Creating a Web Application to Upload & Access the Results of ArcCheck Radiotherapy Tests

UNIVERSITY OF
LINCOLN

Alexandra Hawthorne
HAW15618835

15618835@students.lincoln.ac.uk

School of Computer Science

College of Science

University of Lincoln

Submitted in partial fulfilment of the requirements for the
Degree of BSc(Hons) Computer Science

Supervisor: Dr. Mark Doughty

April 2021

Abstract

This report details the design and development of both a server-side web application and an SQL database created for the purpose of extracting, storing, and accessing key values from ArcCheck Radiotherapy Quality Assurance (QA) test scans, in addition to providing context for this sector as a whole, discussing relevant literature, and detailing Software Development Lifecycle (SDLC) methodologies.

The results of this project are mixed – both necessary products have been created, and tag extraction from certain files is functional, albeit rudimentary in part due to a lack of testing material required for developing these features. A database schema has been fully developed and integrated into the web application and extracted data from files is able to be submitted to this. Authentication will also be discussed and has been successfully implemented both in the web application and in storing credentials in a secure manner inside the database.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Brief Description of Area & Goals

This project is in collaboration with Lincoln County Hospital (LCH) and aims to create a web-based application to archive, upload and access the key information generated by the ArcCheck, a Radiotherapy treatment verification software used by LCH staff, to provide staff members a faster and more intuitive & reliable method of viewing and comparing these data.

The uploading & storing of this information includes implementing a method of submitting various files and algorithmically extracting necessary values, creating a location to store these entries, and adding the extracted information to this data store.

1.2 Additional Deliverables

Other requirements consist of employing security methods in the form of authentication – ensuring only authorised users are able to access the main functionality of this web application – normalisation of any database or equivalent used in order to minimise space used and designing the application itself to be as intuitive & simple to operate as possible.

Additional aims include the ability to view this information at a later date, potentially including the ability to filter and search for specific values or entries, the ability to update old information or correct any incorrect data, and the automatic movement of any relevant files into an archive folder.

Chapter 2

Literature Review

2.1 Background

In order to fully understand this project's requirements and create a suitable product, it was necessary to study the area and investigate relevant academic research that had been undertaken. In this there were several sections, with the most overarching being to understand the field of Radiotherapy as a whole.

2.1.1 Overview

Radiotherapy is a technique typically used to cure or treat patients suffering with cancer. The process uses a Linear Accelerator (RadiologyInfo.org, 2019) (Zhdanov, et al., 2019) to deliver small doses of radiation specifically to the affected area and can be used in conjunction with other cancer treatment methods to increase effectiveness – also known as neo-adjuvant Radiotherapy (NHS, 2020). The process is occasionally used as an accompanying treatment for other ailments including Dupuytren's disease (British Dupuytren's Society, 2013), however the evidence supporting the efficacy of this method has been referred to as weak for a variety of reasons, mainly due to a lack of data collected during studies undertaken (Kadhum, et al., 2017).

The Radiotherapy (RT) department at Lincoln County Hospital is required to frequently carry out Quality Assurance (QA) testing on their equipment to ensure its functionality, reliability, and accuracy, as these are all crucial to ensure an incorrect or misplaced dosage of radiation treatment does not occur. There are many variations of treatment the RT department provides, such as Intensity Modulated Radiotherapy (IMRT) – where radiation beams are used at multiple fixed angles around the patient, rotating the device's gantry (the 'arm' of the radiotherapy machine which holds the Linear Accelerator and allows the angle of radiation to be precisely varied (Oppelt, 2005)) – and Volumetric Modulated Arc Therapy (VMAT) – where doses are applied dynamically during the device's rotation (Zhdanov, et al., 2019) – and as technology develops these are becoming progressively more complex. This means it's extremely difficult for a department to conduct their own validation to ensure that radiation doses are applied accurately and reliably, instead turning to specifically designed products to perform this – one of the most prolific and well-used being the ArcCheck system.

2.1.2 ArcCheck

One product designed specifically to perform QA testing on RT equipment is ArcCheck, a system created by Sun Nuclear Corporation (Sun Nuclear Corporation, n.d.). This package is a combination of software and hardware, namely the ArcCheck Phantom. The Phantom is designed for testing the dose accuracy of a radiotherapy device by detecting many metrics including the distribution of radiation and 'fluence rate' – the amount of energy (in radiotherapy, usually the number of gamma ray photons) passing over a given surface area per unit time, with units , or (Ionactive, 2018). For an area of technology with such great risks, ensuring the accuracy of this system is a crucial step in its maintenance.

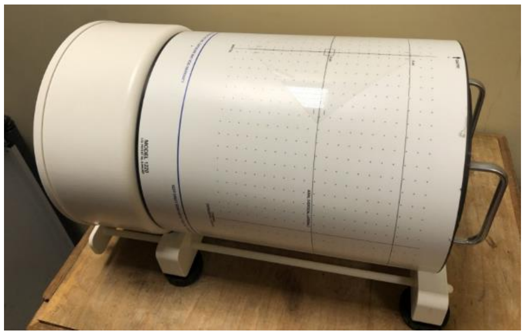


Figure 1 - The ArcCheck Phantom device (Butler, 2020)

ArcCheck is also shipped with a Dose-Volume Histogram analysis tool called 3DVH (Song, et al., 2015). This software is able to create visualisations of the dose information acquired by the Phantom, which can be extremely useful for QA testing. Figure 2 shows one of these graphics, demonstrating the dose volume for an IMRT scan. In (a), the red target area is where the dose is required, while the cyan and green concentric circles imitate Organs At Risk (OAR), for which it is crucial to minimise the exposure to radiation to reduce risks to patient welfare. (b) shows the level of actual radiation applied, strategically using five radiation beams at different angles in order to balance maximizing the target's exposure to the radiation treatment with minimising the dosage applied to nearby Organs At Risk and other sensitive areas. The number of radiation beams used – in addition to the dosage applied per beam – are affected by many factors and are generally calculated on a case-by-case basis. In recent years algorithms such as 'Sensitivity-Based Beam Number Selection' (SBBNS) have been developed to automatically determine the optimal values for these (Ranganathan & Maria Das, 2016). Naturally, using higher dosages and more radiation beams can present a severe risk of a patient's health, increasing the risk of other complications and potential malignancies later in life (Hall & Wuu, 2003).

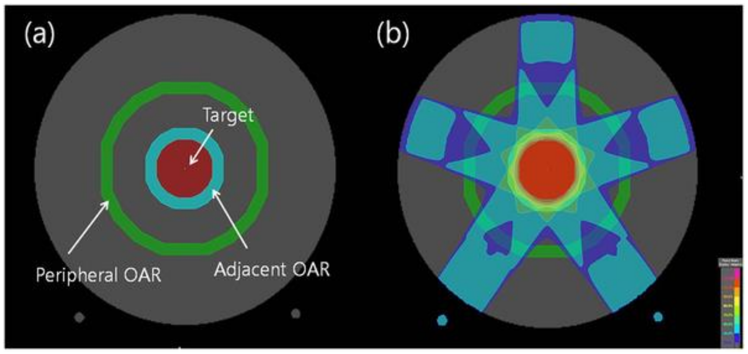


Figure 2 - A diagram of the dose accuracy for a Radiotherapy QA test performed on the ArcCheck Phantom. Organs At Risk are avoided while the target area is dosed with the maximum amount of radiation. (Song, et al., 2015)

As illustrated, it's crucial that radiotherapy doses are as accurate and reliable as possible, and thus testing & verification solutions such as ArcCheck are extremely important to ensuring patient safety when it comes to radiation exposure. However, an issue with using the ArcCheck system is that with each scan performed, a large amount of data is outputted. This includes a DICOM file (described below) and a PDF with graphical results detailing the success rate & accuracy of the scan.

This large amount of data can prove vital for maintaining the linear accelerator and ensuring there are no major problems, however an issue arises if a Radiotherapy team wants to either analyse the characteristics of multiple ArcCheck verification scans or archive them in such a way that the key information can be easily accessed and searched for at a later date. This is precisely the use case for the Lincoln County Hospital's RT team, and their current method for archiving these key data is arduous and time-consuming.

2.2 Related Literature

2.2.1 Current Approach

At present, the RT team creates custom-made plans for patients in Varian Medical Systems' Eclipse Treatment Planning System. These plans include key data such as where on the patient's body the scan will be focussed, the total radiation dose to be delivered, the number of fractions (Using a process known as fractionation, doses are generally split into several radiotherapy sessions, called fractions, to allow healthy cells to recover from the radiation among other reasons (The Royal College of Radiologists, 2019) (Boylan, 2013)), and a small amount of data related to each patient such as their name and NHS Number. In cases of QA testing there are other data stored such as the number of 'Control Points', which are the areas that the ArcCheck software will test to ensure that radiation dosages and dose accuracy are in order. The ArcCheck software also provides methods for graphical analysis of these Control Points (Sun Nuclear Corporation, 2014). This testing plan is executed by the RT device and exported into the ArcCheck system, which compares the planned radiation doses with the actual radiation distribution detected by the ArcCheck Phantom, and creates a multitude of files and folders, including DICOM and PDF files, detailing scan metadata in addition to the 'percentage pass rate' which determines how accurate the test was, based on how many Control Points were deemed close enough to planned value to pass, and thus whether the equipment would be safe to use on patients.

The data from these files is then written up into a large Excel spreadsheet containing specifics of the scan such as date & time, planned location on the 'body' and the type of scanning method used. This process is not only extremely onerous and time-consuming, but also yields large and cumbersome data which cannot be easily manipulated nor quickly understood. There are many thousands of records of these scans, with around 500 being performed every year (Butler, 2020), and so this process is unfit for documenting all of these.

The main contact for this project with regards to collaborating with Lincoln County Hospital was Tim Butler – the Lead of Clinical Computing for the United Lincolnshire Hospitals NHS Trust Radiotherapy Department. This is someone who has many years' worth of experience of utilising Radiotherapy equipment and managing its functionality. As will be discussed in other sections, meetings were held, and files were shared including a project brief and a list of DICOM tags to be extracted.

A DICOM (Digital Imaging COmmunications in Medicine) file is a universal standard for transmitting and storing a variety of medical images (Medical Imaging Technology Association, 2021), usually in the context of PACS (Picture Archiving and Communication Systems) (Choplin, et al., 1992). These files are used in many areas of medical imaging, including CAT and MRI scans, in addition to the field of Radiotherapy. They are denoted with the file extension ".dcm", and usually contain much more information than the image(s) themselves, in the form of DICOM tags. These are key-value pairs where the key denotes a certain piece of information, also given in the DICOM standard, for example the Patient's ID Number (in the UK, this is likely to be their NHS Number) is stored in the tag (0010, 0020) (DICOM Library, n.d.).

2.2.2 Proposed Solution & Ethical Requirements

The project proposed is a system which will automate the process of archiving these scans and saving key information in a much more sustainable way, as a long-term solution to these issues. This ensures that the Radiotherapy department can prioritise other tasks, in addition to being able to view the results of all executed scans in a straightforward system.

This solution is in the form of a web application and database, which stores all relevant data used in the previous Excel spreadsheet and more.

In addition to the automation of tag extraction and storage, another benefit to this approach would be the ease of data analysis and decreasing the requirement for intensive Quality Assurance investigations. Currently this QA process is performed on all patient scans, however it could potentially be dropped to every two or three if all recent QA testing has passed successfully (Butler, 2021).

In terms of the ethical implications of the proposed solution, there are several considerations to take into account. There are two distinct components in this project: a database containing key data from the ArcCheck verification software, and a web-based application which interfaces with this.

As the information stored in this database would all be data already held by the Lincoln County Hospital and should therefore not present an ethical issue with regards to whether the collection of these data will have a harmful or negative impact. However, it is arguable that providing an easy-to-access storage system for important data would allow malicious parties to locate information much more easily and give them faster access to more records than previously. This information is likely to contain patient details such as names and NHS Numbers in addition to the data related to the ArcCheck results themselves. Due to this, it’s key that all steps that can be taken to mitigate the potential for unauthorised access and misuse of data are implemented as best as possible.

One preventative action would be to ensure any user input to the database is “sanitised” before being queried or applied. This means writing code which ensures that users cannot interact directly with the database, no matter what they enter, effectively disallowing the possibility of SQL

Injection (or SQLi) - a form of code injection wherein a bad actor inputs a given devised string in a query being sent to an SQL database (Manh Thang, 2020). This string is designed to change the way in which the database executes the SQL query, thus giving the malicious user unauthorised access to not only data creation but also editing, deleting, and even viewing of potentially personal, private data (Jang & Choi, 2014).

Input sanitisation, when performed correctly, removes the risk of a user interacting with the database in unexpected ways by changing the input string – examples include deleting characters that could interfere with the way the SQL is executed, and even 'doubling up' single quotation mark symbols to ensure they're not parsed as part of the raw SQL code (Oracle, n.d.). There are many prewritten methods and functions for this in a variety of applications & languages, such as the "*mysql_real_escape_string()*" function in PHP (Weiss, 2012).

There is also a risk of unauthorised access through the network. This is a very valid concern, as currently the archived ArcCheck data are stored locally in a PC's file system, which is password-protected and generally only powered on when it is required. These steps allow for much tighter control of who can access this information and at what time. However, the proposed solution involves running both the database and a web application on a local server – a permanently accessible network location, able to be connected to by any device on the same network (TechTerms, 2014).

This implementation inherently poses a security risk, albeit small, as it involves creating another point of vulnerability where both the database and application are stored. There are many ways to mitigate this risk, though – one of the simplest is by implementing login and authentication systems in order to access devices on the network, in addition to ensuring the network itself is as secure as possible, for example using wireless security methods such as WPA2 to protect against unauthorized network connections (Adnan, et al., 2015).

Another issue to consider would be the potential risks during development of the project itself. It's likely that in order to develop algorithms to extract information and create a database schema, 'sample' information would & files have to be provided by the Radiotherapy department – such as DICOM image files containing relevant information – to be used when testing. In the case that any sample data would be transferred, whether online via email or through physical storage such as flash drives, the RT team would be required, legally and ethically, to take steps to either anonymise or pseudonymise these data.

Data anonymisation is a relatively well-known procedure wherein any personal information characteristics are removed from the data entries in question, in order to protect their identity (Sun, et al., 2011). There is also an alternative to this process, namely data pseudonymisation. Under 2016's General Data Protection Regulation (GDPR), pseudonymisation is defined as "*the processing of personal data in such a manner that the personal data can no longer be attributed to a specific data subject without the use of additional information, provided that such additional information is kept separately and is subject to technical and organisational measures to ensure that the personal data are not attributed to an identified or identifiable natural person*" (GDPR.eu, 2016).

Rather than simply removing these identifiers, they are instead modified to ensure that the individual cannot be associated to the data. This is sometimes implemented by generating random unique IDs and assigning them to each data element (Syed, et al., 2021). These IDs and their matching identifiable data would then be securely stored elsewhere: in the case of this project this could mean the Radiotherapy team changing patients' NHS Numbers – for example – to random strings or numbers before transferring data anywhere. This results in the RT department still having access to all necessary data while safeguarding patient information. Considering that the proposed solution includes the extraction and storage of this patient information, pseudonymisation would be much more suitable than anonymisation as it allows experimentation with these data fields.

In conclusion, even with these measures in place there is still a possibility of bad actors viewing and misusing these data, however given the steps taken the likelihood of this was deemed incredibly small, with the benefit and usefulness of the application to the Radiotherapy staff outweighing these risks. It's likely that allowing the Radiotherapy department to access and compare verification results in a more time-efficient and accessible way would have a positive impact overall. Thus, the project was continued with these safety precautions in mind. Any fully functional software officially implemented into the Radiotherapy department's workflow would likely be subject to much stricter NHS-based risk assessments and ethical approvals to ensure that there was minimal chance of issues arising.

2.2.3 Summary

The application will perform all of the standard Create, Read, Update and Delete procedures required for the maintenance of a database. It must also be simple & intuitive in its design, to ensure Radiotherapy staff are able to become accustomed to the new system easily and can perform necessary tasks effectively. In addition, it must be secure – as real patient scans are generally uploaded to the ArcCheck software, private information such as patient names, NHS Numbers and dates of birth may be stored. Unauthorised access to these must be prevented by implementing user authentication.

Chapter 3

Methodology

3.1 Project Management

3.1.1 Overview

There were a variety of factors which were taken into consideration when considering Project Management methodologies. One piece of software that proved particularly useful during this process and throughout the project creation was Notion (Notion, n.d.), an application designed for personal & collaborative productivity and project management, among other tasks.

Notion was used as a large part of the Project Management, from writing quick notes on what features and functionality were required, to larger summaries of systems & libraries that needed to be learnt & implemented. However, possibly the most important use case of the software was the creation of a Kanban board. This will be discussed further in 3.2 – Software Development. A Gantt chart was also created, using Microsoft Excel (Microsoft, n.d.).

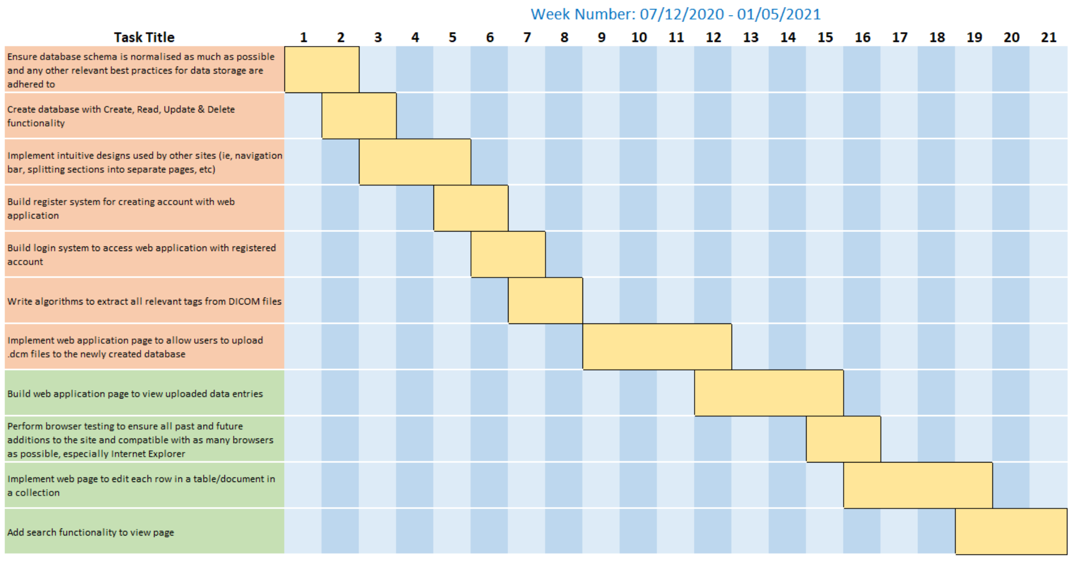


Figure 3 - Gantt Chart showing tasks to be completed, and estimated timeframe for each

According to Atlassian's Agile Coaching site, the Gantt chart is a "horizontal, timeline-based bar chart that represents a project plan in time" (Gebicz, n.d.). It's considered a universal tool of project management, regardless of the software development principal utilised. For this reason, it's widely regarded as a fundamental device for managing a vast array of tasks, even after more than one hundred years after its conception (Davidson, 2019). Gantt charts are designed to visualise a project's timeframe, which can be vital for comprehending what tasks can realistically be completed, in addition to often proving useful when it comes to keeping on track with relevant tasks and staying on schedule.

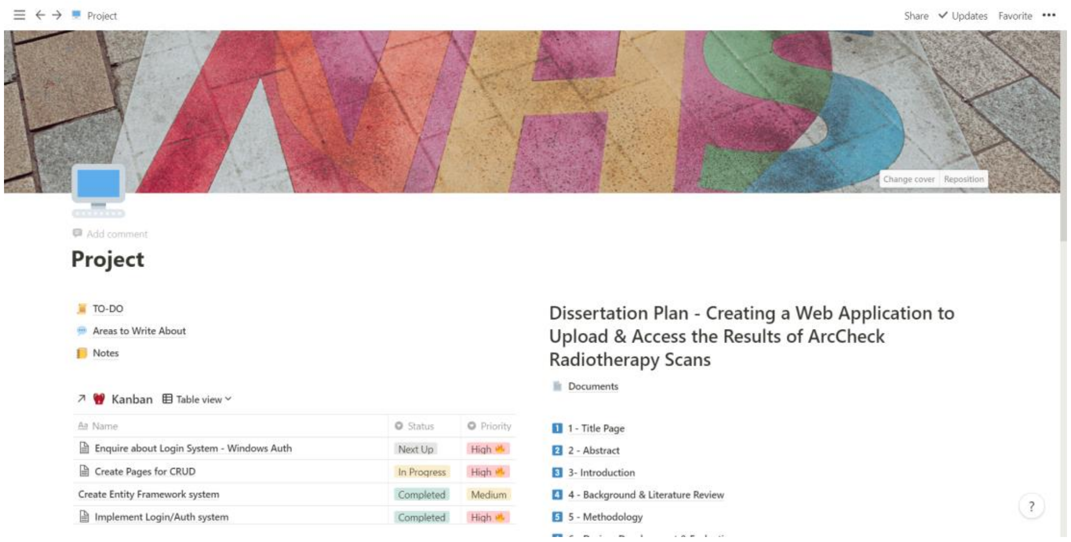
The former of these was certainly applicable to this project, since as discussed in this report's Software Development section, the initial planning of the project's requirements created three separate sections of tasks to complete. However, when adding these tasks into a Gantt chart and estimating the cumulative time cost, it was determined that the given timeframe for this project did not allow the third section of tasks to be realised. This allowed for these tasks to be dismissed from the overall goals of the project, although they were still kept in mind as other deliverables that could be completed if other tasks took less time than expected. In the case that these were not implemented they could be passed back to the client for the Radiotherapy department to create at a later date if needed.

The creation of the Gantt chart proved to be incredibly useful when it came to avoiding ‘scope creep’, sometimes referred to as feature creep. This is defined as “change or growth of project scope or the pressure to deliver more than what was agreed to originally” (Amoatey & Anson, 2017). It’s often a slow process which occurs over a long period of time, where small increments to the project’s aims & objectives occur. These increments are generally requirements that weren’t initially planned, and although the time cost of each individual item may be small, they can accumulate, pushing back original features and creating a large rift between the expected product and what was delivered. Poor definition of the project’s initial requirements is one of the biggest factors in allowing scope creep to take hold (Turk, 2010), and thus ensuring the project has well-laid out goals and deliverables in mind is key to avoiding this.

Besides the goals discussed in the Software Development section below, one major influence on the time constraints estimated during creation of the Gantt chart were the technologies potentially involved. After a few initial meetings with Tim Butler, it seemed likely that the frameworks and “stack” the RT team were familiar with, and thus would want the application to utilise, hadn’t been taught throughout the university course. This provided a great opportunity to put the skills learnt throughout university into practise and become familiar with alternative applications, however it did mean that learning these new architectures and approaches had to be taken into consideration when assessing the time given tasks were likely to take.

3.1.2 Why Notion

There are other alternatives to Notion that were also considered in the initial stages of the project, which included Trello (Trello, n.d.) and monday.com (monday.com, n.d.). Both of these products are frequently used both in and out of the Software Engineering field and feature similar kinds of functionality. However, as described above, many aspects of the systems & frameworks used in this project were not familiar and thus it proved useful to have a more versatile workspace which allowed for integrating & embedding relevant pages of notes into project management areas such as the Gantt chart.



Project

TO-DO
Areas to Write About
Notes

Kanban Table view

Name	Status	Priority
Enquire about Login System - Windows Auth	Next Up	High
Create Pages for CRUD	In Progress	High
Create Entity Framework system	Completed	Medium
Implement Login/Auth system	Completed	High

Dissertation Plan - Creating a Web Application to Upload & Access the Results of ArcCheck Radiotherapy Scans

Documents

1 - Title Page
2 - Abstract
3- Introduction
4 - Background & Literature Review
5 - Methodology

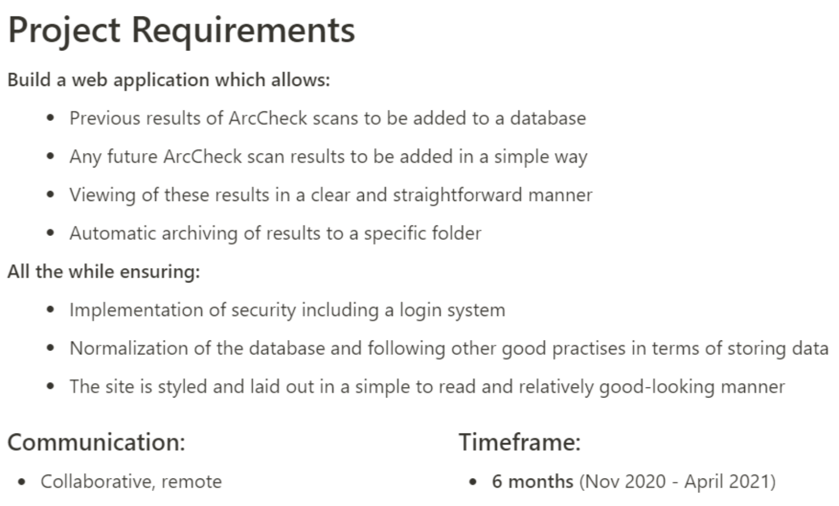
Figure 4 - Project Homepage in Notion

In addition, the above products have a much larger focus on collaboration with a team. Notion does also have these features; however, it is also designed with a focus on personal productivity & general management - which suited the requirements of this project well since all collaboration with radiotherapy team members was either verbal or written communication via emails.

3.2 Software Development

3.2.1 Investigating Requirements

One of the first steps in deciding a Project Management strategy was to create a brief list of the project's general requirements. Included in this were the basic functionalities listed in the Background section above – implementation of the standard Create, Read, Update and Delete functionality of data storage and ensuring security by way of user authentication. Also included was the automatic movement of uploaded files into an archive folder, and efficiency details such as normalising any databases used to minimise redundancy.



Project Requirements

Build a web application which allows:

- Previous results of ArcCheck scans to be added to a database
- Any future ArcCheck scan results to be added in a simple way
- Viewing of these results in a clear and straightforward manner
- Automatic archiving of results to a specific folder

All the while ensuring:

- Implementation of security including a login system
- Normalization of the database and following other good practises in terms of storing data
- The site is styled and laid out in a simple to read and relatively good-looking manner

Communication:

- Collaborative, remote

Timeframe:

- **6 months** (Nov 2020 - April 2021)

Figure 5 - A write-up of the project's potential requirements

The writeup of these rough requirements proved key in the decision-making process for which Software Development Lifecycle (SDLC) methodology to use – neatly consolidating the goals and other key information into one area allowed easier and quicker conclusions to be drawn based on the overall project context.

When making a decision as to which Software Development process to follow, the two main frameworks taken into consideration were Waterfall and agile approaches including Scrum and Kanban, however the Spiral methodology was also investigated.

3.2.2 Waterfall

The Waterfall model was the first to be considered. This process is generally regarded as one of the most inflexible approaches since it is comprised of a set number of fixed phases, including but not limited to requirements analysis, design, implementation & unit testing, integration & system testing, and operation & maintenance (Tarhan & Yilmaz, 2013). This fixed-period approach was deemed to suit this project's requirements in particular, as there was a predefined 'development length', by which time the product should be fully completed. Using the Waterfall methodology would then allow much simpler planning of how to spend this time, potentially dividing the overall timeframe into its component parts, and calculating appropriate costs & lengths before even beginning the project (Surbakti, et al., 2019).

