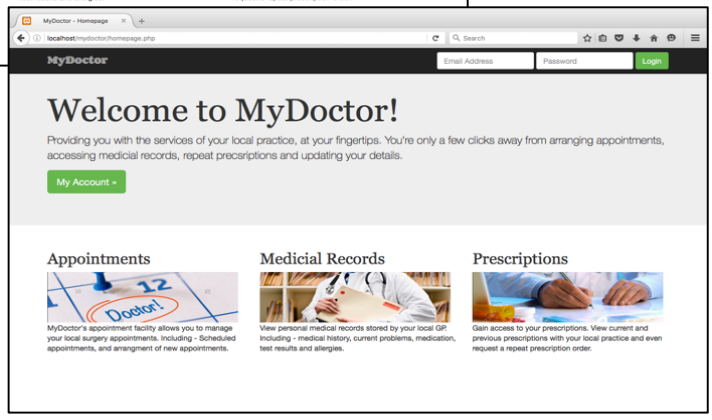
Testing was undertaken in three of the most popular web bowser environments – Google Chrome, Safari and Mozilla Firefox. Tests involved viewing each of the systems features within all three browsers to evaluate how it adapted.



Safari



Google Chrome

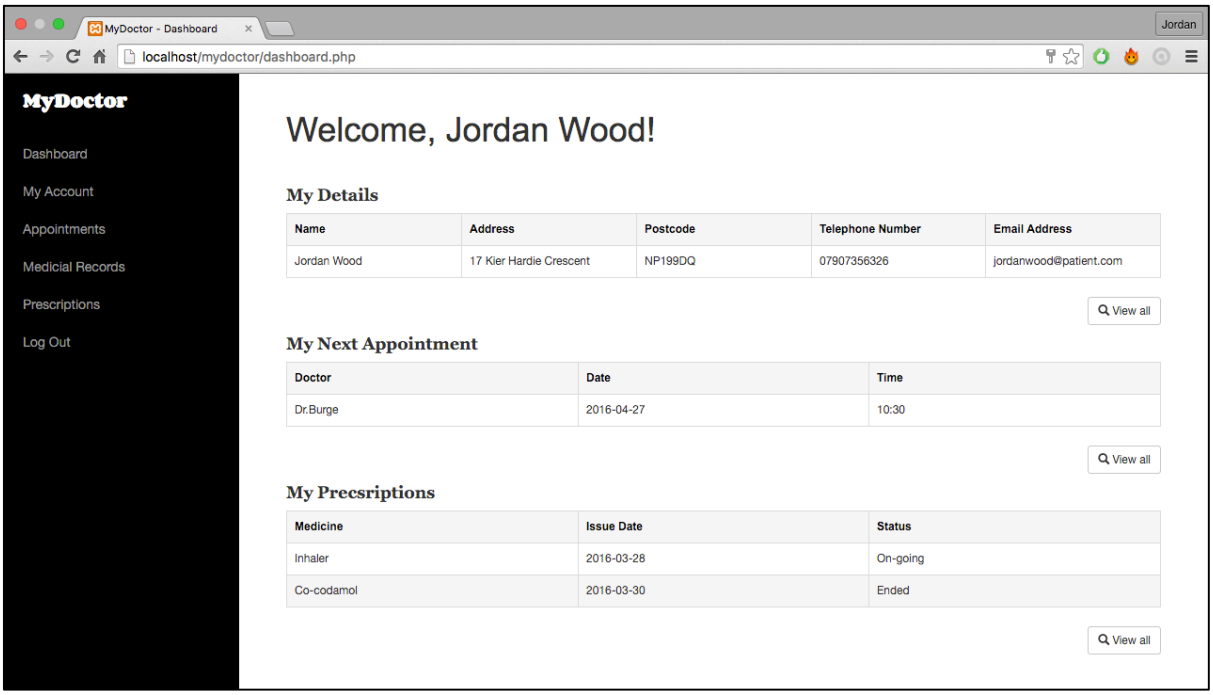


Mozilla

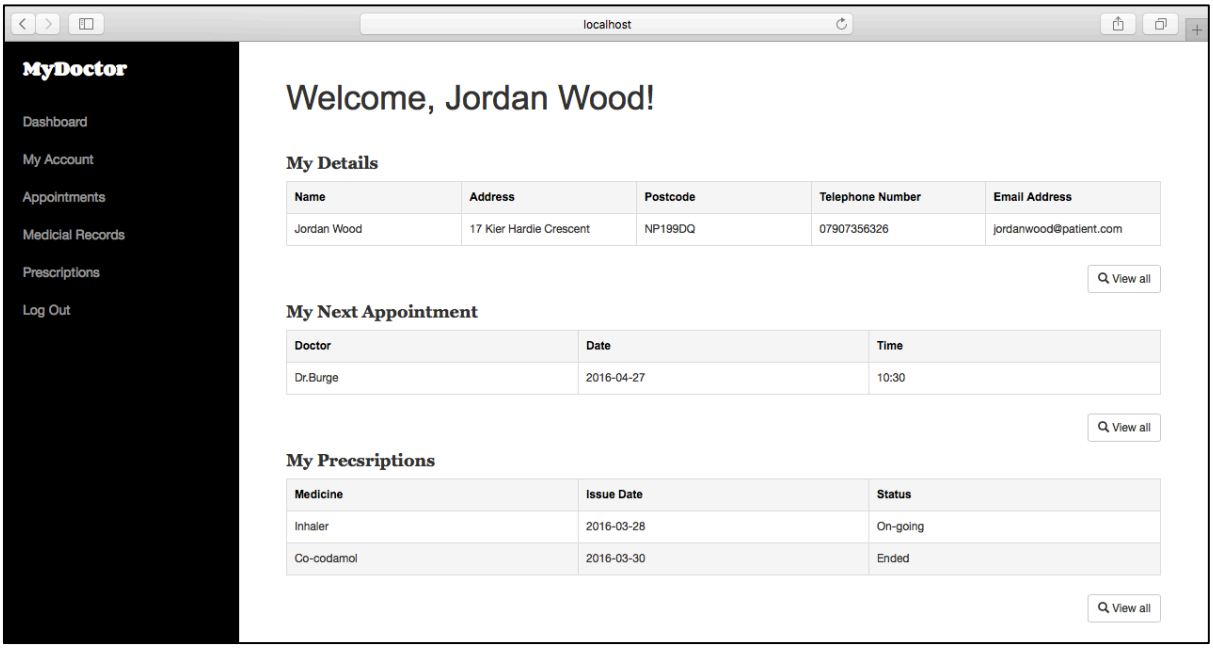
Due to time constraints I was unable to document the whole process, however, full testing was carried out by myself for each page, feature and function within the system. Results showed that the system implemented adapted to each browser

