Designing and developing a points-based algorithm and online application to identify high-risk individuals who should be prioritised for a future COVID-19 vaccination

Sam Byford
15620782

15620782@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. Tryphon Lambrou

May 2021

Abstract

Since its discovery in 2019, COVID-19 has spread across the globe resulting in over 100 million deaths (World Health Organization, 2020a). Its high mortality and infection rates have engulfed the world in mass hysteria with the only hope of returning to normality reliant on the success of vaccination programmes. This project aims to develop an algorithm to highlight certain individuals who should be prioritised and fast-tracked for a COVID-19 vaccination. The process of vaccinating the ~7 billion global population and deciding who gets priority is one of the largest logistical challenges the world has ever faced. This project improves upon existing age-based prioritisation strategies by analysing a multitude of patient preconditions to produce a risk score specific to the individual. The algorithm will be accessed through a secure, online, application that will be functional across a multitude of devices. Algorithm accuracy will be analysed, and the application will be evaluated using the System Usability Scale (SUS). Results from these evaluation methods displayed accuracy of ~87% for the algorithm and a mean SUS score of 76.4. These metrics suggest that the application is straightforward to use and can successfully flag a large portion of at-risk individuals.

Table of Contents

List of Figures

Chapter 1

Introduction

First described in 1966 Coronaviruses are single-stranded RNA viruses that infect both humans and animals (Velavan and Meyer, 2020). Specifically, the coronavirus SARS-COV-2 and the disease it causes, COVID-19, is believed to have originated in animals and then made a successful transition to humans in late 2019. Structurally, the SARS-COV-2 virus is closely related to the other two coronavirus outbreaks that have occurred since the turn of the millennium, SARS (2002-2003) and MERS (2012-Present). However, what makes SARS-COV-2 more threatening to global health is that it is much more contagious - the speed of spread and the infectivity of COVID-19 is far and above that of MERS and SARS (Goh et al., 2020). This has led to a situation where COVID-19 has spread to 215 countries and territories (Worldometer, 2021). This is compared to SARS and MERS which only spread to 5 (World Health Organization, undated) and 27 (European Centre for Disease Prevention and Control, 2021) countries respectively. Most cases of COVID-19 are minor with symptoms including fever, cough, and fatigue; however, some people are not so lucky with severe cases leading to multiple organ failure. With a ~4.2% mortality rate, (Hu et al., 2020) over 100 million deaths have so far been recorded as a direct result of the disease (World Health Organization, 2020a).

To combat the spread of the virus, countries across the globe have placed some of the toughest travel and social restrictions that have ever been seen in the modern era. The cold war, the war on terror and the credit crunch of 2008 have all caused lasting impacts on society but none of them has been as wide-reaching and impactful to so many lives as that of COVID-19. Almost the entire population of the world is affected by the virus in some way or another.

The only realistic path out of the instability the virus has caused is by triggering herd immunity through vaccination. The first and most promising vaccinations to be introduced globally were the Pfizer-BioNTech and Oxford-AstraZeneca vaccinations. They have great efficacies of 95% (Olliaro, 2021) and 63.09% (World Health

Organization, 2021) respectively. However, with these vaccines being developed, a major logistical problem now arises. What is the best way to ensure a fair and accurate distribution of doses to the worlds ~ 7 billion population? Due to risk increasing exponentially as you get older (GOV.UK, 2021), many governments, including in the UK, quickly created priority groups based almost entirely on age. This does ensure a quick rollout of the vaccine; however, it is not entirely fair as it does not necessarily guarantee the most high-risk individuals are protected first. Younger patients with pre-existing conditions that are badly affected by COVID-19 are vaccinated after healthier elderly patients.

This is where 'CoV-Risk', the application built for this project, comes into effect. CoV-Risk aims to efficiently flag and notify individuals who should be prioritized for a COVID-19 vaccination and present the data in a clear, logical manner. It does this by analysing patients on a more individual basis. Instead of simply stating that patients over 80 get the vaccination first, CoV-Risk utilises a custom points-based algorithm to examine the patients' pre-existing conditions, sex, and age to produce a risk score. The higher the risk score, the higher the patient's priority. This custom-built algorithm is accessed by medical professionals using a fully secure, interactive, website application.

To complete the aim of the project, objectives were created to help guide the development (see below). These objectives lay out clear markers for the projects programming with the development conducted using R, Angular, ASP.NET Core and SQL. The created website was then evaluated in two stages (mid and post-development) using the System Usability Scale to measure the usability of the application.

1. Compile an extensive list of the factors that lead to series illness with COVID-19.
2. Assign point values (weights) to the factors based on how influential they are to causing a bad case of COVID-19 and calculate a 'high-risk threshold' value.
3. Develop the points-based algorithm that analyses supplied data and uses the weighted factors to mark patients that are high risk.
4. Design and develop a secure and intuitive website that will provide a user interface for accessing and displaying the algorithm.

5. Build a database to house the results of the algorithm.
6. Deploy the algorithm onto a server and create an API to link the website to the server.
7. Create an email system to notify users who have been marked as a vaccine priority.
8. Ensure the results from the algorithm are presented understandably on the website using graphs, tables, and charts.

The report will further expand on the points discussed in this section and outline what research & development was conducted to meet the objectives and in turn the overall aim of the project.

Chapter 2

Literature Review

The disease COVID-19 is one of the greatest threats the united global community has ever faced. Originating with a handful of cases in the Chinese city of Wuhan in December 2019, the disease has spread rapidly across the globe, infecting 111,102,016 people, and causing 2,462,911 deaths as of 22nd February 2021 (World Health Organization, 2020a). Hundreds of studies have been conducted into the threat COVID-19 presents, identifying trends and patterns within the disease and possible strategies to help return life to normality. The following section will examine a selection of these studies to contextualise the work carried out in this project and provide justification for why it was necessary.

2.1 Background and Justification

Meo et al. (2020) conducted a study into certain biological trends in mortality due to COVID-19 outbreaks. The study emphasizes how the virus spread at an unprecedented speed due to it being a zoonotic disease (spread from animals to humans) that can transmit from person to person through droplets or contact (Meo et al., 2020). This creates a situation where merely standing close to an infectious individual for a prolonged period puts someone at risk. Social distancing and restrictive quarantine policies were adopted across the world to try and curve transmission, however, the results from Meo et al.'s study show that these strategies were not introduced quickly enough. By March 2020, just three months after the initial outbreak, the epicentre of the virus had shifted from central China and into the west, with European countries and the United States of America (USA) providing most cases. This is highlighted by the authors' results, as of March 31st, 2020, the USA held 22% of the global coronavirus cases (164,744). Italy (101,739) and Spain (94,417) were just behind with 13%. On its own an increase in cases was not a huge issue, however, as seen in figure 1 deaths were also exponentially increasing, going way above the linear trend also seen on the graph. Meo et al. (2020) conclude from their results that the number of cases was positively linked to the death rate of COVID-19 infection. This was more than likely due to the

virus' prevalence among the older generation with the median age of infection being 59 years between December 29th, 2019-March 31st, 2020 (Meo et al., 2020). The virus was deadly and had spread to uncontrollable levels. Due to the lack of knowledge and vaccine research, a lockdown of almost global proportions was required.

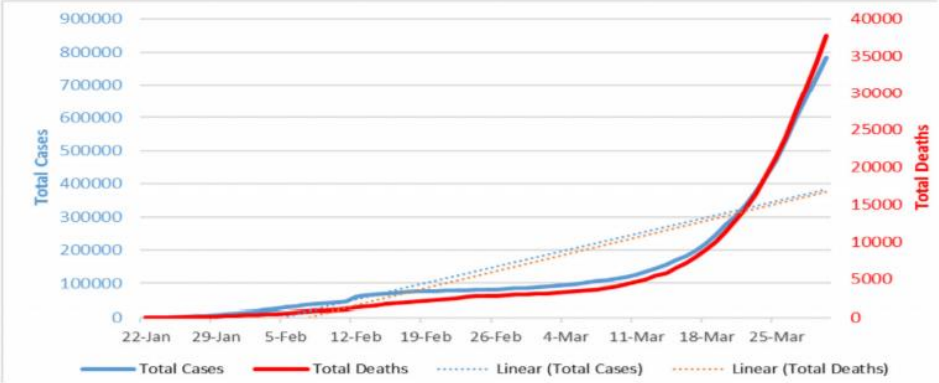


Figure 1: A Graph showing the total number of deaths and cases due to COVID-19 from the end of January 2020 to the end of March 2020 (Meo et al., 2020)

The study conducted by Meo Et al. demonstrates how COVID-19 spread so easily and where it became most focussed during its infancy. Although the data presented in the study is outdated in some regards, examining how the virus was able to dominate globally in its first 12 weeks provides stark evidence of just how contagious this early variant was. Meo et al. mentioned how the virus was spreading with mutable prevalence (2020) a statement that held when multiple mutations of the virus began to appear towards the end of 2020. Current mutations (as of February 2021) of the virus, such as 501Y.V2 which was first identified in South Africa, have been shown to make the virus more transmissible due to their higher viral loads (Mahase, 2021). This strengthens the case for conducting the work carried out as part of this project. With such a transmissible and deadly virus, it is vital that those most at risk are identified, protected, and vaccinated first. Improving on existing strategies by accurately calculating who is most at risk on an individual basis, 'CoV-Risk' can advise people to shield themselves from any form of person-to-person transmission while they wait to receive their vaccination.

With both cases and deaths rising exponentially it was clear that a vaccination was going to provide the only definitive way out of the pandemic. Although effective while in place, lockdowns and quarantines could not be enforced indefinitely as they would crush the global economy and put a huge strain on individuals' freedom and mental wellbeing. Frederiksen et al. (2020) discuss the needs for a vaccination and issues surrounding vaccinating enough of the world population to trigger herd immunity. The authors expand on the points previously made by Meo et al. on how deadly the virus is by stating that as of July 2020 it had a mortality rate of ~3.7% and that it can develop into severe pneumonia and multiple organ failure in elderly patients and those with pre-existing health conditions. A separate study conducted by Liu et al. on patients with severe COVID-19 in China found that patients who are critically ill with the disease were more likely to have conditions such as hypertension and diabetes (2020). This justifies the project as it shows that there is a clear need to identify those with these pre-existing conditions and to find other conditions that would be affected by COVID-19. In turn, this will then allow for an accurate and fair distribution of the vaccine.

The World Health Organization (2020b) state that immunization prevents 2-3 million deaths every year and Frederiksen et al. describe vaccines as representing, "One of the most successful and cost-effective health interventions in human history" (2020). As stated by the authors, herd immunity is the immunisation of a large portion of the population to protect the non-vaccinated and immunocompromised by reducing the number of vulnerable hosts to a level below the transmission threshold (Frederiksen et al., 2020). There is also a separate type of herd immunity known as natural herd immunity which is caused when a certain percentage of the population develops immune protection after exposure to the virus. However, letting the virus spread through the population without a vaccine is very dangerous and ineffective. In an article by Moghneih et al. (2020) the authors discuss how European studies had reported low levels of seropositivity ranging from 0.0% to 8.5%. This means there was only a very small number of antibodies, the cell the body uses to fight the virus, present in the tested patients who had previously had COVID-19. This is not a high enough level for indirect protection from the virus (Moghneih et al., 2020). Herd immunity via immunisation is much safer and has long been a successful strategy, preventing the spread of virus' such as smallpox. Over 80% of the population has been vaccinated against this virus resulting

in its near extinction (Frederiksen et al., 2020). The goal set for COVID-19 herd immunity is a much lower value at ~ 67% (Kwok et al., 2020). This would require 12-16 billion doses of the vaccination in the case of multi-dose vaccines such as the ones that are currently in circulation from Pfizer-BioNTech and Oxford-AstraZeneca.

Producing this many vaccinations is a huge logistical challenge. Although reports of a shortage of medical glass used to package the vaccine are unfounded, there is still a bottleneck in the production of the doses with even the fastest filling robotic arm being too slow to keep up with demand (Feinmann, 2020). This highlights the key problem addressed in the project, how is it decided who gets these first doses of the vaccine? A prioritisation strategy must be implemented, and high-risk individuals must be classified. The study by Clark et al. (2020) looks at doing exactly this. It examines the underlying health conditions affected the most by COVID-19 and calculates the percentage of the population at an increased risk from the virus. The authors used the conditions listed in the Global Burden of Diseases, Risk Factors, and Injuries Study (GBD) and other global health agencies to produce 11 categories of ailments that are associated with severe COVID-19 - these 11 categories include conditions such as cardiovascular disease, kidney disease, HIV/AIDS and diabetes (Clark et al., 2020). Clark et al. were then able to estimate that 1.7 billion people (22%) of the world's population are at a higher risk of severe COVID-19 with at least one underlying health condition (2020). The goal of this project is to help identify these individuals. Vaccinating the most vulnerable patients first will result in a sharp decrease in the global death rate, lowering the threat of the virus to be comparable with diseases such as seasonal flu, and thus authorise the reopening of society. As Campo-Matos et al. (2020) also explain, protecting the most vulnerable helps to maintain resilience in health services such as the NHS. In turn, this further protects the most vulnerable by ensuring these services can provide care for non-COVID conditions.

Unfortunately, current government strategies lean away from the specific precondition approach and adopt a more general system with a heavy focus on patient age.

2.2 Current Implementations and Strategies

Campos-Matos et al. (2021) assess the inequalities in the UK governments prioritisation strategy of ensuring care home residents, elderly people, health and social care works, and people with underlying medical conditions are all vaccinated first. The authors expand on this by stating that, “Individuals with underlying health conditions at highest risk are those considered ‘clinically extremely vulnerable’ and are determined by a clinical panel independent of JCVI (The Joint Committee on Vaccination and Immunisation which provides advice to the UK government)” (Campos-Matos et al., 2021). On the surface this strategy sounds very good as many major risk groups are addressed first, however, as the authors' highlight, the current prioritisation actively targets those of older age to reduce the health inequalities between age groups (Campos-Matos et al., 2020). This leads to situations where a healthy 70-year-old patient could receive the vaccination before a younger, more vulnerable, 30-year-old. Age is the main focus of this strategy when instead it should simply be a factor. The reason for this policy is due to JCVI’s key goal being to administer a vaccination at pace. Delivering the vaccine sporadically across age groups would add complexity to the scheme (Campos-Matos et al., 2020). Yet, to save the most lives possible, we should give the vaccination to those most in need first, regardless of age. The strategy should be based on effectiveness and not on speed; Patient-specific conditions should be considered in line with their age.

Research conducted by Shang et al. (2020) looks to address the issues caused by existing government strategies by creating a scoring system for predicting the mortality of patients with COVID-19. The study uses data provided by three hospitals in Wuhan to analyse the effects of certain pre-conditions and whether they lead to a patient dying from severe COVID-19. The authors' research found that of the patients analysed there were far more men than women with the disease and that the most common comorbidities were hypertension, heart disease and diabetes (Shang et al., 2020). Shang et al. then generated their scoring system by selecting these comorbidities and performing Least Absolute Shrinkage and Selection Operator (LASSO) regression to simultaneously identify the most influential and nullify the least influential factors in severe COVID-19 cases (2020).

To understand LASSO, basic logistic regression must be discussed. Logistic regression allows for the prediction of a discrete outcome from a set of variables that may be continuous, discrete, dichotomous or a mix of all (Franklin and Emmanuel, 2014). In other words, logistic regression can be used to predict if a patient is a high or low risk for developing severe COVID-19 by examining the risk factors (age, sex, blood pressure etc) of the patient. It does this by producing coefficient values for each factor that help to highlight which are significant for prediction. LASSO is a regularization method that manages overfitting and reduces the complexity of the logistic model by shrinking the least significant variables (factors) coefficients' to zero (Ranstam and Cook, 2018). These variables are then excluded from the prediction model.

After performing LASSO, Shang et al. converted the remaining coefficients into the scoring system of COVID-19 (CSS) as seen in figure 2. The factors are assigned a score between 0 and 2 depending on how significant the factor is. If the patient's score rises above 2, they are considered high risk. As shown in the figure, age still plays a key role in COVID-19 severity with age over 75 giving a patient the highest score of 2. However, there is now another factor that is considered on par, the procalcitonin biomarker - the presence of procalcitonin in the blood is used to identify patients with sepsis and full-body bacterial infections (Jin and Khan, 2010). This means that unlike current prioritisation strategies it is now possible for a younger vulnerable patient to be classed as higher risk than older healthier patients and therefore receive a vaccination quicker. For example, using the authors' system a 40-year-old with high levels of procalcitonin and coronary heart disease would be prioritised over a healthy 75-year-old.

Variables		Score	Risk levels
Age	<60 years	0	High risk: Total score > 2 points Low risk: Total score ≤ 2 points
	60-75 years	1	
	>75 years	2	
Coronary heart disease		1	
Percentage of lymphocytes < 8%		1	
Procalcitonin > 0.15 ng/ml		2	
D-dimer > 0.5 ug/ml		1	

Figure 2: Table showing the COVID-19 Scoring System (CSS) (Shang et al., 2020)

The study by Shang et al. provides the biggest influence on this project's creation, however, it is still not perfect. Due to the limited data provided to the researchers at the early stages of the pandemic, only a small number of conditions were considered significant and analysed. As will be demonstrated later, to produce a thorough system additional conditions need to be examined by accessing more complete datasets.

Chapter 3

Methodology

3.1 Project Management

This project has three main sections as listed below:

1. Research and data-collection
2. Website development
3. Evaluation of the artefact

The first section involved the gathering of COVID-19 patient datasets which were then analysed to create the high-risk prediction algorithm. The second section was concerned with the design and development of a fully interactive website. This website allows users to access the previously created prediction algorithm. The artefact was then evaluated with user testing for the third and final section.

To help manage the projects three section's, a few different project management methods were implemented. The first of which was the Gannt chart as seen below. Gannt charts are one of the most popular tools for project management in use today (Geraldi and Lechter, 2012) as they provide a time-controlled graphic visualisation of important tasks within a project. In other words, they allow project managers to easily see how long key tasks will take, when they should begin and when the whole project will be finalised by.

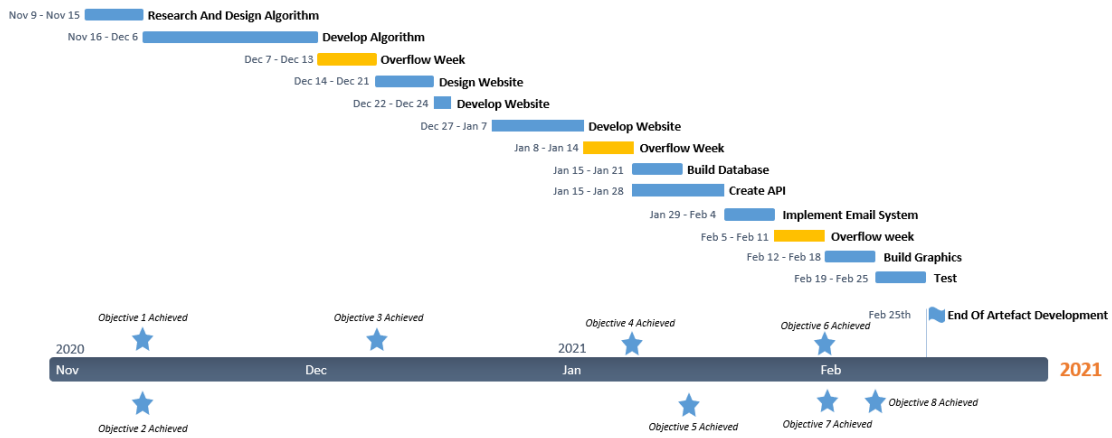


Figure 3: Project Gantt Chart

This chart was used to a relatively good effect during the project to keep on track with tasks and see what needed tackling next. The star icons at the bottom of the chart visualise when each project objective was successfully met. This motivated development as it helped to highlight when key goals were achieved and what needed to be done to achieve the next goal. However, despite their advantages and usefulness, Gannt charts do have some disadvantages that did become clear when managing this project. As Geraldi and Lechter state, "project managers may create an exaggerated time pressure and stress, encouraging unnecessary concurrent work or rework" (2012). Stress to meet the time constraints set out by the chart became a real issue during the project, causing development to slow and subsequently even more stress to pile on. To help mitigate this problem 'overflow' weeks were added to the chart, as shown by the orange boxes. These weeks provided a means to catch up if development was behind schedule as no new development was conducted. They helped to get the project back on track if it had previously stalled.

Arguably the most important tool used to manage the project was the development of checklists which were broken down into sections and sub-sections. These checklists were developed using Microsoft OneNote and covered every section of the project. Each checklist controlled an individual chunk of the project with lists such as 'Configuration' describing the base setup of the artefact environment.

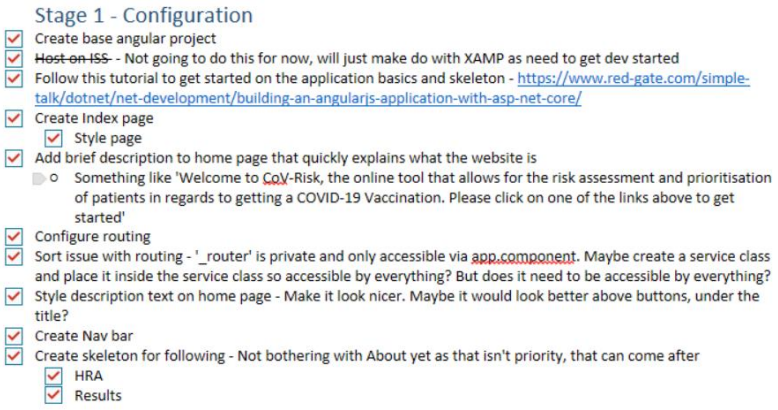


Figure 4: Example of a checklist used to manage the project

The lists were brilliant for managing the project; however, they too had some flaws. Due to the large number of ways the project was broken down, there were many lists, each with a huge number of points. This made visualising the tasks and understanding where they fit into the overall project quite hard to manage. Here is where the Gannt chart helped considerably. The checklists described in detail what tasks needed completing, the Gannt chart then visualised where in the project this task was located, thus helping to see what objective was being worked towards.

From all these project management methods and tools, it is clear to see the guiding principles and demands of the project. Due to its multi-layered structure, the project had to be broken down into standalone chunks, known as a divide and conquer approach. This aided the development from getting overwhelming, provided a clear structure to the assignment and made the overall objectives of the project more manageable to achieve. As well as this the project was under strict time restraints meaning it had to be managed in a time-efficient manner to ensure its full completion.

The demands of the project, as well the tools and methods used, show similarities to the Scrum management methodology. Scrum is a management framework used for existing systems or production prototypes (Schwaber, 1997). It is an agile methodology – "Group of software development methods that are based on iterative and incremental development" (Kumar and Bhatia, 2012) - that breaks projects into timed (2-4 week) sprints. These sprints are extracted from a product backlog, managed by a small team, and focussed on a key feature of an application.

The projects checklists were considered the ‘product backlog’ in that each list focused on a key feature of the project. Individually each checklist was a separate sprint, taking around a few weeks to complete, and the points on the list were the sprint backlog. Being a one-man team there were some limitations to this methodology, the most notable being that there were no separate roles for different team members. The project manager and the scrum leader were the same people meaning there was a lot more to manage on an individual basis. Despite this, lone meetings/discussions were still held when planning what work could be committed to in a sprint and at the start of each development day, the previous day's work would be analysed so any problems could be addressed.

Overall, the time-bound nature of the scrum methodology was perfect for this project as it ensured work was kept on schedule, meeting the demand of the time restrained assignment. The separation of work into sprints highlights the divide & conquer approach required in the project characteristics. Focussing on one key feature at a time, the project could be managed in a way to ensure steady progress was continuously made whilst avoiding any development burnout. Although this methodology does not fully fit with the project, it is by far the best match out of other similar methodologies.

3.2 Software Development

The artefact developed for this project is a website application that will be accessible on a vast array of devices. Being cross-platform is an important demand of this project as it ensures that the application can be used by as many people as possible. An admin employee working at a desk and a nurse working on a ward could use the application to the same effect. The application itself was developed in the separate phases listed below:

A. Initial development of key features
B. User testing and evaluation of usability
C. Further development based on user feedback and additional quality of life improvements
D. Final user testing to evaluate the usability of the application

User testing provides valuable insight into usability problems in both finalised and in-progress applications (Tan Et al., 2009). It was carried out in this project through the

means of the System Usability Scale (SUS) which is a simple 10 item questionnaire that has become an industry standard for measuring the usability of applications (Usability.gov, undated). User testing is vital to this project as it guaranteed the application remained easy to use and showed which areas needed improvements. Most importantly it protected against inadvertently developing the application feature-centred as opposed to user-centred. If the application were developed as feature-centred it would result in their being parts of the tool that are potentially useless to the end-user. User-centred design helps to guard against this by only developing the exact features that users require.

User testing also massively helped with the requirements capture of the project. Due to COVID-19 restrictions, it was hard to interview healthcare professionals directly. Therefore, although not perfect, the initial requirements were obtained through research around the problem domain via academic sources and governing bodies. These requirements were then refined through the user testing of the application - any requirements that should have been considered were highlighted and added.

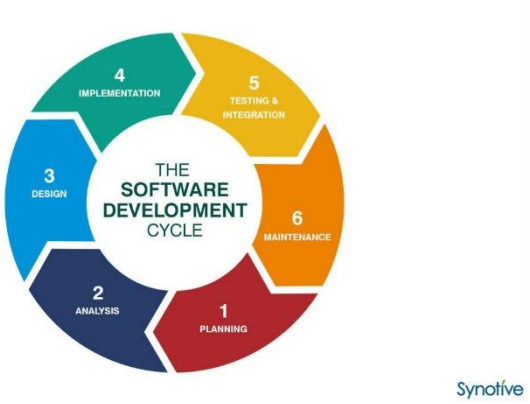


Figure 5: Software Development Lifecycle (Pinherio, 2018)

The use of development followed by multiple stages of user testing and further development indicates the adoption of the Software Development Lifecycle (SDLC) (figure X). As explained by Ruparelia (2010) SDLC is a framework that considers the structure of the stages involved in the development of an application from its initial conception, through to its deployment in the field and maintenance. This project follows the outlined framework as the application was planned, designed through wireframes, developed, and tested by users. It was then maintained and improved via this user feedback, which then looped back into the planning stage and so on.

Combined, the characteristics of the application and the demands of the project from a software development viewpoint validate choosing Scrum as the main methodology for the project. As it is an iterative methodology the artefact is improved upon gradually as development progresses. This matched perfectly with the demand for user testing. Features are developed in sprints, tested by users, and then improved upon in future sprints. Scrum provides flexibility to the project that other methodologies such as the waterfall model would not allow. The waterfall model follows a strict 'flow' of steps from requirements to maintenance that, unlike agile methodologies, cannot be repeated. Petersen et al., state that a commonly accepted problem with the waterfall model is its inability to cope with change (2009). For this project that is not appropriate as it means feedback from user testing will not be integrated into the application. The artefact at the final stage of the model is the *final* design.

Another way in which Scrum enabled the requirement of user testing was by making sure the most valuable features were worked on first. Therefore, after the first phase of development, users were able to test a polished version of the application. This led to constructive feedback that was less focused on what was missing from the website and more focused on what could be improved with the existing features.

However, much like from the project management side, Scrum does have its drawbacks from a software development perspective. The main issue was that the entire development process was carried out by one developer. This could have resulted in the artefact being rushed to stick to the timeframe, in turn, leading to it not being as polished as it could be. Other methodologies such as Kanban might have been more useful in this regard as they are much better at visualising work (Wakode et al., 2015). By utilising Kanban boards that separate work into 'ToDo', 'Doing' and 'Done', Kanban's visualisation would have provided a huge aid to relieve the rush and pressure of developing as a one-man team. It is easy to see what is currently being done and what needs to be done next, development is not contained within many sprints.

However, even though Kanban would have provided better visualisation to the development of the artefact, it was not a good fit for this project since the methodology is not time-restricted. Scrum's time restrains were necessary as the project needed to be completed within a strict window. It is possible that with Kanban progress would have been much slower, leading to a half-complete application. To mitigate the lack of team

structure within the project, sprints were slightly longer than normal at around 5 weeks. This allowed plenty of time to carry out work and maintain a high standard of development whilst simultaneously making sure the project stayed on track.

3.3 Toolsets and Machine Environments

3.3.1 Management Tools

The two main project management tools that were selected for this project were Microsoft OneNote and Microsoft OneDrive. The latter was utilised to keep backups of code files, ensuring there was no single point of failure. If the development machine failed, work was not lost and it could have been carried on, on another device. OneDrive was chosen over competitors such as DropBox due to its seamless integration with other Microsoft apps and the fact that University of Lincoln students are given 1TB of storage for free. DropBox does have superior file version history, with 180 days compared to OneDrive's 30 (Microsoft Undated). However, for this project cloud storage will not be used for version control and will be purely looked to for file restoration - an aspect in which OneDrive excels.

OneNote acted as the focal point where all useful information for the project was collated – Tutorials, development logs, wireframes, and other project-related data were all stored on OneNote. The reason for using OneNote was mainly due to the organisational power that it provides, allowing users to sort work into notebooks, sections, and pages. This structure was perfect for managing the project; there is a clear separation between different aspects of the project and finding required information is quick.