Another possible reason for using Waterfall was the security implications and necessities discussed previously. Due to the rigidity of each phase only lasting a set length, in addition to generally having a higher level of project documentation overall than many other SDLC methods (Kavlakoglu, 2020), there is a much slimmer chance that any security measures and data handling best practices are overlooked, as these would likely have their own subsections inside the analysis, design and implementation phases. This advantage would be particularly relevant for this project due to the various considerations for the prevention of unauthorised access – as this is a project to be utilised by the NHS, and will likely be used to store personal information, it's crucial that all considerations are taken into account and fully implemented and tested.

However, there are also drawbacks to Waterfall as a process – the first of these being that it would be incredibly easy to under- or overestimate the time consumption for any of the phases and doing so would likely throw the entire project's actual timeline out of sync with what was planned. This is especially the case in this situation due to the mentioned learning curve of the systems & frameworks: if a particular topic was found to be particularly difficult or time-consuming to understand, it may pose a risk to the length of the implementation phase.

In addition, after a small amount of research into the web framework which the RT team currently utilises, there are a variety of pre-written methods and architectures which allow for abstraction of many authorisation and authentication procedures. The details of these will be discussed in section 6 – Design, Development & Evaluation. This illustrates, however, that many of the security concerns will possibly be handled by other systems & libraries, and thus may not require a large amount of time to design and implement.

3.2.3 Spiral Methodology

One approach that was also briefly explored was Spiral. This is a much more iterative method than Waterfall, containing multiple phases which are repeated as both the product and its prototypes expand in fidelity. Unlike Waterfall, this process includes risk analyses as dedicated phases, generally performed before creating mock-ups and prototypes (Ray, 2013). This would be particularly useful for the project's use case for the reasons stated above related to the security implications of the overall product.

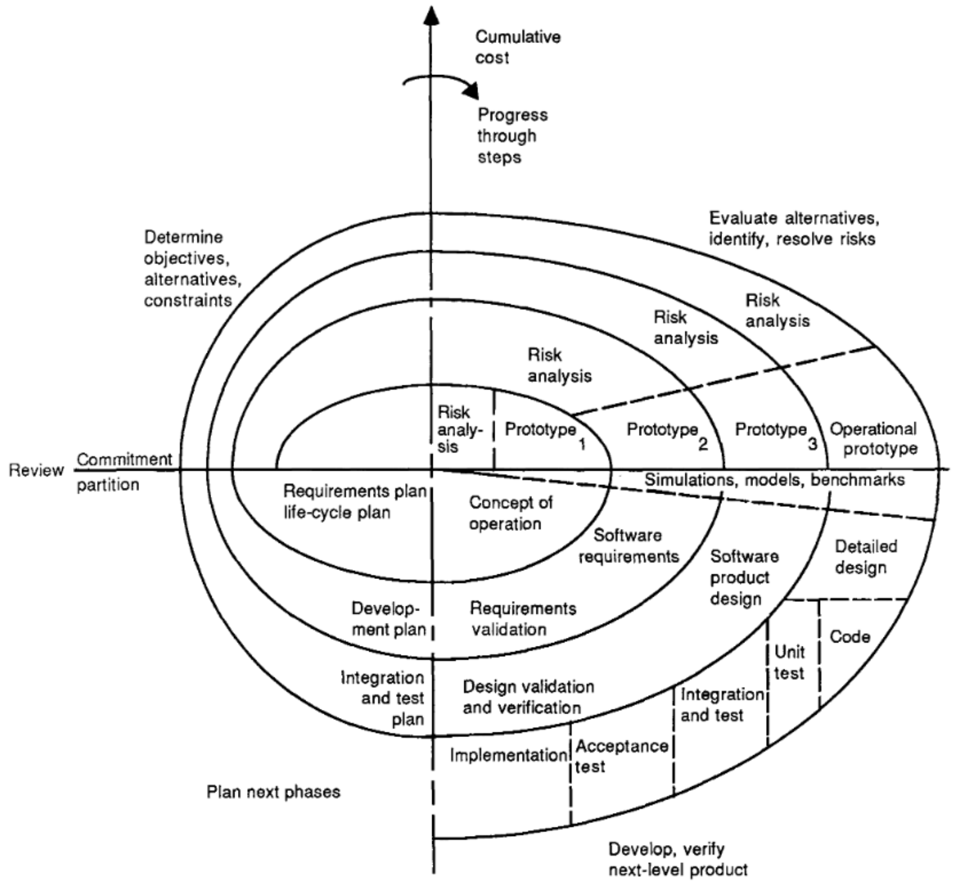


Figure 6 - An example of phases in the Spiral model (Boehm, 1988)

However, as shown in Figure 6**Error! Reference source not found.**, there a re many phases and sections to this methodology. For this reason, it is generally advised against usage in small-scale projects with little moving parts (Upadhyay, 2020), in addition to usually requiring a high level of expertise in order to effectively run this methodology (TestBytes, 2019). For these reasons, this methodology was quickly excluded.

3.2.4 Agile Analysis: Kanban & Scrum

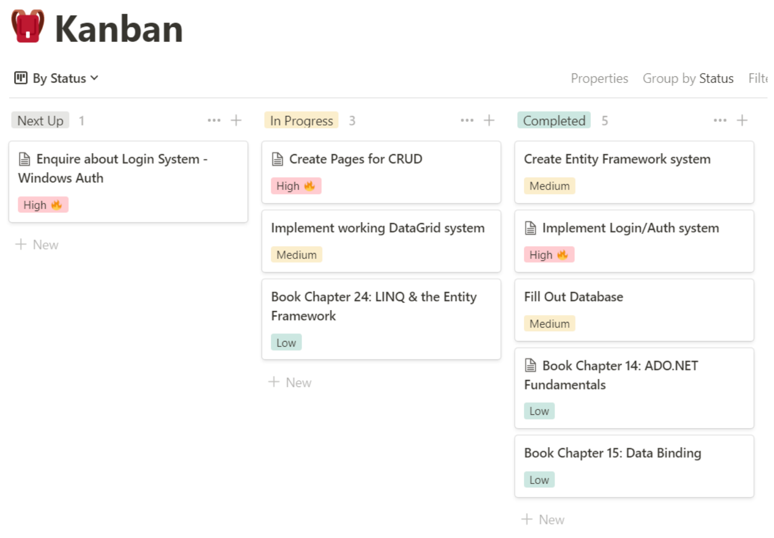
In Software Development, the term ‘agile’ usually describes any methodology or process which applies a set of philosophies popularised in the Agile Manifesto. The four core values of this are as follows: (Beck, et al., 2001)

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

This is a much more flexible and responsive approach designed for changing competition, rapid development, and environments of varying requirements. It’s generally focussed on iteratively building a product in a similar way to Spiral, however with a lot less of a highlight on fixed phases throughout this process.

The techniques used in many agile methodologies, including Scrum and Kanban (GOV.UK Agile Delivery Community, 2016), can have large overlaps. For the most part Scrum was investigated, however parts of the Kanban methodology were also explored.

Kanban is a methodology that heavily takes from the principles laid out in the Agile Manifesto as described above. The most noticeable feature of this process is the inclusion of a Kanban Board, a tool which helps visualise and comprehend project tasks – upcoming, in progress and completed (Rehkopf, n.d.). This is a flexible project management system which can be used for small and large products alike and was utilised in this project due to its simplicity and ease of use. While Kanban is often used in team environments with multiple people working collaboratively, it’s also applicable in single-person projects, and can even be implemented for small tasks in everyday life (Henry, 2015).



Kanban

By Status

Properties Group by Status Filt

Next Up 1

Enquire about Login System - Windows Auth
High

+ New

In Progress 3

Create Pages for CRUD
High

Implement working DataGrid system
Medium

Book Chapter 24: LINQ & the Entity Framework
Low

+ New

Completed 5

Create Entity Framework system
Medium

Implement Login/Auth system
High

Fill Out Database
Medium

Book Chapter 14: ADO.NET Fundamentals
Low

Book Chapter 15: Data Binding
Low

+ New

Figure 7 - Implementation of Kanban board to track task progress throughout project

However, one issue that was noted with the Kanban methodology was that it seemed to be incredibly vague in its development steps, essentially going the opposite direction to inflexible and high-documentation processes such as Waterfall and Spiral (Alquda & Razali, 2017).

Another example of an agile-based Software Development methodology would be Scrum. This is an inherently team-based methodology, usually consisting of five to nine people (Visual Paradigm, n.d.), including a Product Manager, a Scrum Master, and a Development Team.

The Product Owner is responsible for developing a Product Backlog – a master list of all features & functionality the final product should consist of, in the form of User Stories, ranked by their importance and time cost. The Scrum Master essentially oversees the Development Team and aids with any issues they may face during development.

The development period is split into sections, or Sprints, and the items in the Product Backlog are divided between them based on their priority. The process of considering & prioritising tasks would be particularly useful for this project since there are many potential deliverables, and it was believed that the scope of the entire project as a whole would be longer than the time period for this module, thus it would be crucial to analyse which tasks are more necessary to implement than others.

Another benefit of using the Scrum methodology would be its relative simplicity when it comes to starting to use it for the first time. Unlike processes such as Spiral, Scrum is described as having the ability to "be introduced to a company with relatively little hassle" (Turner, 2018). This would be particularly useful due to the relatively short time constraint for this module.

An additional advantage to this approach would be the ability to build iteratively, developing smaller features and adding functionality as time passes. This would likely prove particularly useful since as described above, there are many facets of this project which hadn't been learned before. The Scrum process would then potentially give the ability to implement small sections while learning the fundamentals of the frameworks and libraries and improving and adding to them with each Sprint.

The final benefit to this process would be the inclusion of a "Daily Scrum". These are regular meetings, normally every morning, between the members of a Scrum team to discuss progress and any issues they may be facing (Stray, et al., 2013). While daily meetings with Tim Butler would most likely be unnecessary, regular meetings would be useful; in particular shorter, more numerous discussions as opposed to long but infrequent ones.

However, there is a prominent issue with using the Scrum methodology as part of this project –the process is designed for working in a team environment, generally with at least five members as described above, allowing for the ability for the different roles and responsibilities. While this is an obvious issue with this specific project, it was felt that many of the features and benefits of this process would be well suited. Research was conducted and a Scrum-based alternative for single-person teams was found, often called Personal Scrum. This is described as "an Agile methodology that adapts and applies Scrum practices to one-person projects. It promotes personal productivity through observation, adaptation, progressive elaboration, prioritizing and sizing work, and time-boxing" (Pruitt, 2011).

3.2.5 Applying Kanban & Scrum Processes

The main idea behind Personal Scrum involved taking beneficial steps, philosophies and ideas from Scrum and adapting them for a team of one. The first of these was the creation of User Stories to develop the brief requirements initially created, mentioned in 5a – Project Management. These were sorted into two main categories, shown in Figure 8**Error! Reference source not found.**, illustrating the two types of staff who would be using the application.

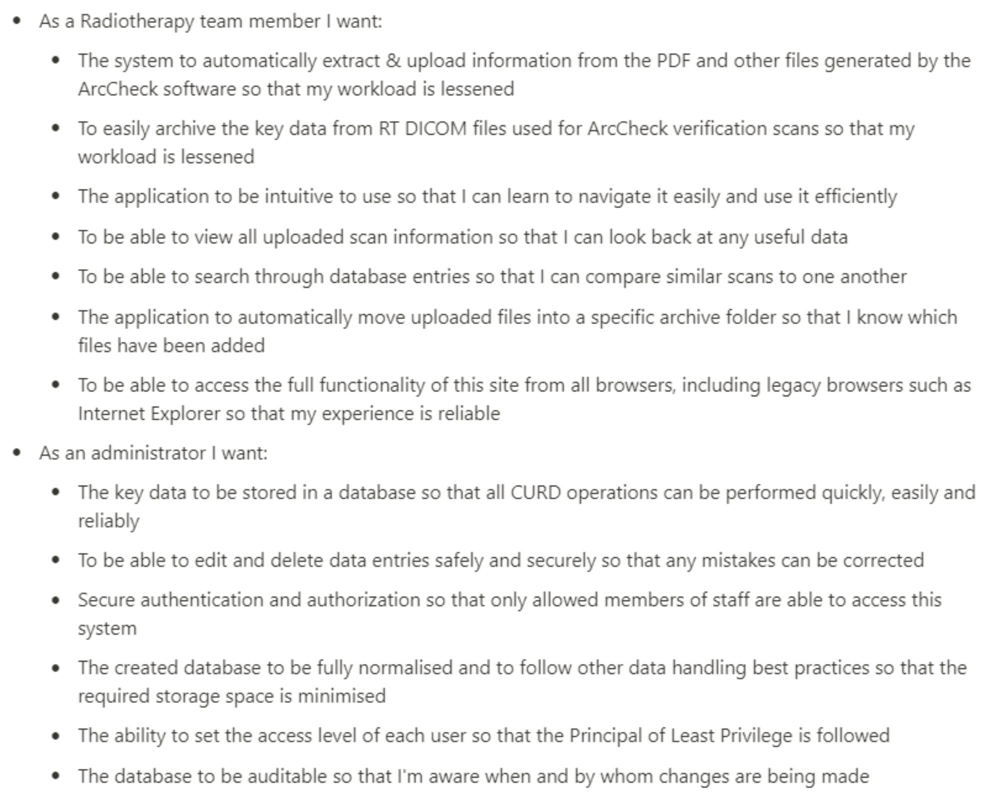
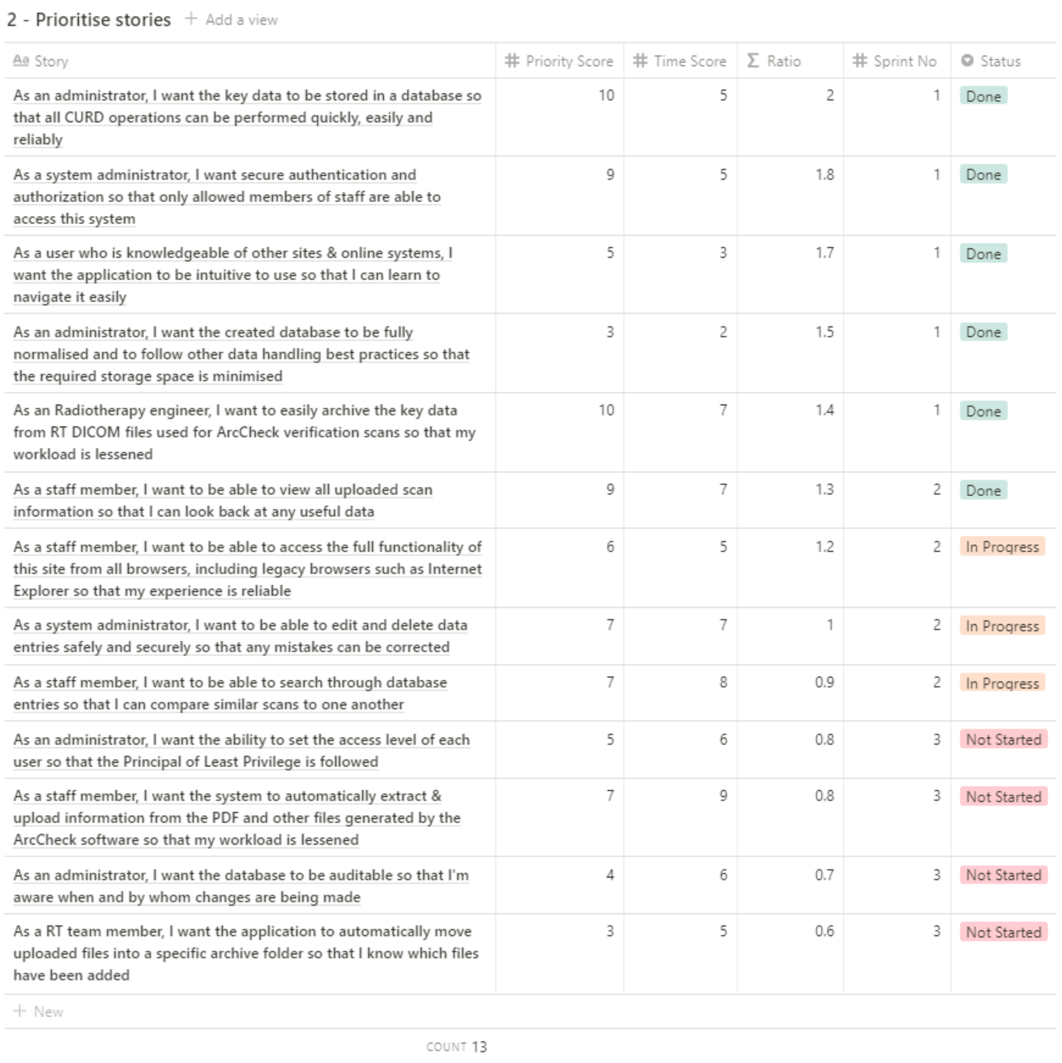

- As a Radiotherapy team member I want:
  - The system to automatically extract & upload information from the PDF and other files generated by the ArcCheck software so that my workload is lessened
  - To easily archive the key data from RT DICOM files used for ArcCheck verification scans so that my workload is lessened
  - The application to be intuitive to use so that I can learn to navigate it easily and use it efficiently
  - To be able to view all uploaded scan information so that I can look back at any useful data
  - To be able to search through database entries so that I can compare similar scans to one another
  - The application to automatically move uploaded files into a specific archive folder so that I know which files have been added
  - To be able to access the full functionality of this site from all browsers, including legacy browsers such as Internet Explorer so that my experience is reliable
- As an administrator I want:
  - The key data to be stored in a database so that all CURD operations can be performed quickly, easily and reliably
  - To be able to edit and delete data entries safely and securely so that any mistakes can be corrected
  - Secure authentication and authorization so that only allowed members of staff are able to access this system
  - The created database to be fully normalised and to follow other data handling best practices so that the required storage space is minimised
  - The ability to set the access level of each user so that the Principal of Least Privilege is followed
  - The database to be auditable so that I'm aware when and by whom changes are being made

Figure 8 - List of all created User Stories, sorted into two categories for the users of the software

These User Stories were then formed into a Product Backlog, and given scores based on their estimated overall importance and time cost. A simple equation divided these to create a general priority score – using Notion Formulas (Notion.VIP, n.d.) and rounding to one decimal place, this was: . This score determined where each User Story was positioned. Stories were then assigned to Sprints. Three sprints were developed, beginning with the highest priority Items, as shown in Figure 9**Error! Reference source not found.**.



2 - Prioritise stories + Add a view

Story	Priority Score	Time Score	Ratio	Sprint No	Status
As an administrator, I want the key data to be stored in a database so that all CURD operations can be performed quickly, easily and reliably	10	5	2	1	Done
As a system administrator, I want secure authentication and authorization so that only allowed members of staff are able to access this system	9	5	1.8	1	Done
As a user who is knowledgeable of other sites & online systems, I want the application to be intuitive to use so that I can learn to navigate it easily	5	3	1.7	1	Done
As an administrator, I want the created database to be fully normalised and to follow other data handling best practices so that the required storage space is minimised	3	2	1.5	1	Done
As an Radiotherapy engineer, I want to easily archive the key data from RT DICOM files used for ArcCheck verification scans so that my workload is lessened	10	7	1.4	1	Done
As a staff member, I want to be able to view all uploaded scan information so that I can look back at any useful data	9	7	1.3	2	Done
As a staff member, I want to be able to access the full functionality of this site from all browsers, including legacy browsers such as Internet Explorer so that my experience is reliable	6	5	1.2	2	In Progress
As a system administrator, I want to be able to edit and delete data entries safely and securely so that any mistakes can be corrected	7	7	1	2	In Progress
As a staff member, I want to be able to search through database entries so that I can compare similar scans to one another	7	8	0.9	2	In Progress
As an administrator, I want the ability to set the access level of each user so that the Principal of Least Privilege is followed	5	6	0.8	3	Not Started
As a staff member, I want the system to automatically extract & upload information from the PDF and other files generated by the ArcCheck software so that my workload is lessened	7	9	0.8	3	Not Started
As an administrator, I want the database to be auditable so that I'm aware when and by whom changes are being made	4	6	0.7	3	Not Started
As a RT team member, I want the application to automatically move uploaded files into a specific archive folder so that I know which files have been added	3	5	0.6	3	Not Started

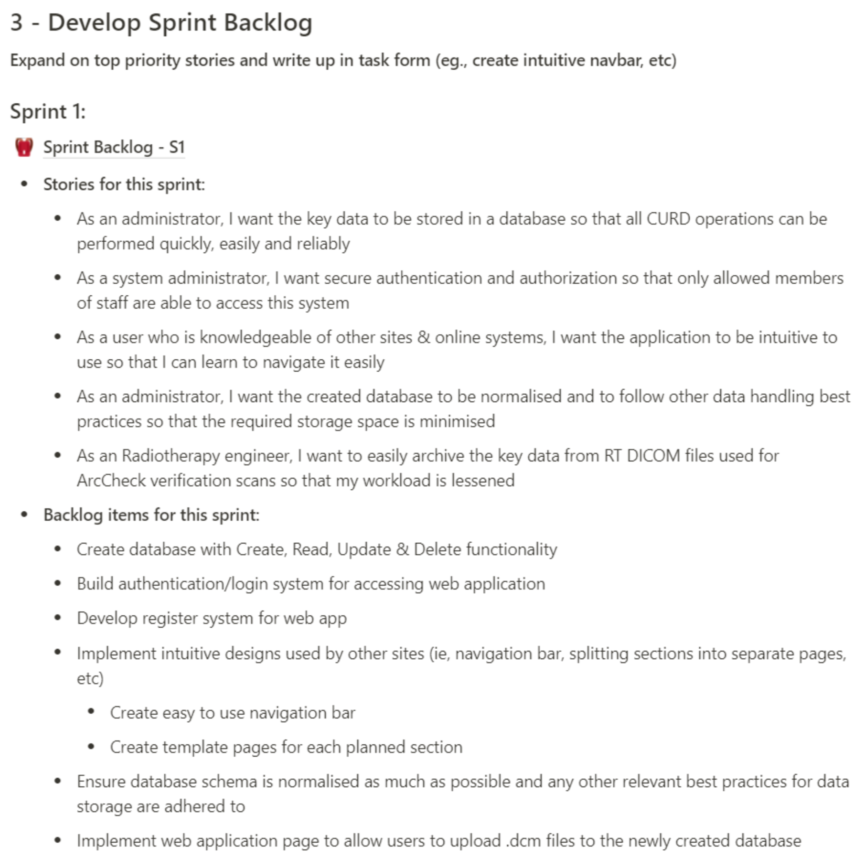
+ New

COUNT 13

Figure 9 - User Stories tabulated and given importance & time scores, and overall priority was calculated from this

These sprints were planned to be executed throughout the year. However, as described above, the creation of a Gantt chart, which involved estimating the time taken for each of these User Stories and plotting them against the project's timeframe, illustrated that there would likely not be enough time to allow for a third sprint, thus these requirements were mostly skipped. If there was any excess time at the end of the project period, however, they would likely be reconsidered and potentially implemented.

For each sprint, the relevant User Stories were rewritten into a Sprint Backlog. The first step in this was rewriting each User Story in the form of one or multiple deliverable tasks, usually called Backlog Items.



3 - Develop Sprint Backlog

Expand on top priority stories and write up in task form (eg., create intuitive navbar, etc)

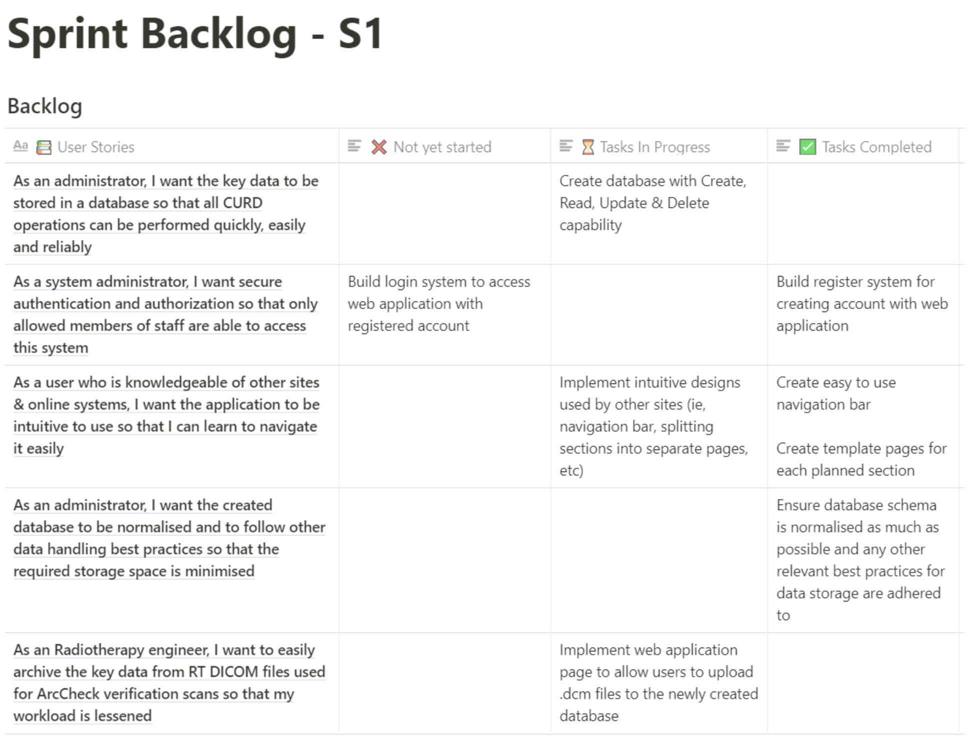
Sprint 1:

Sprint Backlog - S1

- **Stories for this sprint:**
  - As an administrator, I want the key data to be stored in a database so that all CURD operations can be performed quickly, easily and reliably
  - As a system administrator, I want secure authentication and authorization so that only allowed members of staff are able to access this system
  - As a user who is knowledgeable of other sites & online systems, I want the application to be intuitive to use so that I can learn to navigate it easily
  - As an administrator, I want the created database to be normalised and to follow other data handling best practices so that the required storage space is minimised
  - As an Radiotherapy engineer, I want to easily archive the key data from RT DICOM files used for ArcCheck verification scans so that my workload is lessened
- **Backlog items for this sprint:**
  - Create database with Create, Read, Update & Delete functionality
  - Build authentication/login system for accessing web application
  - Develop register system for web app
  - Implement intuitive designs used by other sites (ie, navigation bar, splitting sections into separate pages, etc)
    - Create easy to use navigation bar
    - Create template pages for each planned section
  - Ensure database schema is normalised as much as possible and any other relevant best practices for data storage are adhered to
  - Implement web application page to allow users to upload .dcm files to the newly created database

Figure 10 - The User Stories and respective backlog items for the first sprint

Once tasks had been established, a Sprint Backlog table was used (Scrum.org, n.d.). This is similar to a Kanban board in terms of its ability to categorise each Backlog Item into its progress – Not Started, In Progress or Completed.