extremely well with only minimal size differences as expected, some examples can be seen below and within appendix.

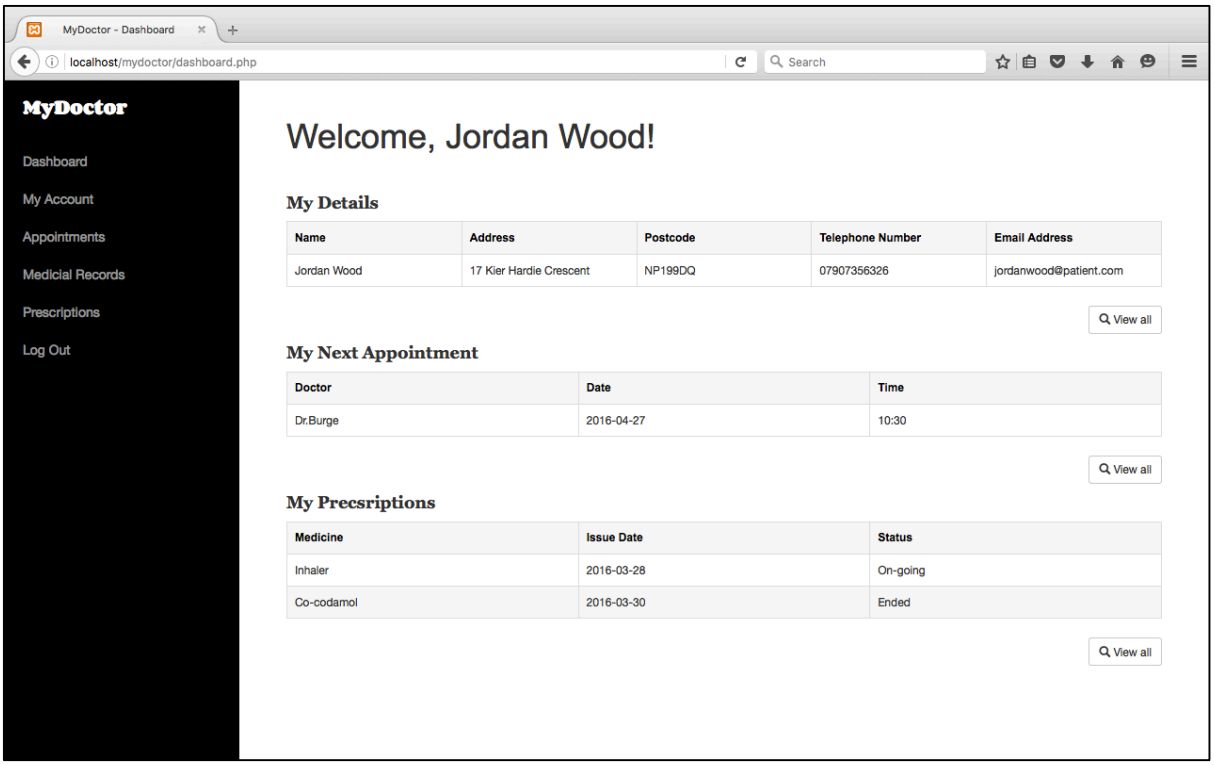
Patient dashboard displayed on user login provides expected data and layout across all three browsers.