However, OneNote is not a dedicated task-tracker, and it does lack features that other productivity tools such as Trello showcase. Trello is a project management tool that provides users with the means to create 'boards' (ToDo lists) and 'cards' (sub-lists) to manage projects (Trello, 2021). The cards can be integrated with calendars, images, and maps. This creates a system that provides users with an easy way to see what tasks are currently underway, what is completed and what needs to start. Unfortunately, Trello fails in its ability to efficiently organise and store 'other' information such as

development logs and wireframes. With the card format, these pieces of information would be harder to locate as they would get lost amongst all the various lists on the board. OneNote provided a best of both worlds approach. Although lacking in task management, it still supplied ample features to monitor development progress.

3.3.2 Development Tools

3.3.2.1 Frameworks and Libraries

Due to the large nature of this project's website application, it became clear that using a framework and/or a selection of programming libraries would massively help with development. As described by Salas-Zárate et al., a framework is 'a high-level solution for the reuse of software pieces, a step forward in simple library-based reuse that allows for sharing common functions and generic logic of a domain application' (2015). This means developers can create more sophisticated applications without directly increasing the complexity of development.

For the website creation, the framework Angular and the JavaScript library React were considered. The main characteristics and features of these tools can be seen in table 4 of appendix 1.

As is clear from this table, the core difference between the Angular framework and the React library is that the framework comes with many built-in modules whereas React requires features, such as routing, to be imported externally. This means that Angular is much quicker to get started with as it requires less installation than React. However, there is an advantage to Reacts more lightweight approach – it is flexible and customisable. You can pick and choose exactly which modules and features to integrate. There is no point in having a module for web forms installed when you are not creating any forms in your application.

To fully compare these toolsets, we must look at the type of machine environments in which the application will be required to operate. One of the major user requirements of this application is that it can be used quickly to assess the priority of patients; therefore, in a time-critical scenario, medical professionals would want to be able to run the application on whichever device they have closest to them, regardless of its performance abilities. The

website must work well on lower-powered devices. In this sense, React may have a slight edge due to its less demanding structure. Reacts credibility as a mobile supported library can also be seen in the mobile framework React Native which uses the React Library as its main foundation.

Being open-source, React is much more flexible than Angular and has a huge amount of community support. It allows you to use a plethora of libraries & frameworks, with fewer restrictions and less emphasis on file structure (ReactJS, 2021). With a more simplistic approach, React also has a smaller learning curve than full frameworks like Angular, you only need a decent understanding of JavaScript.

This all creates the image that Angular is a big complex framework that is very hard to understand and customise. However, that could not be further from the truth. Angular does come with 'out of the box' modules, but it also supports third-party modules, giving developers a 'best of both worlds' approach. Additionally, Angular separates the HTML and typescript code into separate files which creates a more straightforward structure than React which relies on JSX (a mixture of HTML and JavaScript in one file). This separation means that interrupting Angular projects is usually easier than understanding projects built with React.

Despite its steep learning curve, development for this project was carried out using Angular due to its 'all in one' nature and clear structure when compared with React. A large number of development hours were saved using it as time did not have to be spent researching and integrating additional libraries.

3.3.2.2 Server-Side Technologies

It was decided early on that a custom API would need to be created to handle the websites logic and data manipulation. The API would contain the algorithm logic, security authentication, database CRUD, file I/O operations and error handling. By housing the data logic in an API separate from the website code, we can dynamically display different content when needed and easily store information in databases for use across multiple sessions (E.g., saving uploaded patient records so they will be accessible when next using the application) (MDN Web Docs, 2021). This API will be created with C# using the ASP.NET Core framework, which was chosen due to its rich features, in-depth

documentation, cross-platform support, and easy integration with the Angular framework (Ragupathi et al., 2017).

Visual Studio (VS) was utilised to programme all aspects of the application. However, Visual Studio Code (VSC) was also considered as its lightweight modular structure meant that there was a large amount of freedom to customise the IDE. As they are both created by Microsoft, they share the same core features with IntelliSense code suggestion, code refactoring, git version control and debugging tools. However, VS was chosen over VSC for similar reasons as picking Angular over React. By using VS, the time to start development was quicker with its huge array of project templates and time-saving features that VSC does not have built-in. VSC would have required time to be spent researching and installing the best extensions for the application. VS also has built-in local server functionality, meaning a development server can be set up to view and debug the application extremely quickly. A full comparison of the tools can be seen in table 3 of appendix 1.

As previously described, a database was needed for persistent storage that would be accessible across sessions. Microsoft SQL Server Management Studio (SSMS) was the software used to create the database on the local development machine and then to modify the live Azure SQL database (to be discussed later). SSMS was chosen due to its large feature set, ease of use and integration within the Microsoft ecosphere. Visual Studio and the .NET framework provide tools to consolidate databases created in SSMS. Faster non-SQL options were considered but for the relational datasets that this application stores, SSMS was adequate.

3.3.2.3 Cloud Providers

Due to COVID-19, gathering users to test the developed application was tricky. The application was hosted online so users could access and test it remotely. To do this cloud service providers (CSPs) were examined as they allow applications to be hosted on virtual resources and then accessed by users on any device. Only two resources were required, a server to host the application and a SQL database to store the patient and user information.

Configuring cloud hosting was not key to meeting the project objective's and as such whichever provider offered the most simplistic solution was the one that was selected. This ensured more time was focused on the artefact development. With the view of simplicity,

only PaaS tools offered by these CSPs were evaluated (AWS Elastic Beanstalk, Google App Engine and Azure App Service). As stated by Microsoft Azure, what makes PaaS tools special is that they automatically manage the complex infrastructure and middleware of cloud resources (2021). This meant as much time as possible was spent on the more important programming and development aspects than on the hosting and maintenance of the application. The chosen service for this project was Microsoft Azure as it offered a good balance of security, student credit allowance and most importantly it had integration built into Visual Studio. Although being the leading provider in the world, (Kamal et al., 2020) AWS only offered $35 free student credits (Bar, 2015) and therefore was ruled out of contention as it would have required the most basic, slow, configuration for the server, sacrificing usability considerably. Google offered $300 of student credits (University of Michigan, 2021) but ultimately was let down due to its complexity. It required additional third-party visual studio extensions to be utilised.

Being forced to host the application did help considerably with usability testing and with monitoring how well the app functioned in a real production environment. Although developed on a windows machine using Google Chrome, the application had to be accessible on other machine environments like macOS and safari. Hosting the application on a CSP meant any issues in these other environments that may not have been present locally were identified.

3.3.2.4 Summary

The complete list of selected tools can be seen below. Microsoft Excel was used to create patient data files for testing and debugging the application. It was chosen due to its simplicity and that it provides a clear way to visualise data. RStudio is a specialist statistical IDE that was chosen for its data analysis tools and vast community guides on logistic regression.

Framework	IDEs	Cloud Service Provider	Web Browser	Database Management System	Documentation	Storage
Angular	Visual Studio 2017 Community Edition, RStudio	Microsoft Azure	Google Chrome	Microsoft SQL Server Management Studio 17	OneNote, Microsoft Excel	OneDrive

Table 1: Complete breakdown of the tools used in the project

3.4 Research Methods

One of the aims of the project is to "Design and develop a secure and intuitive website that will provide a user interface for accessing and displaying the algorithm". To determine if the project had met this goal, research was conducted to measure if the artefact is indeed secure and intuitive. This research was carried out through the SUS that has been described in previous sections. The reason for choosing a quantitative survey such as SUS is because these methods facilitate the discovery of quantifiable information (Carr, 1994) - they help to measure and describe variables. Specifically, quantitative methods easily describe how usable the application is through simple numerical representation. If an application has high usability, it is very intuitive. The SUS produces a figure ranging from 0-100 where scores over 90 are considered exceptional, 80-90 are good, 70-80 are acceptable and anything under 70 is poor (Bangor et al., 2009). Therefore, it can be said that if the application scored greater than 70, the previously described aim had been met.

However, the data gathered from these surveys was also used to develop future iterations of the application. To do this, the SUS survey was slightly modified to gather qualitative information as well. Under each question, there was a text box asking users to explain why they answered the question how they did. This question had no impact on the scores and therefore the integrity of the SUS was protected. Through gathering qualitative information such as this, it was clear to see the exact specifics of what needed further work rather than just relying on quantitative scores and guessing what needed changing.

The SUS surveys were developed using google forms which also handled the visualisation of the results. The online tool automatically created bar charts that

displayed the results of each question, making it easy to compare results from both surveys. Simple statistical analysis was then performed on the results to find the overall average scores of each survey.

Chapter 4

Design, Development and Evaluation

4.1 Requirements Elicitation, Collection & Analysis

This section will outline the requirements that were gathered for this project and the methods in which they were elicited. To adequately discuss these points, the background behind requirements gathering and why it is so important must first be examined.

Requirement's elicitation is part of the first stage of the SDLC, 'planning'. As stated by Zowghi and Coulin (2005) requirements elicitation is more specifically the process of uncovering and acquiring requirements for computer-based systems. Requirements gathered through these means help to lay the foundation on which the whole artefact is built by showing clear goals that highlight exactly what the artefact needs to do. In this regard, they are similar to the objectives of the project except they are focused purely on the artefact development and not on the project as a whole. Research has shown that many projects fail due to inadequate requirements (Goguen and Linde, 1992). Without careful consideration of this aspect of the project, it is likely that the artefact produced would be substandard and not meet its aim as certain key features may be discarded. The success of the artefact and the gathering of satisfactory requirements are intertwined so appropriate elicitation methods must be utilised.

One of the main forms of requirements elicitation chosen for this project was the examining of academic sources to analyse the full background of the issue. Some of these points have already been touched on in the previous 'Literature Review' section of this report, however, we will now go into more details on exactly what these sources show in terms of requirements. Sources such as Feinmann (2020) demonstrate how huge the demand was for COVID-19 doses paving way for the most important of all the requirements of this project - it must successfully calculate a risk score to prioritise patients. Although obvious, this must be noted as a requirement as it is the crux of the project. Meo et al. (2020) highlighted the seriousness of COVID-19 and from this, it is

strongly implied that the algorithm created must have high accuracy when predicting who is likely to get severe COVID-19. The artefact cannot afford to produce false negatives (i.e., patients being marked as low risk when they are high risk) as it will cause individuals to be delayed in receiving a vaccination and they may get severely ill from COVID-19 while they are waiting. To calculate these predictions the project must analyse a vast array of patient preconditions. This became clear when combining the previously mentioned article by Meo et al. with articles by Clark et al. (2020) and Liu et al. (2020). All three of these studies present evidence that deaths and severe cases of COVID-19 were linked to the pre-existing conditions of patients. More specifically, to ensure this artefact improves upon existing government strategies (as discussed by Campos-Matos et al. (2020)) it must analyse as many factors as possible and not bias on age. This will produce scores that are individual for every patient and not generalised to a group, allowing more detailed prioritisation.

Requirements could be further elicited from the actual objectives of the project. They mention how the website will 'provide a user interface for accessing and displaying the algorithm'. Users, therefore, had to be able to upload patient information for analysis and the results had to be displayed back to them in a coherent format. For this, it was decided that the results would be shown in a set table format with key information easily readable. The objectives also revealed two other requirements, the need for the application to contact high-risk patients and graphically visualise the results. Using the objectives to elicit requirements made sure that the project stayed focused on its aim by concentrating on what was needed and not on additional less important aspects.

One potential way to gather requirements could have been through the use of surveys and user interviews. Surveys allow for quick collection of data from a large, geographically diverse sample (Cope, 2014). Requirements can be obtained directly from clients, so an application is built exactly as they desire. However, due to there being no specific end-client there was no need for user input at the start of development. Despite this, the SUS was used mid-way through development to help consolidate requirements. All previous methods of elicitation were used to generate an initial list which was worked towards. Then, after several had been achieved, the SUS was used to gather any further requirements directly from users that may have been missed. Even though user input was not vital to the requirements elicitation of the project, gathering

user feedback at the mid-stage of development guaranteed all bases were covered. It cemented the need for the initial requirements and ensured the artefact was as convenient to users as it could be, meeting the objective of developing an intuitive website.

The complete list of requirements can be seen below. Requirements 6 and 9 were obtained from the risk assessment. Due to the sensitive information on the site, authentication measures had to be in place to prevent any unregistered user's from accessing patient data. To aid with privacy, only necessary information would be stored, and records had to be modifiable and delible so patients could request changes or the removal of their information. Requirement's 5, 8 and 12 then came from general knowledge of usability, all were features that would enhance user experience.

1. Due to the severity of COVID-19, the algorithm needs to have high predictive accuracy.
2. The Application must successfully calculate risk scores to prioritise individual patients by analysing their pre-existing conditions.
3. Users must be able to upload patient information for analysis.
4. Must display all patient information in set table format with patient name, risk score and other key information easily readable.
5. Allow users to search for specific patients and sort the data in whichever way they see fit.
6. Patient data must be mutable.
7. Visualise results with statistical graphs for easier understanding.
8. Must be cross-platform (work on a large array of devices).
9. Due to the confidential information, the site must be secure and only accessible to authenticated users.
10. Must contact high-risk patients if required.
11. The website has to demonstrate how the algorithm works.
12. Must have guides to help users with the sites functionality and assist them if they get stuck.

4.2 Development Research

To meet the requirement of calculating high-risk scores, certain preconditions and their relationships with COVID-19 had to be analysed to understand how they affected case severity. Analysis was conducted with a prediction model that mapped the most significant factors to predicting severe COVID-19. Before this model was created, datasets were researched to provide the model with accurate input data. Much like requirements elicitation, research such as this is also considered to be in the planning stage of the SDLC. There are many publicly available datasets published by organisations such as the WHO and the UK government. However, these datasets are generalised and describe the broader picture of COVID-19 (e.g., number of cases and number of deaths) rather than specific patient by patient data. To build an accurate prediction model the datasets required had to contain information regarding individual patients, their medical history, and if they had severe COVID-19. Due to the sensitivity of this kind of data, the number of available datasets was limited dramatically. Discussions were held with this projects supervisor to potentially contact the authors of a similar study (Shang et al., 2020) and ask them for the dataset they examined. Fortunately, after some time, a dataset supplied by the Mexican Government was found that proved to be a perfect fit for the algorithm's requirements. The dataset was download and accessed from a Kaggle project by Mukherjee (2020) and contained around 500,000 anonymised COVID-19 medical records from Mexico. Although in Spanish, the dataset came with a dictionary that described what each column represented and a key that explained what the values in the columns meant.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	id	sex	patient_ty	entry_dat	date_sym	date_died	intubed	pneumon	age	pregnanc	diabetes	copd	asthma	inmsupr	hypertens	other_dis	cardiovas	obesity	renal_chr	tobacco	contact_o	covid_res	icu
2	16169f	2	1	########	########	9999-99-9	97	2	27	97	2	2	2	2	2	2	2	2	2	2	2	1	97
3	1009bf	2	1	########	########	9999-99-9	97	2	24	97	2	2	2	2	2	2	2	2	2	2	99	1	97
4	167386	1	2	########	########	9999-99-9	2	2	54	2	2	2	2	2	2	2	2	1	2	2	99	1	2
5	0b5948	2	2	########	########	9999-99-9	2	1	30	97	2	2	2	2	2	2	2	2	2	2	99	1	2
6	0d01b5	1	2	########	########	########	2	2	60	2	1	2	2	2	1	2	1	2	2	2	99	1	2
7	1beec8	2	2	########	########	########	2	1	47	97	1	2	2	2	2	2	2	2	2	2	99	1	1
8	1.75E+56	2	2	########	########	9999-99-9	2	2	63	97	2	2	2	2	1	2	2	2	2	2	99	1	2
9	0ce1f5	1	1	########	########	9999-99-9	97	2	56	2	2	2	2	2	1	2	2	2	1	1	1	1	97
10	12b57a	1	1	########	########	9999-99-9	97	2	41	2	2	2	2	2	2	2	2	2	2	2	99	1	97
11	006b91	1	2	########	########	9999-99-9	1	1	39	2	2	2	2	2	2	2	2	1	2	2	99	1	2
12	0dfc54	1	2	########	########	9999-99-9	2	1	46	2	2	2	2	2	2	2	2	2	2	2	99	1	2
13	17e2a5	1	1	########	########	9999-99-9	97	2	45	2	2	2	2	2	2	2	2	2	2	2	99	1	97
14	126f5c	2	2	########	########	9999-99-9	2	2	28	97	2	2	2	2	2	2	2	2	2	2	99	1	2
15	11c741	1	1	########	########	9999-99-9	97	2	34	2	2	2	2	2	2	2	2	2	2	2	1	1	97

Figure 6: Screenshot of the Mexican COVID-19 Patient dataset

As seen in the above figure, the main patient file contains 23 columns that include details on the patient's age, sex and 12 separate preconditions such as hypertension and asthma (an adequate number of preconditions for a project of this size). With over half a million

patients an accurate model could be created with high validity as there were a vast number of records to help train and test models.

Due to it being found on Kaggle there were doubts about the dataset validity. Kaggle allows users to upload datasets without any real moderation. This meant that although described as a dataset from the Mexican government, it could have been created by a random user. Luckily, the dataset could be trusted as the author (Mukherjee, 2020) had supplied a link to its source, the Mexican governments General Directorate of Epidemiology (Gobierno De México, 2021).

With the validity of the set confirmed there was consideration about if data supplied by the Mexican government could be used to good effect internationally. It is possible that due to only being from one country the set may not provide a good prediction on the global picture. However, the dataset was chosen simply because at the time of the creation of the algorithm it was the only complete, publicly available one that could be found. Other datasets did exist, but they were usually missing key information or were too generalised to a larger population. Settling on the data allowed for development to begin in good time and provided at least some confidence in the efficacy of the algorithm.

4.3 Design

The next stage of the SDLC approached in this project was Design. Due to the structure of the project, this stage was split into a few subsections. First, the design of the algorithm will be discussed, the report will then move into examining how the website was designed.

4.3.1 Algorithm Design

The algorithm for this project was designed in two separate sections. The prediction model was created in R studio and the results were then transferred over to the .NET API where conditional statements were used to calculate patient risk scores.

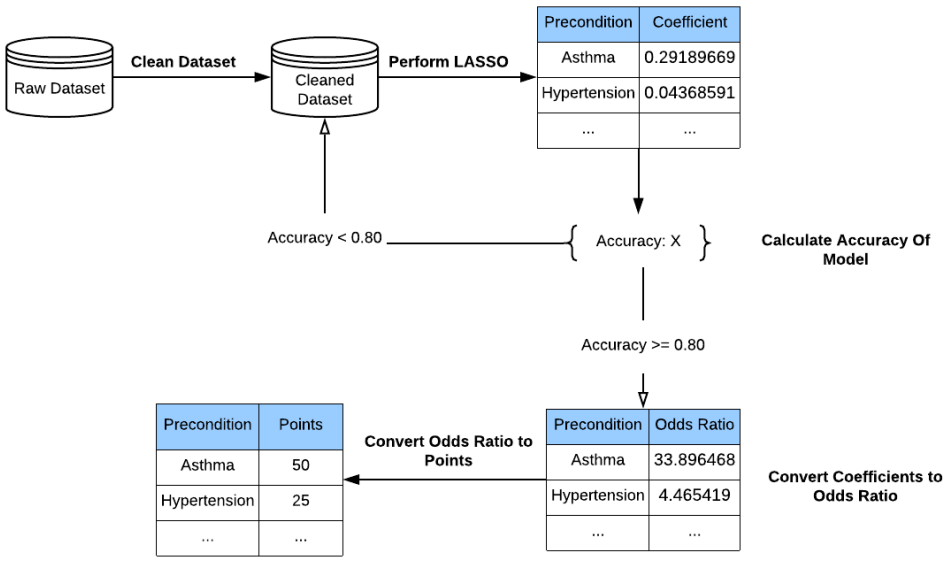


Figure 7: Algorithm Design Flowchart

The above diagram illustrates this design process. To start with the dataset was cleaned to remove any inconsistencies and ensure it was in the correct format. This is a task known as data munging, or data wrangling, and is defined by Endel and Piringer as an initial and fundamental process of examining and transforming data into a usable form (2015). By carefully managing what kind of data is used the accuracy of the prediction model can be improved as fewer unnecessary and erroneous records are analysed.

After cleaning the data comes the creation of the LASSO model. Once the model was created, it was tested and the accuracy of it was calculated. If the accuracy was less than 0.80 (80%) the previous two steps were repeated. It is possible that too much, or too little, data could have been removed during the cleaning process. If too much, the model may not have had enough information to make correct predictions or, if too few, there could have been erroneous records still present skewing the predictions. 0.80 (80%) was chosen as the desired threshold due to the project requirement of high predictive accuracy.

If the model had high accuracy the next stage commenced, converting the coefficients into odds ratios. The odds ratio (OR) measures the association between an exposure (pre-condition) and an outcome (high or low risk). More importantly, they can also be used to determine if a particular exposure is a risk factor for a particular outcome (Szumilas, 2015). As Szumilas explains, the “OR represents the odds that an outcome will occur given a

particular factor, compared to the odds of the outcome occurring in the absence of that exposure" (2015). For example, given a user has asthma what are the odds that they will have severe COVID-19? OR = 1 means the exposure does not affect odds of outcome, OR > 1 means exposure is associated with higher odds of outcome, OR < 1 means the exposure is associated with lower odds of outcome (Szumilas, 2015). These OR's are then used to calculate the points of each factor. The more points, the more significant that factor is in causing severe COVID-19. By converting to an Odds Ratio first, it was much easier to transition to a points-based system as the OR explains each factor in a more straightforward manner than decimal coefficients.

Once the points had been calculated the second, stage of the algorithm was developed. The design for this is conditional statements that compare users pre-existing conditions to the factors list produced from the model. For each pre-condition, the user has on the factors list, that pre-conditions points are added to their risk score. The pseudo-code for this can be seen below.

```
RiskScore = 0
For every condition in patients conditions {
      SWITCH(condition){
            CASE "CONDITION X":
                  RiskScore = RiskScore + PointValForConditionX
                  BREAK
            CASE "CONDITION Y":
                   RiskScore = RiskScore + PointValForConditionY
                  BREAK
            …
            DEFAULT:
                  RiskScore = RiskScore + 0
                  BREAK
      }
}
```

Figure 8: Algorithm Design pseudo code

4.3.2 Application Design

The design process of the application was similar in many ways to the five planes of user experience, a conceptual framework that helps to design an application. It starts with a high-level, abstract description and ends in the concrete visual design of the first prototype. Each stage of the framework is referred to as a plane, with each plane taking the design of the application one step further. The five planes themselves, are strategy, scope, structure, skeleton and surface (Garrett, 2010, 36). As further explained by Garrett, the strategy layer is a high-level description of the application which describes the direction of the project (2010, 36). The scope plane then provides a list of its features whilst the structure plane deals with the relationships between the application elements (Garrett, 2010, 36). The skeleton plane is a more concrete visualisation of the previously described elements and finally, the surface plane deals with the sensory aspects of the application – "the visual, auditory and tactile stimuli" (Garrett, 2010, 36). Following the process of creating each plane helps to iteratively design an artefact that is rooted in the requirements of the project.

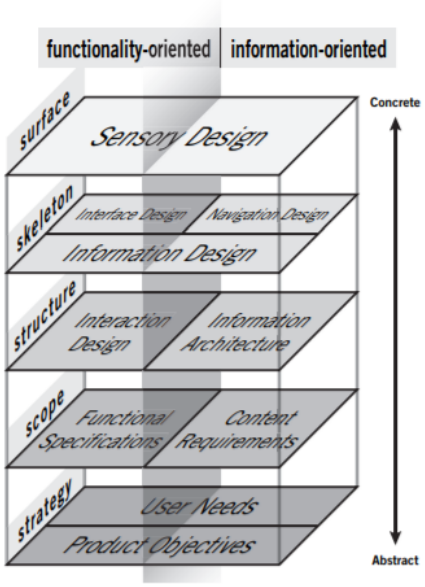


Figure 9: Visualisation of the Five Planes Framework (Garrett, 2010, 37)

4.3.2.1 Artefact Structure

The actual design of the website did not strictly follow each plane and instead took inspiration from the framework. The first piece of design created was a project brainstorm, inspired by both the strategy and scope planes. It contains a high-level description of what the app will do and a list of its features. The document helped to provide the baseline for the website and made it easier to understand the path that the design must follow.

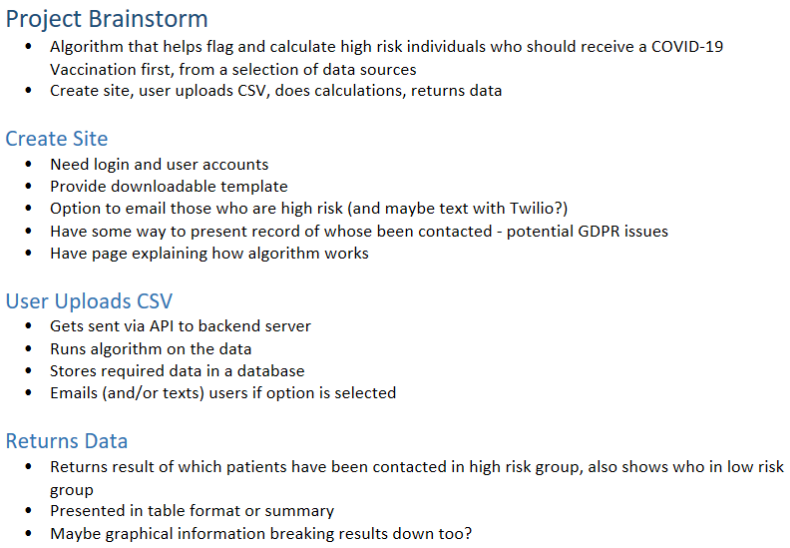


Figure 10: Project website brainstorm

From this document, a list of required pages was extracted. Each page handles features that were highlighted in the brainstorming document. These pages are described below:

- Home
    - Explains the purpose of the tool and motivation behind it
- High-Risk Analysis
    - Will need to describe how to use the tool and present the user with the results of their upload
- Results
    - Presents results and stats related to all previous uploads
- About
    - Explains how the algorithm functions
    - How did I analyse the data to determine the point values?
    - Will contain ‘How to’ Guides

This list is heavily inspired by the structure plane in the five planes framework as it describes each element inside the website, however, it does not explain their relationships. For this, the sitemap seen below was created. It should be noted that the login pages are not included in either of these two methods. Although a requirement, these pages were not vital to the first prototype of the application.

Sitemaps, such as the one below, conceptualise the framework of a website (Bernard, 1999). They show the paths of navigation across the application and, although very basic, this specific map helped massively with the development of the website's routing infrastructure. Each element's relationships are clearly highlighted.

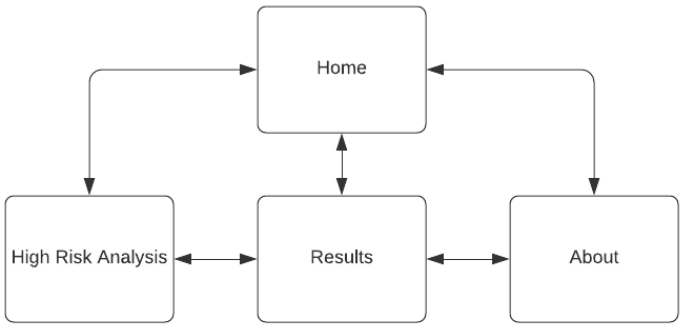


Figure 11: Sitemap for project website

4.3.2.2 Wireframes

The next of the five planes tackled was the skeleton plane which took the form of wireframes. These are blueprints that help to communicate the structure of a piece of software or website that is being built (Balsamiq, undated). The next sub-section will present these wireframes and explain why they were designed in that particular manner.

4.3.2.1.1 Home Page

'Home' acts as a landing page for this application. Its minimalist design stops users from being overwhelmed the second they access the website. They will be greeted with a large title, big navigation buttons and a COVID-19 dashboard creating an inviting experience. The dashboard will list the current COVID-19 cases and deaths in certain countries to provide users with context to the application in a visually appealing manner. The reason for choosing a minimalist layout such as this is for accessibility reasons. "CoV-Risk" is

intended to be used by all medical workers regardless of age or technical ability. Large “call to action” buttons and minimal text across the pages means even those hard of sight will be able to use the application. The navigation bar was designed with help from the sitemap that was previously described. The map illustrated which pages had to be connected and therefore which navigation elements should be accessible on which page.

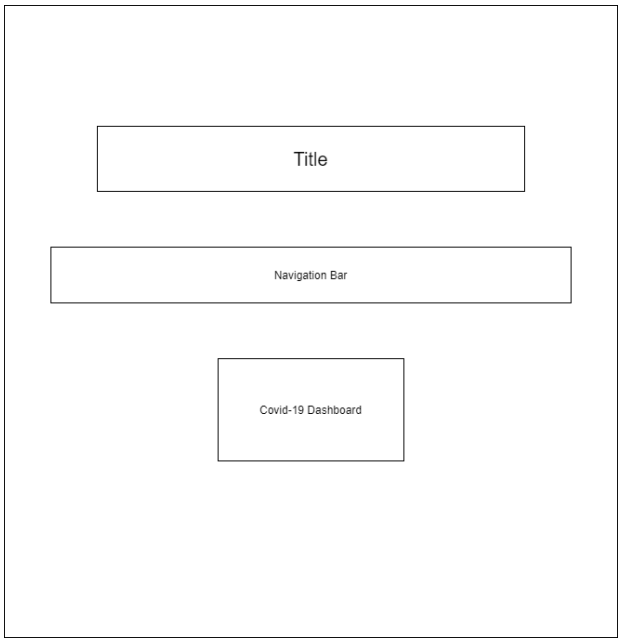


Figure 12: Home page wireframe

4.3.2.1.1 High-Risk Analysis

Three separate wireframes were created for the High-Risk Analysis (HRA) page with each one showing the page in a different state. This page is responsible for uploading a user’s file to the algorithm and calculating patient risk scores. The wireframe below shows the page after a user has uploaded data and is presented with tabular results. The other two stages of the page can be seen in figures 31 and 32 in the Appendix (Appendix 2).