Sprint Backlog - S1

Backlog

User Stories	❌ Not yet started	⏳ Tasks In Progress	✅ Tasks Completed
As an administrator, I want the key data to be stored in a database so that all CURD operations can be performed quickly, easily and reliably		Create database with Create, Read, Update & Delete capability	
As a system administrator, I want secure authentication and authorization so that only allowed members of staff are able to access this system	Build login system to access web application with registered account		Build register system for creating account with web application
As a user who is knowledgeable of other sites & online systems, I want the application to be intuitive to use so that I can learn to navigate it easily		Implement intuitive designs used by other sites (ie, navigation bar, splitting sections into separate pages, etc)	Create easy to use navigation bar Create template pages for each planned section
As an administrator, I want the created database to be normalised and to follow other data handling best practices so that the required storage space is minimised			Ensure database schema is normalised as much as possible and any other relevant best practices for data storage are adhered to
As an Radiotherapy engineer, I want to easily archive the key data from RT DICOM files used for ArcCheck verification scans so that my workload is lessened		Implement web application page to allow users to upload .dcm files to the newly created database	

Figure 11 - Sprint Backlog part way through Sprint 1

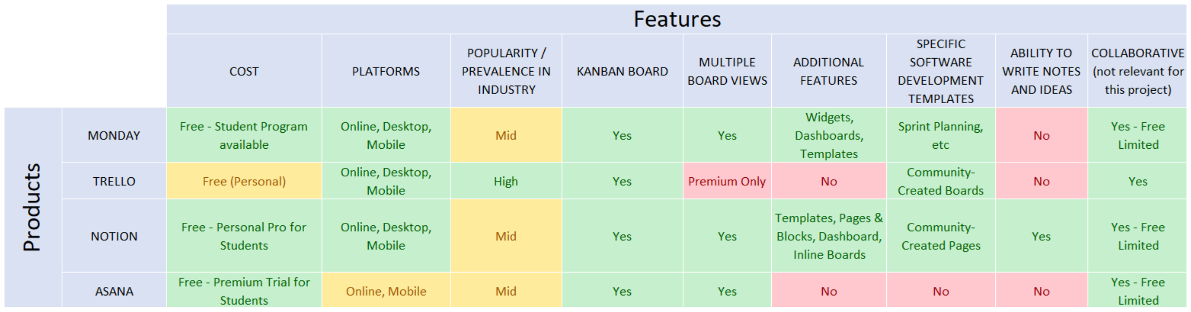
3.3 Toolsets and Machine Environments

One of the major toolsets that was used during Project Management has been discussed in the Project Management section was Notion. However, this was not the only product that was considered to serve the purpose of managing project tasks and keeping on track.

3.3.1 Toolsets in Project Management

There are a plethora of alternative Project Management software packages available online, including ClickUp (ClickUp, n.d.), KissFlow Project (Kissflow Project, n.d.), and Smartsheet (Smartsheet, n.d.) (Aston, 2021). However, the four that were chosen for analysis were monday.com, Notion and Trello as described above, in addition to another application called Asana (Asana, n.d.). With the exception of Notion, these were all found by checking a variety of lists and reviews of the best Project Management software (Delos Santos, 2021) (Fearn, et al., 2021). Notion was known through previous use in personal life.

Various factors were considered when deciding the final application to use for this project. The main functionality which was expected from each of these applications was to create Kanban boards in order to manage the project workflow & various tasks. This was featured in every piece of software, which left none excluded. A Matrix Diagram was created (Lucidchart Content Team, n.d.), using an ‘L-shaped’ matrix since there were two main groups being compared to each other – the applications and their features (Ashur, 1992), to aid with making a decision between these.



		Features								
		COST	PLATFORMS	POPULARITY / PREVALENCE IN INDUSTRY	KANBAN BOARD	MULTIPLE BOARD VIEWS	ADDITIONAL FEATURES	SPECIFIC SOFTWARE DEVELOPMENT TEMPLATES	ABILITY TO WRITE NOTES AND IDEAS	COLLABORATIVE (not relevant for this project)
Products	MONDAY	Free - Student Program available	Online, Desktop, Mobile	Mid	Yes	Yes	Widgets, Dashboards, Templates	Sprint Planning, etc	No	Yes - Free Limited
	TRELLO	Free (Personal)	Online, Desktop, Mobile	High	Yes	Premium Only	No	Community-Created Boards	No	Yes
	NOTION	Free - Personal Pro for Students	Online, Desktop, Mobile	Mid	Yes	Yes	Templates, Pages & Blocks, Dashboard, Inline Boards	Community-Created Pages	Yes	Yes - Free Limited
	ASANA	Free - Premium Trial for Students	Online, Mobile	Mid	Yes	Yes	No	No	No	Yes - Free Limited

Figure 12 - Decision Matrix for each Project Management software considered

This matrix included factors of varying importance, such as whether it was possible to view the Kanban boards in multiple other ways, for example as a timeline of when tasks were due.

Two features that weren't valued as much were the total platforms the product was accessible on, and whether collaboration with teams was possible. This was mostly disregarded as all had the ability to access through an online site and mobile app, and a desktop application would likely not prove all that beneficial. Team collaboration wasn't particularly relevant since this was mostly a personal undertaking despite the product being with a client in mind.

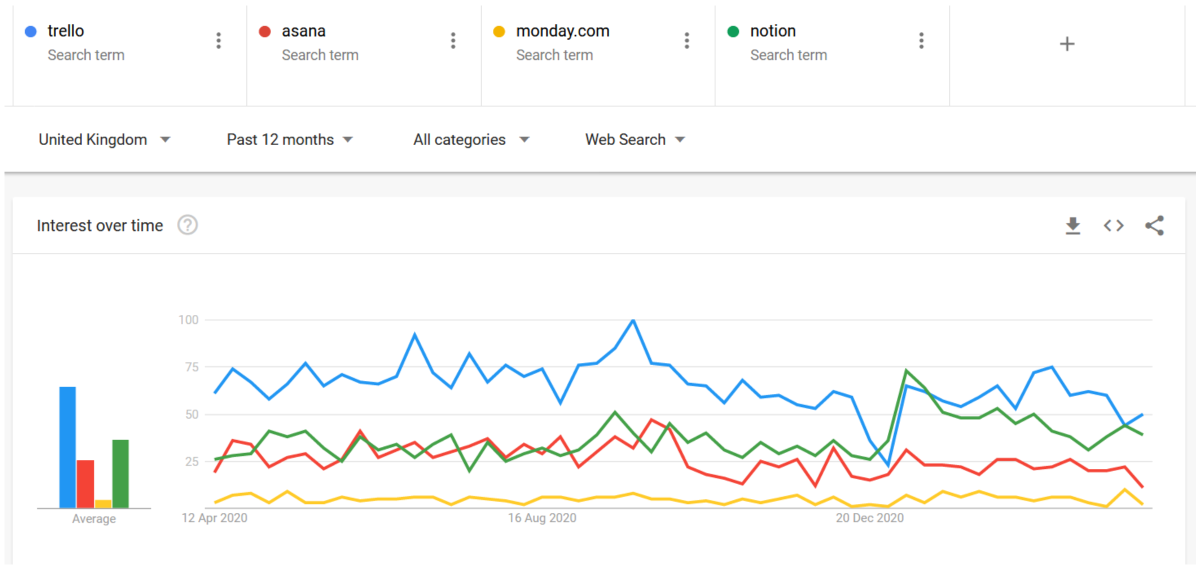


Figure 13 - Google Trends comparison roughly showing the company's popularities over time (Google, 2021). Possibly inaccurate due to multiple meanings of words such as 'Notion', and specificity of 'monday.com'.

Another factor taken into consideration was how well-used the application was in the field overall. Although Trello had by far the highest user count at 50 million in October 2019 (Trello, 2019), many sites seemed to have stopped recommending it in favour of more recent & flexible applications such as monday.com and Asana (Delos Santos, 2021), with Asana's current user count at over 3 million (Shieber, 2020) and monday.com having a similar paying customer count to Asana (Shieber, 2020) (Konrad, 2019).

The ability to write and embed notes and other information into boards was very highly valued since it allowed the consolidation of information, general notes, and Project Management plans into one application. It was for this reason that Notion was deemed best to suit this specific project and thus was used.

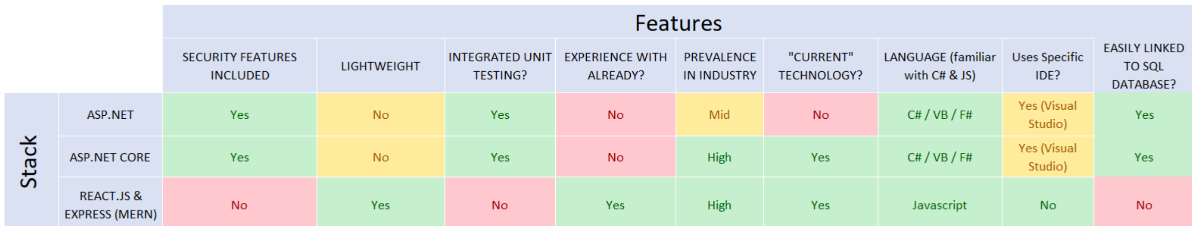
3.1.2 Toolsets & Machine Environments in Software Development

As discussed in the Background section of this report, this product will likely be hosted on a local server running in the Radiotherapy department of Lincoln County Hospital. As this product is planned to be used to store potentially personal information, security regarding who can access which areas is a high priority. Therefore, when deciding which toolsets to utilise, ease of implementation of security features would play a major part in the decision process.

As a result of this project being developed for a client, who will likely be performing maintenance and possibly be adding & editing features after delivery, it was necessary to use the applications and frameworks that the Radiotherapy department were familiar with. This included developing a relational database via Microsoft's SQL Server (Microsoft, n.d.), in addition to using ASP.NET (Microsoft, n.d.) to create the web application itself.

ASP.NET is an open source, server-side web application framework (Wiseley, 2018) which allows developers to easily create dynamic full stack (including both front-end and back-end (Lexico, n.d.)) websites and apps using languages they may already be familiar with – C#, Visual Basic and F# - by making use of the .NET Framework. Its first release was in 2002 (Wikipedia, n.d.) and has been continually improved since then. However, its successor ASP.NET Core was released in 2016 and unifies various models of ASP.NET development including MVC (Model, View, Controller) and Web Pages, in addition to being able to utilise the new .NET Core runtime – which allows for cross-platform use on MacOS and Linux (Heddings, 2020) – as well as the original .NET Framework (Chowdhuri, 2016).

Although the main decisions for applications and frameworks were made by the Radiotherapy department, other options were still explored and analysed as hypotheticals. The three options that were compared were ASP.NET, the more contemporary ASP.NET, and another method of creating full stack web applications commonly called MERN after the libraries, frameworks, and applications that it uses (Mongo.DB, Express.JS, React.JS and Node.JS (Aggarwal & Verma, 2018)). Another L-shaped diagram was utilised, as it was found to be particularly helpful in the decision for Project Management applications.



		Features								
		SECURITY FEATURES INCLUDED	LIGHTWEIGHT	INTEGRATED UNIT TESTING?	EXPERIENCE WITH ALREADY?	PREVALENCE IN INDUSTRY	"CURRENT" TECHNOLOGY?	LANGUAGE (familiar with C# & JS)	Uses Specific IDE?	EASILY LINKED TO SQL DATABASE?
Stack	ASP.NET	Yes	No	Yes	No	Mid	No	C# / VB / F#	Yes (Visual Studio)	Yes
	ASP.NET CORE	Yes	No	Yes	No	High	Yes	C# / VB / F#	Yes (Visual Studio)	Yes
	REACT.JS & EXPRESS (MERN)	No	Yes	No	Yes	High	Yes	Javascript	No	No

Figure 14 - Another matrix diagram comparing features of prospective "stacks" to use

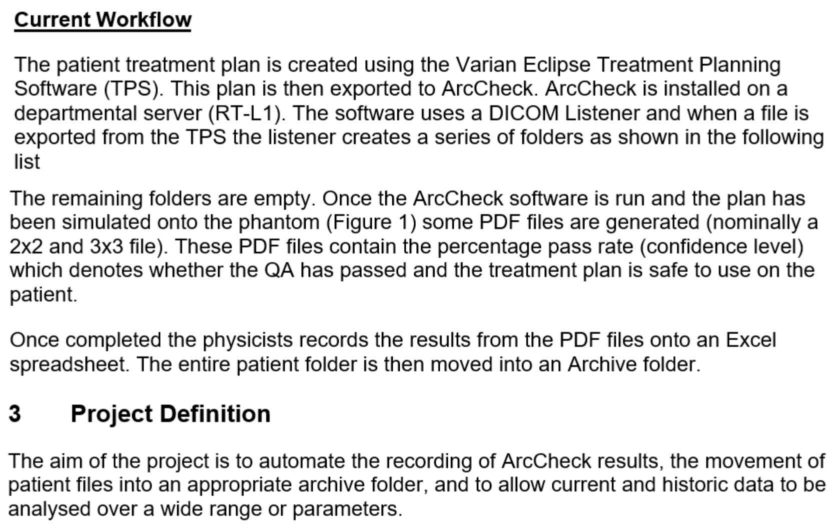
The main feature that would have been considered would likely be the addition of built-in security features to easily manage authentication and levels of access, which is included in ASP.NET and ASP.NET Core. For this reason, if the development tools hadn't been predetermined, ASP.NET Core would likely have been chosen as this contains all the features from the older ASP.NET, implemented in a more modern way.

Chapter 4

Design, Development and Evaluation

4.2 Requirements Elicitation, Collection & Analysis

The first step in planning to create this application was the development of the specific requirements. These were formed as a result of both video interviews and other communication with Tim Butler as discussed in previous sections. The current process involves writing key scan information manually into a master Excel document so that it may be accumulated and analysed later on, and this application's main function would be to remove this arduous task by automating the extraction process, among other tasks.



Current Workflow

The patient treatment plan is created using the Varian Eclipse Treatment Planning Software (TPS). This plan is then exported to ArcCheck. ArcCheck is installed on a departmental server (RT-L1). The software uses a DICOM Listener and when a file is exported from the TPS the listener creates a series of folders as shown in the following list

The remaining folders are empty. Once the ArcCheck software is run and the plan has been simulated onto the phantom (Figure 1) some PDF files are generated (nominally a 2x2 and 3x3 file). These PDF files contain the percentage pass rate (confidence level) which denotes whether the QA has passed and the treatment plan is safe to use on the patient.

Once completed the physicists records the results from the PDF files onto an Excel spreadsheet. The entire patient folder is then moved into an Archive folder.

3 Project Definition

The aim of the project is to automate the recording of ArcCheck results, the movement of patient files into an appropriate archive folder, and to allow current and historic data to be analysed over a wide range or parameters.

Figure 15 - Excerpts from Project Brief (Butler, 2019)

From these communications a brief list of requirements was created (see Figure 5). These included the core functionality of being able to upload the results of ArcCheck verification scans to a database and the ability to view these data at a later date, in addition to other beneficial features such as the transferring of uploaded files and folders to a specific archive folder. It was also discussed that there was often a backlog of many previously completed scans that have yet to be added to the Excel document, and as such the uploading process should be streamlined and intuitive in order for the Radiotherapy team to archive scans as quickly as possible, while still remaining reliable and giving users the ability to preview data before uploading it.

These requirements were then consolidated further in a video interview with Tim Butler to ensure each was correct and nothing was excluded. Once these were confirmed, they were re-written and expanded upon as detailed in section 5b – Software Development (see Figure 8).

However, several other User Stories were added after additional discussions with Butler. These included the ability to set each users' access level, making the database auditable (logging all database actions, allowing administrators to keep track of what changes are being made and when (Microsoft, 2020)), and also search functionality for the user to be able to find specific results in the database.

During analysis of these User Stories, they were ranked in order of importance and it was found that deliverables such as allowing auditing were of a lot less importance than many other requirements. For this reason, their position in the Sprint order was lowered in comparison to the more core functionality such as allowing the user to upload and view scans.

4.2 Design

4.2.1 Database Schema Design

One of the main features which required designing was the schema to be used for the database. A schema defines how every table in the database is structured, from column names & types to its relationships with other tables (Database.Guide, 2016).

This was created by first analysing the information that would need to be stored inside the database itself. A table was provided (Butler, 2020) which detailed the elements that were required, in addition to notes on where this information could be extracted from and how it may be able to be acquired.

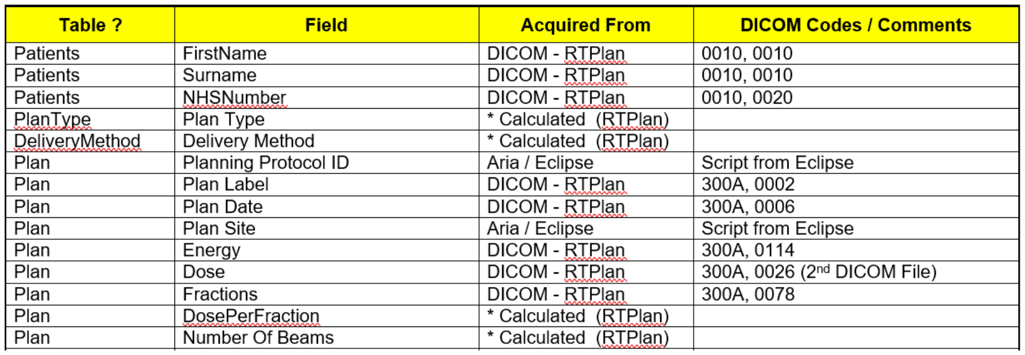


Table ?	Field	Acquired From	DICOM Codes / Comments
Patients	FirstName	DICOM - RTPlan	0010, 0010
Patients	Surname	DICOM - RTPlan	0010, 0010
Patients	NHSNumber	DICOM - RTPlan	0010, 0020
PlanType	Plan Type	* Calculated (RTPlan)	
DeliveryMethod	Delivery Method	* Calculated (RTPlan)	
Plan	Planning Protocol ID	Aria / Eclipse	Script from Eclipse
Plan	Plan Label	DICOM - RTPlan	300A, 0002
Plan	Plan Date	DICOM - RTPlan	300A, 0006
Plan	Plan Site	Aria / Eclipse	Script from Eclipse
Plan	Energy	DICOM - RTPlan	300A, 0114
Plan	Dose	DICOM - RTPlan	300A, 0026 (2nd DICOM File)
Plan	Fractions	DICOM - RTPlan	300A, 0078
Plan	DosePerFraction	* Calculated (RTPlan)	
Plan	Number Of Beams	* Calculated (RTPlan)	

Figure 16 - Examples of key data which would need to be extracted, and the source of each of these data

Some of these items were directly accessible, generally from the "RTPlan" DICOM file for each patient, however many others were classed as calculated fields. This meant there was likely some computation that would be required to determine their values – this will be discussed more specifically in 4.3 - Building and Coding.

From this list of key data, the SQL schema (a collection of tables that outline the "shape" of the data and share a namespace (Melton & Simon, 2002)) was developed, placing each column in a table, attempting to normalise as best as possible. Database normalisation involves organising each table in such a way that minimises redundant data storage, and all related data are stored together (Mendjoge, et al., 2016). This included separating out each patient's information into a dedicated table in order for the data not to be repeated if the same patient had multiple scan entries in the database.

In order to create a prototype of this schema more quickly, Microsoft Access (Microsoft, n.d.) was used to develop these tables and relations. This allowed for a rough version to be sent and feedback could be provided before fully developing an SQL command for this schema.

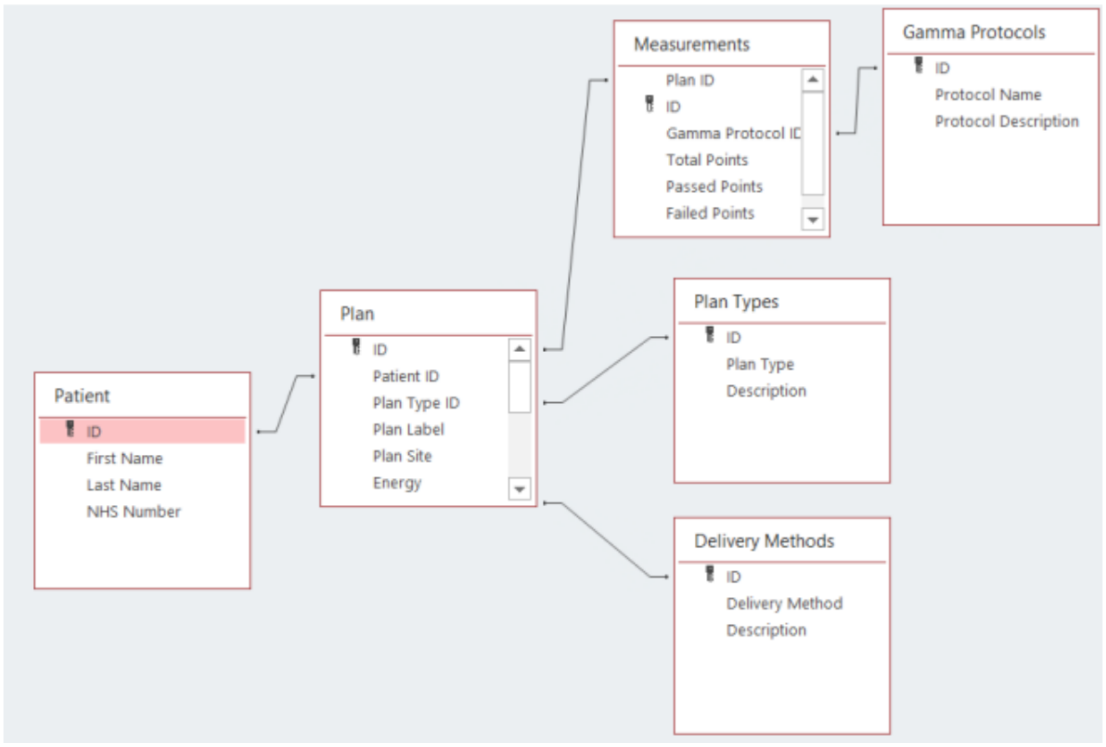


Figure 17 - Database schema developed in Microsoft Access

After developing this model and ensuring everything was correct, an SQL script was developed to create this database, and running this in SQL Server Management Studio (Microsoft, 2019) allowed for reliable and quick creation of this running on a locally hosted server on a personal computer. See Appendix I for the full SQL script. After a little more communication with Tim Butler, it was decided that several more key data should be stored in this database, including an additional table called ArcClinicalPlanField which contained information such as how many segments there were in each IMRT scan. A modified script was provided (Butler, 2020) – viewable in Appendix II.

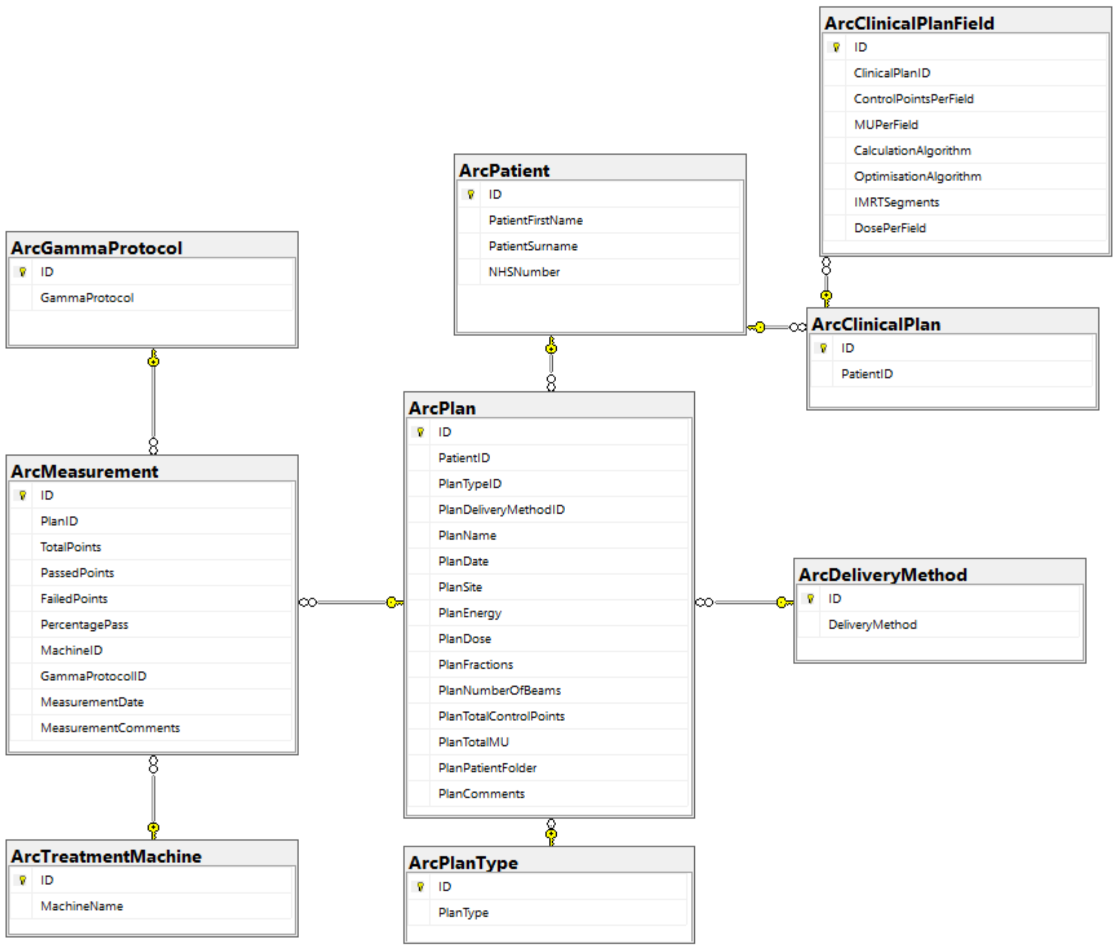
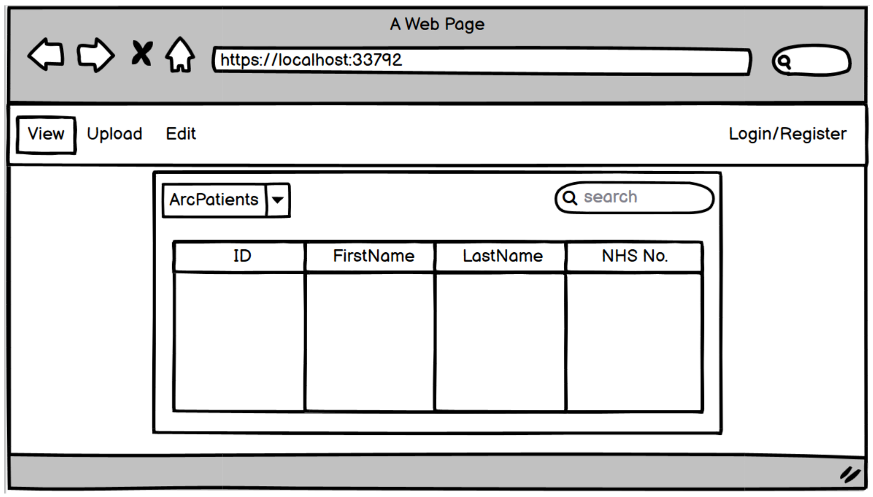
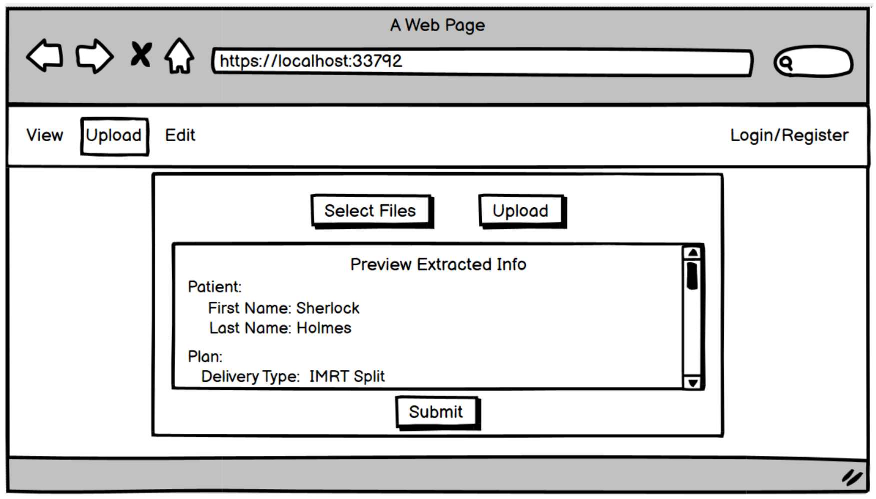


Figure 18 - Final SQL Server Database Diagram

4.2.2 Web Application Design

With regards to the planning and design of the web application itself, it was quickly decided that the main functionality of the site should be split into three separate pages – for uploading scan information, viewing stored data, and editing entries in the case of incorrect information. While A fourth page was required for logging the user into their account or registering to create a new account. These screens were roughly designed in low fidelity using mock-up software Balsamiq Wireframes (Balsamiq Studios, LLC, n.d.), shown in Figure 19.