Google Chrome

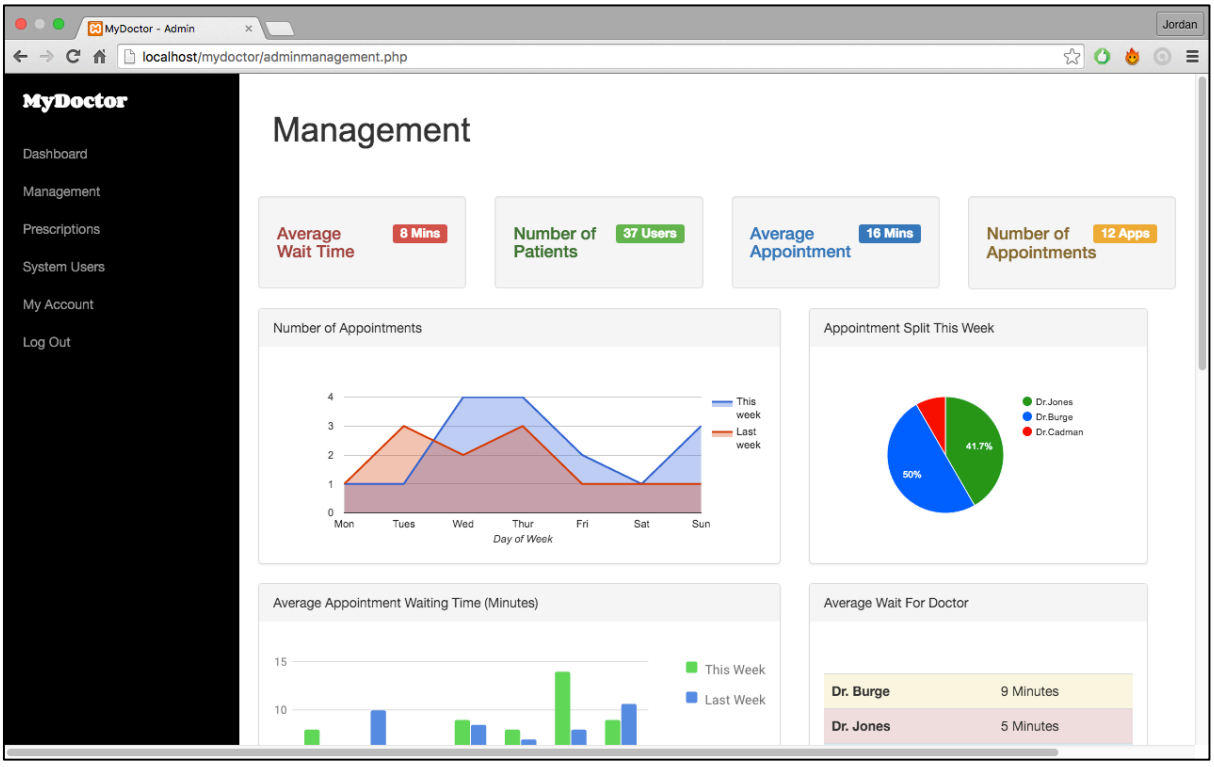


Safari

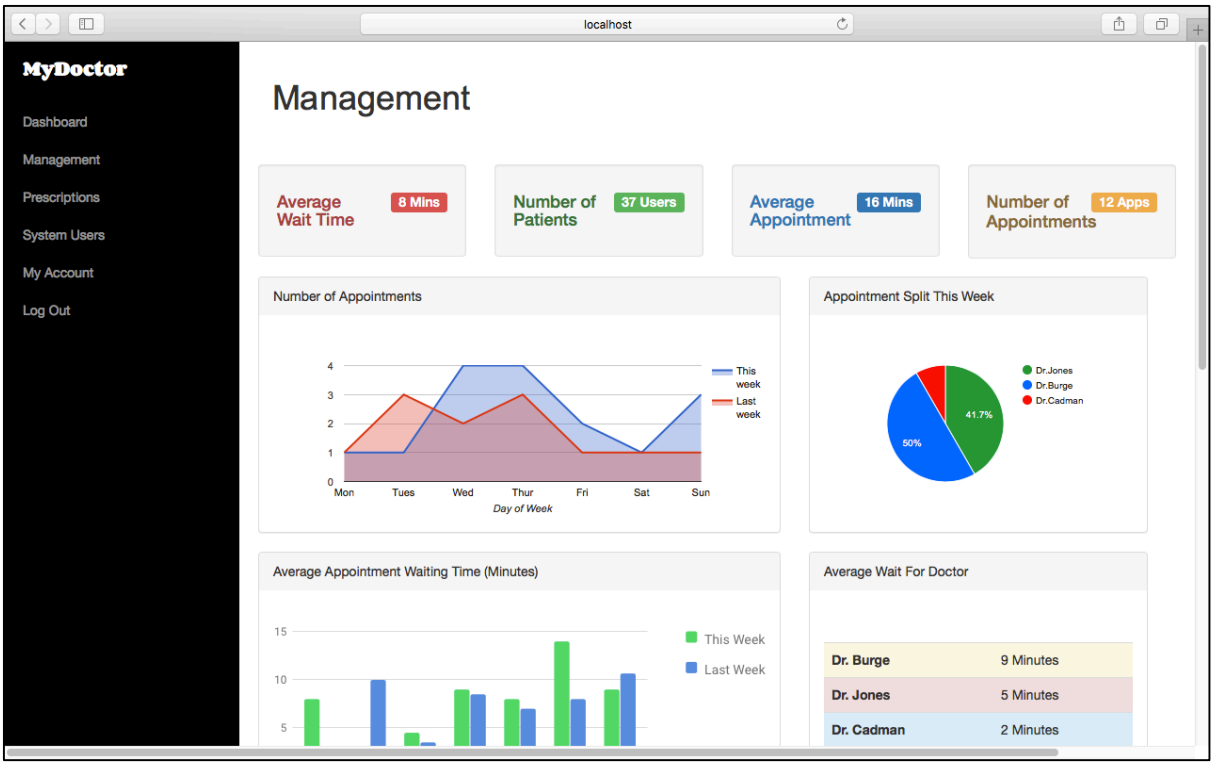


Mozilla Firefox

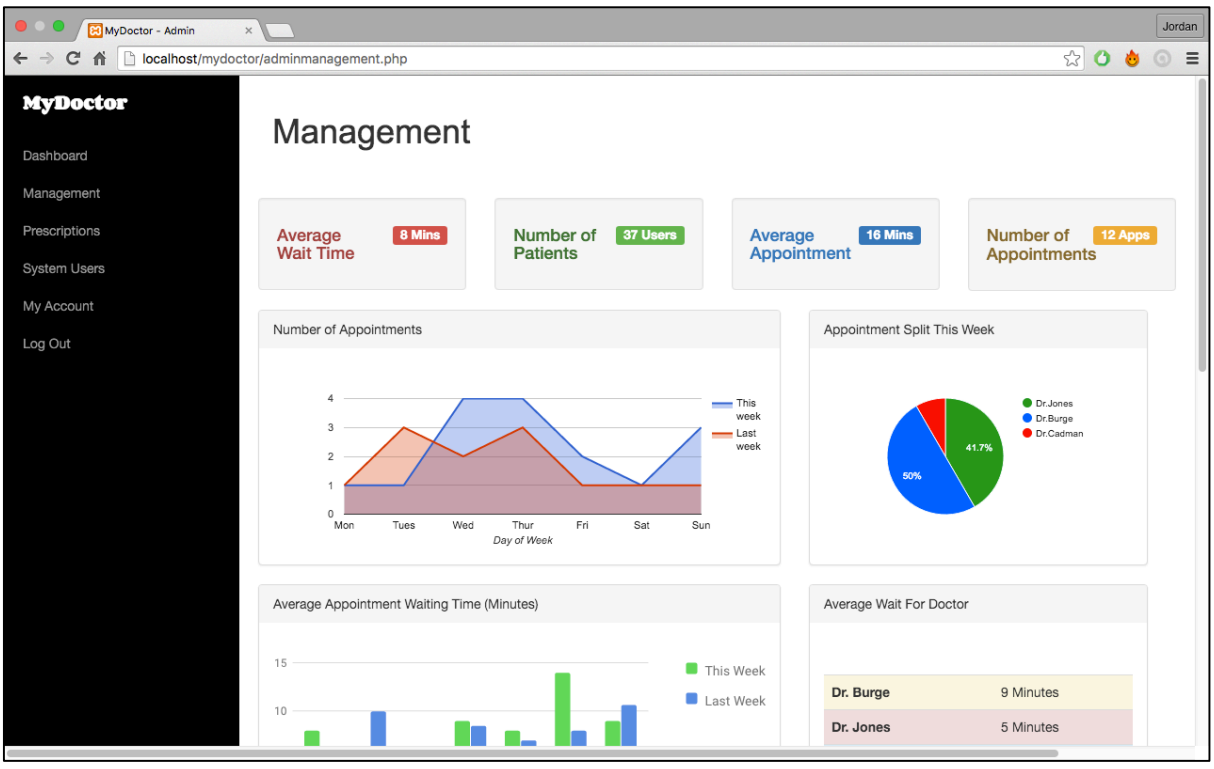
Data analysis features display data and layout as expected across all browsers.



Google Chrome



Safari



Mozilla Firefox

Usability Analysis

After testing I decided the next step I needed to undergo, to evaluate the system, was some form of usability analysis. I decided that to do this I was going to use the usability measurement tool SUS or System Usability Scale (13).

SUS is a reliable and quick tool for measuring usability, consisting of a ten item questionnaire with five response catergorises, ranging from Strongly Agree to Strongly Disagree. Its benefits include:

- Very easy scale to administer to participants
- Can be used on small sample sizes with reliable results
- Is valid – it can effectively differentiate between usable and unusable systems

To use SUS, I had to provide test participants with this set of ten questions in which they would respond to after using the system. As time was limited for the testing stage, I decided to use the same participants involved in the previously mentioned Think-Aloud testing. After these participants had finished completed their Think-Aloud tasks I asked them a series of questions:

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

Response Key
1 - Strongly disagree
2 - Disagree
3 - Neither Disagree or Agree
4 - Agree
5 - Strongly Agree

User 1							
No.	**Question**		**1**	**2**	**3**	**4**	**5**
1	I think that I would like to use this system frequently					x	
2	I found the system unnecessarily complex			x			
3	I thought the system was easy to use				x		
4	I think that I would need the support of a technical person to be able to use this system		x				

5	I found the various functions in this system were well integrated					x	
6	I thought there was too much inconsistency in this system		x				
7	I would imagine that most people would learn to use this system very quickly					x	