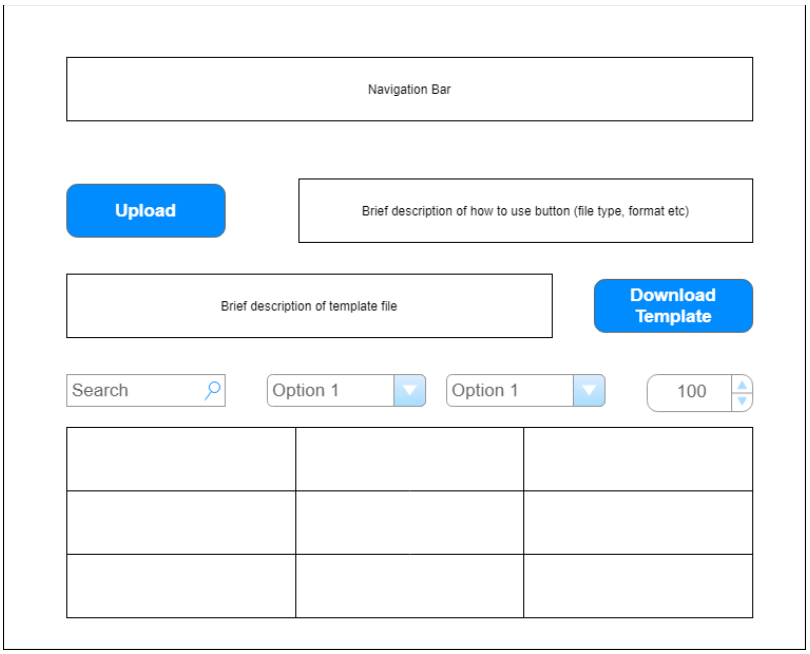


Figure 13: High-Risk Analysis page wireframe

As will be clear from all other pages from now on, the navigation buttons have moved to the top of the page. This is to provide room for the application's main content. There are two large buttons on this page, 'Upload' and 'Download Template'. 'Upload' allows the users to select a patient data file and upload it to the algorithm whereas 'Download Template' provides the users with a '.CSV' template file. This template has been created for ease of use as it helps users to format data in the desired manner.

The information from the current user upload is displayed in tabular format with key fields clearly visible. Filters are provided above the table to help users search and refine the results down to specific patients. This table will be paginated so only a certain number of patients are shown at any one time, preventing the page from overwhelming the user with data. A table was chosen as the best format to present information as it helps to break down large data sources into a much more readable and intuitive format.

4.3.2.1.1 Results

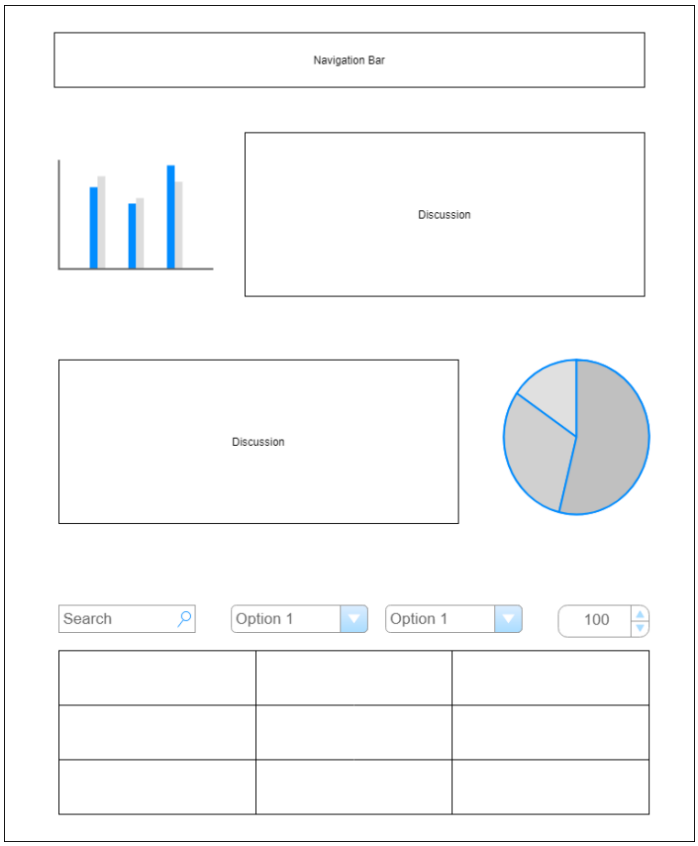


Figure 14: Results page wireframe

The 'Results' page shows the results of all uploads from all users. At the top of the page is the graphical visualisation of the results. This has been included as per the project requirements and helps to summate the results in a unique way for easier interpretation. Next to each graph, there will be a short description of the key information that the graph is displaying – Any trends, unexpected results, or patterns within the data. Finally, an identically formatted table to the HRA page will be generated below the graphs.

4.3.2.1.1 About

The about page acts as the general encyclopaedia for the application. It contains material on how the algorithm works and provides guides on how to use the website. Because of the plethora of information on this page, lots of images will be utilised to make reading more manageable. Each guide will be broken into various subsections that all follow a similar template, providing a sense of uniformity across the page.

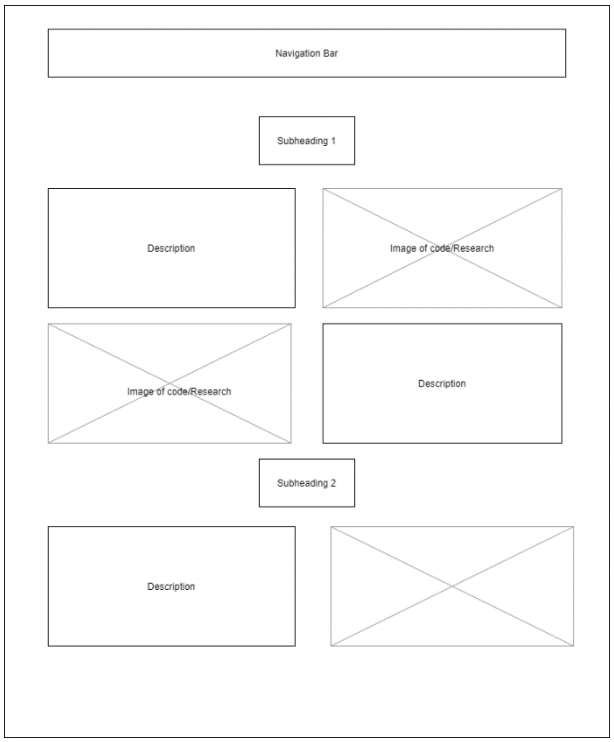


Figure 15: About page wireframe

Overall, wireframes solidified the prototype design and were extremely useful in developing the artefact. With set layouts and elements defined, development was able to run much smoother as less time was spent stopping to consider the design whilst programming.

4.3.2.3 Surface Design

After wireframing was complete the specifics of the visual design, for example, the colour palette, had to be considered. The five planes framework's final plane, surface, considers these visual stimuli by adding more details to the skeleton plane's wireframes. However, for this project, the surface plane was not completed in this manner. This was due to not needing a detailed visual prototype to begin development, the wireframes themselves proved enough. As well as this, a large chunk of time was saved not designing the plane – Time which was then better spent on the development.

Despite not creating full surface plane prototypes, a few, smaller, visual design elements were settled on before development began. A contrasting white and blue colour palette was chosen to make the application both visually appealing and accessible.

Figure 16: Example of the colour theme used across the application

There was some consideration as to whether icons and graphics were appropriate for a tool related to a serious issue such as COVID-19. Yet despite its seriousness, the application still needed to be appealing for end-users - few people would feel happy using a bland, lifeless application.

4.4 Development

After the designs for both the website and algorithm were finalised came the next, and arguably most important, stage of the SDLC, development. Again, due to the nature of this project, this section will be split into two parts, algorithm development and website development.

4.4.1 Algorithm Development

4.4.1.1 Data Pre-processing

As shown in the previous design section, the first stage in the algorithm development was data pre-processing. This ensured that the dataset was in an appropriate format for the prediction model. It was also the most time-consuming stage of the algorithm development due to it making sure that every aspect of the dataset was checked for inconsistencies. Many data cleaning techniques were used in this project, this section will highlight the most important of them.

The file was read into the R studio environment using the package ‘readxl’ and then any patients who had not tested positive for COVID-19 were removed from the dataset. This is shown on line 4 of figure 17, the dataset is made equal to all records that have a ‘covid_res’ value of 1 (true). The set was then checked for any duplicate ‘patientIDs’ using the ‘duplicated()’ function. Any duplicate patients were removed to prevent them from potentially negatively impacting the results. The column ‘date_died’ was then replaced with a new ‘died’ column which was initialised to 0 and set to 1 if the patient had a date of death (I.e., ‘date_died’ not equal to ‘9999-99-99’’). The reason for this column switch was to standardise the dataset (all other feature columns are 1 and 0 values) and make it easier to create the subsequent ‘severe_covid’ column. ‘severe_covid’ was added to the dataset and

provided an easy way to group all those patients who had severe COVID-19. For the sake of this dataset, severe COVID-19 was defined as any patient who either died, had been placed in intubation ('intubed') or into the intensive care unit (ICU). This column is the target variable used in the LASSO model to establish high-risk factors. Another additional column 'age_high' was also created to specifically measure what impact older ages have on COVID-19 outcomes. It was set to 1 if the patient was 50 or over and 0 if they were under 50.

The value '97' was shown in the provided key to mean non-applicable. This value was used in situations where the field did not relate to the patient, for example, pregnancy in males. There is no way this value can be anything other than false as it does not apply, therefore it was set to 0. Unfortunately, there is one big issue with doing this, most of the dataset is male and therefore most patients will have pregnancy = 0. This creates biases in the results as so few people will be pregnant and even fewer will be pregnant with severe COVID. It could have been left as 97 but this would have led to inconsistencies from the LASSO results as the model would compare true and false values to an arbitrary 97 value.

The entire cleaning process (as seen below) trimmed the dataset down from 566602 records to 217605 records, still, an adequate number to create an accurate model. See Appendix 3a for examples of what the dataset looked like before and after it was cleaned (Figures 33 and 34).

```r
library(readxl)
library(anchors)
MexData <- read_excel("Algorithm/Mex-Data-changed.xlsx")
MexData <- MexData[(MexData$covid_res == 1),]
MexData <- MexData[(!duplicated(MexData$id)),]
MexData$died <- c(0)
MexData[(MexData$date_died!="9999-99-99"), "died"] <- 1
MexData$severe_covid <- c(0)
MexData[(MexData$died==1 | MexData$intubed==1 | MexData$icu==1), "severe_covid"] <- 1
MexData$"age_high" <- c(0)
MexData[(MexData$age>=50), "age_high"] <- 1
CleanedMexData = subset(MexData, select = -c(patient_type,entry_date,
                                             date_symptoms, contact_other_covid,
                                             covid_res, date_died, other_disease))
CleanedMexData <- replace.value(CleanedMexData, c("pneumonia", "sex", "pregnancy", "diabetes", "copd", "asthma", "inmsupr",
                                "hypertension", "cardiovascular", "obesity", "renal_chronic", "tobacco",
                                "intubed", "icu"),97,0)

CleanedMexData <- CleanedMexData[!(CleanedMexData$pregnancy == 1 && CleanedMexData$age < 11),] # Removing people too young to be pregnant
CleanedMexData <- CleanedMexData[!(CleanedMexData$age > 100),] # Removing people too old
CleanedMexData <- CleanedMexData[(CleanedMexData$pneumonia != 99 & CleanedMexData$pneumonia != 98) &
                                 (CleanedMexData$pregnancy != 99 & CleanedMexData$pregnancy != 98) &
                                 (CleanedMexData$diabetes != 99 & CleanedMexData$diabetes != 98) &
                                 (CleanedMexData$copd != 99 & CleanedMexData$copd != 98) &
                                 (CleanedMexData$asthma != 99 & CleanedMexData$asthma != 98) &
                                 (CleanedMexData$inmsupr != 99 & CleanedMexData$inmsupr != 98) &
                                 (CleanedMexData$hypertension != 99 & CleanedMexData$hypertension != 98) &
                                 (CleanedMexData$cardiovascular != 99 & CleanedMexData$cardiovascular != 98) &
                                 (CleanedMexData$obesity != 99 & CleanedMexData$obesity != 98) &
                                 (CleanedMexData$renal_chronic != 99 & CleanedMexData$renal_chronic != 98) &
                                 (CleanedMexData$tobacco != 99 & CleanedMexData$tobacco != 98) &
                                 (CleanedMexData$intubed != 99 & CleanedMexData$intubed != 98) &
                                 (CleanedMexData$icu != 99 & CleanedMexData$icu != 98),]

CleanedMexData = subset(CleanedMexData, select = -c(icu, intubed))
```

Figure 17: R code for cleaning dataset

4.4.1.2 Model Creation

With the data cleaned the model itself, seen in figure 18, was created using LASSO regression. To start with, the data was split into a training and test set, 70% of the data was used to train the model and 30% of the data was held back to test its accuracy. This 70/30 split ensures a model is created that does not over or under fit. An overfitted model occurs when the training set is too large, causing the model to learn the input data too well leading to it being highly inaccurate on other datasets. An underfitted model is the inverse and occurs when the training set is too small. The model fails to learn the input data and as such does not make accurate predictions on any datasets.

```
library(glmnet)

index = sample.int(n=nrow(CleanedMexData), size= floor(.7*nrow(CleanedMexData)), replace = F)

train = CleanedMexData[index,] # Create the training data
test = CleanedMexData[-index,] # Create the test data

x <- model.matrix(severe_covid~pneumonia + sex + age + age_high + pregnancy
                  + diabetes + copd + asthma + inmsupr + hypertension + cardiovascular
                  + obesity + renal_chronic + tobacco, train)
y <- train$severe_covid


#LASSO regression
LassoReg <- cv.glmnet(x,y,alpha=1,family="binomial",type.measure = "mse" )
Lambda_min <- LassoReg$lambda.min
LassoModel <- glmnet(x, y, alpha = 1, family = "binomial",
                     lambda = Lambda_min)
LassoCoeff <- coef(LassoModel)

OddsRatio <- (exp(LassoCoeff) - 1) * 100
```

Figure 18: R code for creating the LASSO model

The training set was converted to a matrix with 'severe_covid' as the target variable. This matrix was fed into the 'cv.glmnet()' function along with a few parameters that marked the type of regression to be used. The cross-validation function calculates the best lambda value to use in the model. Lambda is a hyperparameter that controls the shrinkage of non-significant coefficients. Choosing an appropriate lambda is critical, too small and none of the coefficients will shrink, too large and all of them will be set to 0 (kassambara, 2018). The calculated lambda value is fed into the 'glmnet()' function which creates the final LASSO model. From there, the coefficient values (seen in figure 35 of appendix 3a) are extracted, and the odds ratio (OR) was calculated by getting the exponent of these values. The formula below placed the odds ratios into a percentage which made them easier to interrupt (User1607, 2018):

$$\text{Percent Change in the Odds} = (\text{Odds Ratio} - 1) \times 100$$

For example, the odds of having severe COVID with pneumonia (1) over the odds of getting severe COVID without pneumonia (0) is exp(2.37756627) = 10.77864. The percentage change in the odds is (10.77864 – 1) x 100 = 977.863867. We can say that the odds for pneumonia patients getting severe COVID are ~978% higher than the odds for non-pneumonia patients. The complete odds ratio can be seen in figure 36 of appendix 3a.

4.4.1.3 Points System

The final stage of the algorithm development was converting the OR into a simplistic points system. The points system and the reasoning behind each factor's value can be seen in table 5 of appendix 3a. Any factor not in the table had a coefficient (and therefore OR) value of 0, meaning it was not a significant factor in predicting severe COVID-19.

The threshold value for this algorithm is 50. This value was selected to ensure that the most serious conditions will automatically get flagged as high risk. Inversely, it makes sure that less serious conditions do not get flagged as high-risk on their own and have to be combined with other conditions to push them over the threshold. All of this means only those most in need get the vaccination straight away.

4.4.2 Website Development

As previously described in the methodology section, the development for the website was split into separate sprints that enabled a few core features to be developed at a time. The sprints were:

1. Configuration
2. Data Handling
3. Improvement
4. Optimisation

4.4.2.1 Configuration

The first sprint involved getting the Angular project setup with the correct file structure. It was also where the initial page architecture was developed. Thanks to the Angular CLI (command-line interface) setting up the Angular project was extremely simple, all that was required was running the command 'ng-new <ProjectName>' in the chosen directory for the project. Angular then automatically installs all the required npm packages and

dependencies (Angular, undated a). Once the initial file structure was created, a few custom folders were added to help manage the requirements of this specific project. The finalised app structure can be seen in figure 39 of appendix 3b.

The first page tackled was the home page (seen in the following image). As the first page users will see when they log into the application it had to be eye-catching and inviting. The design utilised the elements discussed in the wireframes, however, it was decided that to add more depth, CSS3 animations would be used sparingly across the application. The main use of animations was in the title of the application which drops down the screen and is then underscored. The code for this animation is shown in figure 40 of appendix 3b. These animations make the site feel more polished as well as adding extra touches to make it more engaging. The 'Coronavirus Dashboard' is from elfsight, a website that provides hundreds of widgets for a variety of different needs. This specific widget keeps track of covid deaths and cases worldwide and across three countries, the USA, UK and Mexico. As it is not a key feature of the project there was little point in attempting to create a custom version of the widget, time was better focussed on the apps core features.

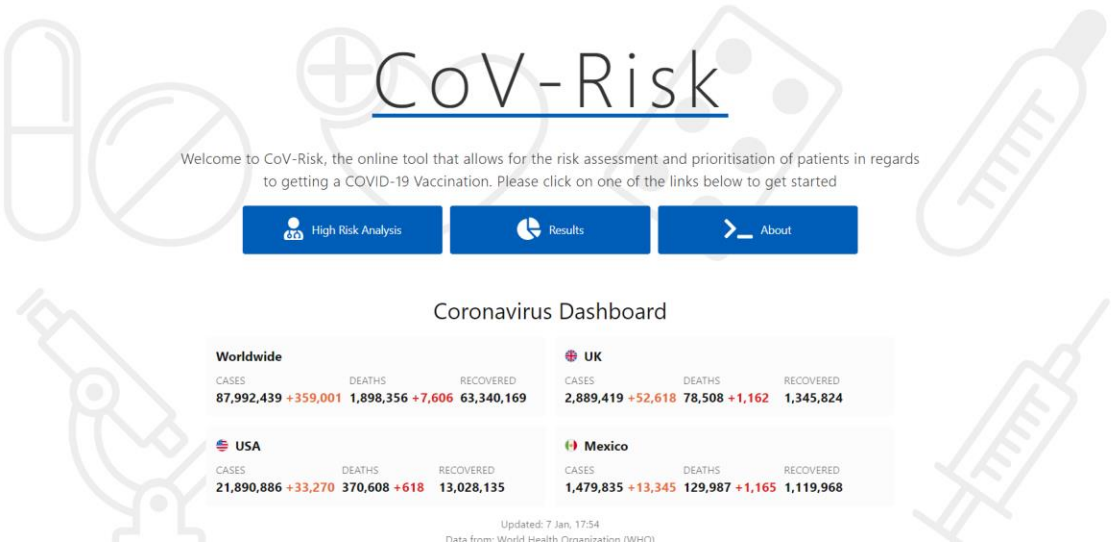


Figure 19: One of the first iterations of the home page

As well as setup and structure, this section was also where the navigation/routing of the application was created. Angular has its own built-in routing component called 'Angular Router' and therefore creating the navigation for this project was fairly straightforward. As seen in figure 41 (appendix 3b), an array of all possible routes and their associated

components (pages) was created. Whenever a navigation button is clicked a function is called, such as one of those seen in appendix 3b figure 42, which changes the application's route to one of the created paths. The associated component is then be displayed in the body of the page. This is known as a single page application (SPA); each component is displayed one at a time on the same page with the active component depending on the application's route. SPA's provide vastly superior load times over generic multi-page websites and allow for the easy sharing of data between components. This SPA effect is achieved by utilising the <routing-outlet> tag on the 'app.component' page (seen in appendix 3b, figure 43). This page contains the navigation bar with the selected component then rendered between the <routing-outlet> tag. Almost every page requires the navigation buttons therefore, instead of writing the code for it multiple times, it is written once with the page contents then displayed underneath it.

The initial structure of the HRA and Results page can be seen in figures 44, 45 and 46 in appendix 3b. The about page was not developed at this stage as it was not a top priority.

4.4.2.2 Data Handling

The main focus in this sprint was developing the ASP .NET Core side of the application, integrating the algorithm into the site and creating the database.

The ASP.NET Core application was created using the built-in visual studio template with the Angular project files then being manually copied into it. Much like with the Angular application, custom folders were added to the project to help manage the projects specific requirements. The finalised structure of the artefact can be seen in figure 58 of the appendices (appendix 3c).

The algorithm was then developed in C# using the previously described pseudo-code and points system. A list of the custom class 'Patient' (shown in appendix 3c figure 60) is supplied to the algorithm and each separate patient is iterated over. Their risk score is updated and they are marked as high risk if they are above the threshold. The code for the algorithm can be viewed in figure 59 of appendix 3c.

With the algorithm and custom classes created in C# the next stage was uploading a user file and extracting the patient data. A lot of time was spent on this stage as not only did the

user data have to be uploaded to the API, it also had to be correctly validated to make sure they had only entered credible information that the algorithm could use. To start with, functionality was added to the upload button on the HRA page which allowed the user to select a CSV file from their local computer. This file was then uploaded to the server, the API code for this is shown in the next image:

```
// POST: api/Algorithm
[HttpPost]
[Authorize(AuthenticationSchemes = JwtBearerDefaults.AuthenticationScheme)]
[Route("UploadPatients")]
public ReturnedPatients UploadPatients(IFormFile file)
{
    var feilds = _algorithm.ExtractPatients(file);
    var response = _algorithm.ValidateFile(feilds);
    if (response.ErrorMessage == "")
    {
        response.ErrorMessage = _data.UploadPatients(response.Patients);
    }
    return response;
}
```

Figure 20: API Post Request

The data from this file was extracted using C#'s 'StreamReader()' function and placed into the custom 'Patient' class before it was then validated. The data validation required for the application involved:

- precondition validation - checking the user had entered one of the preconditions the algorithm supports.
- Null checks – Ensuring user-entered information into all required columns.
- Age validation – making sure a number between 0-100 was entered in the age column.
- Sex validation – checking the user entered either 'M' or 'F' in the sex column.
- Duplicate ID check – checking the user did not enter the same patient ID more than once.
- Phone number checks – checking the value in the phone number column was a numeric value.

One of the main issues that occurred from creating these validation checks was the amount of development time they consumed due to the sheer number of checks that had to be performed. Almost two weeks were spent going over every possibility of user input and

how erroneous inputs could be caught before breaking the application. An example of some of the validation code can be seen in figure 61 of appendix 3c.

Once the code was correctly validated it then needed to be saved into the database. Before this, the database itself had to be created on the development machine using SSMS. The 'CoV-Risk' database was created with a table 'Patients'. The structure of this table can be seen below and the code to create it can be found in figure 64 (appendix 3c):

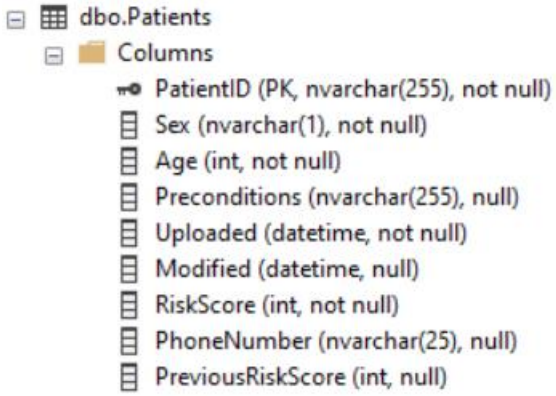


Figure 21: Structure of the 'Patients' table in the database

To save the user uploads into this database Microsoft Entity Framework Core (EF Core) was used. EF Core provides a quick and easy way to access a database through its model format. As Microsoft Docs (2020a) state, "A model is made up of entity classes and a context object that represents a session with the database. The context object allows querying and saving data". A code snippet showing EF Core in action can be viewed in the appendices (appendix 3c, figure 62). Although simple to integrate, EF Core did take a while to understand. This was due to its conversion of databases into classes and objects which made it hard to visualise what data was going where. It took a few days of research to get familiar with the framework.

Once the data was uploaded to the database it then needed to be displayed to the user. A method was added to the API to extract patient data from the database using EF core's querying abilities. As per the wireframes, a table was added to the HRA page, to show the results of the user upload. For simpler display purposes, this table only contained a few of the key patient fields.

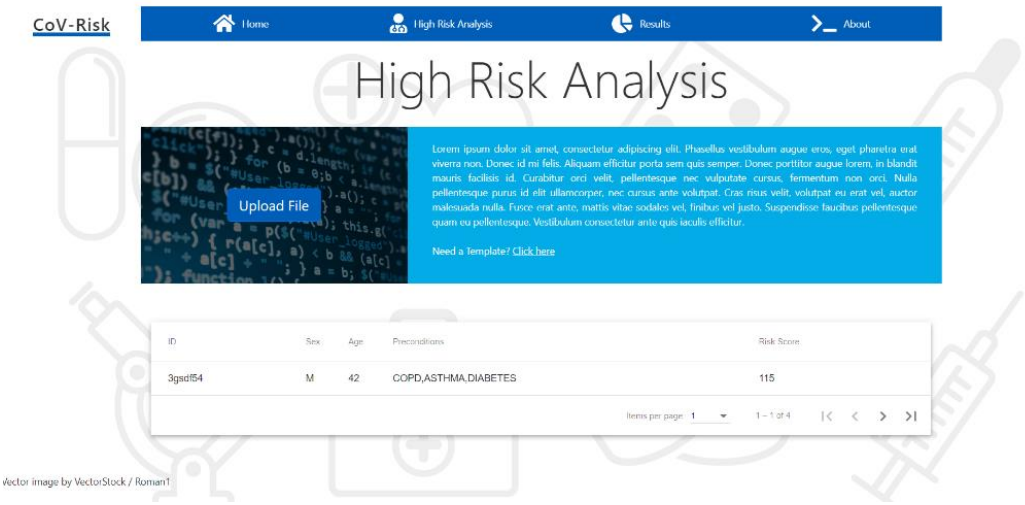


Figure 22: High-Risk Analysis page, early implementation

A larger table was then added to the results page which showed the results of all patient uploads. To provide more detail, this table displayed additional fields to the previous HRA table. For example, dates are shown which helps the user identify the patients they uploaded, meeting the user requirement of 'finding specific patients'. The code for creating this table is accessible in appendix 3b, figure 47 (Note that this code is for the finalised version of the table that will be shown later).

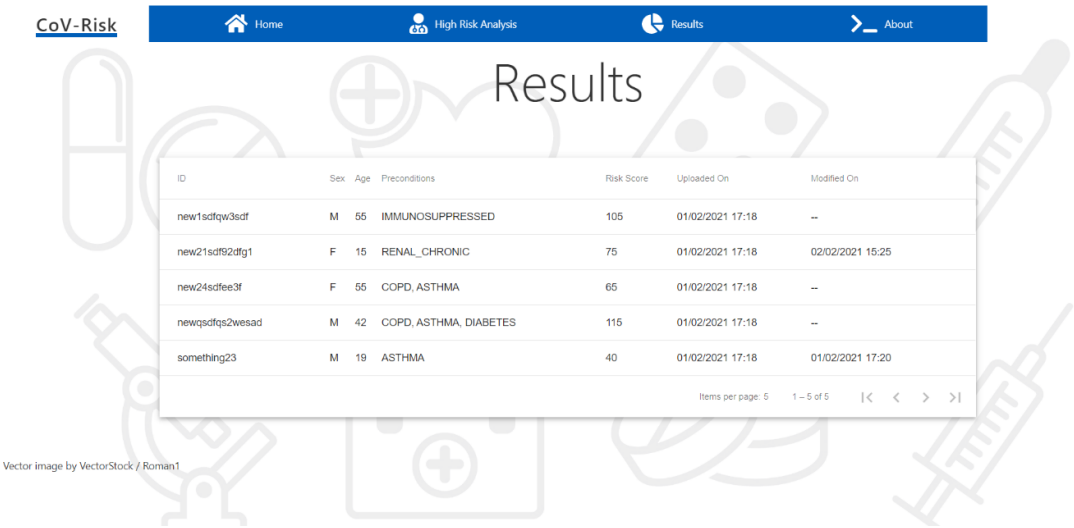


Figure 23: Results page initial development

To help users in uploading patient data, a template file was added to the project which contained all the necessary information to correctly format patient data as well as examples that users can follow. Adding these instructions to the template file reduces the word count

on the website itself, meaning users are less affected by large chunks of text all in one go. The file can be viewed in the appendices (Appendix 3c, figure 63).