Patient:
First Name: Sherlock
Last Name: Holmes
Plan:
Delivery Type: IMRT Split

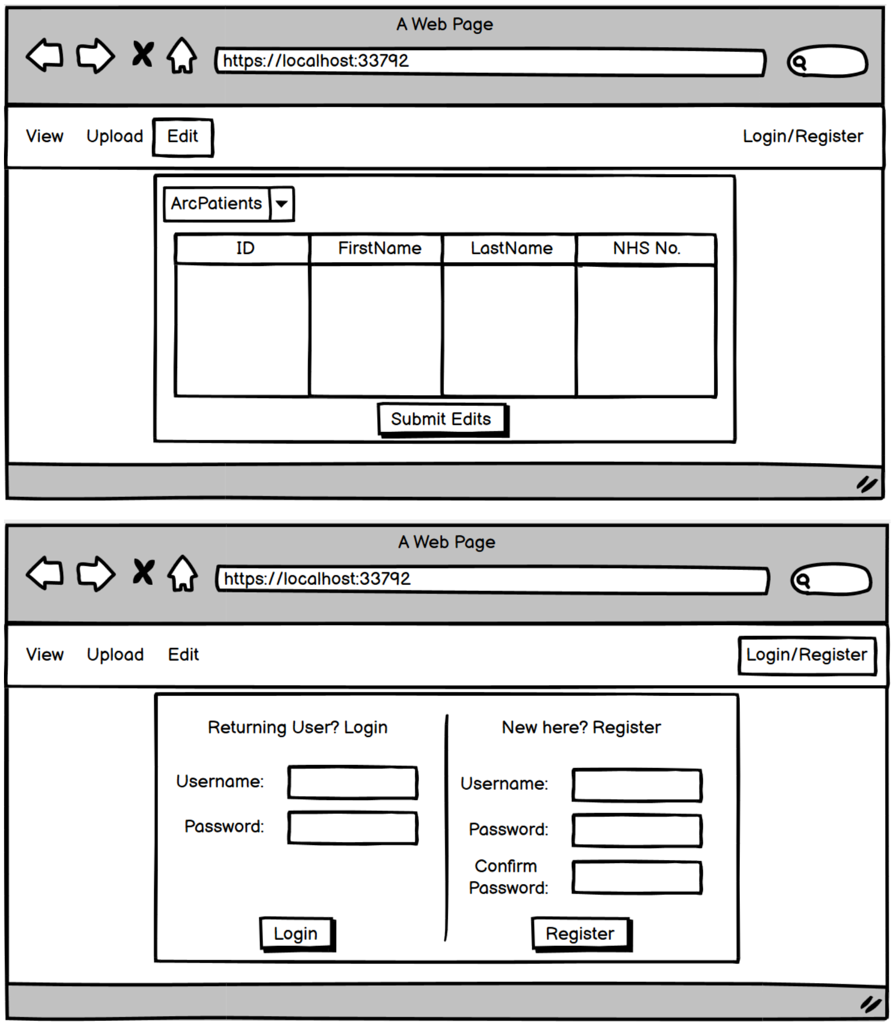


Figure 19 - Wireframes showing mock-ups of application's proposed screens

As shown in these rough sketches, there are additional features such as switching tables & searching through results on the View page and previewing uploaded information on the Upload page. This also features a navigation bar which allows the user to easily traverse through these pages – a common feature on many websites and applications. After further consideration, this navigation bar was edited to use universally recognised icons in order to declutter the area, and to allow for a site title to be added.

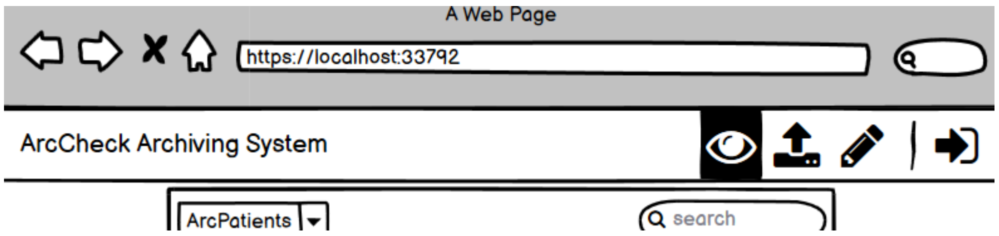


Figure 20 - Redesigned navigation bar

4.3 Building and Coding

4.3.1 ASP.NET Framework

Due to the unfamiliarity with the ASP.NET development stack, many processes, techniques, and systems were learnt about through the course of this module. The first feature was how ASP.NET integrates C# with web-based languages such as HTML. Usually, site markup is described using an HTML file, styled using CSS, and any effects or interactivity can be coded to run in the web browser using JavaScript.

However, when creating an ASP.NET application, the markup is written inside of a .aspx file. The code written inside of this is “translated” into relevant HTML tags and sent to the user when a request is made, but this process allows for the implementation of prebuilt components such as a login element that handles the user’s data in a secure fashion, or a dynamically changing tag that displays the last time the user logged in.

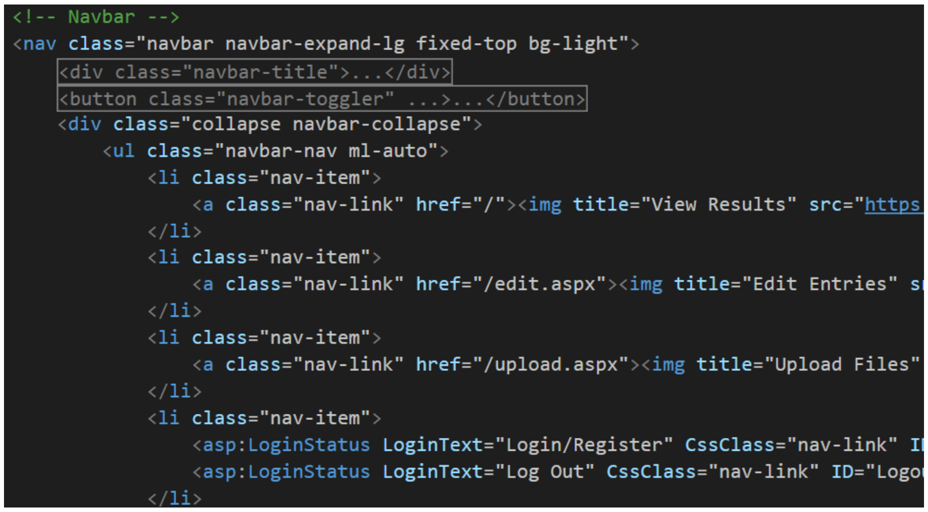
In addition, the ‘ASPX’ files also allow for the usage of “Master” pages. These are wrappers around other pages and allow for easy code reuse due to the ability to write markup for core features such as navigation bars, headers, and footers, along with using embedded code blocks (a way of writing C# code directly inside the ASPX file) to use information such as each page’s title and more.

Besides the use of embedded code blocks, the majority of the server-side C# code is usually stored in “code-behind” files that use an “.aspx.cs” file extension. These are linked to the more front-end focussed pages and can be used to write code that runs either when the page is loaded or based on the user’s interaction with the page and have the ability to edit and read any “server-side” elements in the ASPX page (denoted by the ‘runat=”server”’ property).

ASP.NET also makes use of a ‘web.config’ file – an XML-based settings file – to store information about the application itself. This includes saving database connection information as well as configuring which pages are accessible to authenticated and unauthenticated users, and where to redirect the user in case they visit a page that they’re unable to access. Another standard file included is ‘Global.asax’. As is the case with web.config, only one of these exists per application and consists of several functions which run at specific occasions during the application’s runtime, such as at start-up, whenever a request is sent to the server, or when a user starts a new session.

4.3.2 Project Development – General

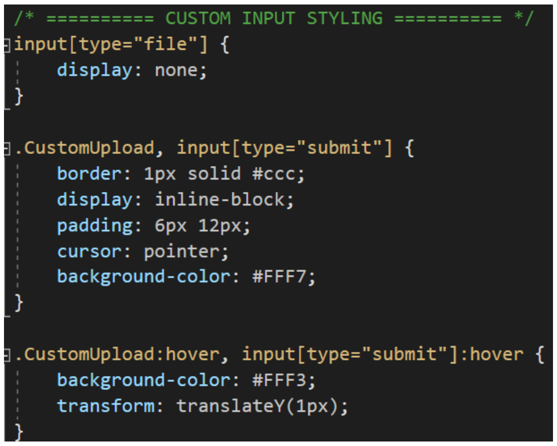
Prior to development it was decided that Bootstrap would be used to allow for a quicker method of creating common site features such as the navigation bar. Bootstrap is a “framework for building responsive, mobile-first sites” (Bootstrap, 2021) which allows classes to be used to apply a variety of CSS styles to HTML elements easily and intuitively. These classes are able to be overwritten in other CSS files, which allows for customisation from the default Bootstrap look and feel.



```
<!-- Navbar -->
<nav class="navbar navbar-expand-lg fixed-top bg-light">
    <div class="navbar-title">...</div>
    <button class="navbar-toggler" ...>...</button>
    <div class="collapse navbar-collapse">
        <ul class="navbar-nav ml-auto">
            <li class="nav-item">
                <a class="nav-link" href="/"><img title="View Results" src="https
            </li>
            <li class="nav-item">
                <a class="nav-link" href="/edit.aspx"><img title="Edit Entries" s
            </li>
            <li class="nav-item">
                <a class="nav-link" href="/upload.aspx"><img title="Upload Files"
            </li>
            <li class="nav-item">
                <asp:LoginStatus LoginText="Login/Register" CssClass="nav-link" I
                <asp:LoginStatus LoginText="Log Out" CssClass="nav-link" ID="Logo
            </li>
```

Figure 21 - Screenshot showing navbar code using Bootstrap classes for styling

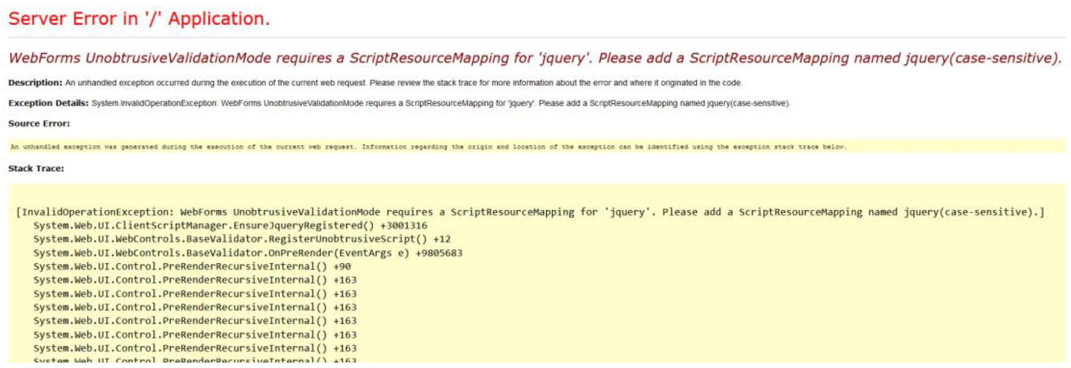
CSS was used to create these customisations, in addition to styling elements without using Bootstrap. One of the main uses was to create a more fitting File Upload element than the default provided by HTML for the upload page. This essentially involved hiding the upload button and using a label to create a much more fitting button (Ibrahim, 2020).



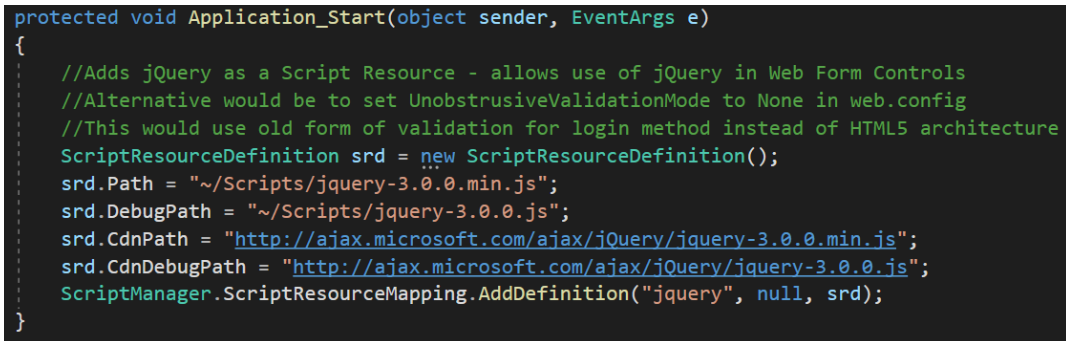
```
/* ========== CUSTOM INPUT STYLING ========== */
input[type="file"] {
    display: none;
}

.CustomUpload, input[type="submit"] {
    border: 1px solid #ccc;
    display: inline-block;
    padding: 6px 12px;
    cursor: pointer;
    background-color: #FFF7;
}

.CustomUpload:hover, input[type="submit"]:hover {
    background-color: #FFF3;
    transform: translateY(1px);
}
```

Figure 22 - CSS used to create custom file upload button

By default, the prebuilt login & register components (discussed in Login section below) that ASP.NET provides utilise validating the user's input (such as checking for a correct email address and matching passwords) before sending the request back to the server. This functionality adds a more streamlined system to the login elements, however as the code used to implement this uses jQuery (a JavaScript library designed to simplify page interactions (jQuery, n.d.)) functions, an error would occur unless this library was linked to the application. The solution for this was to make use of the "Application_Start()" function in the Global.asax file discussed previously, which would link the jQuery library to the application every time it initialised.



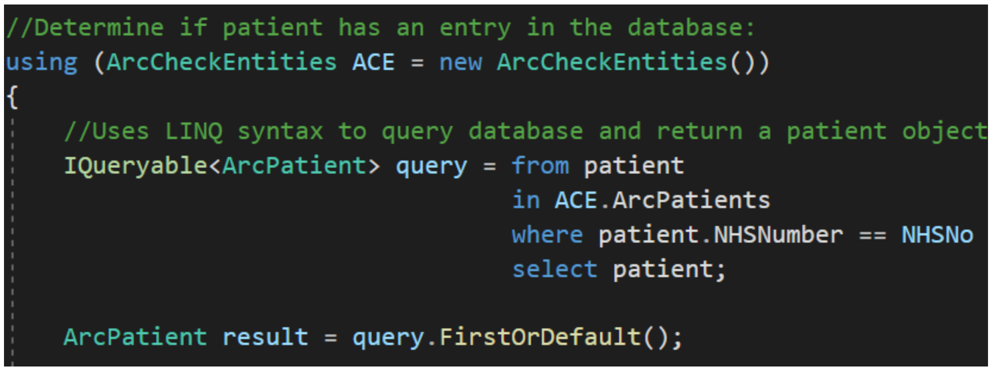
```
Server Error in '/' Application.

WebForms UnobtrusiveValidationMode requires a ScriptResourceMapping for 'jquery'. Please add a ScriptResourceMapping named jquery(case-sensitive).

Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

Exception Details: System.InvalidOperationException: WebForms UnobtrusiveValidationMode requires a ScriptResourceMapping for 'jquery'. Please add a ScriptResourceMapping named jquery(case-sensitive).

Source Error:

An unhandled exception was generated during the execution of the current web request. Information regarding the origin and location of the exception can be identified using the exception stack trace below.

Stack Trace:

[InvalidOperationException: WebForms UnobtrusiveValidationMode requires a ScriptResourceMapping for 'jquery'. Please add a ScriptResourceMapping named jquery(case-sensitive).]
   System.Web.UI.ClientScriptManager.EnsureJqueryRegistered() +3001316
   System.Web.UI.WebControls.BaseValidator.RegisterUnobtrusiveScript() +12
   System.Web.UI.WebControls.BaseValidator.OnPreRender(EventArgs e) +9805683
   System.Web.UI.Control.PreRenderRecursiveInternal() +90
   System.Web.UI.Control.PreRenderRecursiveInternal() +163
   System.Web.UI.Control.PreRenderRecursiveInternal() +163
   System.Web.UI.Control.PreRenderRecursiveInternal() +163
   System.Web.UI.Control.PreRenderRecursiveInternal() +163
   System.Web.UI.Control.PreRenderRecursiveInternal() +163
   System.Web.UI.Control.PreRenderRecursiveInternal() +163
   System.Web.UI.Control.PreRenderRecursiveInternal() +163
```



```
protected void Application_Start(object sender, EventArgs e)
{
    //Adds jQuery as a Script Resource - allows use of jQuery in Web Form Controls
    //Alternative would be to set UnobstrusiveValidationMode to None in web.config
    //This would use old form of validation for login method instead of HTML5 architecture
    ScriptResourceDefinition srd = new ScriptResourceDefinition();
    srd.Path = "~/Scripts/jquery-3.0.0.min.js";
    srd.DebugPath = "~/Scripts/jquery-3.0.0.js";
    srd.CdnPath = "http://ajax.microsoft.com/ajax/jQuery/jquery-3.0.0.min.js";
    srd.CdnDebugPath = "http://ajax.microsoft.com/ajax/jQuery/jquery-3.0.0.js";
    ScriptManager.ScriptResourceMapping.AddDefinition("jquery", null, srd);
}
```

Figure 23 - Error due to not linking jQuery to application & solution involving adding a reference to the jQuery library

Another feature included in ASP.NET is the ability to utilise the Entity Framework to perform database operations. This framework is an Object-Relational Mapping (ORM) tool, based on another data access mechanism called ADO.NET, which allows the creation of C# classes & objects which relate to the tables in an SQL database. This provides high-level methods of easily and intuitively performing queries and commands using functions such as Add() and FirstOrDefault() to insert & read from a database, respectively. This framework also enables the use of LINQ (Language-Integrated Query (Microsoft, 2017)) syntax, a method of querying a database while using C# variables directly. These two technologies combined (often referred to as LINQ to Entities) provide inherent protection against SQL injection attacks described in 4 - Background by bypassing the security risks of string concatenation and manipulation (Microsoft, 2017).



```
//Determine if patient has an entry in the database:
using (ArcCheckEntities ACE = new ArcCheckEntities())
{
    //Uses LINQ syntax to query database and return a patient object
    IQueryable<ArcPatient> query = from patient
                                   in ACE.ArcPatients
                                   where patient.NHSNumber == NHSNo
                                   select patient;

    ArcPatient result = query.FirstOrDefault();
```

Figure 24 - Example of Entity Framework and LINQ to Entities syntax to select patient from database based on NHS Number

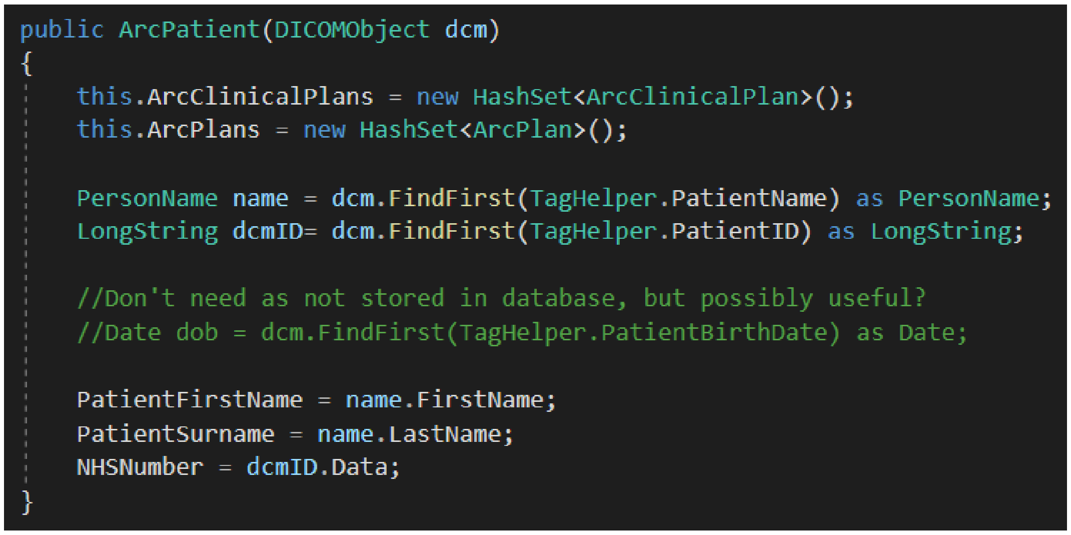
4.3.3 Project Development – Web Pages

4.3.3.1 Upload

The main web page created for this application was the upload section. This is where the user will be able to submit files to be added to the database, which includes extracting the relevant information and storing this. Previously described was how the custom file upload button was designed, however this was only a small part of this page’s functionality. Once the user selects a DICOM file and submits it to the site, it is saved locally to the server and read into a custom C# object via EvilDICOM (Cardan, Rex,

2020) – an open-source library specifically designed for reading & editing DICOM files.

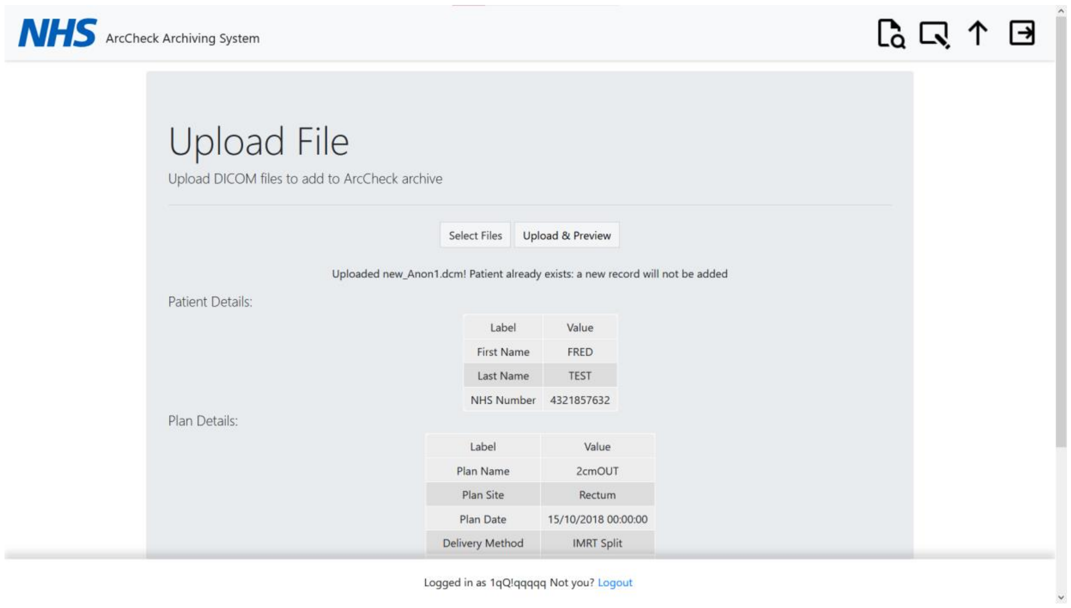
This object is passed into two functions – ‘GetPatient()’ and ‘GetPlan()’. The former starts by extracting the patient’s NHS Number using the PatientID tag (0010, 0020 (DICOM Library, n.d.)). The TagHelper class from EvilDICOM is used as shorthand for the full DICOM tag for readability. Once this has been obtained, a LINQ to Entities query is created as shown in Figure 25**Error! Reference source not found.**, which checks for any entries that already exist for this patient: if a patient with the same NHS Number already exists in the database, naturally a new patient record should not be added and the null-coalescing operator is used to return the result of this query.



```
public ArcPatient(DICOMObject dcm)
{
    this.ArcClinicalPlans = new HashSet<ArcClinicalPlan>();
    this.ArcPlans = new HashSet<ArcPlan>();

    PersonName name = dcm.FindFirst(TagHelper.PatientName) as PersonName;
    LongString dcmID= dcm.FindFirst(TagHelper.PatientID) as LongString;

    //Don't need as not stored in database, but possibly useful?
    //Date dob = dcm.FindFirst(TagHelper.PatientBirthDate) as Date;

    PatientFirstName = name.FirstName;
    PatientSurname = name.LastName;
    NHSNumber = dcmID.Data;
}
```

Figure 25 - New object constructor which handles extraction of DICOM tags

However, if an entry does not exist, a new ArcPatient object (using Entity Framework, classes for each table are created) is initialised using a custom constructor. Inside this, all other tags are extracted such as the patient’s name. This new object is then returned, a Session storage variable is set containing the Patient object (this allows storing this object over the course of multiple requests to the server), the web page preview fields are updated with relevant information and shown to the user.