8	I found the system very cumbersome to use		x				
9	I felt very confident using the system						x
10	I needed to learn a lot of things before I could get going with this system		x				

User 2							
No.	**Question**		**1**	**2**	**3**	**4**	**5**
1	I think that I would like to use this system frequently						X
2	I found the system unnecessarily complex			x			
3	I thought the system was easy to use					x	
4	I think that I would need the support of a technical person to be able to use this system		x				
5	I found the various functions in this system were well integrated						x
6	I thought there was too much inconsistency in this system		x				
7	I would imagine that most people would learn to use this system very quickly						x
8	I found the system very cumbersome to use		x				
9	I felt very confident using the system						x
10	I needed to learn a lot of things before I could get going with this system		x				

User 3							
No.	**Question**		**1**	**2**	**3**	**4**	**5**
1	I think that I would like to use this system frequently						x
2	I found the system unnecessarily complex		x				
3	I thought the system was easy to use						x
4	I think that I would need the support of a technical person to be able to use this system		x				
5	I found the various functions in this system were well integrated					x	
6	I thought there was too much inconsistency in this system		x				
7	I would imagine that most people would learn to use this system very quickly						x
8	I found the system very cumbersome to use		x				
9	I felt very confident using the system						x
10	I needed to learn a lot of things before I could get going with this system		x				

To interpret the participant scores, each question must be converted to a number then:

- For each of the odd numbered questions, subtract 1 from the score
- For each of the even numbered questions, subtract their value from 5
- These new values must then be added together and then multiplied by 2.5 to convert the original scores into scores of 0-100