With these key features developed it was time to move onto the mid-development survey. The survey, seen in appendix 4a (figure 75) was created in google forms and sent to users via email. For the survey participants to access the site it first had to be hosted. An Azure student subscription was set up and the required resources (App Service and SQL database) were created. The configuration for both resources is shown in the appendices (appendix 3c figures 65 and 66). Setting up the resources was extremely smooth, however, issues started to arise when publishing to the app service. When 'publish to Azure' was selected inside visual studio, the files would not be uploaded to the app service. To debug the problem, Azure's Kudu file explorer service was utilised. Upon using the tool, it became clear that the "back-end" ASP.NET Core code had successfully uploaded, what was missing were the Angular files. This issue was fixed by manually uploading the Angular 'dist' folder (the compiled production version of the Angular application) to the app service.

The survey itself was a success and returned 13 responses (these responses can be found in appendix 4a figure 76). Some of the respondents spent a lot of time writing in detail why they scored each question how they did, feedback which proved extremely valuable to application development. The first question on the survey asks users to enter a unique ID, this ID helped to keep each user anonymous whilst also allowing their results to be compared between the mid and post-development survey's. The main piece of feedback given by multiple users was to get the application working on mobile devices. In its current state, the formatting was incorrect on smaller screens. The average SUS score given by the users was calculated as 72.3, which shows even in its infancy the application had good usability. The specific results of both SUS surveys, and how the final scores are generated, will be discussed later in the report.

4.4.2.3 Improvement

This sprint revolved around the improvement of existing features as well as the addition of others. The main improvements made were through styling and quality of life changes. The results page saw the biggest overhaul with an additional text filter being added above the table to allow users to search for specific patient IDs, conditions or even upload dates. Stylistic changes were made to the tables on the results and HRA page (seen in figure 48 of

appendix 3b) to help them stand out more. Colouring high-risk patients made them more easily distinguishable among the data, increasing readability. Sorting was also added to the tables, when a header is clicked the data will be sorted based on the values in that column. As per the requirements and user feedback, statistical graphs were then implemented into the application. ‘Ng2-charts’ and ‘charts-js’ chart packages were utilised with both packages being simple to integrate. Four charts were chosen for the application: a pie chart of the most common pre-conditions, a line graph of age vs risk score, a radar chart of two sex groups and their related preconditions, and a bar chart of sex vs risk score for each sex group. These four were chosen as they all provide a clear and unique visual summation of the data. Some users may find reading information from tables extremely boring and therefore these charts make for a much more engaging way to interrupt the data. The charts were placed into a separate ‘Analysis’ section at the bottom of the page with a show/hide toggle that controlled how many of the charts could be seen at once. By default, only two charts were visible, much like with other features this was to prevent the user from being bombarded with information. They can control exactly what they wish to see. These ‘Results’ page changes can be seen in the following images.

Figure 24: Results page with charts and better formatting

The biggest feature added in this section was the integration of the login system. To do this, three new components were created, a welcome component (seen in the following image), a register component (appendix 3b figure 51) and a sign-in component (appendix 3b figure 50). The authentication itself was handled using JSON web tokens (JWT). When a user signs in they are given a small token that is stored in the browser's local memory. Every request the user sends to the API will include this JWT, allowing them to access routes, services, and resources (JWT.io, undated). An example of this can be seen in the 'AuthGaurd' component utilised in Angular Routing. This component (seen in figure 52 of appendix 3b) is attached to almost every route in the application, and it checks if the user making the routing request has a registered token. If they do, they are permitted onto that

page, if they do not, they are sent back to the '/user' page (welcome page). For this project, a registered token was any token given to a user that was no older than 24 hours. This meant users had to sign in again every day, preventing malicious users from exploiting a device a user had signed in on. Each of the sign-in/register input fields had its own client-side validation making sure only appropriate data was permitted. For example, the password field could not be left blank.

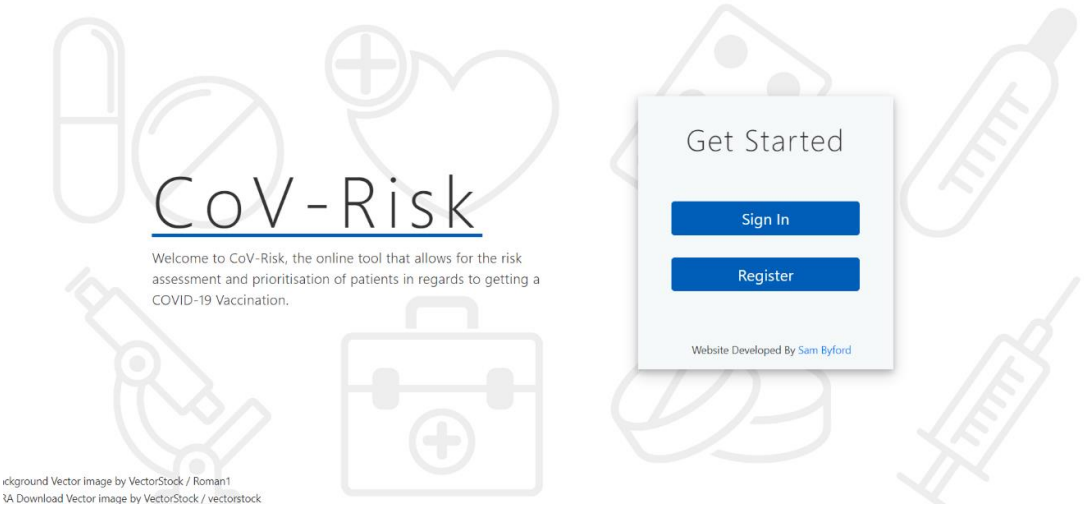


Figure 25: New welcome page of the website

To keep track of users and their registered token, ASP.NET Core Identity (ANCI) was used, which "Manages users, passwords, profile data, roles, claims, tokens, email confirmation, and more" (Microsoft Docs, 2020b). ANCI was used to create tables inside the database that housed the account information of registered users as well as their JWT. A guard, similar to the Angular Auth Guard, was then placed onto each API request (for example line 3 of figure 20) which only permitted the request if the user had a registered token. By placing security onto both the website pages and the API, the privacy and confidentiality of the data was maintained. This authorisation tackled the project goal of creating a 'secure' website. A user icon was added to the top of every page which, when clicked, showed the current user's details, and allowed them to sign out.

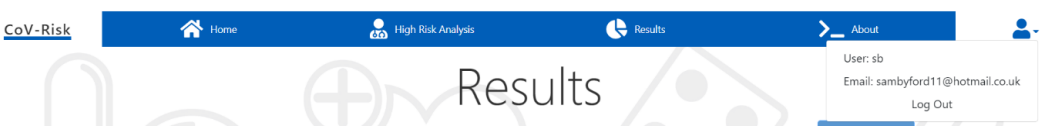


Figure 26: Screenshot highlighting the user portal on the website

The about page did not require any user registration as it contained information on the background and motivation behind the website. Therefore, it needed to be accessible by non-registered users so they could make an informed choice regarding if they wanted to use the application or not. The about page was the most simplistic page on the site, containing a large volume of information. To help users consume this information it was split into sections with contents at the top of the page that, when clicked, took the users to that chosen section. Images and colours were used generously on this page to help make it more interesting and inviting. A guide was written for all the application's main features and there is even a brief section on how the algorithm was developed. Part of the final version of this page can be viewed in the appendices (appendix 3b figure 53).

Another project requirement achieved in this section was the patient contact system. Originally an email system was going to be utilised using C#'s SMTP client (simple mail transfer protocol). However, when setting up and researching the Azure resources it became apparent that Azure no longer supports SMTP directly and instead SMTP relay services must be used. Unfortunately, none of these relay services can be used on Azure's student subscription. It was here that the decision was made to switch to Twilio, a pay-as-you-go SMS, voice, and video API. Twilio allowed the project to automatically text those patients who were high risk and had supplied their phone number. Integration with Twilio was straightforward thanks to the $50 student credit and visual studio helper library offered by the company. The code for sending texts using Twilio is shown in figure 67 of appendix 3c. A dialogue box, seen in the following image, was then added to the website that provided users with the option when uploading a file to select if they wanted high-risk patients to be contacted.

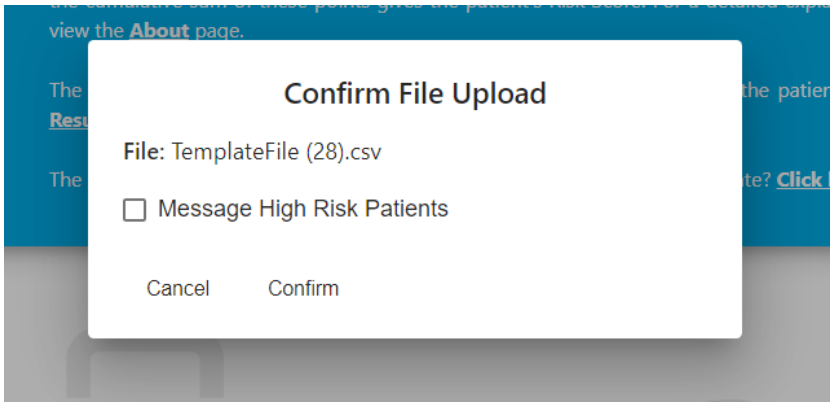


Figure 27: The confirmation dialogue box showing the patient messaging option

One of the user requirements for the application was for data to be ‘mutable’, meaning it could be edited or even removed. Because of this, edit and delete functionality was added to the application. Behind the scenes, this was achieved using EF Core functions to query the database. The table UI was altered to connect to these new API endpoints by allowing users to select patients to delete. When the ‘Delete’ button is clicked, all those patients who have been selected will appear in a confirmation box (figure 54 in appendix 3b) and the user will be able to confirm the deletion. A similar method was utilised for editing patient information. A new ‘More’ column was added to the table that contained an information icon. When pressed this icon would open a dialogue box (figure 55 appendix 3b) containing additional information about the selected patient that could no longer fit in the table. The three mutable columns (phone number, preconditions, and age) could then be edited in this box and the user could choose to message this patient if the new data made them high-risk. Clicking ‘Save’ would upload the modified patient information to the database. The updated table can be seen below.

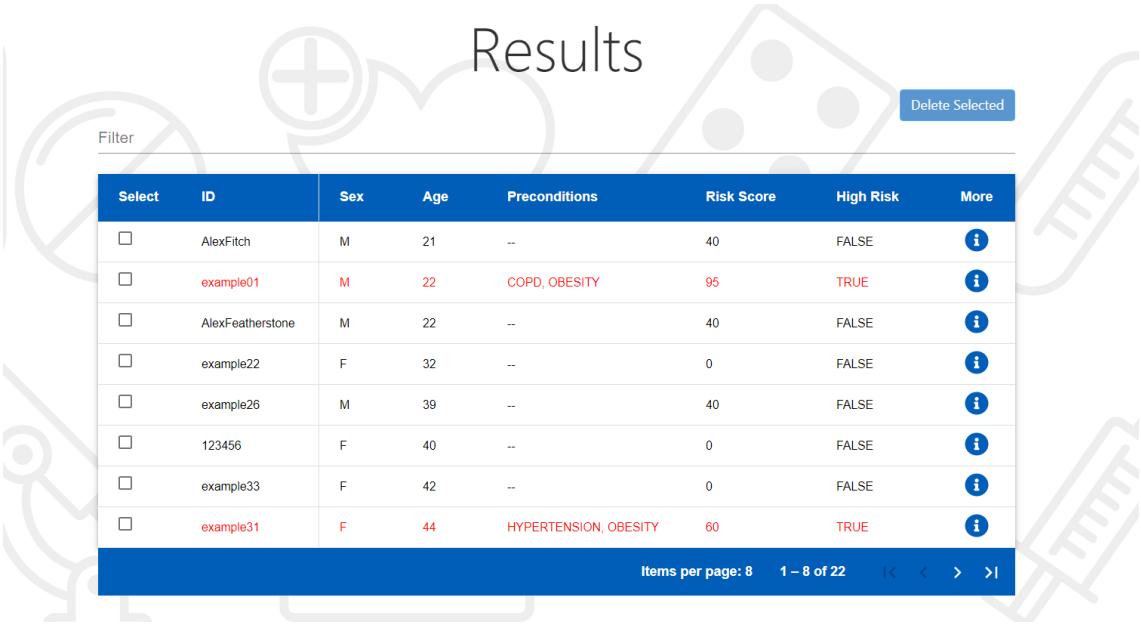


Figure 28: Updated table UI to allow for editing and deleting of patients

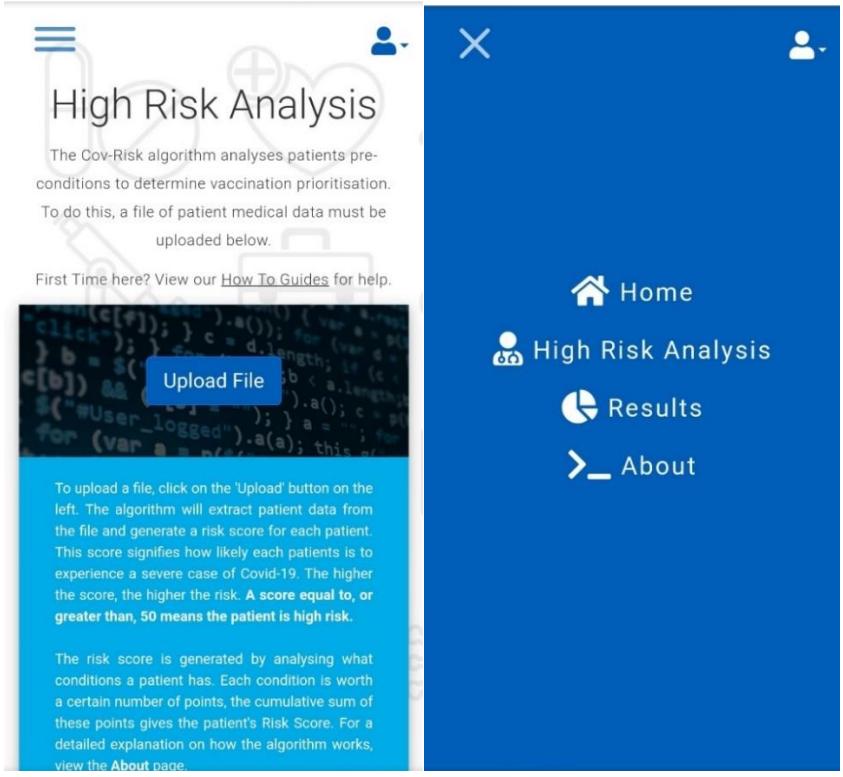
Adding this functionality did cause a slight problem with the generation of the graphs as the updated patient information was not reflected in the charts unless the page was refreshed. This was fixed by fetching the patient data whenever a user(s) had been edited or deleted.

There were many other ‘quality of life’ changes made in this section that provided feedback to the user. For example, the addition of loading spinners (figure 49 of appendix 3b) and success/error pop-ups (figures 56 and 57 in appendix 3b), showing the user that their actions were in progress or had been completed. Unfortunately, due to time constraints, not every feature can be examined.

4.4.2.4 Optimisation

‘Optimisation’ was the final sprint stage and it involved making sure the application was usable across multiple platforms. Bootstrap (a large library of responsive elements) had been utilised across the application to help with the layout, however, manual modifications were needed to ensure every element remained readable across all devices. As well as formatting elements, an entirely new menu had to be created. Navigation bars, such as the one present on the desktop version of the app, do not translate well to mobile devices. Instead, formats such as ‘Burger’ menus are utilised on mobile applications. Burger menus are easier to click and navigate with touch controls than a normal navigation bar due to their larger size. The burger menu designed for this application can be seen in the below screenshots.

Figure 29: Mobile version of the website showcasing the burger menu



Creating a mobile version of the site took a fair amount of time, with the hardest elements to translate to mobile being the tables. Their information had to fit on a small screen whilst also remaining readable. The solution to this was creating scrollable tables that users could horizontally scroll to access more information. To view the full changes made to the mobile version of the site see figures 68-74 in appendix 3d. When the site was completed on both desktop and mobile, the second post-development SUS survey was sent out. The results of this survey will be discussed in the following section.

4.5 Testing

Testing incorporated one of the final stages of the SDLC. For this project, the algorithm and website were tested separately. Both the functionality and the usability of the online application were considered.

4.5.1 Algorithm Testing

The calculated accuracy for the model is ~0.87, the code for calculating this value can be seen in appendix 3a figure 37. This accuracy is greater than the chosen threshold of 0.80, indicating that the algorithm is highly accurate. When given patient datasets, the model correctly identifies ~87% of high-risk individuals (patients who will develop severe COVID-19). To provide context to this value, research was conducted into the accuracy values other Kaggle users had produced when utilising the dataset. Meena (2020) created a similar risk-analysis model using the Mexican dataset and produced an accuracy value of ~0.66. ‘adsuman7’ (2020) used the dataset to predict those patients that are most likely to be placed in ICU given pre-existing conditions, extremely similar to this projects work. The accuracy value they produced for their logistic regression was ~0.84. These accuracy values are slightly less than this project’s algorithm but do show that the developed model is roughly in the expected range. The differences are most likely due to different cleaning techniques and the fact that these users did not use regularisation techniques such as LASSO.

A gradient boost model was produced alongside the LASSO model to provide an alternative if the LASSO model was not effective. The accuracy of this gradient boost model was calculated at ~0.88, slightly higher than the LASSO model. However, it was decided that the model would not be chosen. It was more accurate, but it came at the price of being more

complex to implement and understand. For only a tiny accuracy increase it was not seen as a viable option. Other models such as random forest classifiers and decision trees were also considered, but not developed due to the feedback of other users with the dataset who showed that these models were not as accurate as logistic and LASSO regression.

4.5.2 Website Testing

4.5.2.1 Application Tests

The best way to test the website was by examining each of the requirements for the artefact and testing if it had been successfully met through the applications core features. Testing was carried out throughout the development process; however, the table below showcases the testing on the *final* version of the application. The previous algorithm testing has already shown completion of requirement 1.

Table 2: Requirements testing of website

Requirement No	Test	Input	Expected Result	Test Passed
3	Testing that a patient file is successfully uploaded to the algorithm	.csv file with 1 patient – "test1,M,21, COPD;Obesity,"	A green pop-up will appear on the website confirming that the upload was a success	Yes
2 and 4	This test will ensure the uploaded patients are successfully extracted from the file, their scores are calculated and then displayed to the user in tabular form	See above	A table is generated on the HRA page displaying the results of the uploaded file. The table will show each patient's name, risk score and other key details. The risk score produced is 95 (COPD 25 + M 40 + OBESITY 30) and is highlighted in red due to being above the high-risk threshold.	Yes

Requirement No	Test	Input	Expected Result	Test Passed
5	Test that makes sure the filters & sorting features on the results and HRA tables work as intended	N/A	When ‘test1’ is entered in the search filter on the results page table, the table will be filtered down to only show the recently uploaded patient. Removing the text from the bar and clicking the ‘Age’ heading twice will sort the table by age descending.	Yes
6	Testing if patient data can be updated	Updating patient ‘test1’ to remove ‘COPD’	When the ‘i’ icon is clicked in the patients ‘More’ column a pop-up will appear showing additional patient information related to that patient. When ‘COPD’ is removed and ‘confirm’ is selected, the table will update to show patient ‘test1’ has a risk score of 70.	Yes
6	Testing if the patient data can be deleted	Selecting and deleting patient ‘test1’	When the checkbox of patient ‘test1’ is selected and delete is clicked, a pop-up will appear showing the selected patient and asking the user to confirm the deletion. When ‘yes’ is selected the table will update, showing that this patient is no longer there	Yes
7	Test that confirms that the graphs are correctly updated whenever new patient data is uploaded	.csv file with 1 patient – “test2,M,76, HYPERTENSION,”	When the user upload is completed, the graphs will refresh and display a new data point, ‘ID: test2’	Yes
8	Testing of the mobile version of the website which makes sure the formatting is correct	N/A	When the burger icon is clicked the mobile menu will open, taking up the entire screen width. The results table will be scrollable so the user can view every column and the graphs will be smaller & stacked vertically. The ‘upload’ and ‘download’ buttons on the HRA page will still allow for the selection of a file and the downloading of the template	Yes

Requirement No	Test	Input	Expected Result	Test Passed
9	Security testing on the application's pages	'CoVRisk/Results', 'CovRisk/HRA' and 'CoVRisk/About'	When logged out and the first two URLs are entered, permission will be denied, and the user will be redirected to the '/user' page. When the final URL is entered, the un-registered user will be correctly sent to the about page	Yes
9	Security testing on the applications API	'CoVRisk/Api/GetPatients' and 'CoVRisk/Api/UploadPatients'	When logged out and these two URLs are entered, permission will be denied, and the user will be shown a '401' unauthorised error	Yes
9	Validation testing that ensures only valid users can sign in to the application	Username: 'Blah' Password: '1234'	When attempting to sign in with these details the user will be shown an error informing them they have entered an incorrect username or password	Yes
9	Validation testing to make sure users enter all required information before signing up	Username: 'test' Email: 123@test.com Password: '123' ConfirmPassword: ' '	When the user enters this information, they will be informed that the password must be 4 characters long and the passwords must match in the password and confirm password boxes	Yes
10	Testing to see if high-risk patients are contacted when required	.csv file with 1 patient – "test3,M,78, HYPERTENSION,<UserPhoneNumber>"	When the file is uploaded and the 'message high-risk' patient's option is selected the user will receive a text stating that they are high-risk	Yes
11 and 12	This test will make sure the about page and guides are easily accessible from all required pages	N/A	When viewing the HRA page all links to the guides and about page will send the user to that page. On the about page the hyperlinks on the page contents will take the user to that specific section	Yes

Requirement No	Test	Input	Expected Result	Test Passed
12	Testing that clear error messages are displayed to the user when they upload an invalid file	.csv file with 1 patient – "test4,DOG, 78,HYPER TENSION, <UserPhone Number>"	After uploading the file, the user will be shown an error message telling them that there is an invalid entry in the 'sex' column of patient 'test4'	Yes

4.5.2.2 Usability Tests

Unfortunately, due to time constraints, the second (post-development) SUS survey only returned 11 participant results, down from the 13 of the original survey. However, 11 was still an adequate number of participants to gauge the usability of the site. The mean scores of the surveys were calculated and the differences between the two were analysed. SUS scores are calculated for each patient by subtracting 1 from the score of every odd-numbered question, subtracting the participant's value from 5 for every even-numbered question, adding these values and then multiplying by 2.5 (Thomas, undated). This method produces a score out of 100 where higher scores are better. An Example of this can be seen in figure 38 of appendix 3a.

As previously mentioned, the mid-development SUS survey produced a mean score of 72.3, showing good usability. The post-development survey produced an improved mean score of 76.4 which solidified the strong usability of the application. However, the mean alone does not tell the full story. The median result on the first survey was 75 with a minimum score of 55 and a maximum score of 90. This is compared to the second survey where the median result was 82.5 with a minimum score of 37.5 and a maximum score of 100. This highlights that, although on average the second iteration had higher usability, some users found it vastly inferior to the earlier design with widely ranging scores. User 'COVID2020' increased their score from 57.5 to 75, a dramatic increase of just under 20 points. The reason for this increase was due to the user having difficulties uploading patient files on the first iteration. They did not realise the file had to be of type '.csv' and struggled immensely with using the tool's main feature. This is juxtaposed to the second iteration where plenty of

instructions were provided to the user, participant ‘COVID2020’ stated that after they had read the instructions it was extremely simple to use.

However, the addition of detailed instructions was not taken well by all participants. User ‘looseymoosey96’ downgraded their score from 60 to 37.5, citing the overwhelming information as the main reason for their low score. On the other hand, participants such as ‘Bertie’s Girl’ struggled with a lack of guidance, highlighting the impossibility of pleasing every user. It took them a while until they found the required information on how to format the template. They also provided feedback mentioning that the results table was too ‘static and did not provide filtering options’ – even though there is a text filter above the table and each column can be sorted into ascending or descending order. These features were most likely missed by the participant due to them not being signposted well enough on the site.

Overall, lower scores were expected on the second iteration due to its increased complexity. As more features were added it was likely that certain usability aspects would be impacted. Despite some damning scores from users, the mean score did increase on the whole, highlighting that participants did prefer the more complex and finished version of the application. Participant ‘HowdyPootner’ even gave the application a perfect score of 100, praising the ‘detailed guides’, inviting aesthetic, and speed of the algorithm. The mean score of 76.4 demonstrates that the artefact has met the goal of developing an intuitive website to access the high-risk algorithm and that the aim of presenting data in a clear logical manner has also been addressed. The full detailed results from the second SUS survey can be found in figure 77 of appendix 4b.

Chapter 5

Conclusions

From this report, it is clear that COVID-19 poses a major threat to modern society, with what seems to be an ever-rising death count the disease is wreaking havoc across the globe. However recent developments in vaccinations do mean there is an end in sight. The main motivation behind this project was to help solve the logistical challenge of vaccinating the world's population by creating a prioritisation strategy that was more equitable than existing age-based approaches. From this challenge an aim for the project was created - the algorithm needed to automatically highlight which individuals should be prioritised for a COVID-19 vaccination and the data had to be presented clearly and understandably. To tackle this aim, eight goals (seen in the introduction to the report) were extracted which acted as individual milestones on the way to project completion. Therefore, to determine if the project was a success and the prioritisation problem was addressed, we first have to decide if the goals of the project, and subsequentially its aim, were met.

The first three goals of the project were related to the research, design, and development of the 'CoV-Risk' algorithm. The list of factors that lead to serious illness with COVID-19 was gathered from an open-source dataset provided by the Mexican government. Using logistic regression and regularisation methods, the algorithm was successfully designed in the statistical programming language R and was used to generate point values (as seen in Table 5 appendix 3a) for a variety of pre-existing conditions. As is highlighted by the development section, the algorithm was then converted into a variety of conditional C# methods. These methods, when supplied patient data, analysed each patient's preconditions, and gave them a risk score using the generated condition points. This development highlights how all three of these goals were tackled and the testing of the algorithm's accuracy shows how successfully the goals were met. The designed algorithm produced an accuracy score of ~87%, well above the desired 80% threshold. Users can trust the application as the accuracy provides confidence that the scores produced for each precondition are extremely accurate.

The final five goals were all related to the second half of the project, website development. They were completed through Angular and C# development of the application with user feedback helping to refine certain aspects. Each goal was deconstructed into separate requirement targets that focused on a key feature of the site. As shown by the testing of the application, these requirements were all adequately addressed and, in turn, so were the project goals. The site allows users to upload data, each patient is given a high-risk score and these scores are then presented to the user.

As required by the fourth goal, the final application is fully secure, only permitting registered users to access confidential patient data. This information is adequately stored inside an SQL database that is only accessible with the administrator password. The completion of goal six is shown in the fact that the 'CoV-Risk' application is built on the .NET framework, allowing the site to utilise a custom API to query and extract data from the developed database. Goal 7 was the hardest to meet due to the incompatibility with the chosen email system and the hosting provider used to publish the application (Microsoft Azure). The project had to be changed to instead send an SMS to users who had been marked as a vaccine priority. Some might argue that this goal had not been met successfully, however, this would be an unfair judgment as the application still approaches high-risk users it just now contacts them in a different format.

It is through the completion of the final goal that we can truly see how the project has helped to resolve the biased prioritisation issue that it is built around. The graphs on the site demonstrate that age is not the only deciding factor in determining who needs prioritising. This is notably seen in the following image. Patients across a wide range of ages are marked as high risk, it now depends on what other preconditions the patients have. In the graph, we can see that a 15-year-old patient is actually at higher risk than a 92-year-old patient. This is a huge improvement on existing strategies and demonstrates the exact thing that this project was attempting to achieve; To protect the most lives possible, vaccinations should be given to those most at risk, regardless of age. Age does affect COVID-19 severity, but this developed strategy makes it just a factor rather than the crux of prioritisation.