Upload File

Upload DICOM files to add to ArcCheck archive

Uploaded new_Anon1.dcm! Patient already exists: a new record will not be added

Patient Details:

Label	Value
First Name	FRED
Last Name	TEST
NHS Number	4321857632

Plan Details:

Label	Value
Plan Name	2cmOUT
Plan Site	Rectum
Plan Date	15/10/2018 00:00:00
Delivery Method	IMRT Split

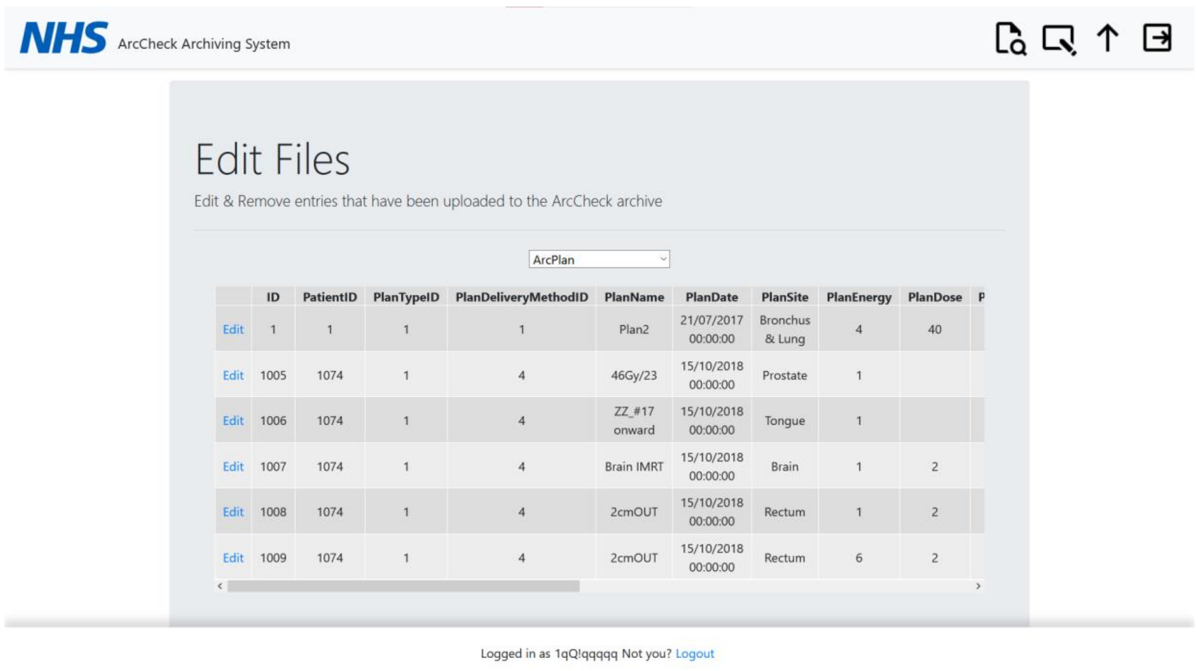
Logged in as 1qQ!qqqqq Not you? Logout

Figure 26 - Screenshot of web application after uploading of DICOM file

The GetPlan() function works in a similar way, using a constructor in the Plan class to create a new ArcPlan object. However, the tag extraction in this constructor is much more complex: using some sample code provided (Butler, 2020), full methods were created to extract values such as the total plan dosage, number of Control Points, and whether the scan was IMRT or VMAT as well as which variation of these it was. The full code is shown in Appendix III, or in ArcPlan.cs file in submission. To quickly test the logic and various methods of extracting values a Python script was created, called “PythonTagExtraction.py” (See Appendix IV). Though the mechanisms for using EvilDICOM differed from Python’s Pydicom (Pydicom, 2020) library, the rapid development a lightweight Python script provided allowed for learning roughly how to collect many tags in an efficient way.

4.3.3.2 Edit & View

Both the Edit page and View page were similar in terms of their functionality, however with respect to their backend implementation they differed significantly.



Edit Files

Edit & Remove entries that have been uploaded to the ArcCheck archive

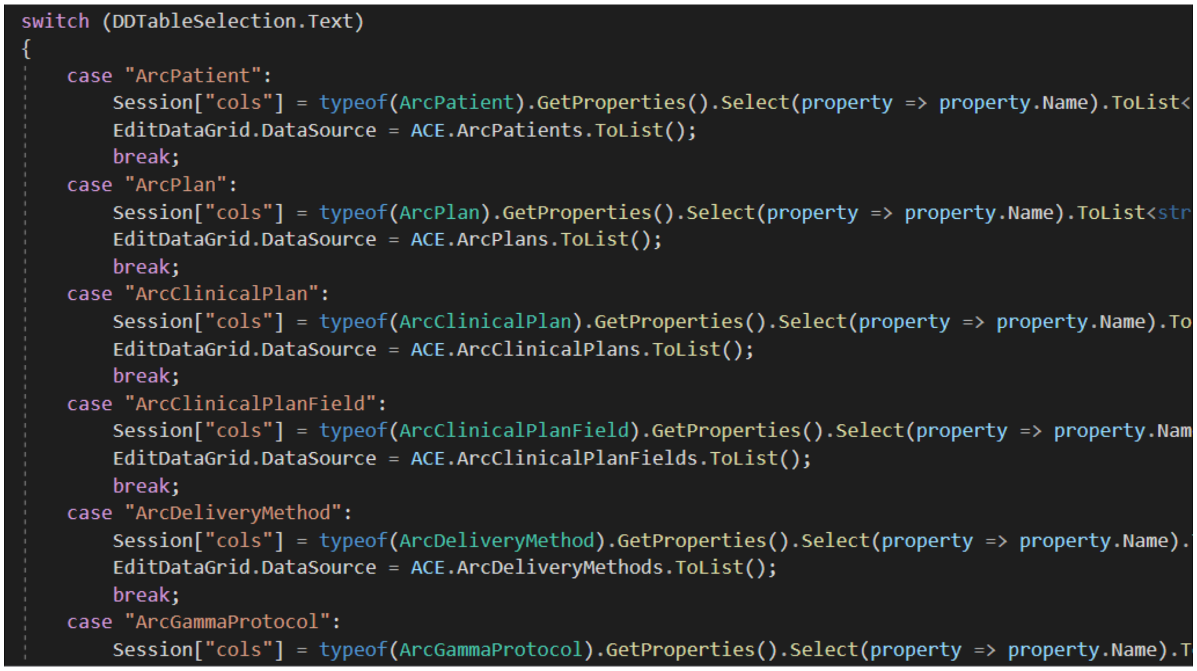
ArcPlan

	ID	PatientID	PlanTypeID	PlanDeliveryMethodID	PlanName	PlanDate	PlanSite	PlanEnergy	PlanDose	P
Edit	1	1	1	1	Plan2	21/07/2017 00:00:00	Bronchus & Lung	4	40	
Edit	1005	1074	1	4	46Gy/23	15/10/2018 00:00:00	Prostate	1		
Edit	1006	1074	1	4	ZZ_#17 onward	15/10/2018 00:00:00	Tongue	1		
Edit	1007	1074	1	4	Brain IMRT	15/10/2018 00:00:00	Brain	1	2	
Edit	1008	1074	1	4	2cmOUT	15/10/2018 00:00:00	Rectum	1	2	
Edit	1009	1074	1	4	2cmOUT	15/10/2018 00:00:00	Rectum	6	2	

Logged in as 1qQ!qqqqq Not you? Logout

Figure 27 - Edit page of the application, with ArcPlan table selected

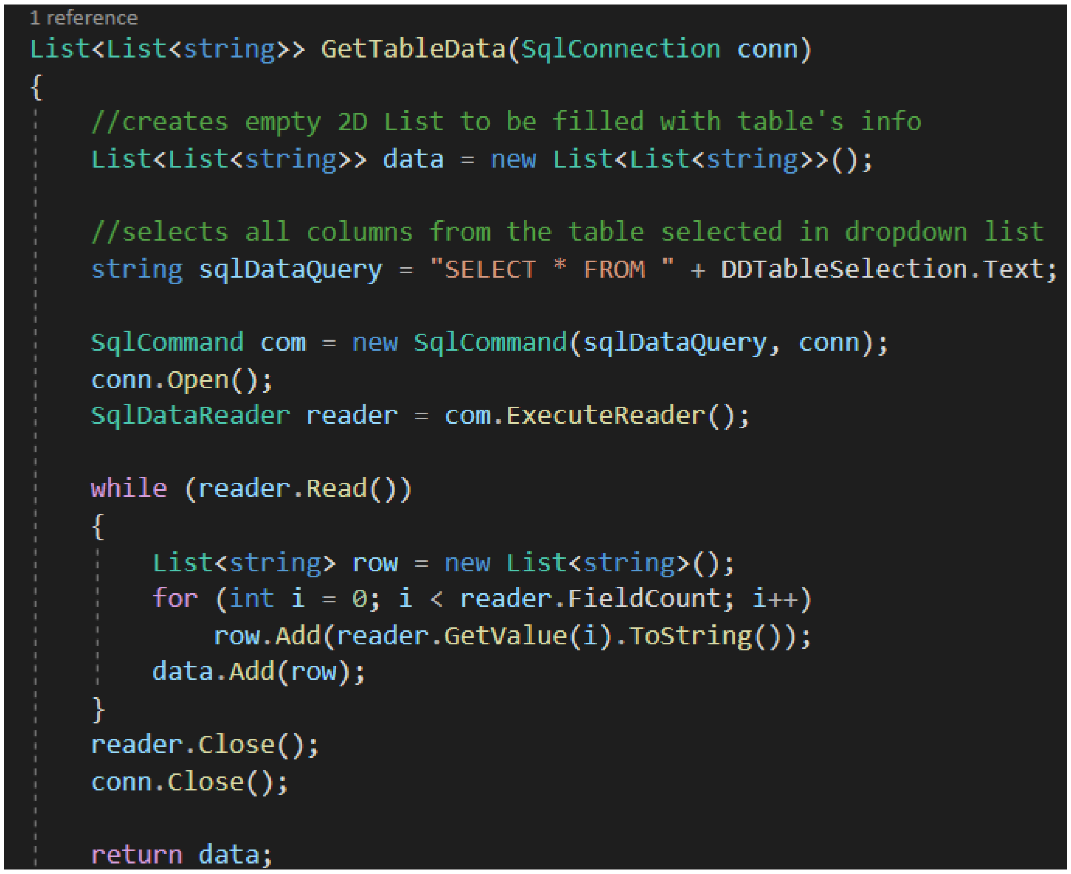
The first step in building both pages was to implement a DataGrid – a method of displaying data which allows abstracted connection to an SQL database using either the Entity Framework or more low-level methods. Initially when creating the Edit page, Entity Framework was used to select data from the relevant tables. However, this presented an issue when adding a dropdown box to select which table the user wanted to view data from, since converting the string value to a C# object isn't a particularly good practise. This resulted in the switch statement shown in Figure 28, and was the main reason that a lower level approach was considered.



```
switch (DDTableSelection.Text)
{
    case "ArcPatient":
        Session["cols"] = typeof(ArcPatient).GetProperties().Select(property => property.Name).ToList<
        EditDataGrid.DataSource = ACE.ArcPatients.ToList();
        break;
    case "ArcPlan":
        Session["cols"] = typeof(ArcPlan).GetProperties().Select(property => property.Name).ToList<str
        EditDataGrid.DataSource = ACE.ArcPlans.ToList();
        break;
    case "ArcClinicalPlan":
        Session["cols"] = typeof(ArcClinicalPlan).GetProperties().Select(property => property.Name).To
        EditDataGrid.DataSource = ACE.ArcClinicalPlans.ToList();
        break;
    case "ArcClinicalPlanField":
        Session["cols"] = typeof(ArcClinicalPlanField).GetProperties().Select(property => property.Nam
        EditDataGrid.DataSource = ACE.ArcClinicalPlanFields.ToList();
        break;
    case "ArcDeliveryMethod":
        Session["cols"] = typeof(ArcDeliveryMethod).GetProperties().Select(property => property.Name).
        EditDataGrid.DataSource = ACE.ArcDeliveryMethods.ToList();
        break;
    case "ArcGammaProtocol":
        Session["cols"] = typeof(ArcGammaProtocol).GetProperties().Select(property => property.Name).T
```

Figure 28 - Particularly unpleasant switch statement setting the data source for the Edit page's DataGrid. Also includes setting Session storage variable of all column names for use in submitting changes.

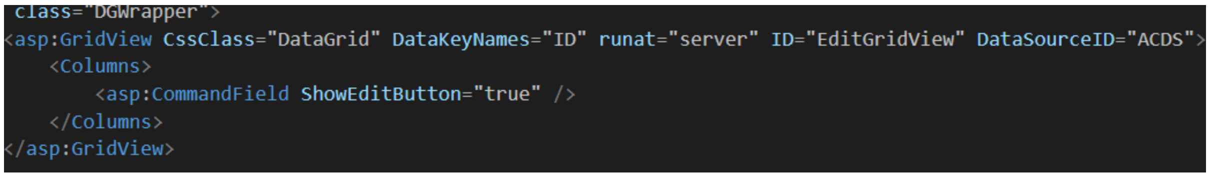
The second revision of the edit page involved utilising ADO.NET classes to manually create an SQL query to retrieve all table data. This involves three main objects: SqlConnection, which opens a connection to a database using a given Connection String; SqlCommand, which takes an SQL query to execute; and SqlDataReader, which acts as a temporary storage for the data returned by the SqlCommand execution (MacDonald, 2012).



```
1 reference
List<List<string>> GetTableData(SqlConnection conn)
{
    //creates empty 2D List to be filled with table's info
    List<List<string>> data = new List<List<string>>();

    //selects all columns from the table selected in dropdown list
    string sqlDataQuery = "SELECT * FROM " + DDTableSelection.Text;

    SqlCommand com = new SqlCommand(sqlDataQuery, conn);
    conn.Open();
    SqlDataReader reader = com.ExecuteReader();

    while (reader.Read())
    {
        List<string> row = new List<string>();
        for (int i = 0; i < reader.FieldCount; i++)
            row.Add(reader.GetValue(i).ToString());
        data.Add(row);
    }
    reader.Close();
    conn.Close();

    return data;
```

Figure 29 - Function that returns all elements from currently selected table, using ADO.NET data objects.

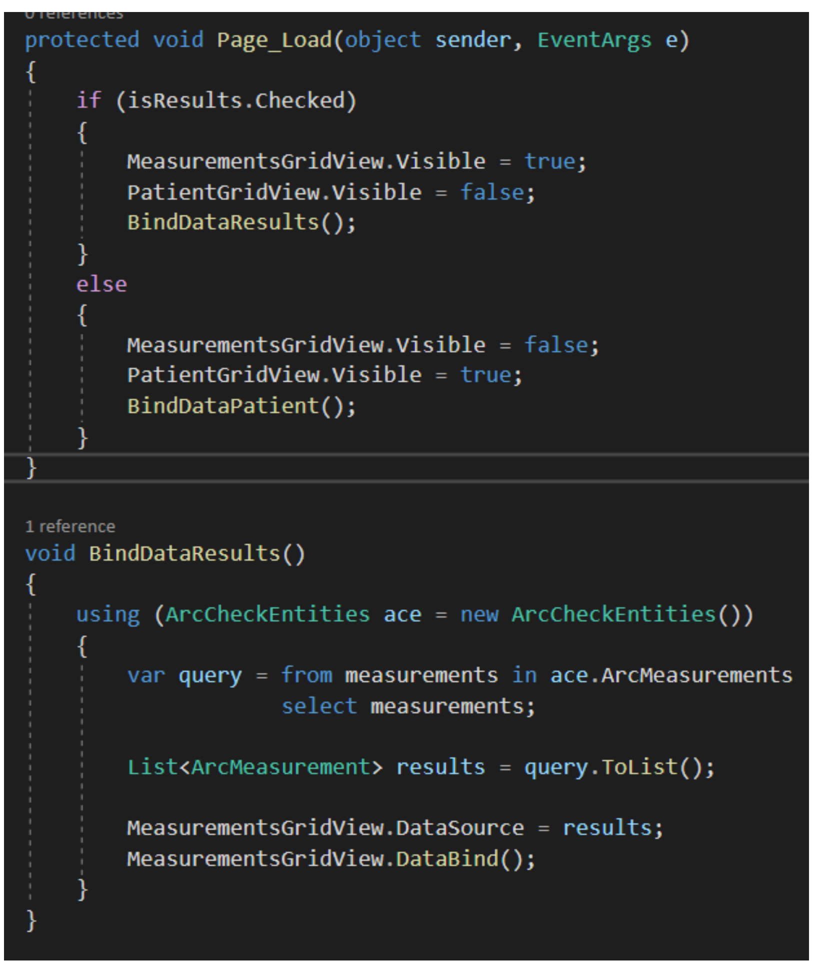
However, when creating this, an alternative method of displaying data in ASP.NET was found, namely a GridView (Năchilă, 2008). This is considered a more contemporary & robust form of tabulating data, and previous methods were adapted to utilise this.



```
 class="DGWrapper">
<asp:GridView CssClass="DataGrid" DataKeyNames="ID" runat="server" ID="EditGridView" DataSourceID="ACDS">
    <Columns>
        <asp:CommandField ShowEditButton="true" />
    </Columns>
</asp:GridView>
```

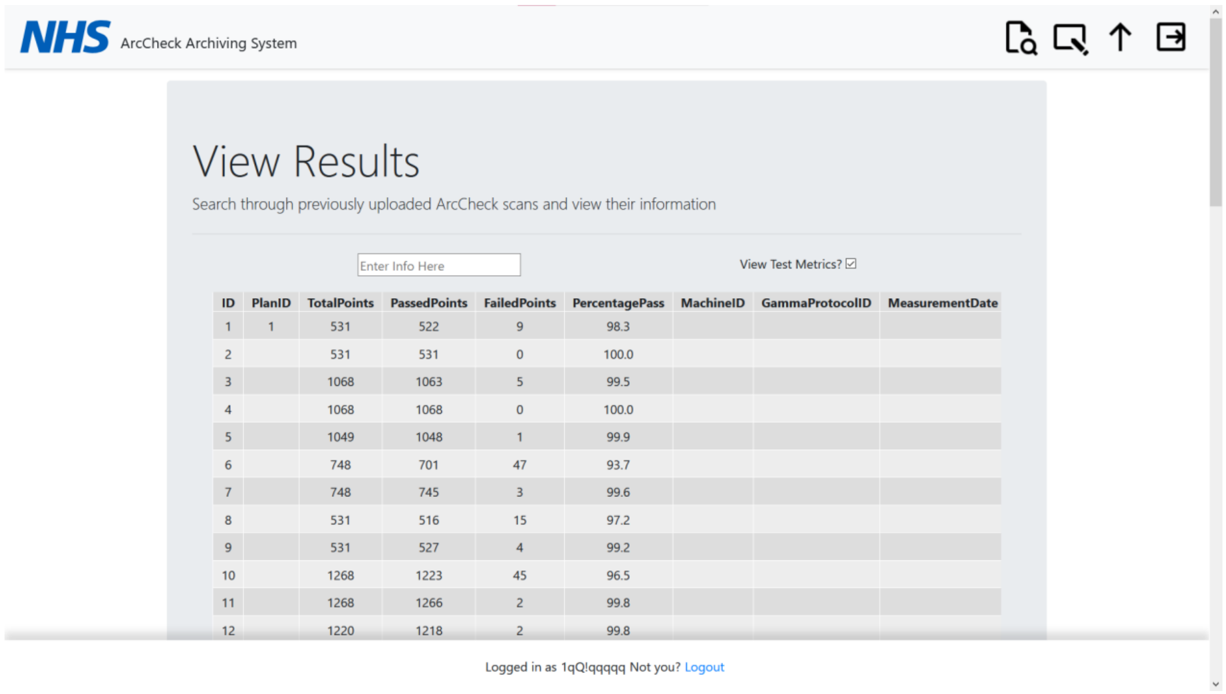
Figure 30 - Front-end code for more modern DataGrid

This GridView was also applied to the View page, though this used two separate GridView elements and a checkbox to switch between them.



```
0 references
protected void Page_Load(object sender, EventArgs e)
{
    if (isResults.Checked)
    {
        MeasurementsGridView.Visible = true;
        PatientGridView.Visible = false;
        BindDataResults();
    }
    else
    {
        MeasurementsGridView.Visible = false;
        PatientGridView.Visible = true;
        BindDataPatient();
    }
}

1 reference
void BindDataResults()
{
    using (ArcCheckEntities ace = new ArcCheckEntities())
    {
        var query = from measurements in ace.ArcMeasurements
                    select measurements;

        List<ArcMeasurement> results = query.ToList();

        MeasurementsGridView.DataSource = results;
        MeasurementsGridView.DataBind();
    }
}
```

Figure 31 - Backend code showing the logic to retrieve data using LINQ to Entities query


NHS ArcCheck Archiving System

View Results

Search through previously uploaded ArcCheck scans and view their information

Enter Info Here

View Test Metrics? ☑

ID	PlanID	TotalPoints	PassedPoints	FailedPoints	PercentagePass	MachineID	GammaProtocolID	MeasurementDate
1	1	531	522	9	98.3			
2		531	531	0	100.0			
3		1068	1063	5	99.5			
4		1068	1068	0	100.0			
5		1049	1048	1	99.9			
6		748	701	47	93.7			
7		748	745	3	99.6			
8		531	516	15	97.2			
9		531	527	4	99.2			
10		1268	1223	45	96.5			
11		1268	1266	2	99.8			
12		1220	1218	2	99.8			

Logged in as 1qQ!qqqqq Not you? Logout

Figure 32 - View page of application

4.3.3.3 Login & Register

These pages were arguably the most important to ensure that the site was secure. In the initial lo-fi design of each page, both of these components were intended to fit on one screen. However, it was decided that it would be more intuitive to include separate screens for each.

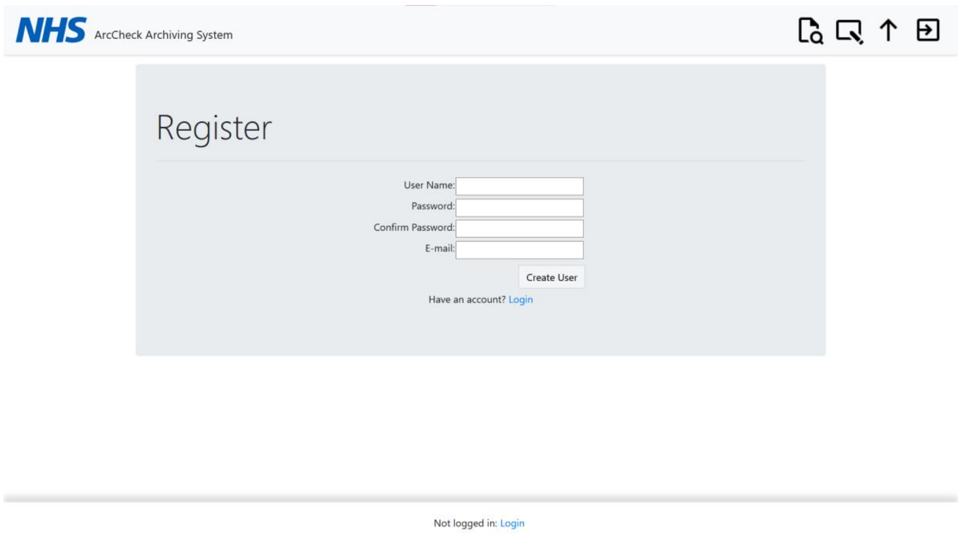
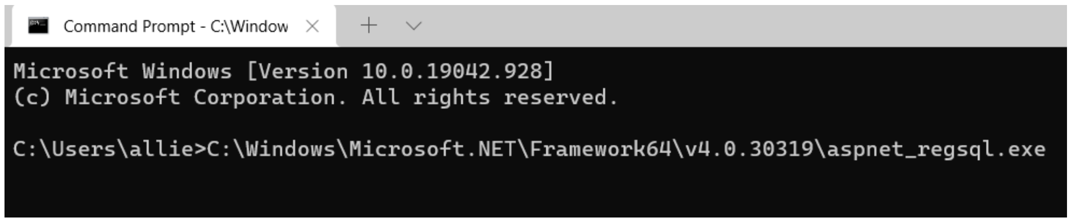


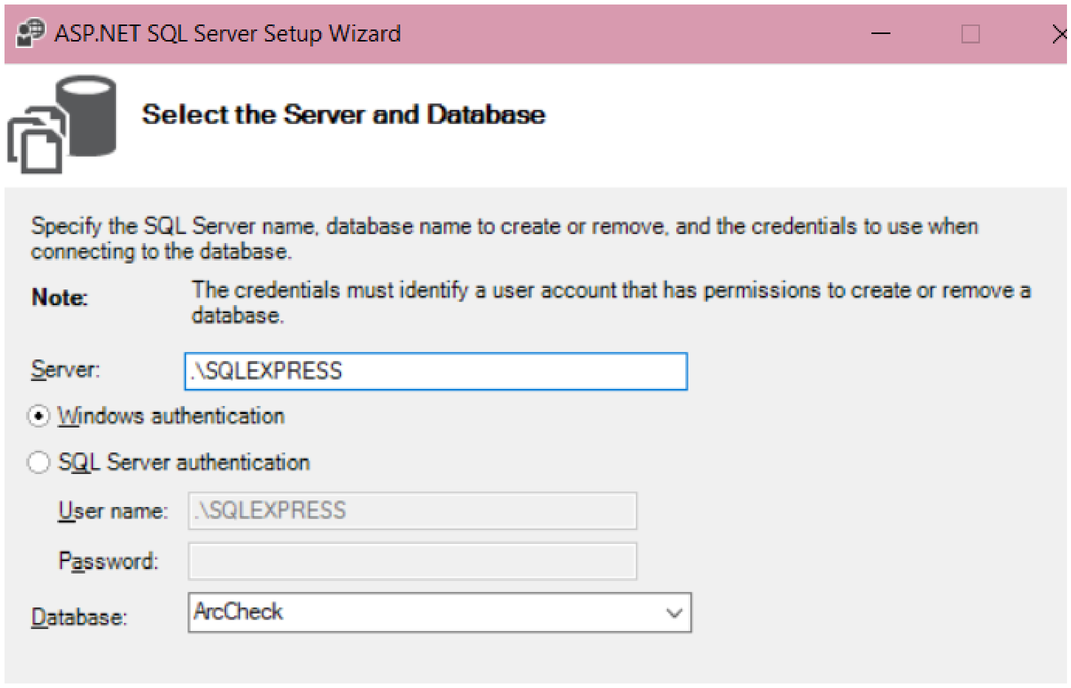
Figure 33 - Register page of application

Creating the login system firstly involved setting up ASP.NET's membership features on the database (MacDonald, 2012). This required running the "aspnet_regsql.exe" file included in the most recent .NET framework folder and runs a graphical setup wizard to connect to a given server & database. This then creates a number of tables, to store data such as account information in 'aspnet_membership', User Roles in the 'aspnet_roles', and so on.



```
Microsoft Windows [Version 10.0.19042.928]
(c) Microsoft Corporation. All rights reserved.

C:\Users\allie>C:\Windows\Microsoft.NET\Framework64\v4.0.30319\aspnet_regsql.exe
```

Figure 34 - Executing command to create membership tables in database


ASP.NET SQL Server Setup Wizard

Select the Server and Database

Specify the SQL Server name, database name to create or remove, and the credentials to use when connecting to the database.

Note: The credentials must identify a user account that has permissions to create or remove a database.