User 1 response calculations are:

Q1: 5 - 1
Q2: 5 - 2
Q3: 3 - 1
Q4: 5 - 1
Q5: 4 - 1
Q6: 5 - 1
Q7: 4 - 1
Q8: 5 - 1
Q9: 5 - 1
Q10: 5 - 1

35 x 2.5 = **87.5/100**

User 2 response calculations are:

Q1: 5 - 1
Q2: 5 - 2
Q3: 4 - 1
Q4: 5 - 1
Q5: 5 - 1
Q6: 5 - 1
Q7: 5 - 1
Q8: 5 - 1
Q9: 5 - 1
Q10: 5 - 1

38 x 2.5 = **95/100**

User 3 response calculations are:

Q1: 5 - 1
Q2: 5 - 1
Q3: 5 - 1
Q4: 5 - 1

Q5: 4 - 1
Q6: 5 - 1
Q7: 5 - 1
Q8: 5 - 1
Q9: 5 - 1
Q10: 5 - 1

39 x 2.5 = **97.5/100**

The average System Usability Scale score is 68, if my score was under this then it would indicate that there were some serious problems with the systems usability that needed addressing. If 80.3 or higher it would be in the top level for usability and shows that participants rated the system as highly usable.

Results show that average response score for my system was 93.33, this indicates it was ranked highly for usability and no problems were identified. Although these results were positive, due to the limited time that could be spent on SUS they may not be 100% accurate.

Future SUS analysis should involve a much larger scale of participants, including those with a mixed variety of experience and skills, in order to gather a more accurate result. For example, due to the time of system older users should also be included to gather an overall sense of usability.

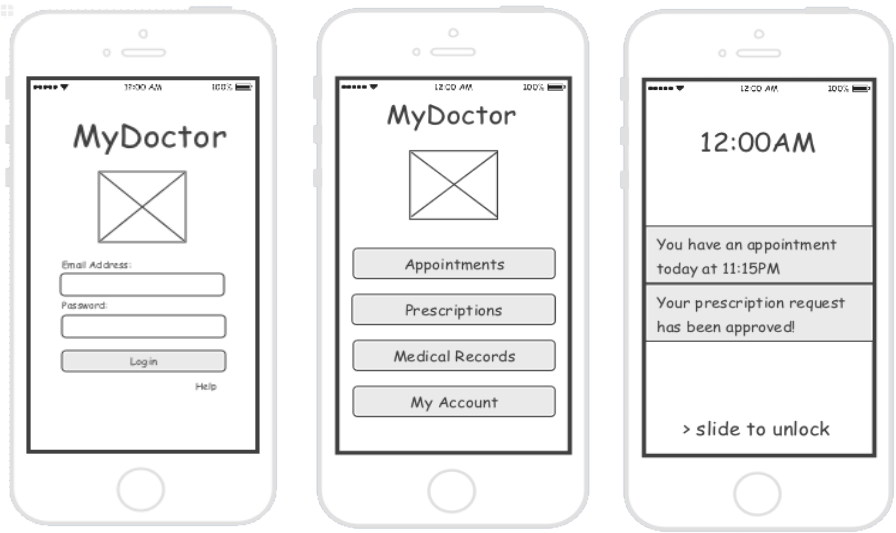
6.0 Future work

Due to the time constraints involved with the project I was unable to implement all of the potential features and functions of the system I desired. However, I believe these can be pursued in the future to further the system. Below I will identify and expand on each of these opportunities.

Mobile Application

One of the main objectives of the project was to increase the accessibility of information shared between patient and doctor, although the introduction of a web application is an effective solution at improving this situation, I believe that future work could be carried out to help introduce an Android and iOS device application.

The number of smartphone users within the UK continues to raise, with last year seeing smartphones overtake laptops as UK users' number one device (14). With two thirds of people now owning a smartphone device the introduction of an application would help to increase the accessibility of the system to its intended audience. The application would include all present features of the web application, however, would need to be redesigned to increase smartphone device usability. I have created some wireframes to indicate what the intended applications interface could appear resemble.

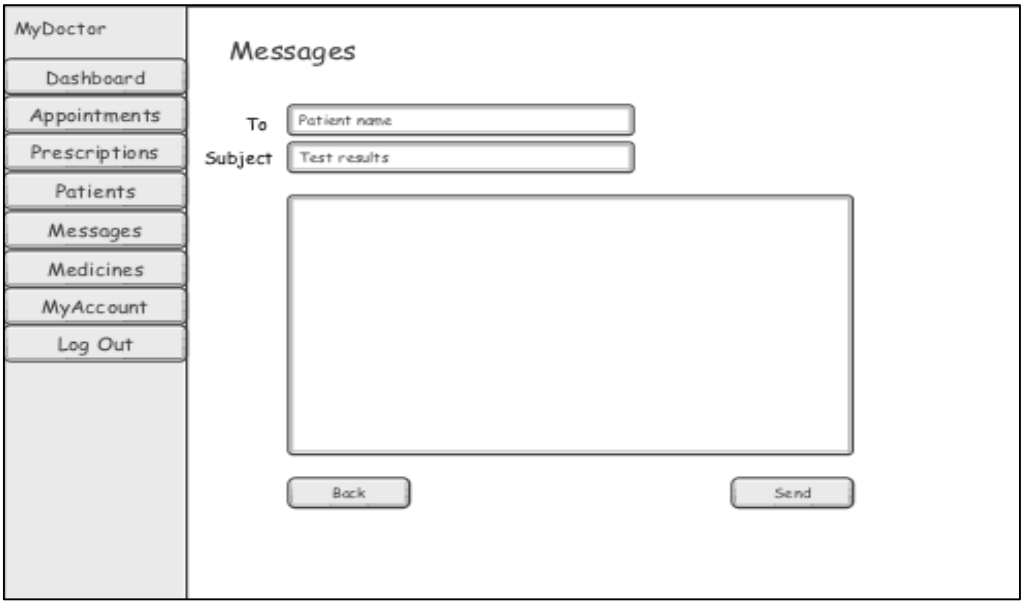


Login screen Dashboard Push notifications

The application would take advantage of smartphone device features, allowing the system to utilise push notifications to alert users of updates to medical records, prescription and appointments etc.