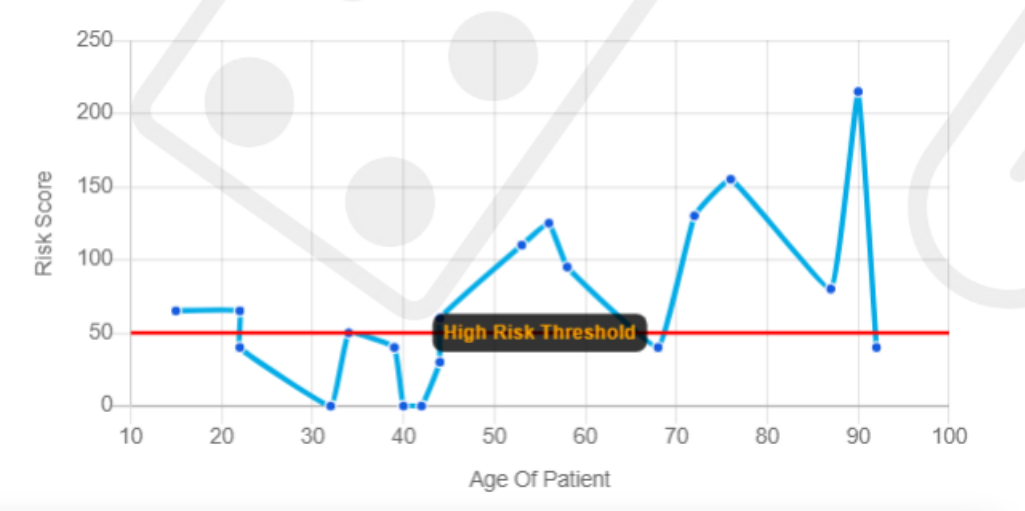


Figure 30: Graph of 'Risk Score' vs 'Age of Patient' from the artefact displaying the results of the algorithm

The artefact was evaluated using the System Usability Scale (SUS) which measures the usability of an application. It produces a score between 0 and 100 where scores greater than 70 indicate good usability. The reason for evaluating usability is to determine if the site is 'intuitive' - the better the usability, the more intuitive it is. The mean usability of the final application was calculated as ~76 from 11 participants. This highlights that it does have very good usability as it is over the 70-point threshold, however, there is still room for improvement. From the evaluation, certain changes are required to the final artefact to improve its usability even further. For example, many users felt overwhelmed by the amount of text on the site. Due to this, it would be beneficial to cut down on the text-based 'How-To Guides' and create video guides instead. These guides would present a more inviting way to learn how to use the application. This could also be combined with more 'Show/Hide' toggles that allow users to hide certain aspects of text that are not important to them. Given more time, changes such as these would have been implemented into the final artefact. Video guides are extremely useful to those users with short attention spans that do not wish to read large blocks of text. 'Show/Hide' toggles have been utilised across the site already but, given the opportunity, additional toggles would be implemented on the about page to help condense the information the user is presented with at any one time.

With the completion of the project's goals and the high usability score given by the post-development SUS, it can be said that the aim of the project was successfully achieved. The algorithm efficiently flags and notifies which individuals should be prioritised for a

COVID-19 vaccination and the data is presented clearly and understandably. The main challenge presented has been addressed in this project and a fairer prioritisation strategy has been developed - one that can be used to help solve the logistical challenge of vaccinating the world's population from COVID-19.

Chapter 6

Reflective Analysis

Overall, the project can be deemed a success with the final artefact delivering what was required. I was extremely pleased with how it turned out and surprised that bigger changes were not enforced. To provide a comprehensive reflection of the project this next section will focus on the specifics of what went well and what went wrong during the project lifecycle.

The front-end development of the Angular project came along smoothly and was developed much easier than I foresaw. I had never built an entire Angular application from the ground up before but, thankfully, there were plenty of online posts that guided me in the right direction. The Angular CLI was also a massive help in building the application as it allowed for the basic structure of components to be created with one command. One of my favourite parts of the development process was finding Angular modules that could be used to provide additional interactivity and functionality to the site. An example of this would be the various graphs supplied by the 'ng2 charts' third-party module. Working with Angular expanded my understanding of software development, I was able to utilise the complex features it provided such as routing, authorisation, and dependency injection to create a sophisticated application. Another part of front-end development that went much smoother than initially thought was the login system. It was one of the last features developed on the site as I believed there was a chance that I would not be able to accomplish it. However, I found some brilliant resources that explained exactly what needed to be done. I now have a rounded knowledge of JWT's, encryption, and user authentication across web-based applications. The project has given me the skills and programming confidence required to say that I could build even more complex Angular applications in the future when given less restrictive time constraints.

Having an accuracy of ~87% the algorithm development also exceeded my expectations. With valuable focus spent on cleaning and pre-processing the dataset, the accuracy was brilliant. I could not have found a better set; it had exactly the right information and it was

in the perfect format required. As previously stated, the algorithm accuracy almost appeared too good to be true. There were some slightly odd records among the set, but with 500,000 patients that was bound to be the case. Through developing the application, I was able to implement the skills of logistic regression and classification I had learnt in university modules. Also, for the first time, I learnt about regularisation techniques (such as LASSO) and how they help improve prediction model validity.

However, for all the effortlessness of certain aspects of the project, others did not go to plan. The initial integration of the ASP.NET Core and Angular applications was very challenging. Developing the API and server-side logic, was straightforward. It was the configuration of the ASP application where I got stuck. A large amount of time was spent researching how to set up the ASP code to launch the Angular application and also what the correct API URL was for the application to utilise. The reason behind these problems was simply lack of experience. Although I had a good understanding of C# before this project, I had never built an ASP.NET Core application. I, rather naively, thought it would not be that dissimilar to other C# applications I had previously developed. Luckily, due to it being such a popular framework, there was plenty of resources online to help resolve the issues I was having. Online forums and blog posts provided a huge hand in getting the setup correctly configured. It was just a case of constructing certain permissions and pointing the application to look in the correct directories.

Cloud hosting was something I had not envisioned being such a big issue at the start of development. One of the major problems with it was that I accidentally spent $50 out of my $100 free student budget within the first few days of setting up the resources. The credits that should have lasted a few months had nearly run out within a week. My lack of Azure knowledge meant I inadvertently configured the most expensive type of app service. Thankfully Azure sent out an email informing me of this resource spend and I was able to figure out how to downgrade the App service. This did mean that for the rest of its lifecycle I had to use very cheap, slow, resources and upon testing the application many users commented on its poor loading times. Although they hindered progress, these problems taught me how to better use cloud services. Having to debug resources, and create new ones from scratch, has led to me now having the expertise to confidently set up applications using cloud providers.

In hindsight there is one major thing that I would have done differently, I would have made designs for the mobile version of the application. The desktop frontend coding went great as I could follow the designs that had been set out through the wireframes and the 'Five planes' method. When it came to building the mobile version, a large portion of the development was spent deciding the best ways to 'convert' each element to be mobile-friendly. Careful consideration had to be paid to what elements should go where and which elements should be replaced entirely. It would have been a much more streamlined process if I had mobile wireframes to base these considerations on. Problems such as this demonstrate the importance of the design stage in building applications. Although the design process leads to development starting later, it provides a blueprint for the entire development, in turn, increasing the speed and efficiency to which an artefact is created.

Given more time there are a few additions I would make to the application. Many participants of the SUS surveys complained that the process of uploading patients in the correct file format was challenging the first time they attempted it. Because of this, I could add functionality to add a small number of patients via a form on the site. This would be easier than formatting an entire '.csv' file and would be more straightforward for new users. This method will not be adequate for larger patient uploads as it would take too long to fill out the form, but for ten or fewer patients it could be a real time-saver. As well as improving the usability for first-time users, it would provide a much quicker way for health care professionals to upload single patient data when they are on the go. Just fill in a form and send it, there is no need to download templates and edit them with specialist software. As well as this, given extra time I could have attempted to research additional datasets for use in the algorithm's creation. Datasets from multiple sources would have increased the validity of results, provide more accurate scores, and potentially would have allowed for more conditions to be analysed. That is one of the major drawbacks of the artefact in its current form, it only checks for a handful of preconditions. Some huge diseases, such as cancer, are not considered at all.

The methodology chosen for this project was a variant of Scrum and, although not fully stuck too, it did help massively with the project's development. The theory behind the sprint structure was adhered to, however, the timeframes of certain sprints were moved around quite freely due to being perplexed on specific aspects of development. Other parts of the project were also completed extremely quickly meaning that occasionally sprints that

should have taken 2 weeks, took two days. Towards the end of development, sprint reflections were almost entirely forgotten due to heavily focusing on the coding itself. This meant some of the features of the application were not tested until the end of development, leading to a handful of broken features from earlier sprints that had to be fixed. Bar these issues the agile approach, and its main principles were extremely useful. Key features were tackled first, allowing the mid-development SUS to be handed out within an almost complete application. The divide & conquer nature of the methodology meant that I was rarely overwhelmed with work and that progress was met at a continuously steady rate.

To conclude, this project can be deemed a huge success as it accomplished all goals that were set out for it and it surpassed expectations in many ways. There were issues along the way, but all problems were methodically resolved promptly. Unfortunately, the application could now be considered obsolete due to many months of vaccinations already taking place since the project's inception. However, I learnt a multitude of skills during each stage of development, skills that will no doubt prove extremely useful in my future career.

References

Adsuman7 (2020) *covid_logistic*. Kaggle Available from https://www.kaggle.com/adsuman7/covid-logistic adsuma7 [accessed 1 February 2021].

Angular (undated a) *Create a new project*. Angular. Available from https://Angular.io/tutorial/toh-pt0 [accessed 30 April 2021].

Angular (undated b) *What is Angular?*. Angular. Available from https://Angular.io/guide/what-is-Angular [accessed 5 April 2021].

Balsamiq (undated) *What Are Wireframes?* Balsamiq Studios. Available from https://balsamiq.com/learn/articles/what-are-wireframes/ [accessed 2 May 2021].

Bangor, A., Kortum, P., Miller, J. (2009) Determining what individual SUS Scores Mean: Adding an Adjective Rating Scale. *Journal Of Usability Studies*, 4(3) 114-123. Available from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.177.1240&rep=rep1&type=pdf [accessed 1st April 2021].

Barr, J. (2015) *AWS Educate – Credits, Training, Content, and Collaboration for Students & Educators*. AWS. Available from https://aws.amazon.com/blogs/aws/aws-educate-credits-training-content-and-collaboration-for-students-educators/ [accessed 26 March 2021].

Bernard, M. (1999) Sitemap Design: Alphabetical or Categorical? *Usability News*, 1(2). Available from https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.510.4237&rep=rep1&type=pdf [accessed 2 May 2021].

Campos-Matos, I., Mandal, S., Yates, J., Ramsay, M., Wilson, J. (2021) Maximising benefit, reducing inequalities and ensuring deliverability: Prioritisation of COVID-19 vaccination in the UK. *The Lancet Regional Health*, 2(March). Available from https://www.thelancet.com/journals/lanepe/article/PIIS2666-7762(20)30021-1/fulltext [accessed 25 February 2021].

Carr, L.T. (1994) The Strengths and weaknesses of quantitative and qualitative research: what method for nursing? Journal of Advance Nursing, 20(4) 716-721. Available from https://pubmed.ncbi.nlm.nih.gov/7822608/ [accessed 1st April 2021].

Clark, A., Jit, M., Warren-Gash, C., Guthrie, B., Wang, H.H.X., Mercer, S.W., Sanderson, C., McKee, M., Troeger, C., Ong, K.L., Checchi, F., Perel, P., Joseph, S., Gibbs, H.P., Banerjee, A., Eggo, R.M. (2020) Global, regional, and national estimates of the population at increased risk of severe COVID-19 due to underlying health conditions in 2020: a modelling study. *Lancet Global Health*, 8(8) E1003-E1017. Available from https://www.thelancet.com/journals/langlo/article/PIIS2214-109X(20)30264-3/fulltext [accessed 24 February 2020].

Cope, D.G. (2014) Using Electronic Surveys in Nursing Research. *Oncology Nursing Forum*, 41(6) 681-682. Available from https://web-a-ebscohost-com.proxy.library.lincoln.ac.uk/ehost/pdfviewer/pdfviewer?vid=1&sid=10fecd68-75bc-4a82-98b8-dc7da844bb04%40sessionmgr4007 [accessed 20 April 2021].

Endel, f. and Piringer, H. (2015) Dat Wrangling: Making data useful again. *IFAC-PapersOnLine*, 48(1) 111-112. Available from https://www.sciencedirect.com/science/article/pii/S2405896315001986 [accessed 10 May 2021].

European Centre for Disease Prevention and Control (2021) *MERS-CoV worldwide overview*. European Centre for Disease Prevention and Control. Available from https://www.ecdc.europa.eu/en/middle-east-respiratory-syndrome-coronavirus-mers-cov-situation-update [accessed 11 March 2021].

Feinmann, J. (2021) Why aren't covid-19 vaccines being manufactured in standard prefilled syringes? *The BMJ*, 372(263). Available from https://www.bmj.com/content/372/bmj.n263 [accessed 24 February 2021].

Franklin, O., Emmanuel, O. (2014) Logistic Regression: A Paradigm for Dichotomous response Data. *The International Journal Of Engineering And Science*, 3(6) 01-05. Available from http://www.theijes.com/papers/v3-i6/Version-6/A03660105.pdf [accessed 26 February 2021].

Frederiksen, L.S.F., Zhang, Y., Foged, C., Thakur, A. (2020) The long road towards COVID-19 Herd immunity: Vaccine platform technologies and mass immunization strategies. *Frontiers in Immunology*, 11(1817) 1-25. Available from https://www.frontiersin.org/articles/10.3389/fimmu.2020.01817/full [accessed 23 February 2021].

Garrett, J.J. (2010) Customer Loyalty and the Elements of User Experience. *Design Management review*, 17(1) 35-39. Available from https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1948-7169.2006.tb00027.x [accessed 2 May 2021].

Geraldi, J., Lechter, T. (2012) Gantt charts revisited: A critical analysis of its roots and implications to the management of projects today. *International Journal of Managing Projects in Business, 5(4)* 578-594. Available from https://www-emerald-com.proxy.library.lincoln.ac.uk/insight/content/doi/10.1108/17538371211268889/full/pdf [accessed 27 March 2021].

Gobierno De México (2021) *Open Data General Directorate of Epidemiology: General Directorate of Epidemiology*. Gobierno De México. Available from https://www.gob.mx/salud/documentos/datos-abiertos-152127 [accessed 15 April 2021].

Goguen, J.A. and Linde, C. (1993) Techniques for requirements elicitation. In: *Proceedings of the IEEE International Symposium on Requirements Engineering*, San Diego, CA, USA, 6 Jan. IEEE, 152-164. Available from https://ieeexplore-ieee-org.proxy.library.lincoln.ac.uk/stamp/stamp.jsp?tp=&arnumber=324822 [accessed 20 April 2021].

Goh, G.K-M., Dunker A.K., Foster J.A., Uversky V.N. (2020) Rigidity of the Outer Shell Predicted by a Protein Intrinsic Disorder Model Sheds Light on the COVID-19 (Wuhan-2019-nCoV) Infectivity. *Biomolecules*. 10(2) 331. Available from 10.3390/biom10020331 [accessed 11 March 2021].

GOV.UK (2021) *Joint Committee on Vaccination and Immunisation: advice on priority groups for COVID-19 vaccination, 30 December 2020*. GOV.UK. Available from https://www.gov.uk/government/publications/priority-groups-for-coronavirus-covid-19-vaccination-advice-from-the-jcvi-30-december-2020/joint-committee-on-vaccination-and-immunisation-advice-on-priority-groups-for-covid-19-vaccination-30-december-2020 [accessed 11 March 2021].

Holas, T. (undated) *Angular vs. React: Which is Better for Web Development?*. Toptal. Available from https://www.toptal.com/front-end/Angular-vs-react-for-web-development [accessed 5 April 2021].

Hu, T., Liu, Y., Zhao, M., Zhuang, Q., Xu, L., He, Q. (2020) A comparison of COVID-19, SARS and MERS. *PeerJ*, 8(August). Available from https://peerj.com/articles/9725/ [accessed 11 March 2021].

JetBrains (undated) *WebStorm*. JetBrains. Available from https://www.jetbrains.com/webstorm/ [accessed 24 March 2021].

Jin, M., Khan, A.I. (2010) Procalcitonin: Uses in the Clinical Laboratory for the Diagnosis of Sepsis. *Laboratory Medicine*, 41(3) 173-177. Available from https://doi.org/10.1309/LMQ2GRR4QLFKHCH9 [accessed 25 February 2021].

JWT.io (undated) *Introduction to JSON Web Tokens*. Auth0. Available from https://jwt.io/introduction - JWT.IO [accessed 2 May 2021].

Kamal, M.A., Raza, H.W., Alam, M.M., Su'ud M.M. (2020) Highlight the Features of AWS, GCP and Microsoft Azure that Have an Impact when Choosing a Cloud Service Provider. *International Journal of Recent Technology and Engineering (IJRTE)*, 8(5). Available from .https://d1wqtxts1xzle7.cloudfront.net/62948566/OrignalPaper-D857311841920200413-101621-1fuw219.pdf?1586846356=&response-content-disposition=inline%3B+filename%3DHighlight_the_Features_of_AWS_GCP_and_Mi.pdf&Expires=1616612509&Signature=G5wK2oxk3cWHtrbXOL4hzEWojb8043C96gUWjbf61P5Ze~2p3sT-4ILz4tlARFdaPj-8zHG0~G94Z6aGysCDKR4S5xvq7Bw7okkTaDylEMKif6br2eGCYUF2cP1OwDZnZIZtWNj73RgudwFE4-XMm~csw340zbrujDfXxSfYdP1LTAq0exouSQOiNzpbZgzgMGy~mSq5OulsnYNGmEutq91gJYDY6ddBauHfHWU7bTYUi-CL7GB1SK4fN5N15EY8m1sbOzfJL3OgIRLTt96ogPgjDZBI9k6ZGtx5exI5XWc5cIbPBY6eo~W72iooaSqf-hrqeHtdq1nqgfKcM0-M9g__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA [accessed 26 March 2021].

Kassambara (2018) *Penalized Regression Essentials: Ridge Lasso & Elastic Net*. Statistical tools for high-throughput data analysis (STHDA). Available from http://www.sthda.com/english/articles/37-model-selection-essentials-in-r/153-penalized-regression-essentials-ridge-lasso-elastic-net/ [accessed 30 April 2021].

Kumar, G., Bhatia, P.K. (2012) Impact of Agile Methodology on Software Development Process. *International Journal of Computer Technology and Electronics Engineering*, 2(4) 46-50. Available from https://d1wqtxts1xzle7.cloudfront.net/52486932/IJCTEE-Impact_of_Agile_Methodology_on_Software_Development_Process.pdf?1491387530=&response-content-disposition=inline%3B+filename%3DIJCTEE_Impact_of_Agile_Methodology_on_So.pdf&Expires=1617364150&Signature=ea7xz6XLYwf3Cxn0RrpplDU4Jp81S7I8Zs1rHzQ9BLSLYoCCXqQPdCdOW26~wwacFZJ-UhqNyFmt3eB9xbdPfmSwND32zkjw2Q42nax0KWCPyC3x-4ZHqWO~hT73uApE3Ka5pE971LmLAhLYRkkakib0xEQD8fFqk-UYf15pNQV8jpZeocNJuc8YgFG~Ouwv9xgQ-HGI6pxfT4C5lTY7umagrMqJy7XN88G3ysHOTR7A2zHBOfYMfZNqbL42Em4hrTlb67HoImI2ZYDiemAtDmG1BDuC5-lwxv4CQJW50PrP2YrUagSh3SQgIDeNWRE9h51XLIHaTb-zMRmzavOygg__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA [accessed 28 March 2021].

Kwok, K.O., Lai, F., Wei, W.I., Wong, S.Y.S., Tang, J.W.T. (2020) Herd Immunity – estimating the level required to halt the COVID-19 epidemics in affected countries. *Journal Of Infection*, 80(6) E23-E33. Available from https://www.journalofinfection.com/article/S0163-4453(20)30154-7/fulltext [accessed 23 February 2021].

Liu, S., Luo, H., Wang, Y., Cuevas, L.E., Wang, D., Ju, S., Yang, Y. (2020) Clinical characteristics and risk factors of patients with severe COVID-19 in Jiangsu province, China: a retrospective multicentre cohort study. *BMC Infectious Diseases*, 20(1) 1-9. Available from https://eds-b-ebscohost-com.proxy.library.lincoln.ac.uk/eds/pdfviewer/pdfviewer?vid=4&sid=7a730b35-e4f7-47ee-98eb-572b8de2d4f7%40sessionmgr4007 [accessed 24 February 2020].

Mahase, E. (2021) Covid-19: What new variants are emerging and how are they being investigated? *The BMJ*, 372(158). Available from https://www.bmj.com/content/372/bmj.n158 [accessed 23 February 2021].

MathWorks (undated) *Overview: Math. Graphics. Programming.* The MathWorks, Inc. Available from https://uk.mathworks.com/products/matlab.html?s_tid=hp_products_matlab [accessed 24 March 2021].

MDN Web Docs (2021) *Introduction to the server side.* Mozilla. Available from https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Introduction [accessed 5 April 2021].

Meo, S.A., Al-Khlaiwi, T., Usmani, A.M., Meo, A.S., Klonoff, D.C., Hoang, T.D. (2020) Biological and epidemiological trends in the prevalance and mortality due to outbreaks of novel coronavirus COVID-19. *Journal of King Saud University - Science*, 32(2020) 2495-2499. Available from https://www-sciencedirect-com.proxy.library.lincoln.ac.uk/science/article/pii/S1018364720301518 [accessed 22 February 2021].

Meena, R. (2020) *Are you in danger of being affected by Corona Virus?* Kaggle. Available from https://www.kaggle.com/ritikmeena/covid-19-prediction [accessed 1 February 2021].

Microsoft Azure (2021) *What is PaaS? Platform as a service.* Seattle: Microsoft. Available from https://azure.microsoft.com/en-gb/overview/what-is-paas/ [accessed 26 March 2021].

Microsoft Azure (undated) *Try Azure for Students free today: Get $100 credit and free developer tools. Sign up without a credit card.* Seattle: Microsoft. Available from https://azure.microsoft.com/en-us/free/students/ [accessed 26 March 2021].

Microsoft Docs (2020a) *Entity Framework Core*. Microsoft. Available from https://docs.microsoft.com/en-us/ef/core/ [accessed 2 May 2021].

Microsoft Docs (2020b) *Introduction to Identity on ASP.NET Core*. Microsoft. Available from https://docs.microsoft.com/en-us/aspnet/core/security/authentication/identity?view=aspnetcore-5.0&tabs=visual-studio [accessed 2 May 2021].

Microsoft (undated) *OneDrive Personal Cloud Storage. Seattle:* Microsoft. Available from https://www.microsoft.com/en-gb/microsoft-365/onedrive/online-cloud-storage [accessed 27 March 2021].

Modi, N. (2020) *Angular vs AngularJS: Differences Between Angular and AngularJS.* Stackify. Available from https://stackify.com/Angular-vs-angularjs-differences-between-Angular-and-angularjs/#:~:text=We%20can%20see%20the%20difference,based%20while%20AngularJS%20uses%20directives. [accessed 5 April 2021].

Moghnieh, R. Abdallah, D. Bizri, A.R. (2020) COVID-19: Second Wave or Multiple Peaks, Natural Herd Immunity or Vaccine – We should be Prepared. *Disaster Medicine and Public Health Preparedness,* (September) 1-8. Available from https://doi.org/10.1017/dmp.2020.349 [accessed 23 February 2021].

Mukherjee, T. (2020) *COVID-19 patient pre-condition dataset: Data obtained from Mexican government data set*. Kaggle. Available from https://www.kaggle.com/tanmoyx/covid19-patient-precondition-dataset [accessed 15 April 2021].

Muşlu, K., Brun, Y., Holmes, R., Ernst, M.D., Notkin, D. (2012) Speculative analysis of integrated development environment recommendations. *ACM SIGPLAN Notices,* 47(10). Available from https://dl.acm.org/doi/abs/10.1145/2398857.2384665 [accessed 5 April 2021].

Olliaro, P. (2021) What does 95% COVID-19 vaccine efficacy really mean? *The Lancet Infectious Diseases,* February. Available from https://www.thelancet.com/journals/laninf/article/PIIS1473-3099(21)00075-X/fulltext [accessed 11 March 2021].

Petersen, K., Wohlin, C., Baca, D. (2009) The Waterfall Model in Large-Scale Development. In: Bomarius F., Oivo M., Jaring P., Abrahamsson P. (eds) Product-Focused Software Process Improvement. PROFES 2009. Lecture Notes in Business Information Processing, vol 32. Berlin. Berlin: Springer-Verlag Berlin Heidelberg, 386-400. Available from https://link.springer.com/chapter/10.1007/978-3-642-02152-7_29 [accessed 1st April 2021].

Pinheiro, J. (2018) *Software Development Life Cycle (SDLC) phases.* Medium. Available from https://medium.com/@jilvanpinheiro/software-development-life-cycle-sdlc-phases-40d46afbe384 [accessed 1st April 2021] .

Pollock, A.M., Lancaster, J. (2021) Asymptomatic transmission of covid-19. *The BMJ*, 371(4851). Available from https://www.bmj.com/content/371/bmj.m4851 [accessed 23 February 2021].

Ragupathi, M. T. S., De Sanctis, V., Singleton, J. (2017) *ASP.NET Core: Cloud-ready, Enterprise Web Application Development* [ebook]. Birmingham: Packt Publishing. Available from https://web-a-ebscohost-com.proxy.library.lincoln.ac.uk/ehost/ebookviewer/ebook/bmxlYmtfXzE1MzMxNTRfX0FO0?sid=c4705572-db94-41a2-8381-8f6536c0c2b7@sdc-v-sessmgr01&vid=0&format=EB&rid=1 [accessed 24 March 2021].

Ranstam, J., Cook, J.A. (2018) LASSO Regression. *British Journal of Surgery*, 105(10) 1348. Available from https://doi.org/10.1002/bjs.10895 [accessed 25 February 2021].

React (2021) *React.* React. Available from https://reactjs.org/ [accessed 5 April 2021].

Romanyuk, O. (2019) *Angular vs React: Which One to Choose for Your App.* freeCodeCamp. Available from https://www.freecodecamp.org/news/Angular-vs-react-what-to-choose-for-your-app-2/ [accessed 5 April 2021].

Ruparelia, N.B. (2010) Software development lifecycle-models. *ACM SIGSOFT Software Engineering Notes*, 35(3). Available from https://dl-acm-org.proxy.library.lincoln.ac.uk/doi/pdf/10.1145/1764810.1764814 [accessed 29 March 2021].

Salas-Zárate, M.P., Alor-Hernández, G., Valencia-García, R., Rodríguez-Mazahua, L., Rodríguez-González, A., Cuadrado, J.L.L. (2015) Analyzing best practices on Web Development frameworks: The Lift approach. *Science of Computer Programming*, 102(May) 1-19. Available from https://www.sciencedirect.com/science/article/pii/S0167642314005735 [accessed 5 April 2021].

Schmitz, H., Wübker, A. (2011) What Determines Influenza Vaccination Take-up of elderly Europeans? *Health Economics*, 20(October) 1281-1297. Available from https://onlinelibrary.wiley.com/doi/abs/10.1002/hec.1672 [accessed 25 February 2021].

Schwaber, K. (1997) Scrum Development Process. In: Sutherland J., Casanave C., Miller J., Patel P., Hollowell G. (eds) *Business Object Design and Implementation*, London. London: Springer-Verlag London Limited, 117-134. Available from https://link.springer.com/chapter/10.1007/978-1-4471-0947-1_11 [accessed 28 March 2021].

Shang, Y., Liu, T., Wei, Y., Li, J., Shao, L., Liu, M., Zhang, Y., Zhao, Z., Xu, H., Peng, Z., Wang, X., Zhou, F. (2020) Scoring systems for predicting mortality for severe patients with COVID-19. *EclinicalMedicine*, 24(2020) 100426-undefined. Available from https://eds-a-ebscohost-com.proxy.library.lincoln.ac.uk/eds/detail/detail?vid=0&sid=03f1d743-7bea-4c9d-b419-4eff924fddc7%40sessionmgr4008&bdata=JnNpdGU9ZWRzLWxpdmUmc2NvcGU9c2l0ZQ%3d%3d#AN=S258953702030170X&db=edselp [accessed 25 February 2021].

StatAnalytica (2019) *R vs Matlab: Which one is more powerful and Why?*. StatAnalytica. Available from https://statanalytica.com/blog/r-vs-matlab/ [accessed 24 March 2021].

Szumilas, M. (2010) Explaining Odds Ratios. *Journal of the Canadian Academy of Child and Adolescent Psychiatry*, 19(3) 227-229. *Available from* https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2938757/ [accessed 10 May 2021].