Server: .\SQLEXPRESS

(● Windows authentication)

○ SQL Server authentication

User name: .\SQLEXPRESS

Password:

Database: ArcCheck

Figure 35 - Interface to select which server & database to add membership to

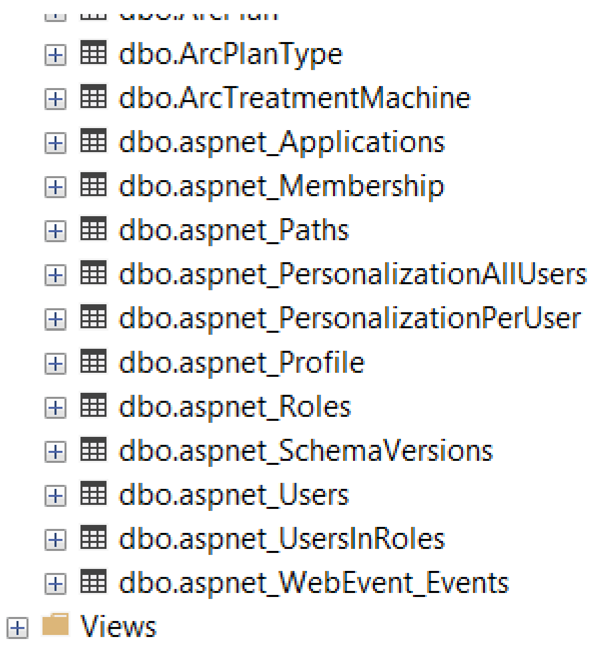
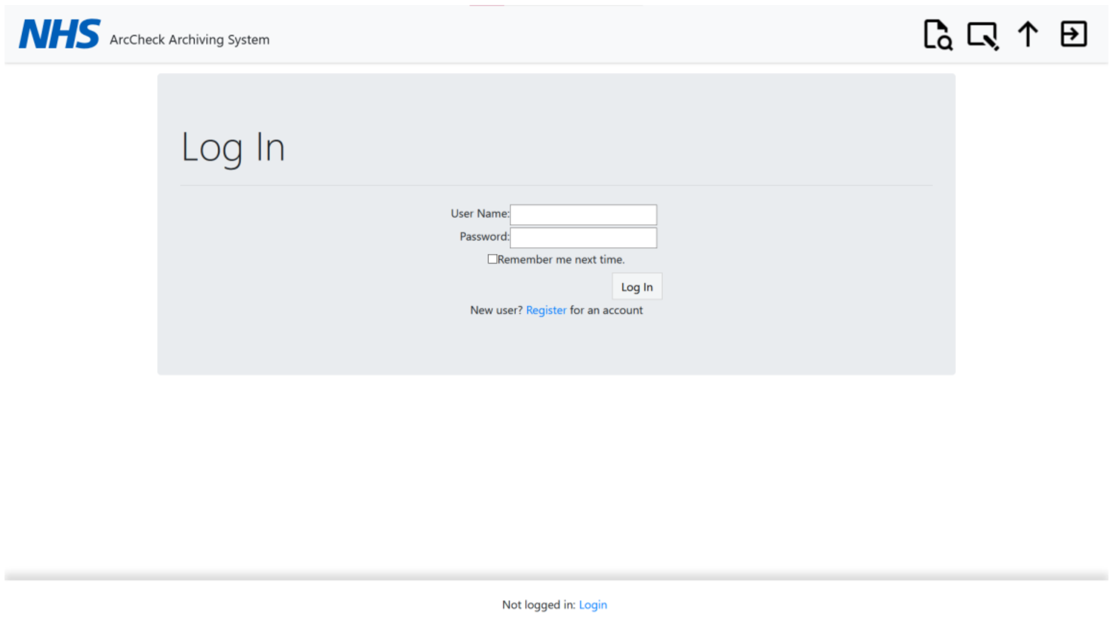

dbo.ArcPlanType
dbo.ArcTreatmentMachine
dbo.aspnet_Applications
dbo.aspnet_Membership
dbo.aspnet_Paths
dbo.aspnet_PersonalizationAllUsers
dbo.aspnet_PersonalizationPerUser
dbo.aspnet_Profile
dbo.aspnet_Roles
dbo.aspnet_SchemaVersions
dbo.aspnet_Users
dbo.aspnet_UsersInRoles
dbo.aspnet_WebEvent_Events
Views

Figure 36 - All generated tables are prefixed with "aspnet_"

Running this executable allows the database to be referenced in the application's web.config file as a membership provider, along with certain information about what is required from the user when registering, whether to hash their password and more. This membership provider is then linked with the main ASP.NET Login component and is ready to be used on the page.



Log In

User Name:
Password:
☐Remember me next time.
Log In
New user? Register for an account

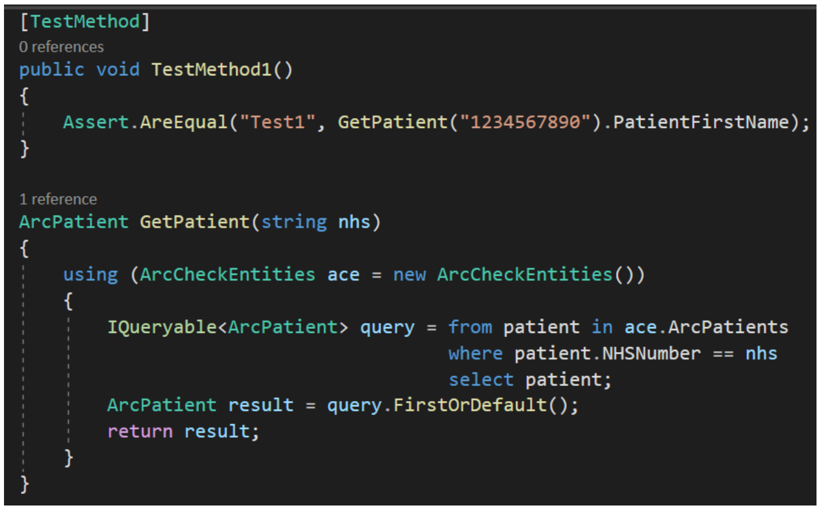
Not logged in: Login

Figure 37 - Login page of application

4.4 Testing

While developing this application, Visual Studio 2019's in-built web server hosting system IIS Express (Microsoft, 2017) was used. This is a variation on the original IIS (Internet Information Services (Microsoft, n.d.)) which allows web applications to easily run on Windows Server machines, and even regular Windows PCs. This system allowed for quick testing of features and code, as with any client-side changes (such as HTML tags and styling edits) the webpage simply had to be reloaded, and with server-side modifications the web server could be restarted at the click of a button right from Visual Studio's IDE.

Unit testing was also experimented with during the development process – while not fully implemented, unit tests were written for querying the database and returning Patient and Plan information. Although not entirely fool proof, these tests are a great way to easily minimise the risk that there's an issue with any given code.

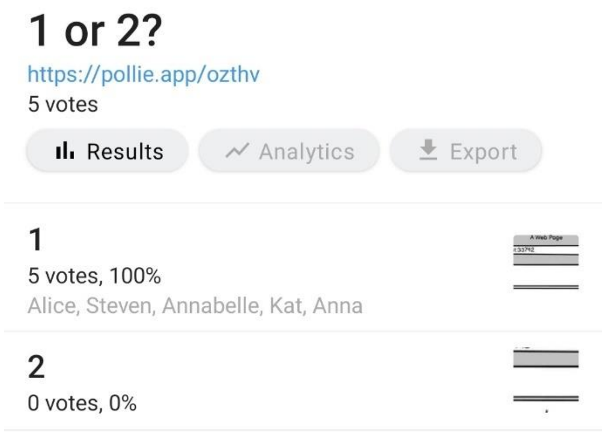


```
[TestMethod]
0 references
public void TestMethod1()
{
    Assert.AreEqual("Test1", GetPatient("1234567890").PatientFirstName);
}

1 reference
ArcPatient GetPatient(string nhs)
{
    using (ArcCheckEntities ace = new ArcCheckEntities())
    {
        IQueryable<ArcPatient> query = from patient in ace.ArcPatients
                                       where patient.NHSNumber == nhs
                                       select patient;
        ArcPatient result = query.FirstOrDefault();
        return result;
    }
}
```

Figure 38 - Unit Test for getting Patient record via Entity Framework

The site design itself was also tested, with a poll being conducted asking users to choose between the two navigation bar designs as described above in 4.2.2 – Web Application Design. This was created using the mobile app Pollie (Pollie, n.d.), a lightweight app which allows the user to create and share polls, and in particular attach images to each answer. This was shared with a few people, and though a sample of five isn't particularly large the results were unanimous, with everyone preferring the redesigned navbar that used icons rather than text.

In addition, a System Usability Scale survey (usabiliTEST, n.d.) was conducted, although there was unfortunately not enough time to make edits to the application as a result. This will be further discussed in 4.6 – Evaluation.


1 or 2?
https://pollie.app/ozthv
5 votes

1
5 votes, 100%
Alice, Steven, Annabelle, Kat, Anna

2
0 votes, 0%

Figure 39: Results of user testing unanimously favoured the redesigned navigation bar

4.5 Operation & Maintenance

In terms of full-time, permanent operation of this web server, the publishing and final launching would fall on the Radiotherapy department of the Lincoln County Hospital as this application would run on their local network. However, this process was still explored, and a test launch was performed on a personal laptop.

As explained in Testing above, IIS Express was used during development and is a more development-focussed variation of the regular IIS (Altvater, 2017). The latter is usually advised for production builds of applications, however, and was used to run the full version of the project. This required using the IIS Manager application which is pre-installed in Windows 10. This is a simple way to view and edit all local systems, sites and applications which are using the IIS Server framework.

The first step in fully launching the application using IIS was to create a new application pool. This is a dedicated allocation of resources specifically for this project, and the creation of a new, dedicated application pool allows for isolation from other apps, including the ability to have different settings for each web application currently published. While this doesn't present any benefit while running this application locally since it is the only ASP.NET project being run, the Radiotherapy department may have other web applications and so this is often recommended in order to keep a separation between them. A drawback of isolating application pools, though, is that it is more memory intensive (Schults, 2019).

Once this had been created in IIS Manager, a new application could then be added to the 'Default Web Site' that was automatically created. This application was linked to the filepath of the ArcCheck ASP.NET project and given the alias "arccheck", which determined the URL that the site was available on ("localhost/arccheck"). Once this was completed the application was available via IIS, however there were multiple issues with several process that had to be remedied, namely the database connection and relative URLs.

Upon attempting to login through the IIS-hosted application, an error was thrown with the message "Cannot open database "ArcCheck" requested by the login… Login failed for user 'IIS APPPOOL\ArcCheckPool'". The solution to this required creating a new login for this Application Pool using SQL Server Management Studio and granting this login several database roles such as "aspnetMembershipFullAccess", "dbDataReader" and "dbDataWriter" to allow for table reading and modification (Microsoft, 2020) (Halsey, 2011). The relative URL's fix was much simpler and involved changing any links to running on the ASP.NET server then prefixing them with "~/" to indicate they're relative to the ASP.NET project (kv-prajapati, 2011).

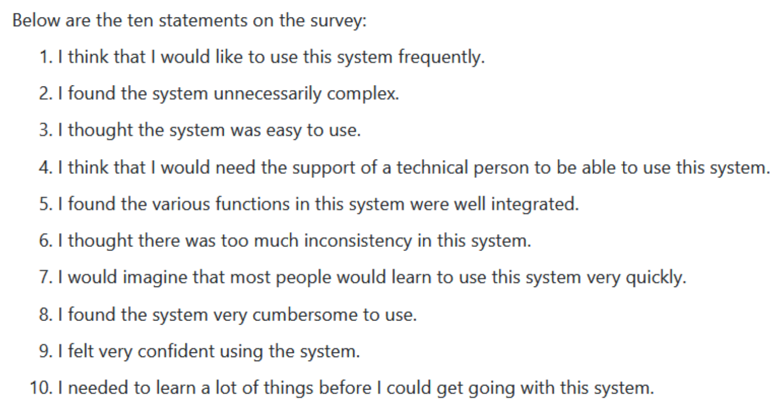
4.6 Evaluation

Overall, this project was successful in its main goals, however there proved to be many setbacks and issues which affected the final product. One of the main difficulties was the lack of sample files provided. In the case of some DICOM tags, this made it much harder to create functions and essentially impossible to perform testing on these, for example in the case of differentiating between VMAT Full and Partial Arc tests since there was only one IMRT DICOM file that was sent. However, since there were no sample ArcCheck PDF files received, it was impossible to develop any methods of extracting this data. However, this is something that was being worked on inside of the Radiotherapy department (Butler, 2019).

An area of success was the extraction of tags from a given DICOM file. Since the process of obtaining this information is conducted in the constructor for each class in the Entity Framework, this could easily be expanded to include the ArcCheck PDF files which would allow other tables to be filled in the database.

One area that could be improved to increase the robustness & maintainability of the project code would be to include more unit testing. Though this was experimented with as discussed above, creating smaller, more specific functions, then developing unit tests for all of these would make finding bugs & other issues much easier. This is one aspect of the agile philosophy that was unfortunately taken into account, as many agile frameworks operate under Test-Driven Development (Dooley, 2017).

A potentially useful piece of feedback was the System Usability Scale (SUS) conducted on the overall usability of the site itself. This is a questionnaire created in 1986 specifically designed to gauge users’ levels of comfort and ease of use when operating a given system (Lewis & Sauro, 2009). The ten questions are an even mix of positive and negative, and applying a calculation to the results of these questions produces a score between 0 and 100, giving a rough guideline of the overall usability of the system. According to UsabiliTEST's information page, a score of 68 is considered average: above this shows good usability, and below indicates there are major usability issues (usabiliTEST, n.d.).


Below are the ten statements on the survey:

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.

Figure 40: Full list of SUS questions (usabiliTEST, n.d.)

A Google Forms (Google, n.d.) document was created containing these questions (accessible at https://forms.gle/61XuQgDQgoJ9tmFB9) and a number of people were invited to test the site's functionality, then fill out the SUS survey. Calculating the averages of all of these responses gave the site an average usability score of 70.6 – above average, though only marginally. Unfortunately, there was not enough time for this to be fully analysed & acted upon, however some potential improvements could be to ensure every element is properly labelled and described, in addition to possibly choosing more relevant/understood icons on the navigation bar. There is a caveat to this, however, insofar as the users that completed this survey had no knowledge of the context of the application – since it is specifically tailored to the Radiotherapy department, there are terms used that an untrained user will likely not be familiar with.

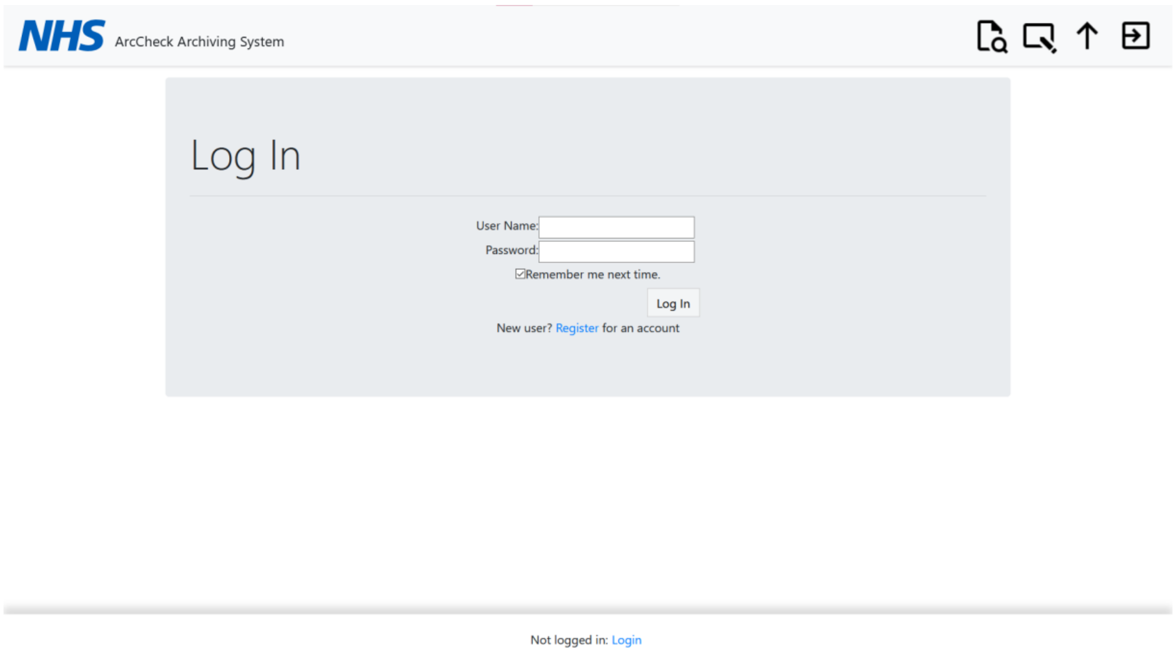
Chapter 5

Conclusions

The overall result of this project in achieving the aims & objectives set out in the Introduction of this report is a success. Two products have been created – an SQL database which allows for the storing of key data of Radiotherapy verification scans, in addition to a web application that interacts with this database, allowing the user to upload files and view past results. However, many of the additional features and goals were unfortunately not met to a considerable standard, though were implemented in a basic form.

Extraction of the relevant DICOM tags functions well, though since no sample files of this kind were able to be sent, a method of extracting & storing information from PDF files generated by the ArcCheck software was not able to be developed. These PDF files contain information on the scan such as how many Control Points were deemed to have "passed" in terms of radiation dose & accuracy, and thus can be used as Quality Assurance to determine how reliable the radiotherapy machine is overall. Implementing a way to easily see that all recent Quality Assurance tests have passed would allow the Radiotherapy team to potentially have to conduct less in-depth testing, for example rather than performing this on every patient it could be reduced to every two or three. However, implementing the tag extraction in the object constructors allowed for much more effective abstraction and simplified any future expansions on the application.

Another feature that could have been added is the automatic archiving of all uploaded files. This would likely not have been too difficult to develop – possibly consisting of creating a folder using the patient's NHS Number or scan date as an identifier and moving all relevant files into this. Two other potential features would be to expand the authentication system to include User Roles – ensuring only designated administrators could access the Edit page, for example – and enabling database auditing, in addition to developing a method of accessing the logs to view all changes.


NHS ArcCheck Archiving System

Log In

User Name:

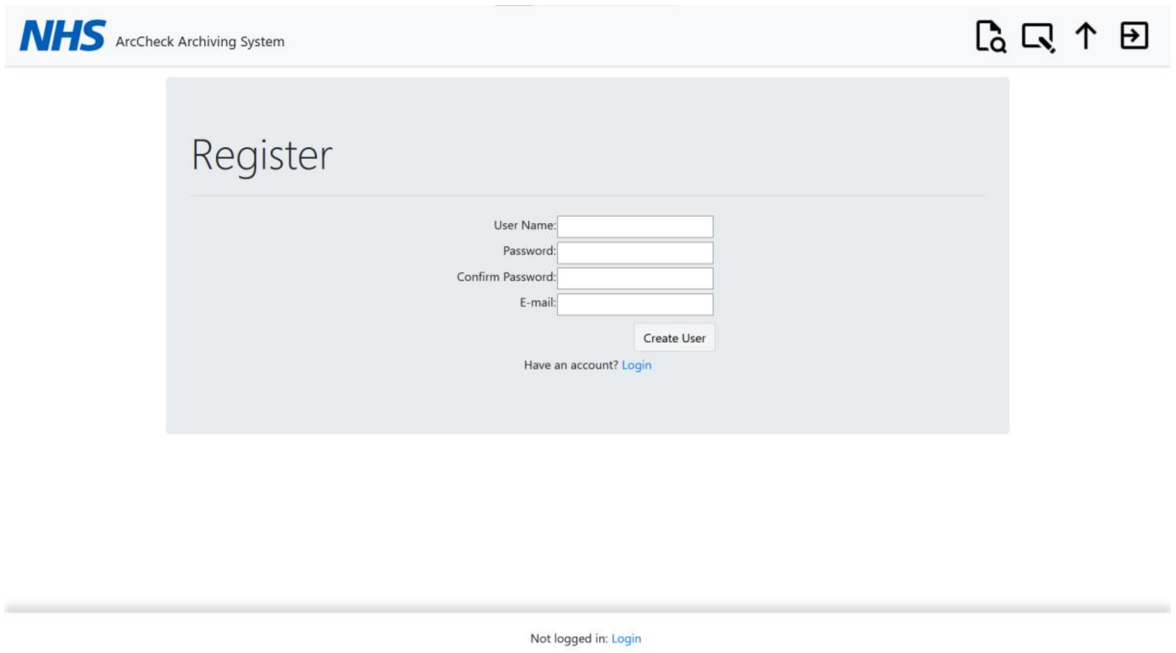
Password:

☑Remember me next time.

Log In

New user? Register for an account

Not logged in: Login


NHS ArcCheck Archiving System

Register

User Name:

Password:

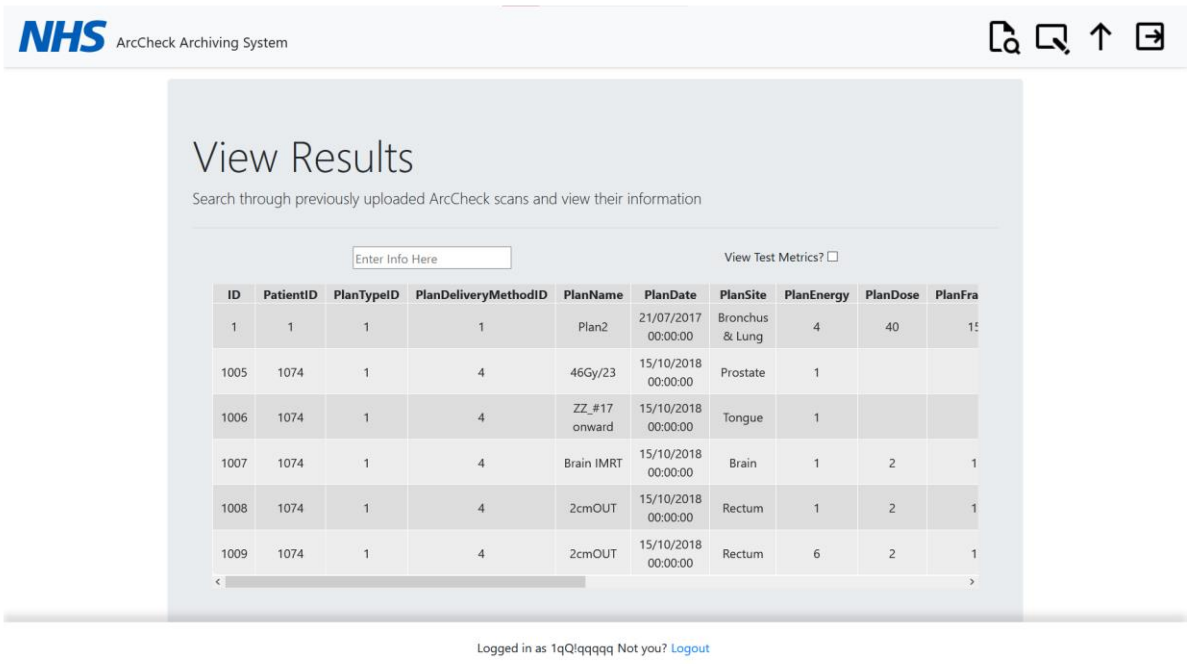
Confirm Password:

E-mail:

Create User

Have an account? Login

Not logged in: Login



NHS ArcCheck Archiving System

View Results

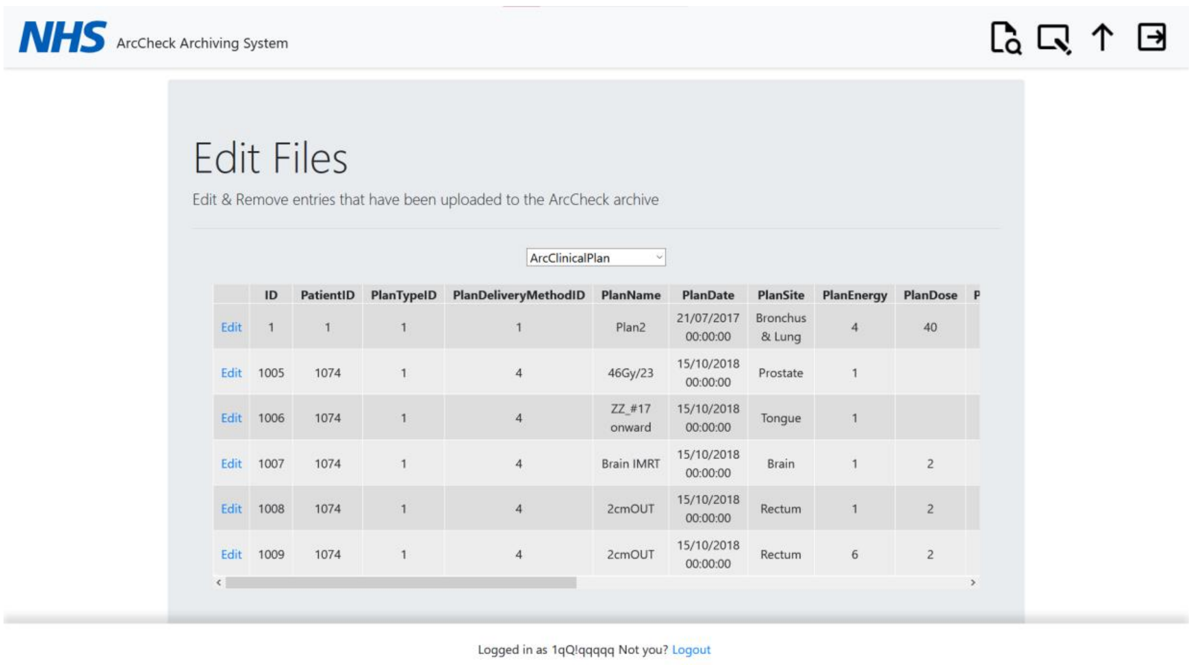
Search through previously uploaded ArcCheck scans and view their information

Enter Info Here

View Test Metrics? ☐

ID	PatientID	PlanTypeID	PlanDeliveryMethodID	PlanName	PlanDate	PlanSite	PlanEnergy	PlanDose	PlanFra
1	1	1	1	Plan2	21/07/2017 00:00:00	Bronchus & Lung	4	40	15
1005	1074	1	4	46Gy/23	15/10/2018 00:00:00	Prostate	1		
1006	1074	1	4	ZZ_#17 onward	15/10/2018 00:00:00	Tongue	1		
1007	1074	1	4	Brain IMRT	15/10/2018 00:00:00	Brain	1	2	1
1008	1074	1	4	2cmOUT	15/10/2018 00:00:00	Rectum	1	2	1
1009	1074	1	4	2cmOUT	15/10/2018 00:00:00	Rectum	6	2	1

Logged in as 1qQ!qqqqq Not you? Logout



NHS ArcCheck Archiving System

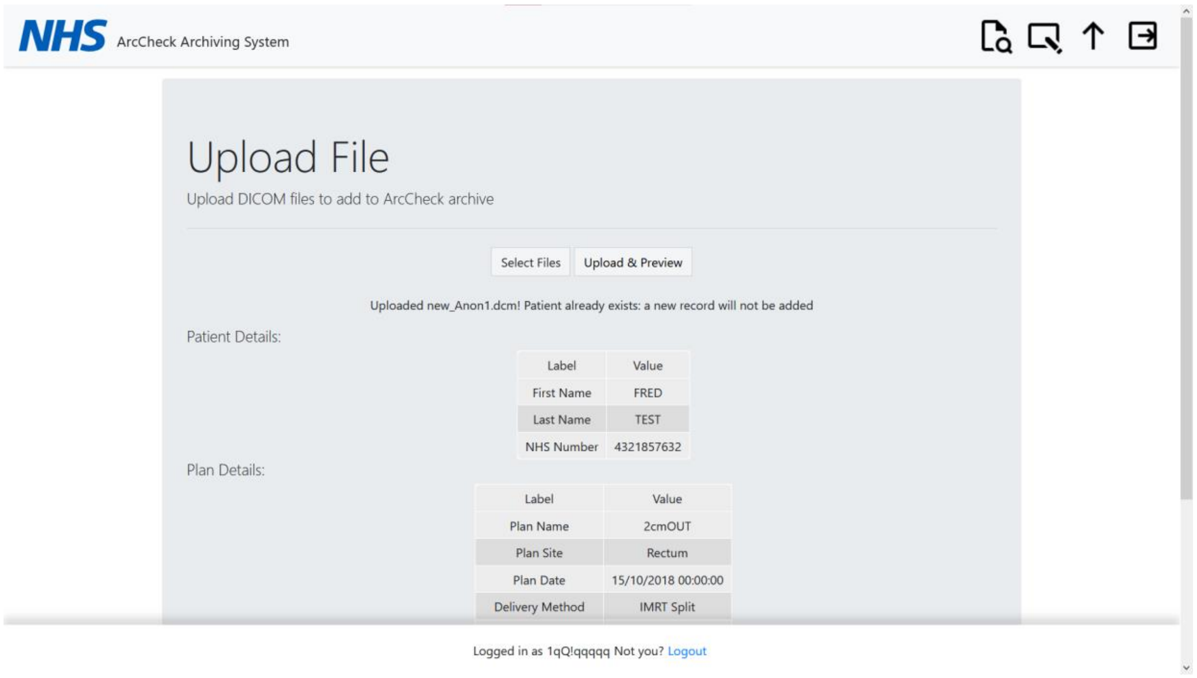
Edit Files

Edit & Remove entries that have been uploaded to the ArcCheck archive

ArcClinicalPlan

	ID	PatientID	PlanTypeID	PlanDeliveryMethodID	PlanName	PlanDate	PlanSite	PlanEnergy	PlanDose	P
Edit	1	1	1	1	Plan2	21/07/2017 00:00:00	Bronchus & Lung	4	40	
Edit	1005	1074	1	4	46Gy/23	15/10/2018 00:00:00	Prostate	1		
Edit	1006	1074	1	4	ZZ_#17 onward	15/10/2018 00:00:00	Tongue	1		
Edit	1007	1074	1	4	Brain IMRT	15/10/2018 00:00:00	Brain	1	2	
Edit	1008	1074	1	4	2cmOUT	15/10/2018 00:00:00	Rectum	1	2	
Edit	1009	1074	1	4	2cmOUT	15/10/2018 00:00:00	Rectum	6	2	

Logged in as 1qQ!qqqqq Not you? Logout



Upload File

Upload DICOM files to add to ArcCheck archive

Select Files | Upload & Preview

Uploaded new_Anon1.dcm! Patient already exists: a new record will not be added

Patient Details:

Label	Value
First Name	FRED
Last Name	TEST
NHS Number	4321857632

Plan Details:

Label	Value
Plan Name	2cmOUT
Plan Site	Rectum
Plan Date	15/10/2018 00:00:00
Delivery Method	IMRT Split

Logged in as 1qQ!qqqqq Not you? Logout

Figure 41 - Screenshots of all application pages

Authorization to the web application was successfully set up using the ‘aspnet_reqsql.exe’ command and developed by using ASP.NET’s membership system & detailing where to redirect logged out users to in the application’s web.config file. This allowed the authentication methods to be created, enabling users to register and login, storing these credentials in a secure manner by making use of password hashing & salting algorithms built into ASP.NET.