Messaging

The solution currently acts as a platform for sharing information between those interacting with a doctor's surgery, however, users aren't provided with a method of interacting with each other directly. Future work should be carried out to introduce a messaging facility similar to Email, that is made available through the system to allow users such as patient and doctor to communicate effectively.

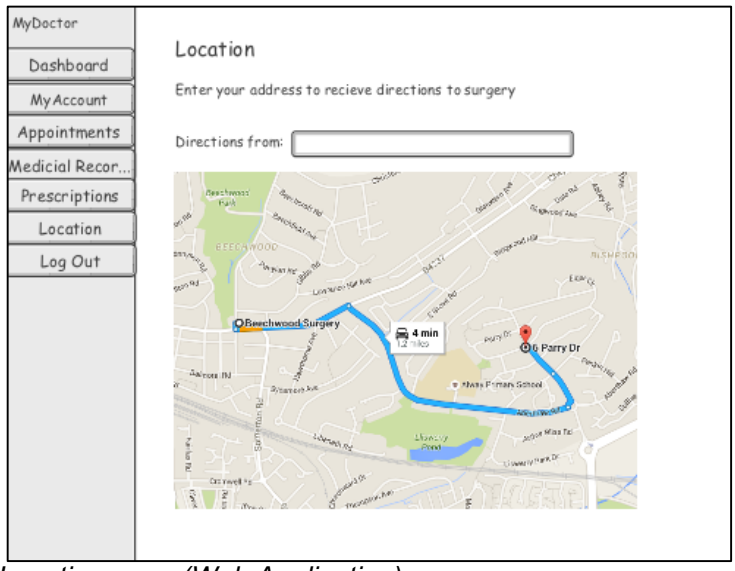


Messaging page

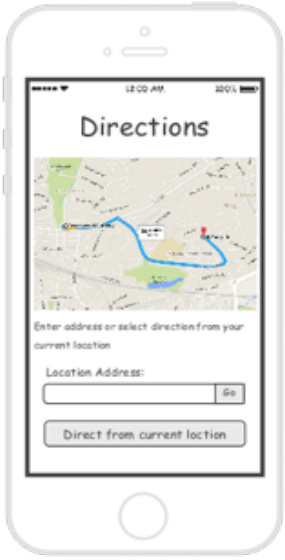
Location

Both the web and future smart device applications could improve their usefulness to the target audience by providing location based services. The implementation of Google Maps API could be used to generate a way point to the doctors' surgery or corresponding chemist etc, helping to provide their location and even directions.

A smartphone application could utilise the devices Global Positioning System functionality to provide directions from the user's current position to the location of the surgery or chemist helping increase its accessibility.



Location page (Web Application)

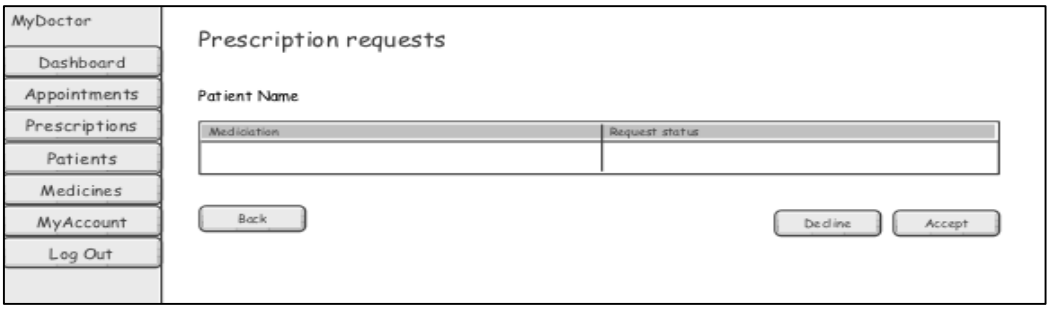


Directions page (Mobile Application)

Prescription authorisation

At the current stage of the system implementation, prescription requests by patients are stored by updating prescription records within the database. This means there isn't a method for doctors to view these requests and determine to accept and decline them. This currently wasn't a essential requirement in achieving my projects aim, however, when deployed into doctor surgeries this would change.

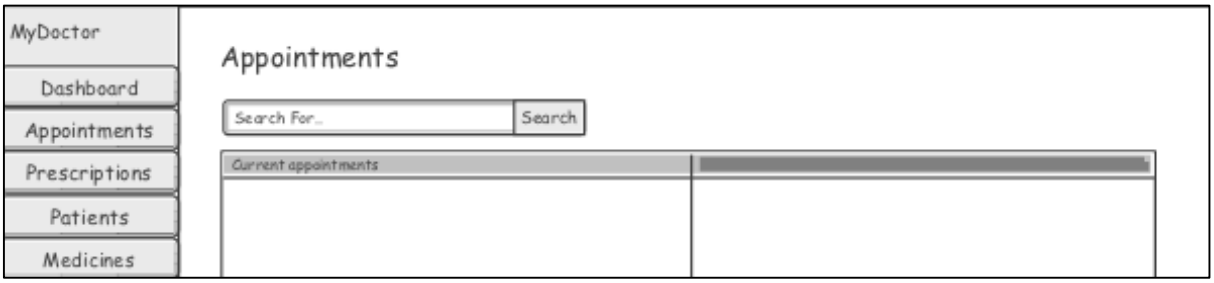
Future development should also focus on the implementation of a doctors area to allow doctor users to view prescription requests and either accept or decline them. An example of this page's layout can be seen below.