Tan, W.S., Liu, D., Bishu, R. (2009) Web evaluation: Heuristic evaluation vs. user testing. *International Journal of Industrial Ergonomics*, 39(4) 621-627. Available from https://pdf.sciencedirectassets.com/271473/1-s2.0-S0169814109X00041/1-s2.0-S016981410800053X/main.pdf?X-Amz-Security-Token=IQoJb3JpZ2luX2VjELz%2F%2F%2F%2F%2F%2F%2F%2F%2F%2FwEaCXVzLWVhc3QtMSJIMEYCIQCH%2FL5TQwwdJhfm30Tcgzi8Env9i2x9zXWyLwfNsn0ZsQIhAIpxUwBGLE31cDmSlDVH1%2BBgUpVqiwtFmO5m9A2ZgqwMKrQDCBQQAxoMMDU5MDAzNTQ2ODY1IgyTXHgGEoISv7kqURUqkQMCtDIs1w41toG2PSkQUMTl1M77f9kO4%2BoP%2BnUGQ0FXaWdkF1wMxNs6UgRT%2BUTfXY0pCaFFvMnt2b7UGfyKxWyf8ObFaESkzruquu7y2ORpu9f6WMz0wpzCNMeT75953OUvRubY1nOwaZL5gNEdSgyO5SnC9lGgDLtWJc%2FXjOGMr9hCFC3xoFtCeNfJzpnDKJr3zCZW6R5pKJ9MmY6R%2BziOUdbiKc%2Bqr4NzXPhgjn8DV43GOEu3oj7Ut7OXpU1a3USpowein69hXj%2FgWoBGIwCbXL4IOOCZjAdx5mMP9brED0%2FJxVw3Plf%2BMLaJisc2rpKt4vQChyhSpUwScFFmCky6yMq5XXR%2B4eQdr0dpeYMVVMxxR2fSTkDBnBaEbS8X0DiHTdWJuTm0YKtPZpj3L52ijYaRiRTVJmHWRacm3XXfggc%2F4ypPakrXMoZtTt9ot15DxabMXig3ldR0bM1dN1LZev4hcrtw%2B3tGFyz5v%2FujYMqfAUpknA9y3MEq4GRBI98%2B5UnDrmc14TKNy%2FNXCPqYxzDA3quDBjrqARvqxcOGzq28ZHFhxWfnCLZi3zDb%2BjBQS%2BhD2whds7gdodF32vQdLl%2FZqUwVUwRF0bM5T3PwAXjir8ZobrTek%2FWj5BQUtPllAjf87Mc7QdyKGY6OeiIlcTlxrC%2FhOel0bl%2BOiipcETqqNwlZ%2FQ4ZQHTI%2B6u2K0G9PRPu%2BpJUxyEOwdxPlZ7pQ6FMIP5INmA3c%2B%2B6Y3PX41U3JO2zMKZQYsqTHaaa%2FZd0p%2FRVI8Z6ctYcWiPhJRKc3pEloqyKH14HEHQG6xuf9bxSLyExPBULYpOrF5MpKMwUQzihxRhiDZfiWZO6hBPZ2hkDag%3D%3D&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20210405T115755Z&X-Amz-SignedHeaders=host&X-Amz-Expires=300&X-Amz-Credential=ASIAQ3PHCVTYULDX4HMU%2F20210405%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Signature=36ac626f969110b72a2ad771525c0e56d8302967e6d9f125ed34e2febb21667f&hash=964bdf0c64fc85a4521b40281827c46500dc52d6cd1c937b646dcbb92635c8e9&host=68042c943591013ac2b2430a89b270f6af2c76d8dfd086a07176afe7c76c2c61&pii=S016981410800053X&tid=spdf-f686224e-bd2a-45ca-ae5d-d4770975b2df&sid=e46cd10c8c2a6243c1993793a0b38f315c29gxrqb&type=client [accessed 28 March 2021].

Techson, M. (2020) *AngularJS LTS Extended in response to COVID-19*. Angular Blog. Available from https://blog.Angular.io/angularjs-lts-extended-in-response-to-covid-19-321b037212f5 [accessed 5 April 2021].

The R Foundation (undated) *What is R?*. The R Foundation. Available from https://www.r-project.org/about.html [accessed 24 March 2021].

Thomas, N. (undated) *How To Use The System Usability Scale (SUS) To Evaluate The Usability Of Your Website.* UsabilityGeek. Available from https://usabilitygeek.com/how-to-use-the-system-usability-scale-sus-to-evaluate-the-usability-of-your-website/ [accessed 25 April 2021].

Trello (2021) *Trello helps teams move work forward.* Trello. Available from https://trello.com/en-GB [accessed 27 March 2021].

University Of Michigan (2021) *Google Cloud for Students*. The Regents of the University of Michigan. Available from https://its.umich.edu/computing/virtualization-cloud/google-cloud/students [accessed 26 March 2021].

Usability.gov (undated) *System Usability Scale (SUS).* Usability.gov. Available from https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html [accessed 29 March 2021].

User1607 (2018) *logit – interpreting coefficients as probabilities* [StackExchange]. 24 August. Available from https://stats.stackexchange.com/questions/363791/logit-interpreting-coefficients-as-probabilities - user1607 [accessed 25 April 2021].

Velavan, T.P. and Meyer, C.G. (2020) The COVID-19 epidemic. *Tropical Medicine & International Health*, 25(3) 278-280. Available from https://doi.org/10.1111/tmi.13383 [accessed 11 March 2021].

Visual Studio Code (undated) *Code editing. Redefined.* Seattle: Microsoft. Available from https://code.visualstudio.com/ [accessed 24 March 2021].

Visual Studio (undated) Visual Studio: Best-in-class tools for any developer. Microsoft. Available from https://visualstudio.microsoft.com/ [accessed 24 March 2021].

Wakode, R.B., Raut, L.P., Talmale, P. (2015) Overview on Kanban Methodology and its implementation. *International Journal for Scientific Research & Development*, 3(02) 2518-2521. Available from https://d1wqtxts1xzle7.cloudfront.net/48171603/Overview_on_Kanban_Methodology_and_its_Implementation.pdf?1471604202=&response-content-disposition=inline%3B+filename%3DOverview_on_Kanban_Methodology_and_its_I.pdf&Expires=1617708071&Signature=B8XNwmTe3diXaoaDfOeUNocckttx3b8875nmHe40wYMWBsczcdWijkoSE85cwjmYwryMkDLap2WKNiNeJCPSp1kq2ja4W~8J-WY~1~EDuNgZJj0MtJX2lC~FFuqamItV84c~kjCFY7xQcHSf912KFWojwXFLLTsYq5cEN7Mx6IowIw2Cotmz-79a8k4fDOnrroE3-krGMqTtlnRjRdrN56NGpzJdFe8qHHDjR~ZkkZ1OAT-3hISExqKYt-mRpg2Ws6-JbnfjCx78MyJUdtZQsDmYwBY3uboWa513G-J5EiN2jDFCWRTWxGemGi3SPacZlt5xSlvEr6WyPo2I6Uyb0Q__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA [accessed 1st April 2021].

WebStorm (2021) *Deploy your application*. JetBrains. Available from https://www.jetbrains.com/help/webstorm/deploying-applications.html [accessed 24 March 2021].

World Health Organization (2021) *The Oxford/AstraZeneca COIVD-19 vaccine: what you need to know*. World Health Organization. Available from https://www.who.int/news-room/feature-stories/detail/the-oxford-astrazeneca-covid-19-vaccine-what-you-need-to-know#:~:text=How%20efficacious%20is%20the%20vaccine,associated%20with%20greater%20vaccine%20efficacy. [accessed 11 March 2021].

World Health Organization (2020a) *WHO Coronavirus Disease (COVID-19) Dashboard*. World Health Organization. Available from https://covid19.who.int/ [accessed 22 February 2021].

World Health Organization (2020b) *Immunization Coverage*. World Health Organization. Available from https://www.who.int/en/news-room/fact-sheets/detail/immunization-coverage [accessed 23 February 2021].

World Health Organization (undated) *Severe Acute Respiratory Syndrome (SARS)*. World Health Organization. Available from https://www.who.int/health-topics/severe-acute-respiratory-syndrome [accessed 11 March 2021].

Worldometer (2021) *Countries where COVID-19 has spread*. Worldometer. Available from https://www.worldometers.info/coronavirus/countries-where-coronavirus-has-spread/ [accessed 11 March 2021].

Zowghi, D. and Coulin, C. (eds) (2005) *Requirements Elicitation: A Survey of Techniques, Approaches, and tools* [ebook]. Berlin: Springer. Available from https://link.springer.com/chapter/10.1007/3-540-28244-0_2 [accessed 25 April 2021].

Word Count

Total: 19471

Appendices

1 - Methodology

	Visual Studio	Visual Studio Code
Price	Free Community edition but also has paid Professional and Enterprise editions with extra features	Free
Debugger	Yes	Yes
Deployment Capabilities	Can directly publish to cloud services	Can directly publish to cloud services
Code Refactoring	Yes	Yes
Auto Complete	IntelliSense code suggestions	IntelliSense code suggestions
Key Languages Supported	C#, HTML, JavaScript, CSS, PHP, Python, Visual Basic, F#, C++	C#, HTML, Python, Java, JavaScript, Ruby, CSS, C++ and hundreds of others supported via extensions
Error Highlighting	Yes	Yes
Templates	Huge range of templates to quickly begin development	None out of the box but can integrate via extensions
Extensions/Plugins	Has a range of extensions however feature-rich out of the box meaning less need for them	Basic features out of the box but customisable with a massive number of extensions
Open Source	Community edition is but paid editions are not	Yes
Version Control	Git or Team Foundation Version Control (TFVC)	Git
References	Visual Studio, undated	Visual Studio Code, undated

Table 3: Comparison of Visual studio and Visual studio code IDE's

	Angular	React
Framework or Library?	Framework	Library
Supported?	Yes	Yes
Complexity	No additional libraries needed, comes with a vast array of modules, first-party libraries, and features out the box	UI development only, additional third-party libraries must be used for other features
Language	Typescript, JavaScript, HTML	JS ES6+, JSX Script
App Structure	Components that are directives with templates. Fast and simpler to understand	Flexible component-based
UI components	Built-in Angular Material modules for UI design	Requires external UI-libraries
Dependency injection	Yes	No
Data binding	Fast Bidirectional Binding	Unidirectional binding
Portability	Suitable for both web and mobile development	Suitable for both web and mobile development
Routing	Built-in angular routing module	Requires external routing modules
Free to Use?	Yes	Yes
Open Source?	No	Yes
References	Angular, undated b Holas, undated Romanyuk, 2019 Modi, 2020	ReactJS, 2021 Holas, undated Romanyuk, 2019

Table 4: Comparison of the Angular framework and the JavaScript Library React

2 - Design

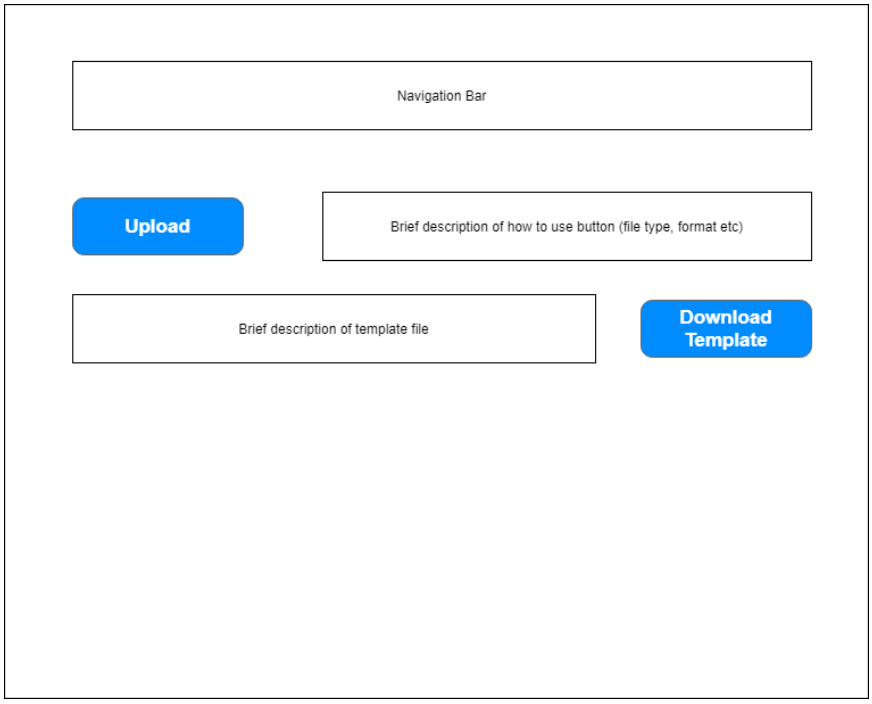


Figure 32: High Risk Analysis Page - Before User Upload

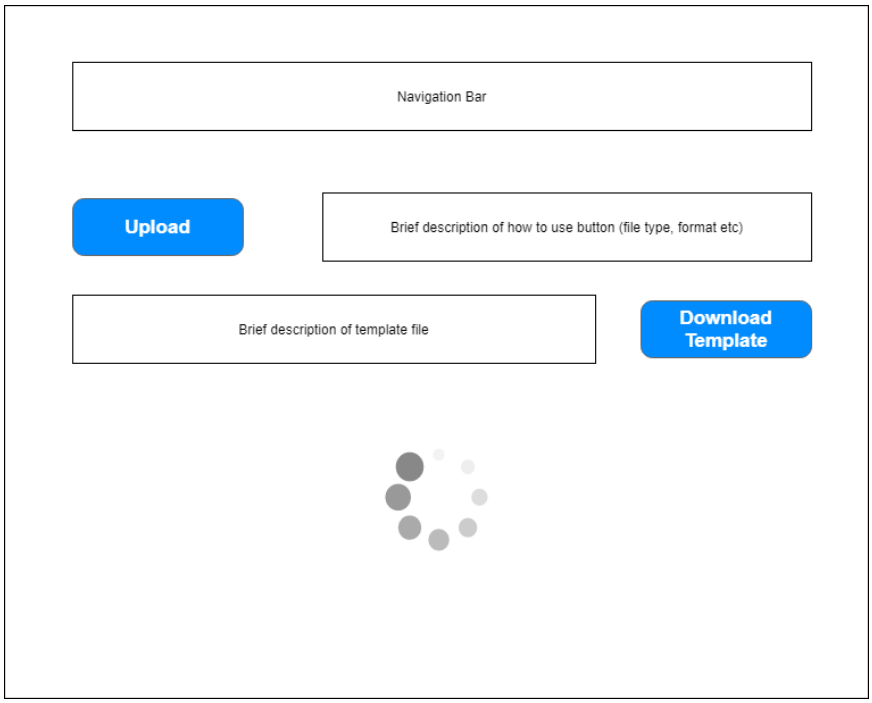


Figure 31: High Risk Analysis Page - During User Upload

3 - Development

a) Algorithm Development

	id	sex	patient_type	entry_date	date_symptoms	date_died	intubed	pneumonia	age	pregnancy	diabetes	copd	asthma	inmsupr	hypertension	other_disease	cardiovascular	obesity	renal_chronic	tobacco	contact_other_covid	covid_res	icu
1	16169f	0	1	2020-05-04	2020-05-02	9999-99-99	97	0	27	97	0	0	0	0	0	0	0	0	0	0	0	1	97
2	1009bf	0	1	2020-03-19	2020-03-17	9999-99-99	97	0	24	97	0	0	0	0	0	0	0	0	0	0	99	1	97
3	0b5948	0	0	2020-04-17	2020-04-10	9999-99-99	0	1	30	97	0	0	0	0	0	0	0	0	0	0	99	1	0
4	1beec8	0	0	2020-04-16	2020-04-16	43950	0	1	47	97	1	0	0	0	0	0	0	0	0	0	99	1	1
5	1.75E+56	0	0	2020-04-22	2020-04-13	9999-99-99	0	0	63	97	0	0	0	0	1	0	0	0	0	0	99	1	0

Figure 33: First 5 records of Dataset before pre-processing

	id	sex	pneumonia	age	pregnancy	diabetes	copd	asthma	inmsupr	hypertension	cardiovascular	obesity	renal_chronic	tobacco	died	severe_covid	age_high
1	16169f	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1009bf	0	0	24	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0b5948	0	1	30	0	0	0	0	0	0	0	0	0	0	0	0	0
4	1beec8	0	1	47	0	1	0	0	0	0	0	0	0	0	1	1	0
5	1.75E+56	0	0	63	0	0	0	0	0	1	0	0	0	0	0	0	1

Figure 34: First 5 records of dataset after pre-processing

LASSO Regression

pneumonia	2.37756627
sex	-0.32744532
age	0.03873916
age_high	0.23988520
pregnancy	.
diabetes	0.29189669
copd	0.04368591
asthma	.
inmsupr	0.11683222
hypertension	0.14833490
cardiovascular	.
obesity	0.14443793
renal_chronic	0.67812725
tobacco	.

Figure 35: LASSO Regression Significant factor Coefficient Results

Odds Ratio

pneumonia	977.863867
renal_chronic	97.018461
diabetes	33.896468
sex	-27.923730
age_high	27.110322
hypertension	15.990129
obesity	15.538998
inmsupr	12.393084
age	3.949931
copd	4.465419
pregnancy	0.000000
asthma	0.000000
cardiovascular	0.000000
tobacco	0.000000

Figure 36: LASSO Regression Odds Ratio

```r
library(magrittr)
library(dplyr)

x_test <- model.matrix(severe_covid~pneumonia + sex + age + age_high + pregnancy + diabetes + copd + asthma + inmsupr
                       + hypertension + cardiovascular + obesity + renal_chronic + tobacco, test)

lasso_prob <- LassoModel %>% predict(newx = x_test)
predicted.classes <- ifelse(lasso_prob > 0.5, 1, 0)
observed.classes <- test$severe_covid
accuracyofLasso <- mean(predicted.classes==observed.classes)

```

Figure 37: Script to calculate the accuracy of the LASSO model

- Participant ID: 1994
  - Original Scores = 3,2,4,2,4,3,4,2,4,2
  - Converted Scores = 2,3,3,3,3,2,3,3,3,3
  - Sum = 28
  - **Score = 28*2.5 = 70**

Figure 38: Example of how SUS Scores are generated

Feature	**Odds Ratio % to 1 d.p**	**Point Value**	**Reasoning**
Pneumonia	977.9	100	Having this condition means you are 977% more likely to get severe-COVID-19. Therefore, it deserves the top marks
Renal_chronic	97	75	This condition is second in the severity ranking and needs to be treated as such with a high score
Diabetes	33.9	50	The last of the big three with it having a 33% odds ratio. Having this condition just about means you are at high risk from severe covid
Male	27.9	40	Just under the high-risk threshold being a male makes you 27% more likely to get severe_covid but is not enough to get a vaccination outright
Age >= 50	27.1	40	As above, being over 50 makes you 27% more likely to get severe covid
Hypertension	16.0	30	Having hypertension puts you at 16% more likely to get severe covid
Obesity	15.5	30	Much like hypertension, obesity puts you at ~16% more likely to get severe COVID
Immunosuppressed	12.4	25	Slightly less than obesity as this only puts you at 12% more likely to get severe COVID
COPD	4.5	25	The least severe condition of all those analysed having COPD only puts you at a 4.5% greater chance of getting severe covid

Table 5: Example of how SUS Scores are generated

b) Angular Development

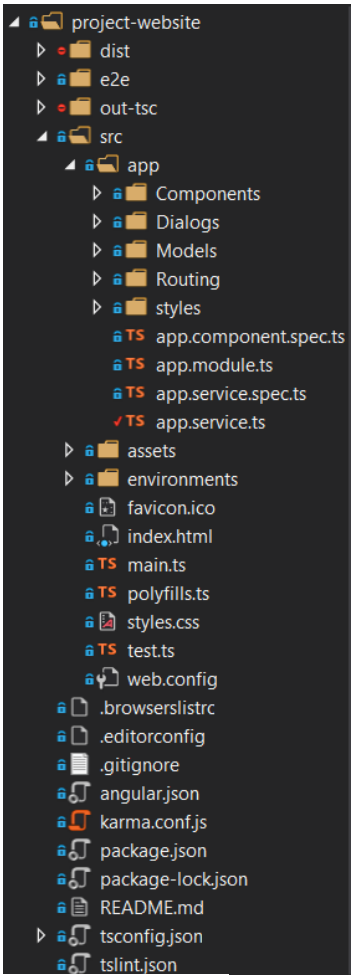


Figure 39: File structure of Angular Application

```css
.index-title {
  transform: translateY(-600px);
  animation: 1.2s slideIn ease-in-out forwards 0.75s;
}

.index-title::before {
  content: '';
  width: 0%;
  height: 5px;
  background: #005EB8;
  position: absolute;
  bottom: 5px;
  -webkit-animation: 1s underline ease-in-out forwards 2s;
  animation: 1s underline ease-in-out forwards 2s;
  mix-blend-mode: screen;
}
```

Figure 40: CSS3 animation of home page title code

```typescript
const routes: Routes = [
  { path: '', redirectTo: '/User', pathMatch: 'full' },
  { path: 'User', component: UserComponent },
  { path: 'Home', component: HomeComponent, canActivate: [AuthGuard] },
  { path: 'HRA', component: HRAComponent, canActivate: [AuthGuard] },
  { path: 'Results', component: ResultsComponent, canActivate: [AuthGuard] },
  { path: 'About', component: AboutComponent }, // can access about without being logged in to find out more about the app
  { path: 'Registration', component: RegistrationComponent },
  { path: 'login', component: LoginComponent },
  { path: '**', component: PageNotFoundComponent },
];
```

Figure 41: Angular Routing

```typescript
HRA_nav() {
  this.burgerActive = false;
  this.appService._router.navigate(['/HRA']);
}
Home_nav() {
  this.burgerActive = false;
  this.appService._router.navigate(['/Home']);
}
Results_nav() {
  this.burgerActive = false;
  this.appService._router.navigate(['/Results']);
}
About_nav() {
  this.burgerActive = false;
  this.appService._router.navigate(['/About']);
}
```

Figure 42: Routing Functions

```
<!-- Desktop Nav Bar-->
<div class="row nav-container" *ngIf="this.showNav()">
  <div class="col-1 d-flex justify-content-center align-items-center logo-container-container" style="padding-right:0;">
    <div class="logo-container d-flex justify-content-center align-items-center">
      <p class="logo">CoV-Risk</p>
    </div>
  </div>
  <div class="col-10 nav-bar-col">
    <div class="row nav-bar">
      <div (click)="Home_nav();" class="col-sm-3 d-flex justify-content-center home-nav cursor" style="height:fit-content;">
        <fa-icon class="home-icon" [icon]="appService.faHome" size="2x"></fa-icon>
        <span class="header-text">Home</span>
      </div>
      <div (click)="HRA_nav();" class="col-sm-3 d-flex justify-content-center HRA-nav cursor" style="height:fit-content;">
        <fa-icon class="" [icon]="appService.faUserMd" size="2x"></fa-icon>
        <span class="header-text">High Risk Analysis</span>
      </div>
      <div (click)="Results_nav();" class="col-sm-3 d-flex justify-content-center Results-nav cursor" style="height:fit-content;">
        <fa-icon class="" [icon]="appService.faChartPie" size="2x"></fa-icon>
        <span class="header-text">Results</span>
      </div>
      <div (click)="About_nav();" class="col-sm-3 d-flex justify-content-center About-nav cursor" style="height:fit-content;">
        <fa-icon class="" [icon]="appService.faTerminal" size="2x"></fa-icon>
        <span class="header-text">About</span>
      </div>
    </div>
  </div>
  <div class="col-1 d-flex justify-content-center align-items-center">...</div>
</div>
<!--Mobile Nav Bar-->
<div class="row mobile-nav-bar justify-content-center align-items-center">...</div>
<!--Content-->
<div class="container" style="min-height:100vh">
  <router-outlet></router-outlet>
</div>
```

Figure 43: App.Component.html showing the desktop navigation bar and router outlet tags

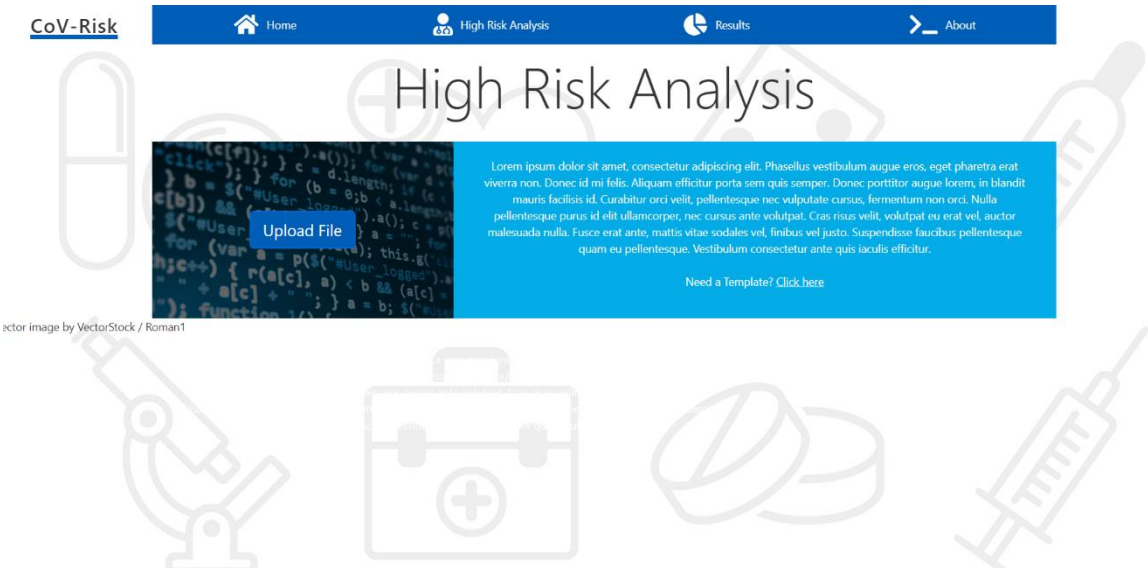


Figure 44: HRA Initial Development - Show less

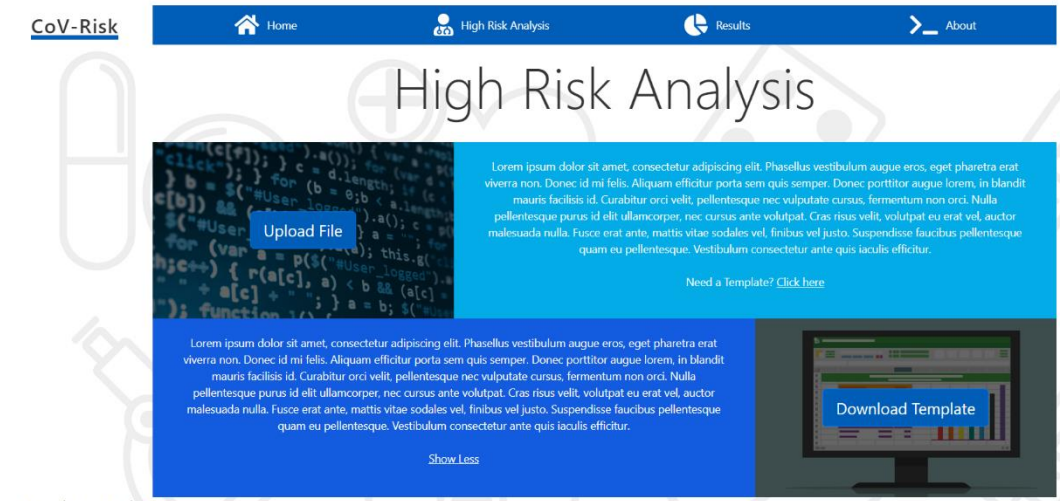


Figure 45: HRA Initial Development - Show More

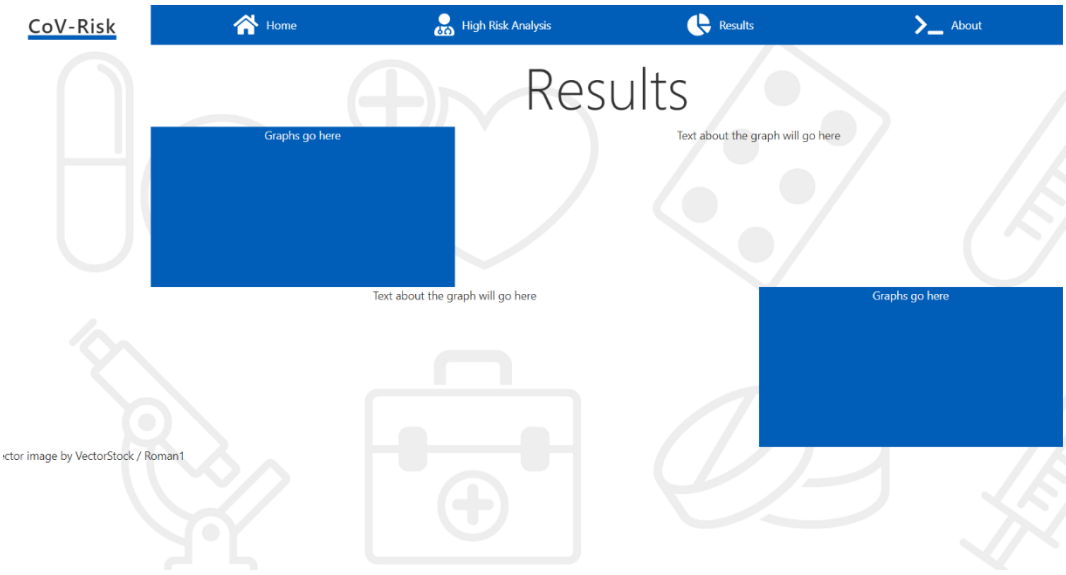


Figure 46: Results Page Initial Development

```
<div [hidden]="patientsForResults.patients.length == 0" class="mat-elevation-z4 resultsTable">
  <table mat-table [dataSource]="ResultsDataSource" style="width:inherit; border-collapse:separate" matSort>
    <ng-container matColumnDef="select">
      <th mat-header-cell *matHeaderCellDef class="table-head">Select</th>
      <td mat-cell *matCellDef="let patient">
        <mat-checkbox [(ngModel)]="patient.deletePatient" (change)="this.DisableDelete()"></mat-checkbox>
      </td>
    </ng-container>
    <ng-container matColumnDef="id" sticky>
      <th mat-header-cell *matHeaderCellDef class="table-head" mat-sort-header> ID </th>
      <td mat-cell *matCellDef="let patient" [style.color]="patient.rowColour"> {{patient.id}} </td>
    </ng-container>
    <ng-container matColumnDef="sex">
      <th mat-header-cell *matHeaderCellDef class="table-head" mat-sort-header> Sex </th>
      <td mat-cell *matCellDef="let patient" [style.color]="patient.rowColour"> {{patient.sex}} </td>
    </ng-container>
    <ng-container matColumnDef="age">
      <th mat-header-cell *matHeaderCellDef class="table-head" mat-sort-header> Age </th>
      <td mat-cell *matCellDef="let patient" [style.color]="patient.rowColour"> {{patient.age}} </td>
    </ng-container>
    <ng-container matColumnDef="preconditions">
      <th mat-header-cell *matHeaderCellDef class="table-head" mat-sort-header> Preconditions </th>
      <td mat-cell *matCellDef="let patient" [style.color]="patient.rowColour"> {{patient.preconditions.length > 0 ? patient.preconditions : '--'}} </td>
    </ng-container>
    <ng-container matColumnDef="riskScore">
      <th mat-header-cell *matHeaderCellDef class="table-head" mat-sort-header> Risk Score </th>
      <td mat-cell *matCellDef="let patient" [style.color]="patient.rowColour"> {{patient.riskScore}} </td>
    </ng-container>
    <ng-container matColumnDef="highRisk">
      <th mat-header-cell *matHeaderCellDef class="table-head" mat-sort-header> High Risk </th>
      <td mat-cell *matCellDef="let patient" [style.color]="patient.rowColour"> {{patient.highRisk.toString().toUpperCase()}} </td>
    </ng-container>
    <ng-container matColumnDef="edit">
      <th mat-header-cell *matHeaderCellDef class="table-head" style="text-align:center;"> More </th>
      <td mat-cell *matCellDef="let patient" style="text-align:center;">
        <fa-icon matTooltipClass="custom-tooltip" matTooltip="View More Patient Information" style="color: #005EB8; cursor:pointer;" class=""
                 [icon]="appService.faInfoCircle" size="2x" (click)="EditPatient(patient);"></fa-icon>
      </td>
    </ng-container>
    <tr mat-header-row *matHeaderRowDef="columnsToDisplayForResults"></tr>
    <tr mat-row *matRowDef="let myRowData; columns: columnsToDisplayForResults"></tr>
    <!-- Row shown when there is no matching data. -->
    <tr class="mat-row" *matNoDataRow>
      <td class="mat-cell" colspan="4">No data matching the filter "{{input.value}}"</td>
    </tr>
  </table>
</div>
<mat-paginator class="paginator table-head" [pageSizeOptions]="[8]" showFirstLastButtons [hidden]="patientsForResults.patients.length == 0"></mat-paginator>
```