The created login & register systems are both fully functional and store users’ information securely, though there is one glaring issue with this method of creating accounts in the sense that anyone is able to create an account, potentially using dummy information to gain access to possibly sensitive data. One solution to this would be to implement an email verification system, though this could still be bypassed easily by using a temporary email address service such as 10 Minute Mail (10 Minute Mail, n.d.). Another approach could be to require a password to create an account, stored in a hashed & salted form, that only the system administrators know.

Chapter 6

Reflective Analysis

Overall, this project certainly has its strong & weak points. One aspect I felt went well was the design of the web application – the navigation bar allowed for easy access to each of the application's main pages, and the use of icons to represent the pages allowed for universal recognition while decreasing the clutter on the bar. This isn't necessarily particularly useful when the project has four pages, however if the application were to be expanded further it would increase the overall learnability of the page. (Bachour, 2021). In addition, I felt that both the colour scheme and the implementation of Bootstrap's Jumbotron (Bootstrap, n.d.) as a container for the main page content gave the application an overall professional appeal. After communication with Tim Butler, the Radiotherapy team will likely make use of this design among other features.

One major issue I had throughout the entire development period for this project was using the framework effectively. Though I had had relatively extensive experience with C# as a language prior to taking on this project, ASP.NET was brand new to me and required a lot of research and time to understand the underlying mechanics, best practices, and processes.

The struggle with this was twofold – since ASP.NET was first released in 2002 and has been succeeded by ASP.NET Core since 2016, there was a lack of comprehensive guides and descriptions. This resulted in an over-reliance on the reference material hosted created by Microsoft, however though the book *Beginning ASP.NET 4.5 in C#* (MacDonald, 2012) did also prove particularly useful.

In addition, I never really felt that the framework particularly 'clicked' with me. The concept of using prebuilt drag-and-drop functionality such as login elements, while speeding up development of the surface, allowed for a level of abstraction which made me too confident to learn the core mechanics – such as Membership Providers for Login and Register pages, and how Data Sources worked in the case of the DataGrid and GridView elements – for quite a while, which caused quite a bit of confusion.

Moreover, using Visual Studio 2019 as an IDE felt busy and cluttered, which clouded my ability to think and develop elements a surprising amount, in addition to VS 2019 performing some tasks slowly. Despite this, the integrated toolsets that this allowed for did prove very useful, such as using IIS Express to quickly relaunch the web application when making changes to backend code.

Both of the above points are in contrast to recent experience I have had with using React, a JavaScript framework which simplifies the creation of web applications; and Visual Studio Code, a (mostly (Kenlon, 2020)) open-source text editor (Microsoft, 2021). The lightweight nature of VS Code allowed me the ability to focus on individual tasks at hand without getting overwhelmed by unnecessary complexity and React felt a lot more natural with respect to creating elements. Though there may have been longer development times since everything would need to be built from the ground up, I believe this would have provided extra insight as to how these elements, and others, functioned.

Experience with any alternate framework or approach is always valuable and I definitely don't regret taking the time to learn this, however it is a shame that the Radiotherapy department at Lincoln County Hospital has not yet 'upgraded' to utilising the more contemporary & frequently updated ASP.NET Core, as this will likely be in much higher demand in other industries.

The Software Development processes and philosophies detailed in the Methodology section of this report proved beneficial in developing a high-level view of the tasks required and gave a rough estimation of all the deliverables that could conceivably be developed in the given timeframe by making use of a Gantt chart. In addition, the User Stories developed from Scrum methodologies helped me narrow down the exact features that should be developed. However, unfortunately these processes were not strictly adhered to throughout the development period, mainly at the fault of my own self-discipline. I did prioritise the most important pages first, though spending too much time on the Upload backend code did push other tasks out of schedule, causing their development to be rushed and not fully implemented.

One issue I had throughout this module's timeframe – and in particular from November-January – was the various impacts of COVID-19. This took quite a toll on me mentally as I did find working from home incredibly difficult, though fortunately the University library was key in providing a space to work on coursework.

However, the main issue with this was on the side of the hospital – in November 2020, a "Critical Incident" was declared at Lincoln County Hospital due to a rising number of COVID-19 patients (BBC, 2020). Though these patients likely wouldn't have directly affected the Radiotherapy department, this did have "a major impact on flow”, and as a result communication with Butler throughout this time was scarce. This state ended in early January (BBC, 2021).

Another effect of COVID-19 was the inability to visit the hospital and have in-person meetings. This would have been a great opportunity to get first-hand insight as to how the team worked, in addition to physically seeing their current workflow, and potentially even the radiotherapy machines themselves. On the other hand, the online nature of the pandemic did prove beneficial in terms of more frequent meetings (excluding the Critical Incident period) for this exact reason – commuting to the hospital (a 45-minute walk) for each meeting would have meant they would have taken place a lot less often.

Thinking of things that I have learnt and would do differently if undertaking the same project again, I would definitely ensure the perceived requirements of the app are clearly communicated and double-checked to be accurate to the actual requirements. I would also keep a stricter eye of adherence to any plan laid out, making sure not to fall behind or become too focussed on one area, while neglecting others. I have also learnt the importance of regular communication with clients, whether in-person or online.

References

10 Minute Mail, n.d. *10 Minute Mail - Free Anonymous Temporary email - 10 Minute Mail - Free Anonymous Temporary email.* [Online]
Available at: https://10minutemail.com/
[Accessed 14 May 2021].

Adnan, A. H. et al., 2015. *A comparative study of WLAN security protocols: WPA, WPA2.* Dhaka, Bangladesh, IEEE.

Aggarwal, S. & Verma, J., 2018. Comparative analysis of MEAN stack and MERN stack. *International Journal of Recent Research Aspects,* 5(1), pp. 127-132.

Alquda, M. & Razali, R., 2017. *A comparison of scrum and Kanban for identifying their selection factors.* Langkawi, Malaysia, IEEE.

Altvater, A., 2017. *What is IIS Express? How It Works, Tutorials, and More – Stackify.* [Online]
Available at: https://stackify.com/what-is-iis-express/
[Accessed 25 April 2021].

Amoatey, C. T. & Anson, B. A., 2017. Investigating the major causes of scope creep in real estate construction projects in Ghana. *Journal of Facilities Management,* 15(4), pp. 393-408.

Asana, n.d. *Manage your team's work, projects, & tasks online • Asana.* [Online]
Available at: https://asana.com
[Accessed 8 April 2021].

Ashur, H., 1992. *Using the Matrix Diagram Method to Build an Organizational Taxonomy for a Developing Multi-Disciplinary Center.* Eatontown, NJ, USA, IEEE.

Aston, B., 2021. *Compare The 15 Best Project Management Software Of 2021.* [Online]
Available at: https://thedigitalprojectmanager.com/best-project-management-software/
[Accessed 8 April 2021].

Bachour, K., 2021. *CMP3035M Cross-Platform Development - Lecture 4: Mobile Usability,* Lincoln, UK: University of Lincoln.

Balsamiq Studios, LLC, n.d. *Balsamiq. Rapid, Effective and Fun Wireframing Software | Balsamiq.* [Online]
Available at: https://balsamiq.com/
[Accessed 19 April 2021].

BBC, 2020. *'Critical incident' at Lincolnshire hospitals after Covid patients rise - BBC News.* [Online]
Available at: https://www.bbc.co.uk/news/uk-england-lincolnshire-54933070
[Accessed 25 May 2021].

BBC, 2021. *Covid: Critical incident stood down at Lincoln hospital - BBC News.* [Online]
Available at: https://www.bbc.co.uk/news/uk-england-lincolnshire-55544485
[Accessed 25 May 2021].

Beck, K. et al., 2001. *Manifesto for Agile Software Development.* [Online]
Available at: http://agilemanifesto.org/
[Accessed 30 March 2021].

Boehm, B., 1988. A Spiral Model of Software Development and Enhancement. *Computer,* 21(5), pp. 61-72.

Bootstrap, 2021. *Introduction - Bootstrap v5.0.* [Online]
Available at: https://getbootstrap.com/docs/5.0/getting-started/introduction/
[Accessed 21 April 2021].

Bootstrap, n.d. *Jumbotron - Bootstrap.* [Online]
Available at: https://getbootstrap.com/docs/4.0/components/jumbotron/
[Accessed 25 May 2021].

Boylan, C. J., 2013. *The Effective Application of Dynamic Arc Radiotherapy*. [Online]
Available at:
https://www.escholar.manchester.ac.uk/api/datastream?publicationPid=uk-ac-man-scw:208322&datastreamId=FULL-TEXT.PDF
[Accessed 8 February 2021].

British Dupuytren's Society, 2013. *Dupuytren's Disease Treatment.* [Online]
Available at: https://dupuytrens-society.org.uk/treatment/dupuytrens-disease/
[Accessed 14 3 2020].

Butler, T., 2019. *Project Brief - ArcCheck*, s.l.: s.n.

Butler, T., 2020. [Interview] (15 December 2020).

Butler, T., 2020. *CreateArcCheckTables.sql,* s.l.: s.n.

Butler, T., 2020. *Database_Fields,* s.l.: s.n.

Butler, T., 2020. *Personal Communication* [Interview] (1 December 2020).

Butler, T., 2021. *Personal Communication* [Interview] (10 March 2021).

Cardan, Rex, 2020. *rexcardan/Evil-DICOM: A C# DICOM Library.* [Online]
Available at: https://github.com/rexcardan/Evil-DICOM
[Accessed 23 April 2021].

Choplin, R. H., Boheme II, J. M. & Maynard, C. D., 1992. Picture Archiving and Communications Systems: An Overview. *RadioGraphics,* 12(1), pp. 127-129.

Chowdhuri, S., 2016. *ASP.NET Core Essentials.* 1st ed. Birmingham, UK: Packt Publishing.

ClickUp, n.d. *ClickUp™ | One app to replace them all.* [Online]
Available at: https://clickup.com/
[Accessed 8 April 2021].

Database.Guide, 2016. *What is a Database Schema?.* [Online]
Available at: https://database.guide/what-is-a-database-schema/
[Accessed 14 April 2021].

Davidson, J., 2019. Gantt Charts. In: *Everyday Project Management.* s.l.:Berrett-Koehler Publishers, pp. 127-135.

Delos Santos, J. M., 2021. *Best Project Management Software for 2021.* [Online]
Available at: https://project-management.com/top-10-project-management-software/
[Accessed 8 April 2021].

DICOM Library, n.d. *DICOM Library - Anonymize, Share, View DICOM files ONLINE.* [Online]
Available at: https://dicomlibrary.com/dicom/dicom-tags/
[Accessed 2 May 2021].

DICOM Library, n.d. *DICOM Library - Anonymize, Share, View DICOM files ONLINE.* [Online]
Available at: https://dicomlibrary.com/dicom/dicom-tags/
[Accessed 23 April 2021].

Dooley, J. F., 2017. Testing in an Agile Development Environment. In: *Software Development, Design and Coding : With Patterns, Debugging, Unit Testing, and Refactoring.* Galesburg, Illinois: Apress, p. 256.

Fearn, N., DeMuro, J. P. & Turner, B., 2021. *Best project management software of 2021.* [Online]
Available at: https://www.techradar.com/uk/best/best-project-management-software
[Accessed 8 April 2021].

GDPR.eu, 2016. *Article 4 GDPR - Definitions.* [Online]
Available at: https://gdpr.eu/article-4-definitions/
[Accessed 10 March 2021].

Gebicz, M., n.d. *What is a Gantt Chart? | Atlassian.* [Online]
Available at: https://www.atlassian.com/agile/project-management/gantt-

chart
[Accessed 4 March 2021].

GOV.UK Agile Delivery Community, 2016. *Agile methods: an introduction.* [Online]
Available at: https://www.gov.uk/service-manual/agile-delivery/agile-methodologies
[Accessed 29 March 2021].

Hall, E. J. & Wuu, C.-S., 2003. Radiation-induced second cancers: the impact of 3D-CRT and IMRT. *International Journal of Radiation Oncology, Biology, Physics,* 56(1), pp. 83-88.

Halsey, S., 2011. *asp.net - The EXECUTE permission was denied on the object 'aspnet_CheckSchemaVersion', database 'XXX' - Stack Overflow.* [Online]
Available at: https://stackoverflow.com/questions/7710868/the-execute-permission-was-denied-on-the-object-aspnet-checkschemaversion-dat
[Accessed 25 April 2021].

Heddings, A., 2020. *What's the Difference Between .NET Framework and .NET Core?.* [Online]
Available at: https://www.cloudsavvyit.com/6314/whats-the-difference-between-net-framework-and-net-core/
[Accessed 10 April 2021].

Henry, A., 2015. *Productivity 101: How to Use Personal Kanban to Visualize Your Work.* [Online]
Available at: https://lifehacker.com/productivity-101-how-to-use-personal-kanban-to-visuali-1687948640
[Accessed 30 March 2021].

Ibrahim, F., 2020. *How to create a custom file upload button - DEV Community.* [Online]
Available at: https://dev.to/faddalibrahim/how-to-create-a-custom-file-upload-button-using-html-css-and-javascript-1c03
[Accessed 12 May 2021].

Ionactive, 2018. *Ionactive | Fluence Rate.* [Online]
Available at: https://ionactive.co.uk/resource-hub/glossary/fluence-rate
[Accessed 8 February 2021].

Jang, Y.-S. & Choi, J.-Y., 2014. Detecting SQL injection attacks using query result size. *Computers & Security,* Volume 44, pp. 104-118.

jQuery, n.d. *jQuery.* [Online]
Available at: https://jquery.com/
[Accessed 21 April 2021].

Kadhum, M., Smock, E., Khan, A. & Fleming, A., 2017. Radiotherapy in Dupuytren's disease: a systematic review of the evidence. *Journal of Hand Surgery,* 42E(7), pp. 689-692.

Kavlakoglu, E., 2020. *Agile vs. Waterfall | IBM.* [Online]
Available at: https://www.ibm.com/cloud/blog/agile-vs-waterfall
[Accessed 15 March 2021].

Kenlon, S., 2020. *7 open source alternatives to VS Code | Opensource.com.* [Online]
Available at: https://opensource.com/article/20/6/open-source-alternatives-vs-code
[Accessed 25 May 2021].

Kissflow Project, n.d. *Collaborative Project Management Software | Kissflow Project.* [Online]

Available at: https://kissflow.com/project/
[Accessed 8 April 2021].

Konrad, A., 2019. *Israel's New Top Unicorn: Monday.com Hits $1.9 Billion Valuation With $150 Million Raise.* [Online]
Available at:
https://www.forbes.com/sites/alexkonrad/2019/07/30/monday-raises-150-million-becomes-israel-top-unicorn/
[Accessed 9 April 2021].

kv-prajapati, 2011. *Relative and absolute paths on ASP.NET/IIS - Stack Overflow.* [Online]
Available at: https://stackoverflow.com/questions/7674117/relative-and-absolute-paths-on-asp-net-iis
[Accessed 25 April 2021].

Lexico, n.d. *FULL STACK | Definition of FULL STACK by Oxford Dictionary on Lexico.com also meaning of FULL STACK.* [Online]
Available at: https://www.lexico.com/definition/full_stack
[Accessed 10 April 2021].

Lucidchart Content Team, n.d. *Matrix Diagrams: What They Are and How to Use Them | Lucidchart Blog.* [Online]
Available at: https://www.lucidchart.com/blog/what-is-a-matrix-chart
[Accessed 8 April 2021].

MacDonald, M., 2012. ADO.NET Fundamentals. In: E. Buckingham, ed. *Beginning ASP.NET 4.5 in C#.* New York: Apress, pp. 440-451.

MacDonald, M., 2012. *Beginning ASP.NET 4.5 in C#.* 1st ed. New York, USA: Apress.

MacDonald, M., 2012. Membership. In: E. Buckingham, ed. *Beginning ASP.NET 4.5 in C#.* New York: Apress, pp. 639-674.

Manh Thang, N., 2020. Improving Efficiency of Web Application Firewall to Detect Code Injection Attacks with Random Forest Method and Analysis Attributes HTTP Request. *Programming and Computer Software,* 46(5), pp. 351-361.

Medical Imaging Technology Association, 2021. *1 Scope and Field of Application.* [Online]
Available at:
http://dicom.nema.org/medical/dicom/current/output/chtml/part01/chapter_1.html
[Accessed 2 May 2021].

Melton, J. & Simon, A. R., 2002. Schemas and Catalogs. In: D. D. Cerra, ed. *SQL: 1999 : Understanding Relational Language Components.* San Francisco: Morgan Kaufmann, pp. 29-30.

Mendjoge, N., Joshi, A. R. & Narvekar, M., 2016. *Intelligent Tutoring System for Database Normalization,* Pune, India: IEEE.

Microsoft, 2017. *Download Internet Information Services (IIS) 10.0 Express from Official Microsoft Download Center.* [Online]
Available at: https://www.microsoft.com/en-us/download/details.aspx?id=48264
[Accessed 24 April 2021].

Microsoft, 2017. *Language-Integrated Query (LINQ) (C#) | Microsoft Docs.* [Online]
Available at: https://docs.microsoft.com/en-

us/dotnet/csharp/programming-guide/concepts/linq/
[Accessed 23 April 2021].

Microsoft, 2017. *Security Considerations (Entity Framework) - ADO.NET | Microsoft Docs.* [Online]
Available at: https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/ef/security-considerations
[Accessed 23 April 2021].

Microsoft, 2019. *What is SQL Server Management Studio (SSMS)?.* [Online]
Available at: https://docs.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver15
[Accessed 12 May 2021].

Microsoft, 2020. *CREATE LOGIN (Transact-SQL) - SQL Server | Microsoft Docs.* [Online]
Available at: https://docs.microsoft.com/en-us/sql/t-sql/statements/create-login-transact-sql?view=sql-server-ver15#examples
[Accessed 25 April 2021].

Microsoft, 2020. *SQL Server Audit (Database Engine).* [Online]
Available at: https://docs.microsoft.com/en-us/sql/relational-databases/security/auditing/sql-server-audit-database-engine
[Accessed 14 April 2021].

Microsoft, 2021. *Github - microsoft/vscode: Visual Studio Code.* [Online]
Available at: https://github.com/Microsoft/vscode
[Accessed 25 May 2021].

Microsoft, n.d. *ASP.NET | Open-source web framework for .NET.* [Online]

Available at: https://asp.net
[Accessed 10 April 2021].

Microsoft, n.d. *Buy Microsoft Access Database & Application Software.* [Online]
Available at: https://www.microsoft.com/en-gb/microsoft-365/access
[Accessed 15 April 2021].

Microsoft, n.d. *Buy Microsoft Excel Spreadsheet Software or Try Excel, Free.* [Online]
Available at: https://www.microsoft.com/en-gb/microsoft-365/excel
[Accessed 30 March 2021].

Microsoft, n.d. *Home: The Official Microsoft IIS Site.* [Online]
Available at: https://www.iis.net/
[Accessed 24 April 2021].

Microsoft, n.d. *SQL Server 2019 | Microsoft.* [Online]
Available at: https://www.microsoft.com/en-gb/sql-server/sql-server-2019
[Accessed 10 April 2021].

monday.com, n.d. *monday.com product page.* [Online]
Available at: https://monday.com/product/
[Accessed 2 March 2021].

Năchilă, C., 2008. Building a dynamically ASP.NET 2.0 GridView control. *Informatica Economică,* XII(1), pp. 114-119.

NHS, 2020. *Radiotherapy - Overview.* [Online]
Available at: https://www.nhs.uk/conditions/radiotherapy/
[Accessed 14 3 2020].

Notion.VIP, n.d. *Meet Notion's Formula Property.* [Online]
Available at: https://www.notion.vip/meet-notions-formula-property/
[Accessed 4 April 2021].

Notion, n.d. *Notion - The all-in-one workspace for your notes, tasks, wikis, and databases..* [Online]
Available at: https://www.notion.so/
[Accessed 26 March 2021].

Oppelt, A., 2005. *Imaging Systems for Medical Diagnostics: Fundamentals, Technical Solutions and Applications for Systems Applying Ionizing Radiation, Nuclear Magnetic Resonance and Ultrasound.* 1st ed. Erlangen, Germany: Publicis Corporate Publishing.

Oracle, n.d. *What Is Input Validation and Sanitization?.* [Online]
Available at:
https://download.oracle.com/oll/tutorials/SQLInjection/html/lesson1/les01_tm_ovw3.htm
[Accessed 9 March 2021].

Pollie, n.d. *Pollie: The easiest way to share a poll.* [Online]
Available at: https://www.pollie.app/
[Accessed 26 May 2021].

Pruitt, J., 2011. *Personal Scrum.* [Online]
Available at: https://jgpruitt.wordpress.com/2011/04/10/personal-scrum/
[Accessed 4 April 2021].

Pydicom, 2020. *Pydicom.* [Online]
Available at: https://pydicom.github.io/
[Accessed 24 April 2021].

RadiologyInfo.org, 2019. *LINAC (Linear Accelerator).* [Online]
Available at: https://www.radiologyinfo.org/en/info.cfm?pg=linac
[Accessed 8 February 2021].

Ranganathan, V. & Maria Das, K., 2016. Determination of optimal number of beams in direct machine parameter optimization-based intensity modulated radiotherapy for head and neck cases. *Journal of Medical Physics,* 41(2), pp. 129-134.

Ray, L. L., 2013. Security considerations for the spiral development model. *International Journal of Information Management,* Volume 33, pp. 684-686.

Rehkopf, M., n.d. *What is a kanban board?.* [Online]
Available at: https://www.atlassian.com/agile/kanban/boards
[Accessed 30 March 2021].

Schults, C., 2019. *What Is an IIS Application Pool – Stackify.* [Online]
Available at: https://stackify.com/what-is-an-iis-application-pool/
[Accessed 25 April 2021].

Scrum.org, n.d. *What is a Sprint Backlog.* [Online]
Available at: https://www.scrum.org/resources/what-is-a-sprint-backlog
[Accessed 5 April 2021].

Shieber, J., 2020. *As losses expand, Asana is confident it has the ticket for a successful public listing.* [Online]
Available at: https://techcrunch.com/2020/08/24/as-losses-expand-asana-is-confident-it-has-the-ticket-for-a-successful-public-listing/
[Accessed 9 April 2021].

Smartsheet, n.d. *Increase Productivity With This Powerful Project Management Software | Smartsheet.* [Online]

Available at: https://www.smartsheet.com/s/project-software
[Accessed 8 April 2021].

Song, J. B., Shin, H.-J., Kay, C. S. & Son, S. H., 2015. Dosimetric Verification by Using the ArcCHECK System and 3DVH Software for Various Target Sizes. *PLoS ONE,* 10(3), p. e0119937.

Stray, V. G., Lindsjørn, Y. & Sjøberg, D. I., 2013. *Obstacles to Efficient Daily Meetings in Agile Development Projects: A Case Study.* Baltimore, MD, USA, IEEE.

Sun Nuclear Corporation, 2014. *ArcCheck and 3DVH - The Ultimate 4D Patient QA Solution.* [Online]
Available at: http://epsilonelektronik.com/wp-content/uploads/2015/05/ArcCHECK-3DVH.pdf
[Accessed 8 February 2021].

Sun Nuclear Corporation, n.d. *ArcCHECK® - Sun Nuclear.* [Online]
Available at: https://www.sunnuclear.com/products/arccheck
[Accessed 2 May 2021].

Sun, X., Wang, H., Li, J. & Zhang, Y., 2011. Injecting purpose and trust into data anonymisation. *Computers & Security,* 30(5), pp. 332-345.

Surbakti, E. E., Purwandari, B., Solichah, I. & Kumaralalita, L., 2019. Analysis of Software Development Method Selection: A Case of a Private Financial Institution. *ICBIM '19: Proceedings of the 3rd International Conference on Business and Information Management*, 12-14 September, pp. 168-173.

Syed, S. et al., 2021. API Driven On-Demand Participant ID Pseudonymization in Heterogeneous Multi-Study Research.. *Healthcare Informatics Research,* 27(1), pp. 39-47.

Tarhan, A. & Yilmaz, S. G., 2013. Systematic analyses and comparison of development performance and product quality of Incremental Process and Agile Process. *Information and Software Technology,* Volume 56, pp. 477-494.