Request authorisation page

Sorting of data

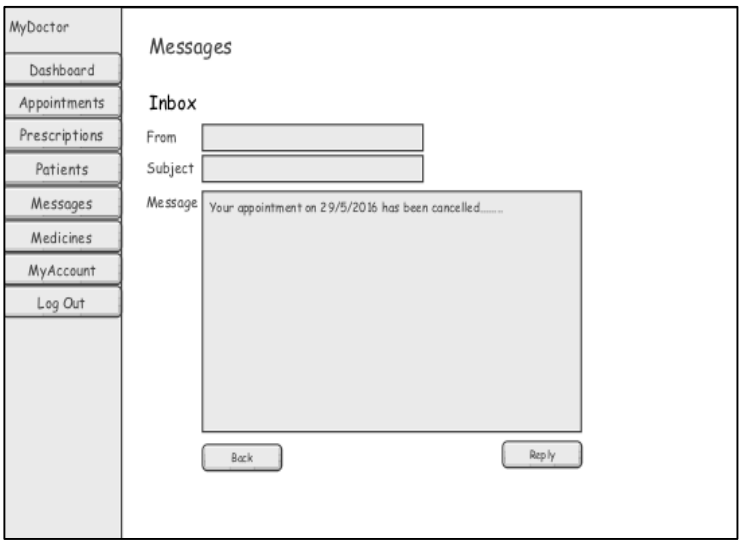
It became apparent when reviewing think-aloud test results that when the system is introduced into real life situations, the volume of patients, prescriptions and appointment data etc will be of a much greater scale than that of current. After analysing user feedback, a method of searching these large collections of data should be introduced, helping users to filter data.



Appointment cancelation

The current system allows patient and doctor users to view their scheduled appointments, however, it doesn't provide any means for cancelation of those appointments. Rendering the system useless in a situation when a cancelation was required, meaning that the user would have to revert to traditional methods such as a telephone call.

To overcome this problem future developments of the system should include a form of cancelation within the appointment sections. Working with the messaging and smart device application proposed, users affected by a cancelation would be informed via automated email and/or push notifications on their device. These methods would then provide the user with the next steps required for example rearrangement of appointment.



Appointment cancelation message (Web Application)

Appointment cancelation notification (Mobile Application)

Security

The current solution allows users to access different areas of the system simply by manipulating the URL, for example a patient user could edit the URL page from dashboard.php to adminmanagement.php and gain access the data analysis page with no problem.

As the present version of the system only contains test data, security isn't currently an issue, however, due to the sensitivity of the real data the system will store once introduced into doctor surgeries, it is essential that the correct security mechanisms are implemented to stop this. For example, user types shouldn't be able to access other user type areas, and on attempting to do so should be provided with unauthorised permission page restricting entry.

Further security measures should be taken to help protect data stored within system database. Currently as I am the only individual carrying out database administration, access to this data is limited, however, in future administration tasks may be carried out by various members of staff. If the database was comprised by one of these individuals or an outsider gained unauthorised access it could have a negative affect on the surgery and my system. To prevent this, it is important that in the future passwords are encrypted. This could be done by using an industry-standard cipher, such as the Message-Digest Algorithm 5 (MD5). This is a one way hashing process that ensures that the passwords stored on the server cannot be deciphered by anyone.

Receptionist user type

It was initially planned that there would be a fourth user type accessing the system. This was receptionist, however, as the project was focused on the implementation of a new method, time constraints meant this was not viable use of efforts.

Future versions of the solution however, should include receptionist user type, allowing reception staff to easily configure and update system resources with information such as scheduled appointments.

Dummy data

When creating the prototype of the solution time constraints meant that it wasn't possible to compile as much data as I would of liked. This meant that the amount of test data may not provide a full representation of how the system will manage it, for example within the data analysis section an accurate demonstration of how tools may be used by administration staff by not be provided. To generate a larger set of realistic data for system demonstrations, in the future I would create an algorithm to simulate dummy appointment, user and prescription data etc.

Visual appearance

As the aim of the project was to implement all requirements the appearance of the system isn't to the standard I would have liked the end product to have been. Although this is not a key requirement I believe that future developments should also concentrate on developing the visual aspects of the system in order to provide a more professional feel before distribution.

System Evaluation

Once all aspects mentioned within future work have been implemented, it is important that they are tested thoroughly, just as current aspects have been, to identify that they work to their proposed purpose within the system, without error. Previously identified requirements for further analysis such as SUS should also be carried out to ensure results are accurate.

8.0 Conclusion

To summarise, the aim of this project was to design and develop a suitable solution that could be used within local doctor surgeries to improve their accessibility and management of resources. Overall, I believe that, whilst there is opportunity for future work, the project was successful at meeting these goals.

The system at its current state of development is a web application supporting all tasks required by its users, within the scope of the project. However, before the system could be introduced and operational in a real life environment, as previously mentioned, further work is very much required.

To arrive at the point in which the system scope was met, I had to begin with conducting some initial research of the existing solutions in place and identify how they could be improved to provide a unique solution to doctor surgeries in Wales. I was able to use this research to then define the systems requirements.