Figure 47: Results table HTML code

Figure 48: Improved HRA page with the better-styled graph

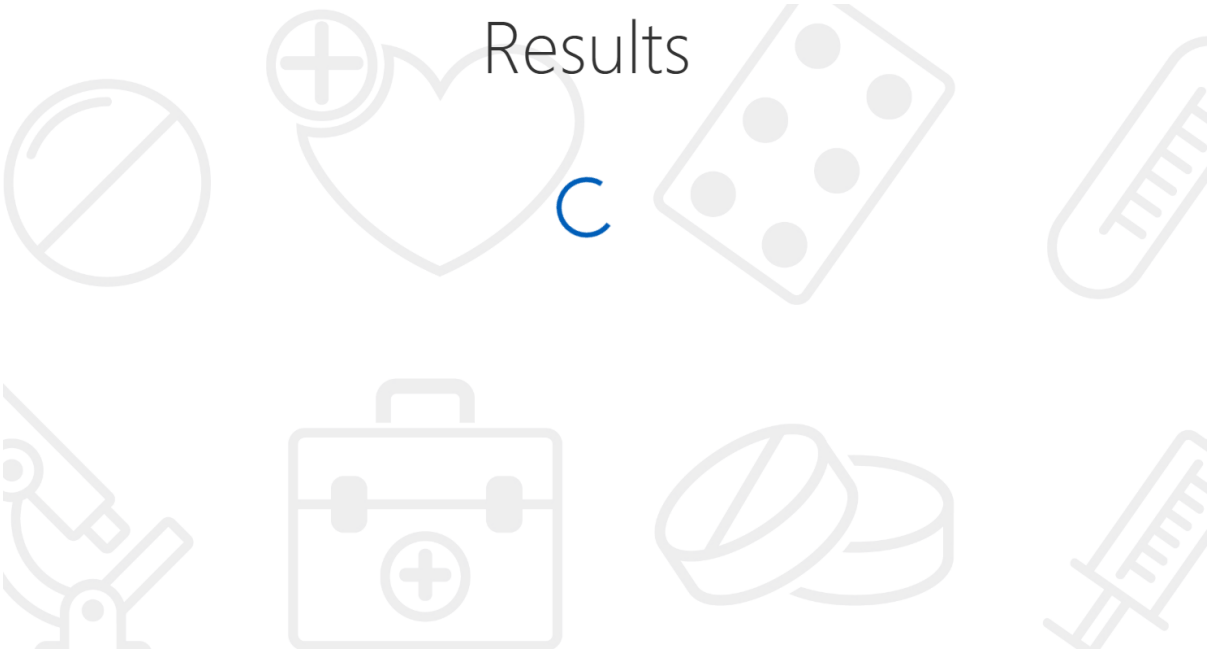


Figure 49: Angular Material spinner showing when the application is processing a request

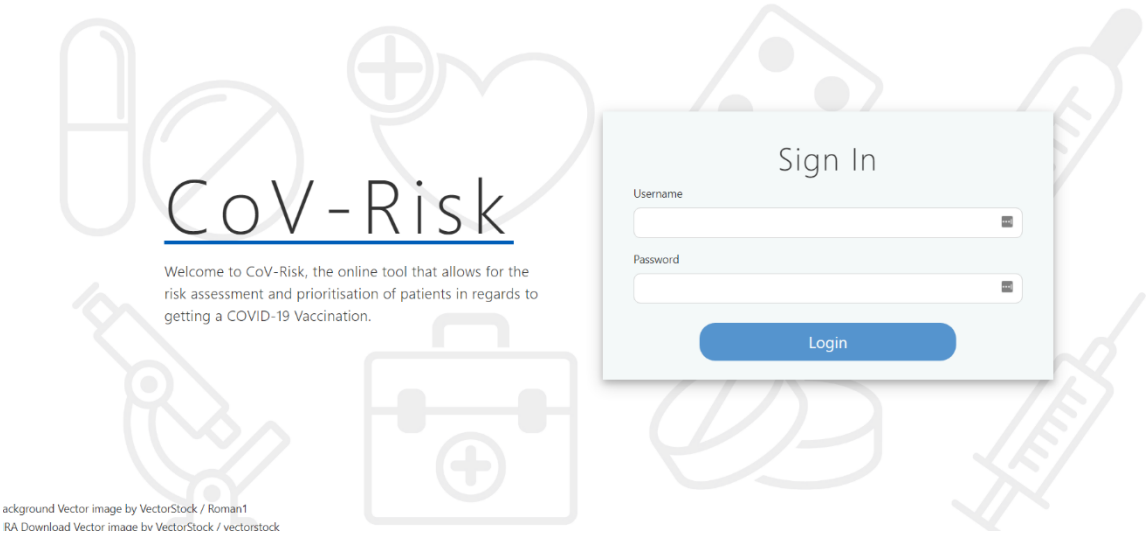


Figure 50: sign-in component

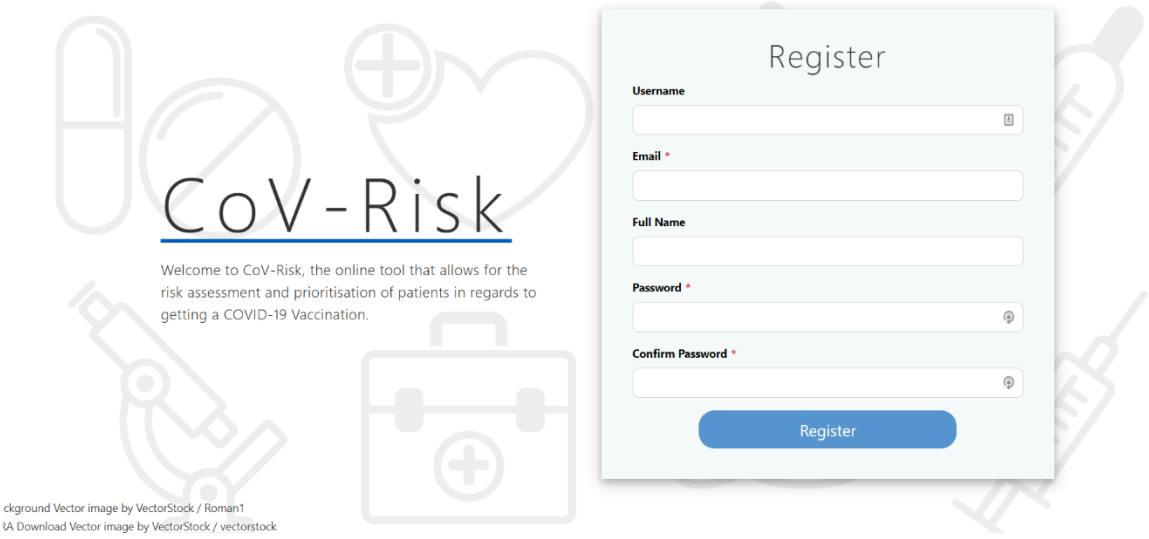


Figure 51: Register component

```
import { Injectable } from '@angular/core';
import { CanActivate, ActivatedRouteSnapshot, RouterStateSnapshot, Router } from '@angular/router';
import { Observable } from 'rxjs';

@Injectable({
  providedIn: 'root'
})
export class AuthGuard implements CanActivate {
  constructor(private router: Router) { }

  canActivate(
    next: ActivatedRouteSnapshot,
    state: RouterStateSnapshot): boolean {
    if (localStorage.getItem('token') != null)
      return true;
    else {
      this.router.navigate(['/User']);
      return false;
    }

  }
}

```

Figure 52: Auth Guard component ensures only users with a registered token can make a routing request

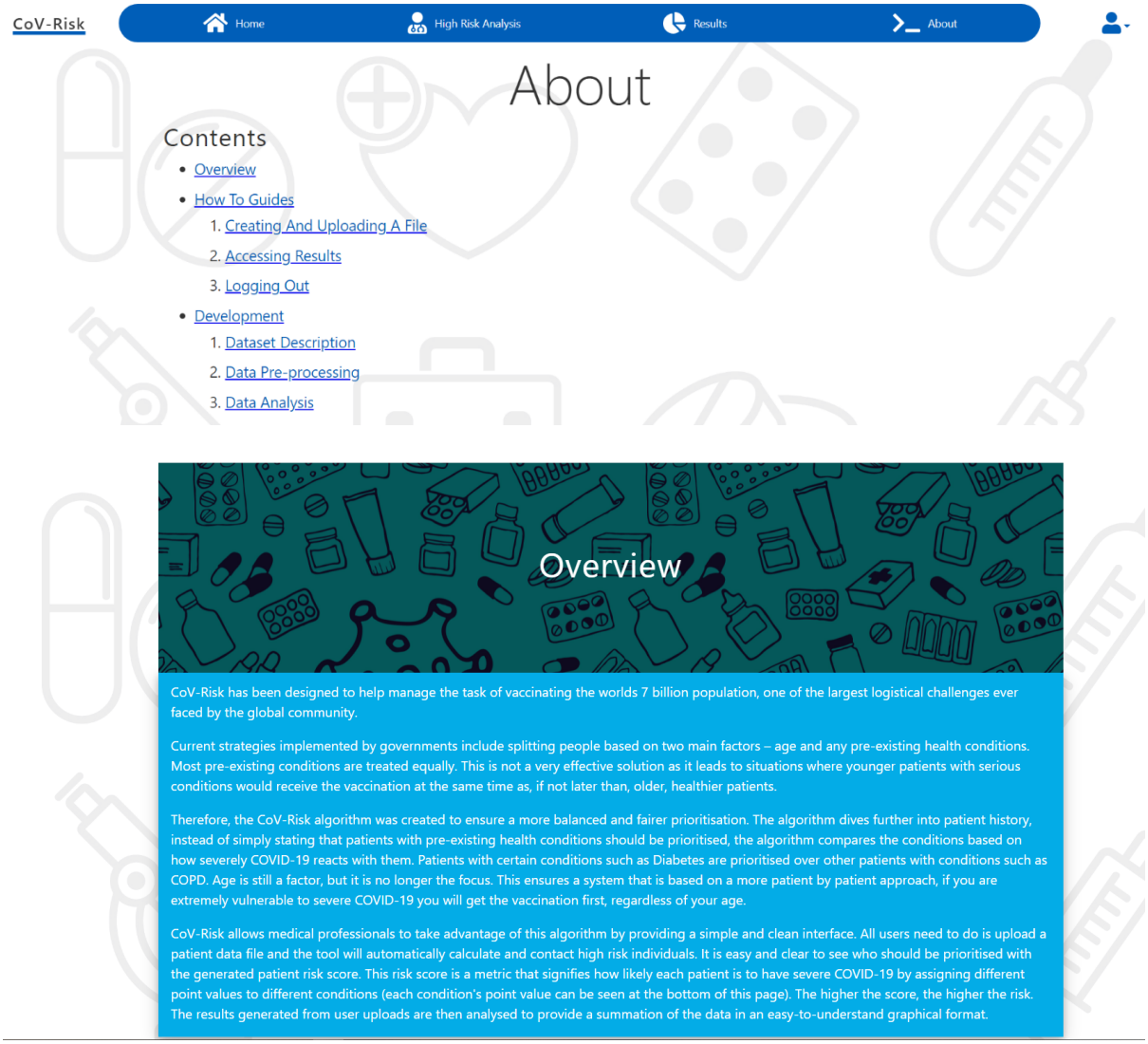


Figure 53: About page

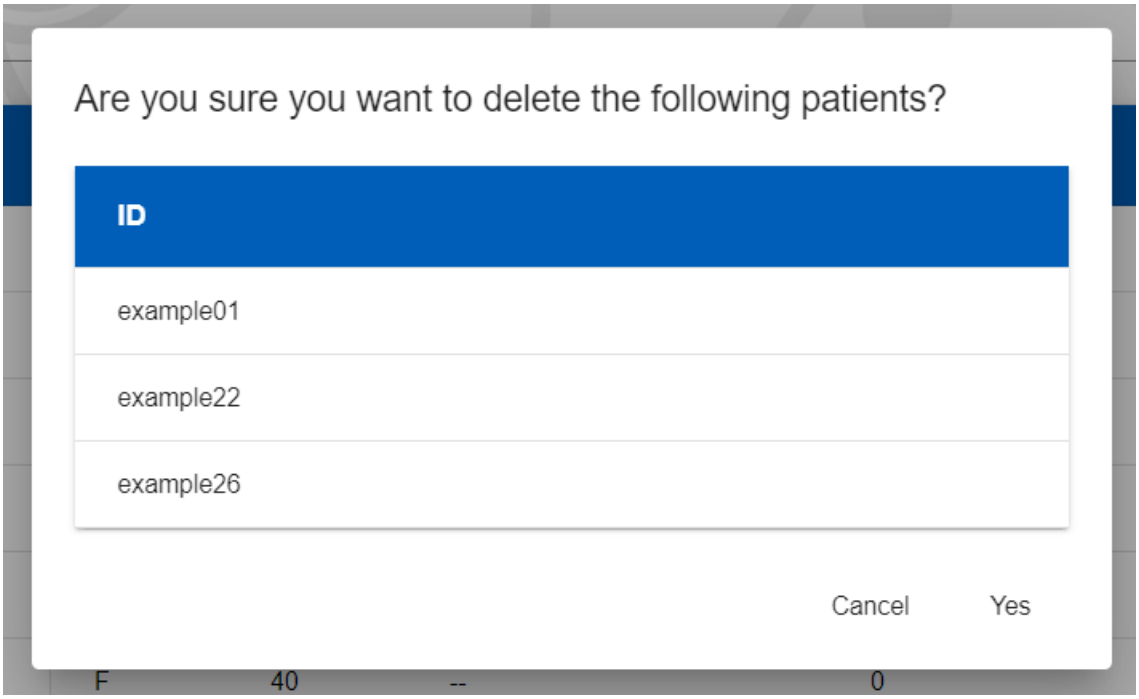


Figure 54: Delete patient confirmation box

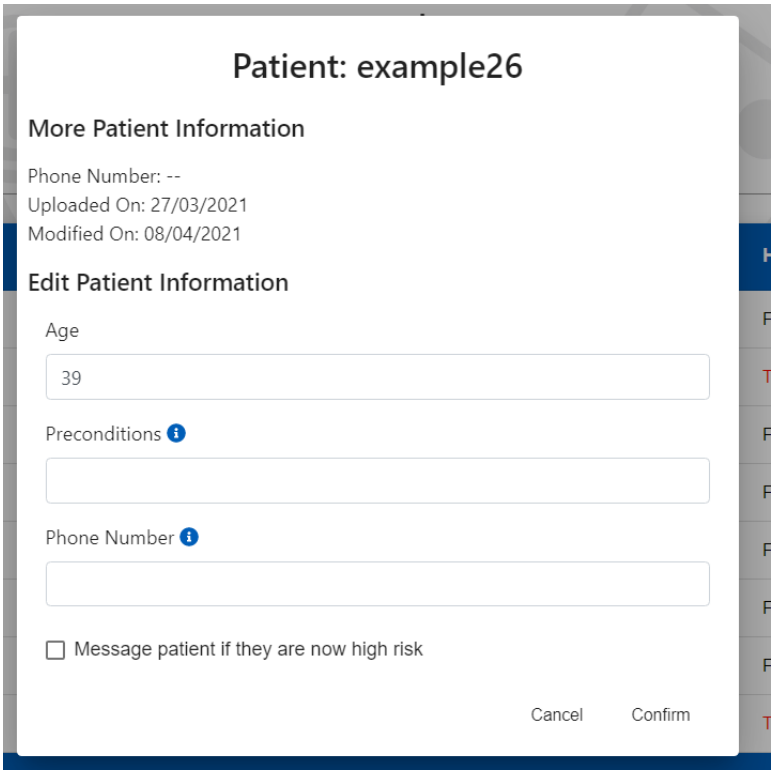


Figure 55: Update Patient dialogue box

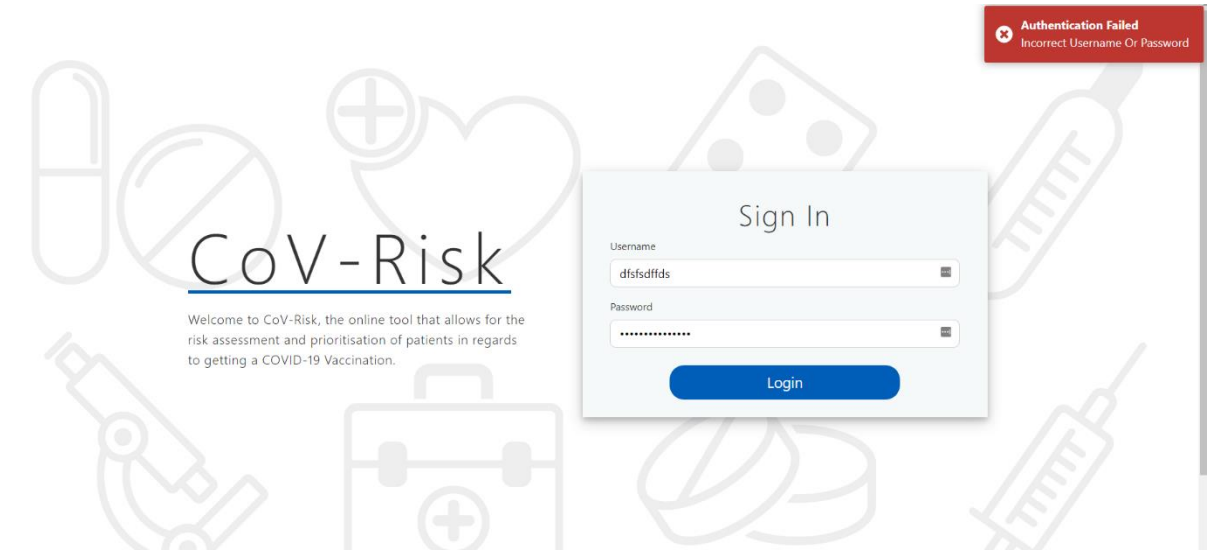


Figure 56: Example of the toaster pop-up with an error message

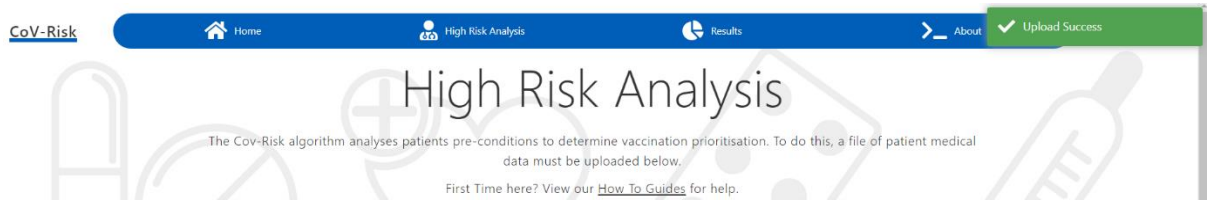


Figure 57: Example of the toaster pop-up with a success message

c) C# Development

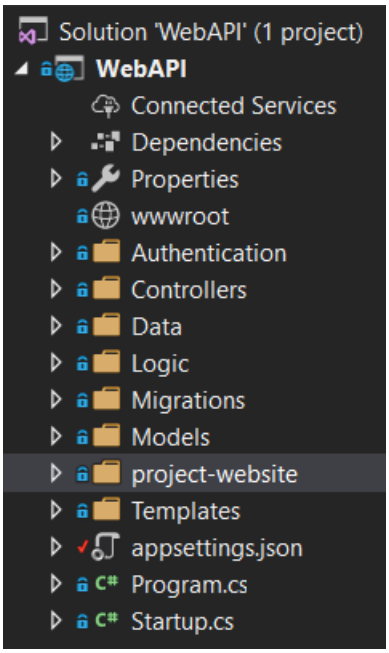


Figure 58: File structure of the full application - the angular code is kept in the highlighted folder, 'Project-website'

```
70      public List<Patient> CalculateRisk(List<Patient> patients)
71      {
72          foreach (var patient in patients)
73          {
74              patient.riskScore = 0;
75
76              //Calculate risk score of patients
77              foreach (var condition in patient.preconditions)
78              {
79
80                  switch (condition.ToUpper())
81                  {
82                      case "PNEUMONIA":
83                          patient.riskScore += 100;
84                          break;
85                      case "RENAL_CHRONIC":
86                          patient.riskScore += 75;
87                          break;
88                      case "DIABETES":
89                          patient.riskScore += 50;
90                          break;
91                      case "HYPERTENSION":
92                          patient.riskScore += 30;
93                          break;
94                      case "OBESITY":
95                          patient.riskScore += 30;
96                          break;
97                      case "IMMUNOSUPPRESSED":
98                          patient.riskScore += 25;
99                          break;
100                     case "COPD":
101                         patient.riskScore += 25;
102                         break;
103                     default:
104                         patient.riskScore += 0;
105                         break;
106                 }
107             }
108             if (patient.sex.ToUpper() == "M")
109             {
110                 patient.riskScore += 40;
111             }
112             if (patient.age >= 50)
113             {
114                 patient.riskScore += 40;
115             }
116
117             // Calculate if the paitent is at high risk
118             // High risk = risk score >= 50
119             if (patient.riskScore >= 50)
120             {
121                 patient.highRisk = true;
122                 patient.rowColour = "red";
123             }
```

Figure 59: High Risk Algorithm

```
public class Patient
{
    public Patient(string[] feilds)...

    public Patient() {}

    public string id { get; set; }
    public string sex { get; set; }
    public int age { get; set; }
    public List<string> preconditions = new List<string>();
    public string preconditonsStr { get; set; }
    public int riskScore { get; set; }
    public bool highRisk { get; set; }
    public DateTime uploaded { get; set; }
    public DateTime? modified { get; set; }
    public string rowColour { get; set; }
    public string phoneNumber { get; set; }
    public bool deletePatient { get; set; }
}
```

Figure 60: Patient Class

```
private static string ValidateAge(string patientAge, string patientID)
{
    var ErrorMessage = "";

    var isAgeNumeric = int.TryParse(patientAge, out int age);

    if ((!isAgeNumeric || age > 120 || age < 0))
    {
        ErrorMessage += ("The value in the Age column for patient " + patientID +
            " should be a number in the range 0-120" + "\n");
    }

    return ErrorMessage;
}
```

Figure 61: Age validation code

```
20      public string UploadPatients(List<Patient> patients)
21      {
22          string ErrorMessage = "";
23          try
24          {
25              List<Patients> patientsForDBAdd = new List<Patients>();
26              List<Patients> patientsforDBUpdate = new List<Patients>();
27              // convert each patient class to DB version 'Patients' and add to list
28              foreach (var patient in patients)
29              {
30                  // convert patient type to DB patient type
31                  var patientForDB = ConvertPatientForDB(patient);
32
33                  //check if patient already exists
34                  var exists = _context.Patients.Count(p => p.PatientId == patientForDB.PatientId) > 0 ? true : false;
35
36                  //if patient exists add that patient to the 'update' list, else add them to the 'add' list
37                  if (exists)
38                  {
39                      patientsforDBUpdate.Add(patientForDB);
40                  }
41                  else
42                  {
43                      patientForDB.Uploaded = patient.uploaded;
44                      patientsForDBAdd.Add(patientForDB);
45                  }
46              }
47
48              foreach (var patient in patientsForDBAdd)
49              {
50                  _context.Patients.Add(patient);
51              }
52
53              ErrorMessage += UpdatePatients(patientsforDBUpdate);
54
55              _context.SaveChanges();
56          }
57          catch (Exception ex)
58          {
59              ErrorMessage = ErrorMessage + "UPLOAD ERROR:\n" + ex;
60          }
61          return ErrorMessage;
62      }
```

Figure 62: EF Core Example - Uploading a patient to the database

	A	B	C	D	E
1	Id	Sex	Age	Preconditions	Phone Number
2	<Unique Patient ID>	<Sex Of Patient>	<Age Of Patient>	<Any Preconditions the Patient has. Must be written in the format - "Condition1; Condition2; Condition3" etc>	<Mobile number of patient written in the format - "country Code_phone number" where "_" represents a space. Note: If using excel an apostrophe ' must be added before the + symbol >
3					
4	example01	M	22		+44 07850823427
5	example02	F	56	COPD; OBESITY; HYPERTENSION	
6	example03	M	72	DIABETES	+44 12345678912

Figure 63: Template file showing how to format patient data

```
USE [Dissertation]
GO

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Patients](
    [PatientID] [nvarchar](255) NOT NULL,
    [Sex] [nvarchar](1) NOT NULL,
    [Age] [int] NOT NULL,
    [Preconditions] [nvarchar](255) NULL,
    [Uploaded] [datetime] NOT NULL,
    [Modified] [datetime] NULL,
    [RiskScore] [int] NOT NULL,
    [PhoneNumber] [nvarchar](25) NULL,
    [PreviousRiskScore] [int] NULL,
 CONSTRAINT [PK_Patients] PRIMARY KEY CLUSTERED
(
    [PatientID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

Figure 64: Script to create 'patients' table in the database

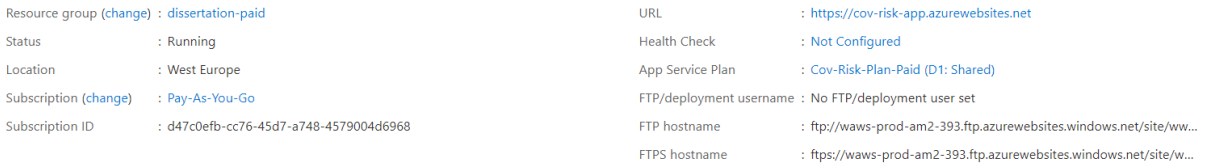


Figure 65: Azure App Service Configuration

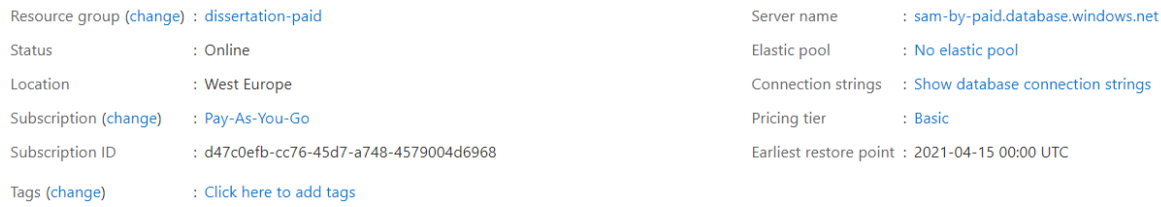


Figure 66: Azure database configuration