TechTerms, 2014. *Server Definition.* [Online]
Available at: https://techterms.com/definition/server
[Accessed 9 March 2021].

TestBytes, 2019. *What's Spiral Model? Advantages and Disadvantages.* [Online]
Available at: https://www.testbytes.net/blog/spiral-model-advantages-and-disadvantages/
[Accessed 29 March 2021].

The Royal College of Radiologists, 2019. *Radiotherapy Dose Fractionation, Third Edition,* London, United Kindom: The Royal College of Radiologists.

Trello, 2019. *50 Million Is Just The Beginning: Automation, Templates, And More New Features To Keep Your Team Building.* [Online]
Available at: https://blog.trello.com/50-million-celebration
[Accessed 9 April 2021].

Trello, n.d. *About | What Is Trello?.* [Online]
Available at: https://trello.com/about
[Accessed 3 March 2021].

Turk, W., 2010. Scope Creep Horror. *Defense AT&L,* 39(2), pp. 53-55.

Turner, J., 2018. Benefits of Scrum. In: *Scrum - The Ultimate Beginner's Guide to Learn Scrum Step by Step.* s.l.:James Turner - Independently Published, p. 30.

Upadhyay, R. K., 2020. *Advantages and Disadvantages of using Spiral Model.* [Online]
Available at: https://www.geeksforgeeks.org/advantages-and-disadvantages-of-using-spiral-model/
[Accessed 29 March 2021].

usabiliTEST, n.d. *System Usability Scale online with analytics.* [Online]
Available at: https://www.usabilitest.com/system-usability-scale
[Accessed 26 May 2021].

Visual Paradigm, n.d. *What is Scrum Team? - Scrum Guide.* [Online]
Available at: https://www.visual-paradigm.com/scrum/what-is-scrum-team/
[Accessed 1 April 2021].

Weiss, A., 2012. *Prevent Web Attacks Using Input Sanitization.* [Online]
Available at: https://www.esecurityplanet.com/endpoint/prevent-web-attacks-using-input-sanitization/
[Accessed 9 March 2021].

Wikipedia, n.d. *ASP.NET | Wikipedia.* [Online]
Available at: https://en.wikipedia.org/wiki/ASP.NET
[Accessed 10 April 2021].

Wiseley, M., 2018. *What is ASP.NET and Why Should I Use It?.* [Online]
Available at: https://www.wakefly.com/blog/what-is-asp-net-and-why-should-i-use-it/
[Accessed 10 April 2021].

Zhdanov, A. E., Borisov, V. I. & Dorosinsky, L. G., 2019. *IMRT Verification, Analysis and Performance Evaluations: Modeling of Dose*

Delivery by Shifting MLC Angel and Phantom Position. Tashkent, Uzbekistan, IEEE.

Word Count

17603 (TOTAL) – 2074 (APPENDICES) = **15529**

APPENDICES

Appendix I – CreateACDB.sql:

```sql
CREATE DATABASE ArcCheck
GO

USE ArcCheck

CREATE TABLE Patient (
    ID  INT IDENTITY(1,1) PRIMARY KEY,
    FirstName NVARCHAR(50),
    Surname NVARCHAR(50),
    NHSNumber NVARCHAR(10)
)

CREATE TABLE PlanType (
    ID INT IDENTITY(1,1) PRIMARY KEY,
    PlanType NVARCHAR(50)
)

CREATE TABLE DeliveryMethod (
    ID INT IDENTITY(1,1) PRIMARY KEY,
    DeliveryMethod NVARCHAR(50)
)

CREATE TABLE GammaProtocol (
    ID INT IDENTITY(1,1) PRIMARY KEY,
    GammaProtocol NVARCHAR(50)
)

CREATE TABLE PlanInfo (
    ID INT IDENTITY(1,1) PRIMARY KEY,
    PatientID INT REFERENCES Patient(ID),
    PlanTypeID INT REFERENCES PlanType(ID),
    DeliveryMethodID INT REFERENCES DeliveryMethod(ID),
    PlanName NVARCHAR(50),
    PlanDate DATETIME,
    PlanSite NVARCHAR(50),
    PlanEnergy NVARCHAR(50),
    PlanDose DECIMAL,
    PlanFractions INT,
    PlanNumberOfBeams INT,
    PlanTotalControlPoints INT,
    PlanTotalMU INT,
)

CREATE TABLE ArcMeasurement (
    ID INT IDENTITY(1,1) PRIMARY KEY,
    PlanID INT REFERENCES PlanInfo(ID),
    GammaProtocolID INT REFERENCES GammaProtocol(ID),
    TotalPoints INT,
    PassedPoints INT,
    FailedPoints INT,
    PercentagePass DECIMAL,
    MeasurementDate DATETIME,
)
```

Appendix II: CreateArcCheckTables.sql

```
/*****************************************************
* THIS SCRIPT ADDS TABLES INTO THE ARCCHECK DATABASE *
*                                                    *
* 1. ArcPatient                                      *
* 2. ArcPlanType                                     *
* 3. ArcDeliveryMethod                               *
* 4. ArcPlan                                         *
* 5. ArcGammaProtocol                                *
* 6. ArcMeasurement                                  *
* 7. ArcTreatmentMachine                             *
* 8. ArcClinicalPlan                                 *
* 9. ArcClinicalPlanField                            *
*                                                    *
* Includes Relationships and Foreign Keys            *
*****************************************************/

PRINT '-------------------------'
PRINT ' Starting ArcCheck Query '
PRINT '-------------------------'
GO

/* Check to see if the database already exists */
IF db_id('ArcCheck') is NULL
CREATE DATABASE ArcCheck
GO


USE ArcCheck
GO

/* ArcPatient Table */
CREATE TABLE ArcPatient
(
    ID INT IDENTITY(1,1) NOT NULL,
    PatientFirstName NVARCHAR(50),
    PatientSurname NVARCHAR(50),
    NHSNumber NVARCHAR(10)
    CONSTRAINT PK_Patient PRIMARY KEY CLUSTERED
    (ID ASC)
)

/* ArcPlanType Table */
CREATE TABLE ArcPlanType
```

```
(
    ID INT IDENTITY(1,1) NOT NULL,
    PlanType NVARCHAR(50)
    CONSTRAINT PK_PlanType PRIMARY KEY CLUSTERED
    (ID ASC)
)

/* ArcDeliveryMethod Table */
CREATE TABLE ArcDeliveryMethod
(
    ID INT IDENTITY(1,1) NOT NULL,
    DeliveryMethod NVARCHAR(50)
    CONSTRAINT PK_DeliveryMethod PRIMARY KEY CLUSTERED
    (ID ASC)
)

/* ArcPlan Table */
CREATE TABLE ArcPlan
(
    ID INT IDENTITY(1,1) NOT NULL,
    PatientID INT,
    PlanTypeID INT,
    PlanDeliveryMethodID INT,
    PlanName NVARCHAR(50),
    PlanDate DATETIME,
    PlanSite NVARCHAR(50),
    PlanEnergy NVARCHAR(50),
    PlanDose DECIMAL,
    PlanFractions INT,
    PlanNumberOfBeams INT,
    PlanTotalControlPoints INT,
    PlanTotalMU DECIMAL,
    PlanPatientFolder NVARCHAR (50),
    PlanComments NVARCHAR(MAX)
    CONSTRAINT PK_ArcPlan PRIMARY KEY CLUSTERED
    (ID ASC)
)

/* ArcGammaProtocol Table*/
CREATE TABLE ArcGammaProtocol
(
    ID INT IDENTITY(1,1) NOT NULL,
    GammaProtocol NVARCHAR(50)
    CONSTRAINT PK_ArcGammaProtocol PRIMARY KEY CLUSTERED
    (ID ASC)
```

```
)

/* ArcMeasurement Table*/
CREATE TABLE ArcMeasurement
(
    ID INT IDENTITY(1,1) NOT NULL,
    PlanID INT,
    TotalPoints INT,
    PassedPoints INT,
    FailedPoints INT,
    PercentagePass DECIMAL,
    MachineID INT,
    GammaProtocolID INT,
    MeasurementDate DATETIME,
    MeasurementComments NVARCHAR(MAX)
    CONSTRAINT PK_ArcMeasurement PRIMARY KEY CLUSTERED
    (ID ASC)
)

/* ArcTreatmentMachine Table */
CREATE TABLE ArcTreatmentMachine
(
    ID Int IDENTITY(1,1) NOT NULL,
    MachineName NVARCHAR(20)
    CONSTRAINT PK_ArcTreatmentMachine PRIMARY KEY CLUSTERED
    (ID ASC)
)

/* ArcClinicalPlan Table */
CREATE TABLE ArcClinicalPlan
(
    ID Int IDENTITY(1,1) NOT NULL,
    PatientID INT
    CONSTRAINT PK_ArcClinicalPlan PRIMARY KEY CLUSTERED
    (ID ASC)
)

/* ArcClinicalPlanField Table */
CREATE TABLE ArcClinicalPlanField
(
    ID Int IDENTITY(1,1) NOT NULL,
    ClinicalPlanID INT,
    ControlPointsPerField INT,
    MUPerField DECIMAL,
    CalculationAlgorithm NVARCHAR(20),
    OptimisationAlgorithm NVARCHAR(50),
    IMRTSegments DECIMAL,
    DosePerField DECIMAL
    CONSTRAINT PK_ArcClinicalPlanField PRIMARY KEY CLUSTERED
    (ID ASC)
)



/*********************************************
* DEFINE RELATIONSHIPS FOR ARCCHECK DATABASE *
*********************************************/
/* ArcPlan Table Relationships */
ALTER TABLE ArcPlan WITH CHECK ADD CONSTRAINT FK_ArcPlan_PatientID_A
rcPatient_ID FOREIGN KEY(PatientID)
REFERENCES ArcPatient(ID)
GO
ALTER TABLE ArcPlan WITH CHECK ADD CONSTRAINT FK_ArcPlan_PlanTypeID_
ArcPlanType_ID FOREIGN KEY(PlanTypeID)
REFERENCES ArcPlanType(ID)
GO
ALTER TABLE ArcPlan WITH CHECK ADD CONSTRAINT FK_ArcPlan_PlanDeliver
yMethodID_ArcDeliveryMethod_ID FOREIGN KEY(PlanDeliveryMethodID)
REFERENCES ArcDeliveryMethod(ID)
GO

/* ArcMeasurement Table Relationships */
ALTER TABLE ArcMeasurement WITH CHECK ADD CONSTRAINT FK_ArcMeasureme
nt_PlanID_ArcPlan_ID FOREIGN KEY(PlanID)
REFERENCES ArcPlan(ID)
GO
ALTER TABLE ArcMeasurement WITH CHECK ADD CONSTRAINT FK_ArcMeasureme
nt_MachineID_ArcTreatmentMachine_ID FOREIGN KEY(MachineID)
REFERENCES ArcTreatmentMachine(ID)
GO
ALTER TABLE ArcMeasurement WITH CHECK ADD CONSTRAINT FK_ArcMeasureme
nt_GammaProtocolID_ArcGammaProtocol_ID FOREIGN KEY(GammaProtocolID)
REFERENCES ArcGammaProtocol(ID)
GO

/* ArcClinicalPlan Table Relationships */
ALTER TABLE ArcClinicalPlan WITH CHECK ADD CONSTRAINT FK_ArcClinical
Plan_PatientID_ArcPatient_ID FOREIGN KEY(PatientID)
REFERENCES ArcPatient(ID)
GO
```

```
/* ArcClinicalPlanField Table Relationships */
ALTER TABLE ArcClinicalPlanField WITH CHECK ADD CONSTRAINT FK_ArcCli
nicalPlanField_ClinicalPlanID_ArcClinicalPlan_ID FOREIGN KEY(Clinica
lPlanID)
REFERENCES ArcClinicalPlan(ID)
GO


/* Execute Script */
GO


PRINT '--------------------------'
PRINT ' Completed ArcCheck Query '
PRINT '--------------------------'
GO
```

Appendix III – ArcPlan.cs:

```
    public partial class ArcPlan
    {
        [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.
Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]
        public ArcPlan()
        {
            this.ArcMeasurements = new HashSet<ArcMeasurement>();
        }

        public ArcPlan(DICOMObject dcm)
        {
            this.ArcMeasurements = new HashSet<ArcMeasurement>();

            string PlanLabel = dcm.FindFirst(TagHelper.RTPlanLabel).
DData.ToString();
            DICOMSelector selector = dcm.GetSelector();

            PlanSite = PlanLabel.Split()[0]; // Assumes site is alwa
ys first word in plan label
            PlanName = PlanLabel.Split()[1]; // Not even sure if thi
s works for the name tbh
            PlanDate = (System.DateTime)dcm.FindFirst(TagHelper.RTPl
anDate).DData;

            PlanDeliveryMethodID = getDeliveryMethodID(dcm, out int
totalControlPoints);
                        // is using 'out' like this best practise? I
 have no idea but it works lol
            PlanTotalControlPoints = totalControlPoints;
            PlanNumberOfBeams = dcm.FindAll(TagHelper.BeamNumber).Co
unt;
            PlanFractions = Convert.ToInt32(dcm.FindFirst(TagHelper.
NumberOfFractionsPlanned).DData.ToString());
            PlanTypeID = PlanFractions == 1 ? 1 : 2; // If only 1 fr
action, it's verification, else it's treatment
            PlanDose = getDose(dcm);
            PlanEnergy = dcm.FindFirst(TagHelper.NominalBeamEnergy).
DData.ToString();
            PlanTotalMU = getTotalMU(dcm, PlanFractions);

            /* ========================= */
            /* === Talk about ways of parsing/converting data? use b
elow as example ===
            IntegerString energy = dcm.FindFirst(TagHelper.NumberOfF
ractionsPlanned) as IntegerString;
            Date planDate = dcm.FindFirst(TagHelper.RTPlanDate) as D
ate;
            CodeString[] rotDirArr = dcm.FindAll(TagHelper.GantryRot
ationDirection).ToArray() as CodeString[];

            PlanEnergy = energy.Data.ToString(); //DOES ENERGY NEED
TO BE A STRING??
            PlanDate = planDate.Data;
            */
        }

        decimal getDose(DICOMObject dcm)
        {
            decimal dose = 0;
            try
            {
                //Simple approach - find maximum dose tag (may not e
xist)
                dose = Int32.Parse(dcm.FindFirst(TagHelper.DeliveryM
aximumDose).DData.ToString());
            }
            catch (Exception e)
            {
                //System.Diagnostics.Debug.WriteLine("Error finding
maximum dose: " + e.Message);
                //System.Diagnostics.Debug.WriteLine("Using second m
ethod");

                DICOMSelector selector = dcm.GetSelector();

                var rbs = selector.ReferencedBeamSequence;
                //uses lambda function to find total amount of "beam
 dose" labels (same as number of beams)
                int beamCount = rbs.Select(s => s.BeamDose_).Count;

                //iterates over all beams
                for (int i = 0; i < beamCount; i++)
                    //increments total dose by the value of this bea
m's dose
                    dose += Convert.ToDecimal(rbs.Select(s => s.Beam
Dose_[i]).Data);
```

```
            }
            return dose;
        }
        int getDeliveryMethodID(DICOMObject dcm, out int totalContro
lPoints)
        {
            totalControlPoints = 0;

            //Gets list of all GRT entries
            var RotDirList = dcm.FindAll(TagHelper.GantryRotationDir
ection);
            DICOMSelector selector = dcm.GetSelector();

            //Iterates over each rotation direction entry
            for (int i = 0; i < RotDirList.Count; i++)
                //If gantry is rotating at all, it's VMAT
                if (RotDirList[i].DData.ToString() != "NONE") return
 getVMATID(selector);

            //If the gantry never rotates it's IMRT
            return getIMRTID(selector, out totalControlPoints);
        }
        int getIMRTID(DICOMSelector selector, out int totalControlPo
ints)
        {
            int imrtID = 1; // IMRT Fixed by default
            List<string> XLeafJaw = new List<string>();
            totalControlPoints = 0;

            // Code taken from Tim's ArcHome.aspx.cs - Sent 03/2020
            for (int myBeamCount = 0; myBeamCount < selector.Control
PointSequence_.Count; myBeamCount++)
            {
                var ControlPointPerBeam = selector.ControlPointSeque
nce_[myBeamCount].DData_.Count;
                totalControlPoints += ControlPointPerBeam;

                // Cycle through Each Control Point
                for (int myControlPointCount = 0; myControlPointCoun
t < ControlPointPerBeam; myControlPointCount++)
                {
                    var DeviceTypeCount = Convert.ToInt32(selector.B
eamSequence.Select(s => s.ControlPointSequence_[myBeamCount]).Select
(s => s.BeamLimitingDevicePositionSequence_[myControlPointCount].DDa
ta_.Count).ToString());
```

```
                    // Cycle through Beam Device Position Sequence -
 only concerned with ASMYX Leaf / Jaw Positions
                    for (int TypeCount = 0; TypeCount < DeviceTypeCo
unt; TypeCount++)
                    {
                        var DeviceType = selector.BeamSequence.Selec
t(s => s.ControlPointSequence_[myBeamCount]).Select(s => s.BeamLimit
ingDevicePositionSequence_[myControlPointCount]).Select(s => s.RTBea
mLimitingDeviceType_[TypeCount].DData);
                        var dcmXLeafPosition = selector.BeamSequence
.Select(s => s.ControlPointSequence_[myBeamCount]).Select(s => s.Bea
mLimitingDevicePositionSequence_[myControlPointCount]).Select(s => s
.LeafJawPositions_[TypeCount].DData);
                        if (DeviceType.ToString() == "ASYMX")
                        {
                            XLeafJaw.Add(dcmXLeafPosition.ToString()
);
                        }
                    }
                }

                //Leaf/Jaw check - rewritten to make use of hashsets
                HashSet<string> LeafJawSet = new HashSet<string>(XLe
afJaw);
                if (LeafJawSet.Count > 1) imrtID = 4; // If more tha
n 1 Leaf/Jaw X position per beam, it's IMRT Split
                XLeafJaw.Clear();
            }

            return imrtID;
        }

        // ========== Can't test out as don't have VMAT dcm file ===
=======
        int getVMATID(DICOMSelector selector)
        {
            int vmatID = 1; // VMAT Full by default

            // Code taken from Tim's ArcHome.aspx.cs - Sent 03/2020
            for (int myBeamCount = 0; myBeamCount < selector.Control
PointSequence_.Count; myBeamCount++)
            {
                var ControlPointPerBeam = selector.ControlPointSeque
nce_[myBeamCount].DData_.Count;
```

```
                if (selector.BeamSequence.Select(s => s.ControlPoint
Sequence_[myBeamCount]).Select(s => s.GantryAngle_[0].DData) == null
)
                    return 0; // ERROR: No gantry angle data for thi
s beam (continue instead of return?)
                var dcmFirstGantryAngle = selector.BeamSequence.Sele
ct(s => s.ControlPointSequence_[myBeamCount]).Select(s => s.GantryAn
gle_[0].DData);

                // Get Gantry Angle for Last Control Point
                var dcmLastGantryAngle = selector.BeamSequence.Selec
t(s => s.ControlPointSequence_[myBeamCount]).Select(s => s.GantryAng
le_[0].DData);
                int LastAnglePosition = 0;

                int startLoop = ControlPointPerBeam - 1;
                bool success = false;

                while (startLoop > 0 && !success)
                {
                    try
                    {
                        dcmLastGantryAngle = selector.BeamSequence.S
elect(s => s.ControlPointSequence_[myBeamCount]).Select(s => s.Gantr
yAngle_[ControlPointPerBeam - 1].DData);
                        success = true;
                    }
                    catch
                    {
                        startLoop--;
                        LastAnglePosition++;
                    }
                }


                double myDifference = Math.Abs(Convert.ToDouble(dcmF
irstGantryAngle) - Convert.ToDouble(dcmLastGantryAngle));
                if (myDifference != 2)
                {
                    vmatID = 2;
                    break;
                }
            }
```

```
                return vmatID;
            }

            decimal getTotalMU(DICOMObject dcm, Nullable<int> fractions)
            {
                DICOMSelector selector = dcm.GetSelector();

                decimal totalMU = 0;
                // Code taken from Tim's ArcHome.aspx.cs - Sent 03/2020
                for (int myFractionCount = 0; myFractionCount < fraction
s; myFractionCount++)
                {
                    // Get the Number of Fraction Group Sequences
                    int MyFractionGroupSequenceCount = Convert.ToInt32(s
elector.FractionGroupSequence_[myFractionCount].DData_.Count);

                    // Go through each Fraction Group
                    for (int myFractionGroupCount = 0; myFractionGroupCo
unt < MyFractionGroupSequenceCount; myFractionGroupCount++)
                    {
                        // Get the Number of Beam Sequences
                        int myBeamSequenceCount = Convert.ToInt32(select
or.FractionGroupSequence.Select(s => s.ReferencedBeamSequence_[myFra
ctionGroupCount].DData_.Count));

                        // Cycle through each beam Sequence and Read all
 the MU Values
                        for (int myBeamCount = 0; myBeamCount < myBeamSe
quenceCount; myBeamCount++)
                        {
                            var MUPerBeam = selector.FractionGroupSequen
ce.Select(s => s.ReferencedBeamSequence_[myFractionGroupCount]).Sele
ct(s => s.BeamMeterset_[myBeamCount].DData.ToString());
                            totalMU += Convert.ToDecimal(MUPerBeam);
                        }
                    }
                }

                return totalMU;
            }
            public int ID { get; set; }
            public Nullable<int> PatientID { get; set; }
            public Nullable<int> PlanTypeID { get; set; }
            public Nullable<int> PlanDeliveryMethodID { get; set; }
            public string PlanName { get; set; }
```

```
        public Nullable<System.DateTime> PlanDate { get; set; }
        public string PlanSite { get; set; }
        public string PlanEnergy { get; set; }
        public Nullable<decimal> PlanDose { get; set; }
        public Nullable<int> PlanFractions { get; set; }
        public Nullable<int> PlanNumberOfBeams { get; set; }
        public Nullable<int> PlanTotalControlPoints { get; set; }
        public Nullable<decimal> PlanTotalMU { get; set; }
        public string PlanPatientFolder { get; set; }
        public string PlanComments { get; set; }

        public virtual ArcDeliveryMethod ArcDeliveryMethod { get; se
t; }
        [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.
Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]
        public virtual ICollection<ArcMeasurement> ArcMeasurements {
 get; set; }
        public virtual ArcPatient ArcPatient { get; set; }
        public virtual ArcPlanType ArcPlanType { get; set; }
    }
```

Appendix IV – PythonTagExtraction.py:

```
"""
Assumptions:
- Delivery Type for each beam will always be the same (not used - ca
n we use this Treatment Delivery Type variable?)
    - Treatment Delivery Type entry? Or Plan Intent??
- "Plan Energy" is the same as each beam's energy (beam energy never
 changes)
- Patients never have one fraction (1 fraction = verification plan)
- Total dose is equal to the sum of all beam doses

"""

import pydicom as pdc

def isIMRT(beamSeq):
    try:
        for i in range(len(beamSeq)):
            direction = bs[i].ControlPointSequence[0].GantryRotation
Direction
            if (direction != "NONE"):
                return False
        return True
    except AttributeError:
        print("Couldn't find gantry direction!")

def getIMRTType(beamSeq):
    totalControlPoints = 0
    IMRTMethodID = 3 # IMRT Fixed by default
    XLeafJaw = []
    numBeams = len(beamSeq)

    # Lots of looping here: each beam -> each control point -
> each BLDPS
    for i in range(numBeams):
        ControlPointSequence = beamSeq[i].ControlPointSequence # thi
s and BLDPS are to just to look neater
        totalControlPoints += len(ControlPointSequence) # increment
total control points by the points in each beam

        for j in range(len(ControlPointSequence)):
            BLDPS = ControlPointSequence[j].BeamLimitingDevicePositi
onSequence # simplify code, same as ControlPointSequence variable
```

```
            for k in range(len(BLDPS)):
                if BLDPS[k].RTBeamLimitingDeviceType == "ASYMX":
                    XLeafJaw.append(BLDPS[k].LeafJawPositions)


        # If any Leaf/Jaw positions are different in the same beam,
it's IMRT Split:
        XLeafJawSet = set(map(tuple, XLeafJaw)) # looks weird but be
st solution to getting set of 2d arrays (list of unique values)
        if len(XLeafJawSet) > 1: # if set contains more than one val
ue it's split
            IMRTMethodID = 4
        XLeafJaw = []

    return [IMRTMethodID, totalControlPoints, numBeams]

def getVMATType(beamSeq):
    VMATMethodID = 1 # VMAT Full Arc by default
    totalControlPoints = 0

    firstGantryAngle = beamSeq[0].ControlPointSequence[0].GantryAngl
e
    lastGantryAngle = 0

    for i in range(len(beamSeq)):
        ControlPointSequence = beamSeq[i].ControlPointSequence

        if ControlPointSequence[0].GantryAngle == None : return [Non
e, None, None] # If there's a null gantry angle we probably have iss
ues
        lastGantryAngle = ControlPointSequence[0].GantryAngle

        totalControlPoints += len(ControlPointSequence)

    if abs(firstGantryAngle - lastGantryAngle) != 2:
        VMATMethodID = 2 # if first and last aren't (basically) the
same it's partial

    return [VMATMethodID, totalControlPoints, 0] # Number of beams i
s 0 as VMAT doesn't use beams (???)

def getDose(beamSeq):
    dose = 0
    for i in range(len(beamSeq)):
        dose += beamSeq[i].BeamDose
    return dose


# ========== CODE BELOW ==========

dcm = pdc.dcmread("new_Anon1.dcm") # Reads in dicom file to object

# ===== ARCPATIENT FIELDS =====
name = dcm.PatientName
[FName, LName] = [name.given_name, name.family_name]
nhs = dcm.PatientID

# ===== ARCPLAN FIELDS =====
# Shorthand Variables
bs = dcm.BeamSequence
fgs = dcm.FractionGroupSequence[0]

# Get Fields
planLabel = dcm.RTPlanLabel.split()
planDateFull = dcm.RTPlanDate
planSite = planLabel[0]
planName = planLabel[1]
planDate = planDateFull[6:8] + "/" + planDateFull[4:6] + "/" + planD
ateFull[:4]


numberOfFractions = fgs.NumberOfFractionsPlanned
dose = getDose(fgs.ReferencedBeamSequence)

planType = 1 if numberOfFractions == 1 else 2 # If 1 fraction then p
lan is verification, else treatment (weird python ternary operator s
yntax)
planEnergy = bs[0].ControlPointSequence[0].NominalBeamEnergy

[deliveryMethodID, totalControlPoints, numberOfBeams] = getIMRTType(
bs) if isIMRT(bs) else getVMATType(bs)



print()
print("===== ArcPatient Fields =====")
print("Patient Name: ", LName + ",", FName)
print("NHS Number: ", nhs)
print()
```

```python
print("===== ArcPlan Fields =====")
print("Plan Name:", planName)
print("Plan Site:", planSite)
print("Plan Date:", planDate)
print("Delivery Method ID:", deliveryMethodID)
print("Control Points per Field:", totalControlPoints)
print("Number of Beams:", numberOfBeams)
print("Number of Fractions:", numberOfFractions)
print("Plan Type:", planType)
print("Energy:", planEnergy)
print("Dose:", dose)
```