The next stage was to design the system architecture and user interface. This was a very important step and helped me gain initial blue prints of how data would be stored and how users would interact with the system. I then progressed to the implementation of these designs. The implementation stage was a slow but steady process; in which I did have some difficulties as expected with my limited technical knowledge. However, I managed to complete the stage and it was a beneficial experience.

In order to evaluate the success of the implemented system, I used various testing techniques such as test cases, and System Usability Scale analysis to gauge its usability. Results indicated that the system operated successfully, however, there was scope for future improvement.

On completion of the project I have developed a system that provides those interacting with a doctor surgery the functionality to perform required tasks with an alternative more accessible method, that doesn't replace others.

Although the project has been a challenge, I am proud to have successfully managed the process and learnt a lot about project management and programming along the way, overcoming the obstacles in front of me.

9.0 Reflection on Learning

Now that the project has come to an end, I can reflect on my individual learning. In order to do this, I must evaluate the techniques and lessons I have learnt throughout the project.

New Tools

I experienced the use of Bootstrap framework for the first time when completing the project. The framework was ideal for the time requirements of the project, helping to reduce implementation time and create a responsive, professional user interface. I would definitely use it again in a similar situation, however, I learnt that Bootstrap is a very popular framework, which can lead to the inadvertent creation of similar, generic and unmemorable website features. If the project was to be done again and time constraints weren't a big feature, I would consider creating the web application without the use of bootstrap, to generate a more unique website.

Time management

The project has helped me further my appreciation for the importance of time management when faced with time constraints. During the beginning of the project, the time period provided, appeared to be ample to myself. Before the project I had never had to manage something of this scale, so work wasn't perceived as to be as urgent as in fact it was.

As time began to progress and milestones I had indicated with my Gantt chart were ignored, I started to fall behind. When I started the implementation stage of the process it hit me, by not regarding my Gantt chart it had led to an accumulation of work. I found that I hadn't scheduled enough time for the implementation stage of the system, however, rather than skip certain elements and limit the desired product, I compacted this work into a stressful few weeks. I had learnt to be more respectful of the time required to complete tasks and dedicated all of my remaining time to the completion of the project.

If I could back to when I defined my time management schedule, I would definitely have condensed the time plan for initial stages of the project, such as background research, instead focusing more time on implementation. Overall, the project has had a large impact on my understanding of time management and has taught me to consider scheduling greatly.

Personal learning

The project spanned over one term and was the longest time spent by myself working on a project individually. I believe, due to this I have gained a lot of experience individually that I wouldn't have been able to gain from a group project. The project length enabled me to experience all stages of a project lifecycle and test my skills learnt from my time at university.

This project has proved to be a great learning opportunity for me, as through its progression, I have developed my technical knowledge considerably. Although I had some knowledge of the required technical tools such as HTML, PHP and MySQL, I did underestimate the challenges that would be involved in the development stages. Whilst undergoing development, due to this limited knowledge I was forced to seek out new information and techniques, gaining the skills to overcome the obstacles in front of me. I believe that if it wasn't for this project I wouldn't have had the motivation to carry out this research and am proud to have successfully produced a solution to the problem at hand.

The experience I have gained through the project I believe, has contributed to my professional development as an individual. I now have gained a greater understanding of each stage of system development and the processes involved, which I can reuse in the future depending on the career path I choose. On completion of the project I have a much clearer view of what I am truly capable of producing with the skills I've gained at university.

10.0 Bibliography

Agile Waterfall hybrid smart approach or terrible solution. (n.d.). Retrieved 02 08, 2016, from http://intland.com/blog/agile/agile-waterfall-hybrid-smart-approach-or-terrible-solution/

Bootstrap homepage template. (n.d.). Retrieved 02 18, 2016, from http://getbootstrap.com/examples/jumbotron/

Bootstrap simple sidebar. (n.d.). Retrieved 02 18, 2016, from http://blackrockdigital.github.io/startbootstrap-simple-sidebar/

Bootstrap Templates. (n.d.). Retrieved 02 08, 2016, from http://designscrazed.org/free-responsive-html5-css3-templates/

Data Protection Act Fine. (n.d.). Retrieved 02 02, 2016, from http://www.bbc.co.uk/news/technology-23286231

Free of Information Act. (n.d.). Retrieved 02 09, 2016, from http://www.nhs.uk/aboutnhschoices/contactus/pages/freedom-of-information.aspx

Intellectual Property. (n.d.). Retrieved 02 09, 2016, from https://www.copyrightservice.co.uk/copyright/intellectual_property

JavaScript Event Calendar. (n.d.). Retrieved 04 02, 2016, from http://fullcalendar.io/

Patient Online: the benefits of online access to records for GP practices and patients. (n.d.). Retrieved 02 07, 2016, from http://www.nhs.uk/aboutNHSChoices/aboutnhschoices/find-and-choose-services/Pages/gp-online-services.aspx

Patient-Access. (n.d.). Retrieved 02 07, 2016, from http://www.stopsleyvillagepractice.co.uk/pages/Patient-Access

Smartphone Users Reasearch. (n.d.). Retrieved 05 01, 2016, from http://media.ofcom.org.uk/news/2015/cmr-uk-2015/

Stakeholder Analysis . (n.d.). Retrieved 05 03, 2016, from https://www.mindtools.com/pages/article/newPPM_07.htm

System Usability Scale. (n.d.). Retrieved 05 02, 2016, from http://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html

Visual Impairment Study. (n.d.). Retrieved 02 02, 2016, from http://www.nhs.uk/Conditions/Visual-impairment/Pages/Introduction.aspx