```
210     public string MessagePatients(List<string> phoneNumbers, TwilioConfiguration _twilioConfig)
211     {
212         var errorMessage = "";
213
214         try
215         {
216             string accountSID = _twilioConfig.AccountSID;
217             string authToken = _twilioConfig.AuthenticationToken;
218
219             // Initialize the TwilioClient.
220             TwilioClient.Init(accountSID, authToken);
221
222             foreach (var number in phoneNumbers)
223             {
224                 var message = MessageResource.Create(
225                     to: new PhoneNumber(number),
226                     from: new PhoneNumber("Cov19Risk"),
227                     body: "COVID-19 High Risk Alert\n\nYou have been marked as high risk for developing a severe form of COVID-19. " +
228                     "This means you are eligible for a COVID-19 vaccination.\n\nPlease contact your GP for more information" +
229                     "\n\nMessage sent from https://cov-risk-app.azurewebsites.net");
230             }
231         }
232         catch (TwilioException ex)
233         {
234             errorMessage += ex;
235         }
236
237         return errorMessage;
238
239     }
```

Figure 67: Code to send a text message to high-risk patients using the Twilio API

d) Mobile Development

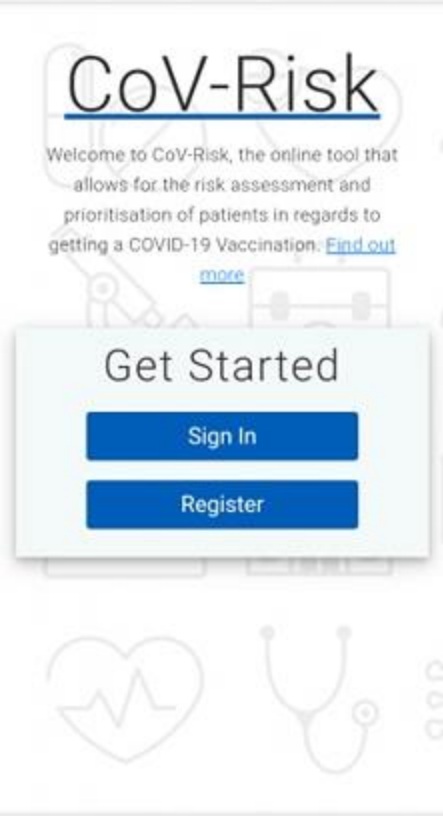


Figure 68: Mobile Welcome Page

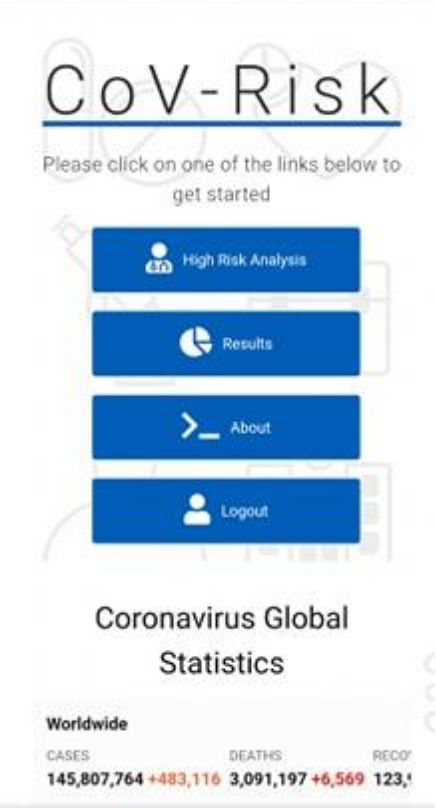


Figure 69: Mobile Home page

Figure 70: Mobile HRA page

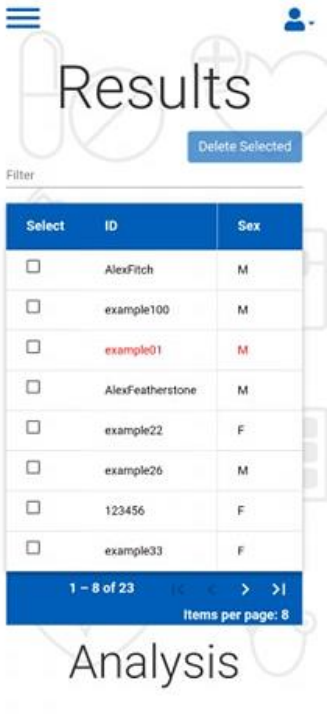


Figure 71: Mobile Results page - Top

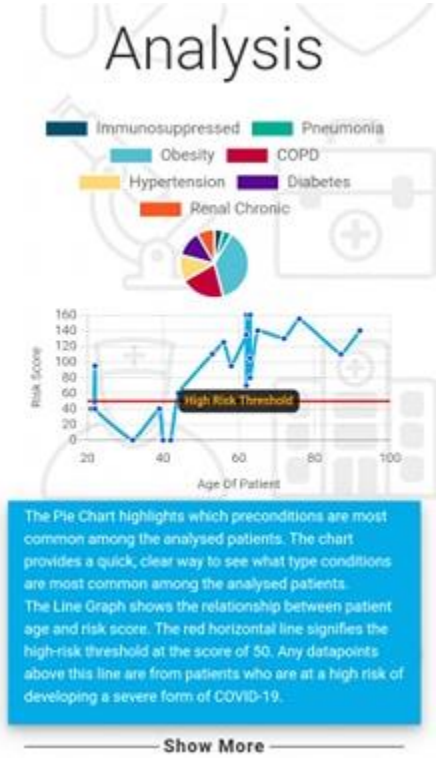


Figure 72: Mobile Results page – Bottom

Figure 73: Mobile About page – 1

Figure 74: Mobile About page – 2

4 -System Usability Scale Surveys

a) Mid development

'CoV-Risk' System Usability Scale - Mid Development Survey

Please go to 'https://cov19risk.azurewebsites.net' and familiarise yourself with the website. Explore the different pages and test the features of the website.

Now please answer the following questions based on your experience with the site.

User ID *

Please enter a short, anonymous, user ID that will be used to compare your results from this survey and the follow-up end of development survey. Do not include your name in the ID, feel free to use ID generators such as this one 'https://www.345tool.com/generator/random-id-generator'. Make sure to keep this ID safe (write it down, copy into a word document etc) so you can use it on the next survey.

Short-answer text

If you were a medical worker trying to distribute vaccinations, would you like to use this website frequently? *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

Please Explain Your Answer

Long-answer text

I found the website unnecessarily complex *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

Please Explain Your Answer

Long-answer text

I thought the website was easy to use *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

Please Explain Your Answer

Long-answer text

I think that I would need the support of a technical person to be able to use this website *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

Please Explain Your Answer

Long-answer text

I found the various functions in this website were well-integrated *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

Please Explain Your Answer

Long-answer text

I thought there was too much inconsistency in this website *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

Please Explain Your Answer

Long-answer text

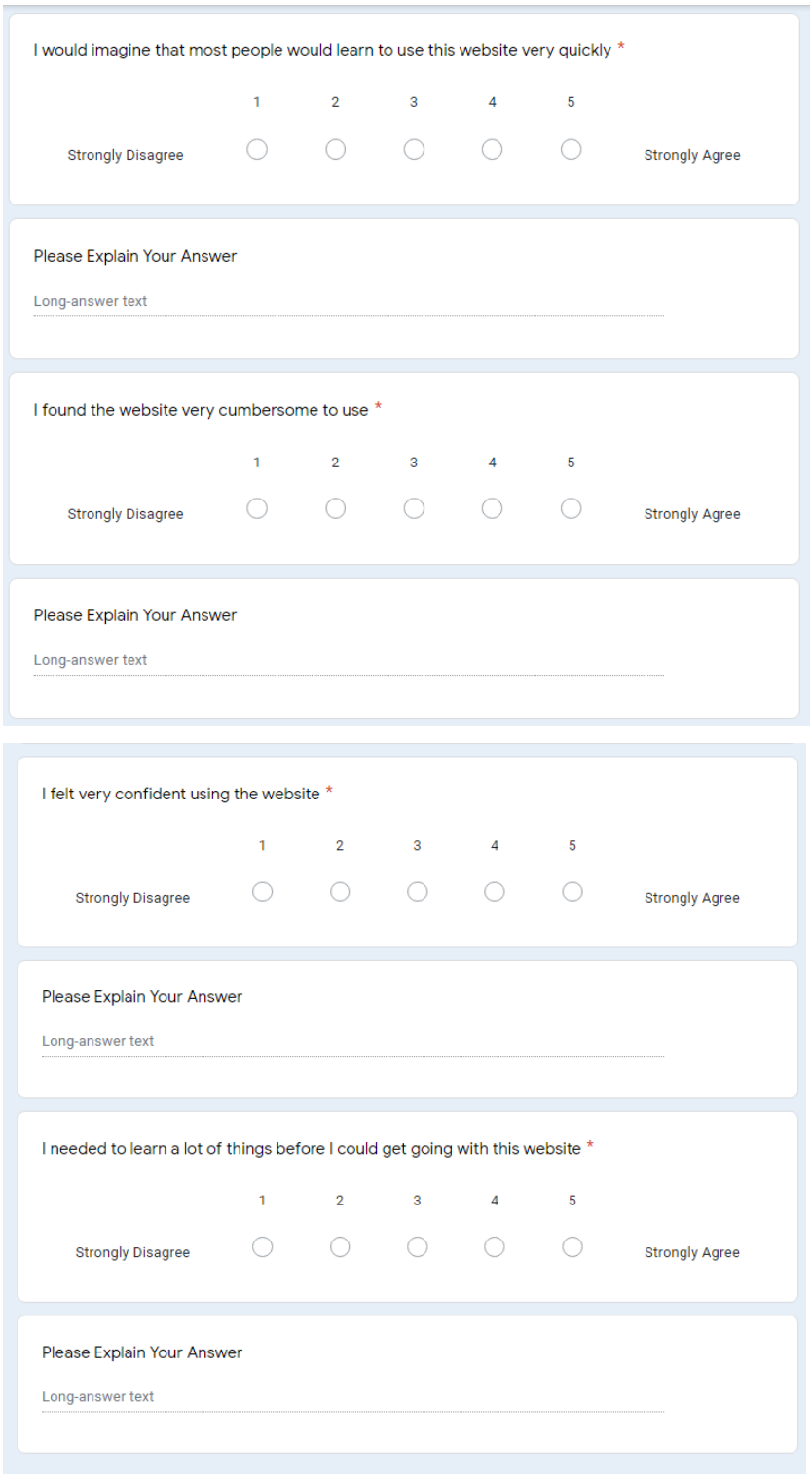
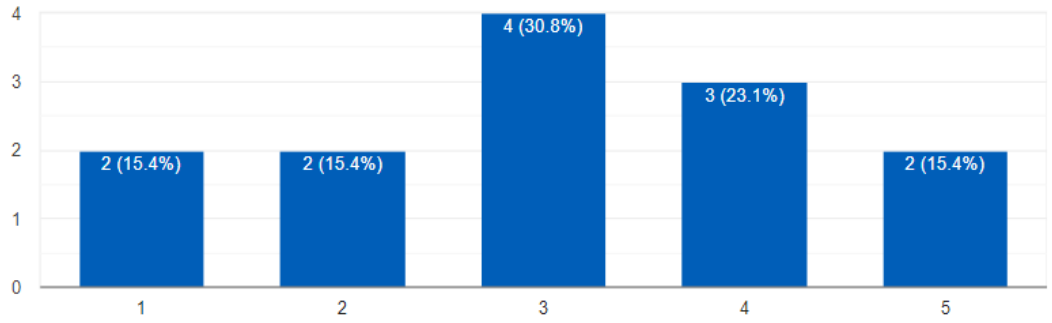


Figure 75: System Usability Scale - Survey Questions

If you were a medical worker trying to distribute vaccinations, would you like to use this website frequently?

13 responses



Please Explain Your Answer

11 responses

the website gives the data needed to be able to priorities patients

Couldn't upload data as data error occurred

The site gives limited information
Too many words particularly in the explanation of the random generator - I don't think this is necessary.
The 'About' link goes no-where
It doesn't inspire much confidence that I'll get what I need
Pre-existing conditions needs to have 'none 'column so you know they've answered the question/not missed it out.
I haven't given it a 1 because I think that the site has great potential; people would feel reassured that they were being treated fairly.
It would be good to actually be able to input our information to test the algorithm.
Who does the analysis once this figure has been generated?

Maybe but I'm not sure what use it would be to determine who need the vaccination when they already have everyone's medical records. I also don't think the death rate in Mexico influences who gets the vaccine.

Please Explain Your Answer

11 responses

Probably if I felt the data input and the algorithm worked to give the right result

Very clear and importantly straightforward to understand

The data it returns is very basic, there doesn't seem to be any way to sort, filter or download the results.
It would be helpful to have the high risk criterion on the results page, instead one has to refer to the HRA page to check.
There isn't a way to indicate who has already received the vaccination so there is a risk of inviting people more than once.
There are no contact details on the results page so one would need to cross-refer to another database in order to contact people.
I am slightly suspicious of the results as the Upload was incorrectly filled in so the website seems to have made an assumption about the sex of one of the patients.

It looks like it could be useful when assessing whether a patient is high-risk or not but unless this website became integrated into the already-used system, it may be too time-consuming in a time-sensitive situation.

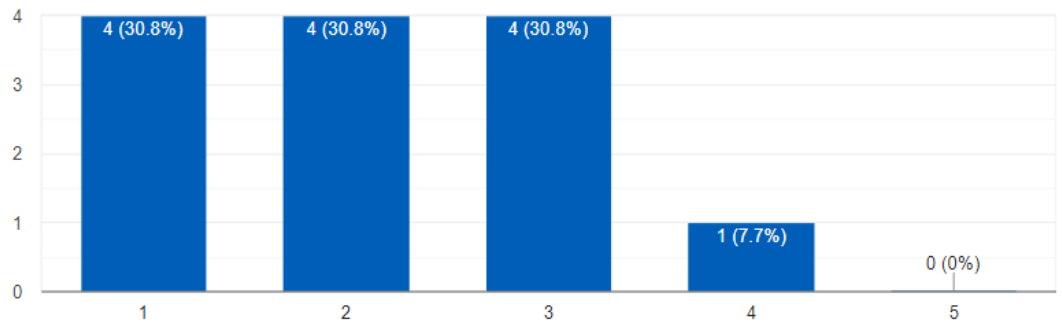
The website would be good at analysing data, not easy to use currently.

To meet the target set to vaccinate then the algorithm needs to run quickly to assess the priority of the patient. It would be very convenient if this worked on iphones and ipads which are used to aid the monitoring of inpatients and outpatients in hospitals.

Easy Succinct way to find results

I found the website unnecessarily complex

13 responses



Please Explain Your Answer

12 responses

website is simple to navigate and not complicated

Unable to use the website in the way it was intended

Too many opportunities for me to wonder if I've missed something. Instructions limited and unclear. It's simple enough to use.

Easy to navigate - only complicated part is what a .csv file is

Not many pages and simple - website itself works well - its the algorithm that's the complex thing but users don't need to know how this works, just that it does.

I certainly did not find the website complex - quite the contrary

It actually seemed very simple, but because of that the results were difficult to use efficiently.

The website is easy to navigate and the figures are explained well.

Not so much complex to use but confusing to understand.

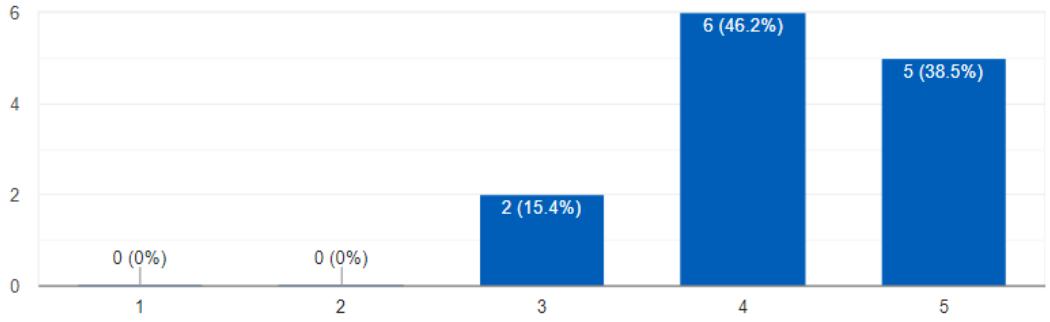
Things are clearly explained and easy to follow.

Very user friendly and self-explanatory

Not too many options and buttons. Mostly clearly labelled

I thought the website was easy to use

13 responses



Please Explain Your Answer

9 responses

Found it OK to find where to click to find information but doesn't seem to show much/ show it clearly. Then spent time checking to see if I'd used it correctly.

Easy to navigate except the "About" page not opening.

When all links are working smoothly should be fine - upload of .csv file not smooth. Couldn't get it to work on the Mac

It was easy to use because all that was required was to upload a file to get your results. However, the data it returned was not easy to use at all.
The data given on the Home page does not make sense and is incomplete (the +/- figures are not explained, so could be +/- on yesterday or last week or last year. Also, there are no +/- figs for the UK.

The website is quite small and well-labelled. However, the about section seems to be out of order as both links I found to it were broken/didn't work.

Functions work and laid out in an easy to use way

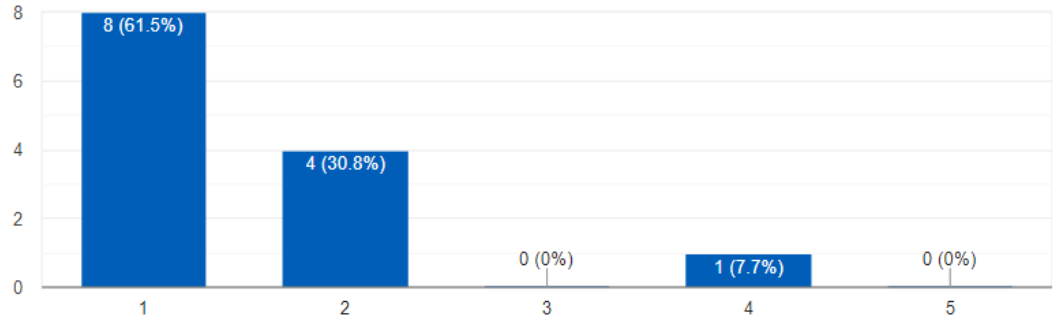
Everything was clearly labelled and instructions were easy to follow.

Easy to navigate, all laid out very simply

I managed to find what I needed

I think that I would need the support of a technical person to be able to use this website

13 responses



Please Explain Your Answer

10 responses

To enable file to be converted to csv file

It's easy enough to use just doesn't tell me very much.
The user just needs the bottom line figure and knowledge of how to access the information inputted for analysis.

Its a website not a nuclear reactor

Pages and functions are clear. Website itself very simple.

It was very easy to upload the file to get the results.

It seems easy to understand.

Probably would need help to understand what results meant.

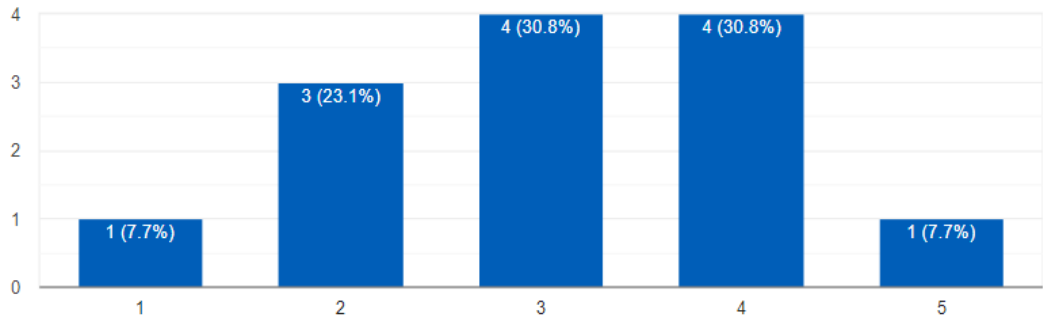
Easy instructions.

Easy for anyone to access

Personally I had no issues using the website

I found the various functions in this website were well-integrated

13 responses



Please Explain Your Answer

11 responses

the About button didn't work on either the home page or other menus

Unable to give answer as unable to use as intended

I wasn't sure what I needed to do and when I did move on to something different I wasn't sure why I was there.
The flow wasn't clear.

It all seems to work well except being able to see other people's results but that might be intentional

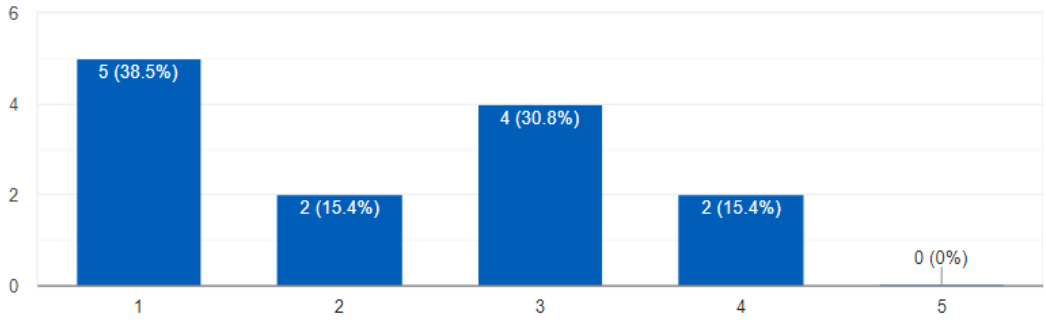
Not easy to assess this as yet to see how each page relates to another and how data input then gives the Risk number. Actually very few functions that the end - user need worry about once the Data has been put in and the site has assessed the Risk

When I reviewed the website the 'about' option was not working as designed and therefore I can't make a positive statement about the full integration.

No idea what the question means. Just taking a guess - the About tab doesn't work.

I thought there was too much inconsistency in this website

13 responses



Some of the pages seemed incomplete and need some information adding.

Some of the items did seem to load in correctly.

Please Explain Your Answer

9 responses

See above

Progression not clear from start to finish.
Little potential for inconsistency as website very limited.

Seems consistent

Sequence to follow from Data input to Risk assessment pages not yet obvious - as a non user. Once the algorithm deals with the data that has been put in should be very clear. No website inconsistency obvious - so long as the algorithm doesn't have inconsistencies.

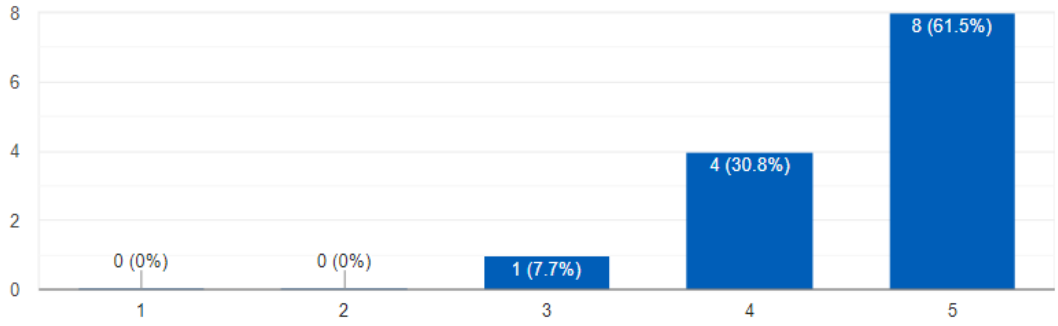
Refer previous comment in relation to the 'about' function not working as designed

Didn't see anything inconsistent on the website. However, the file I uploaded from the email contained an error that the website appears to have corrected by itself (converting 'bob' to 'F') which may have resulted in an incorrect return.

Unsure what the question means? I've seen no inconsistency on the website whatsoever.

I would imagine that most people would learn to use this website very quickly

13 responses



Please Explain Your Answer

9 responses

If the upload worked then I think this would be a simple website to use

Straight forward - limited number of pages so little potential for problems. Not much to learn, just not very clear what needs doing.

Despite knowing what a .csv file is

Page titles / purpose clear

There is only one operation required.

It's fairly self-explanatory.

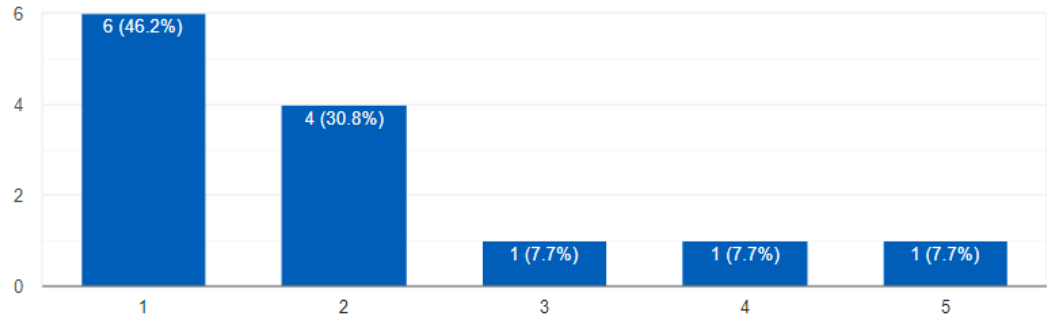
I think with some explanation to use the website it would be quick to learn how to use.

I think it would be easy to use, but some might need some technical support downloading and using the template.

I think most people would be able to work the website out with no problem

I found the website very cumbersome to use

13 responses



Please Explain Your Answer

8 responses

I want something that outlines the purpose, gives me some initial information - stats in charts - then is straight forward in inputting information. I wouldn't have much confidence that I'd done the right thing and got the outcome I needed. Easy enough to use.

About page didn't work, don't know what a .csv file is and unsure about but there is an available template to overcome this.

Seems very staightforward as there are not too many pages/ links

There is only one operation required.

Not at all.

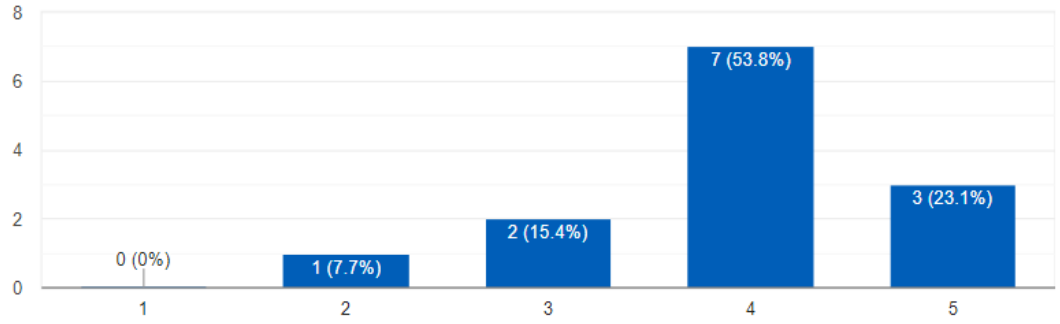
Easy to use and upload documents.

Seems easy enough to use.

Overall intuitive and straightforward

I felt very confident using the website

13 responses



Please Explain Your Answer

8 responses

Simple website

I could follow the instructions OK, I just felt that I might be missing something as I wasn't sure what it was telling me/ what the purpose of it was so then doubted myself.

I feel very confident using the website.

Not yet fully confident - would need to mock up data entry page and see how the results then showed risk for a particular patient and compare results for different patients to assess reliability.

Yes - once the 'about' function has been addressed

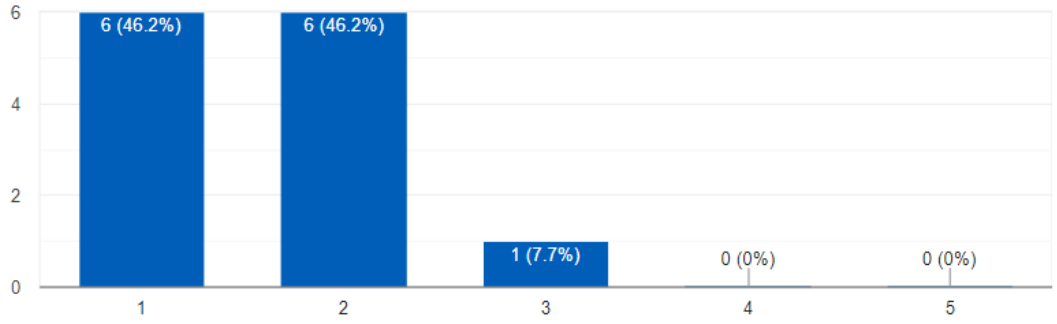
It is easy to upload a file. However, it is not easy to use the data supplied by the website.

Once I played around a little, I felt confident navigating the site.

I think it will be easy to use once the website is finished and has all the components added.

I needed to learn a lot of things before I could get going with this website

13 responses



Please Explain Your Answer

9 responses

Not much to it; if it's just going to help medics to prioritise they just need the site to generate a score - as a medic that's all I need to know so nothing to learn.

To engage with it thoroughly I might do.

Very few pages so everything is clear - assuming the end user would input the data and then be able to assess the risk for each particular patient. Data input does not seem too heavy so should be easy to use, assuming the algorithm does the work to generate the number showing Risk. Website itself is simple to use

The website is very intuitive - however, it would be useful if the download options displayed that the computer was working on the download in the event the computer performance is slow.

I already know how to upload a file.

I think basic computer literacy is all that's needed.

I think it would be easy once learnt, but I feel some would struggle to use the website if they are not

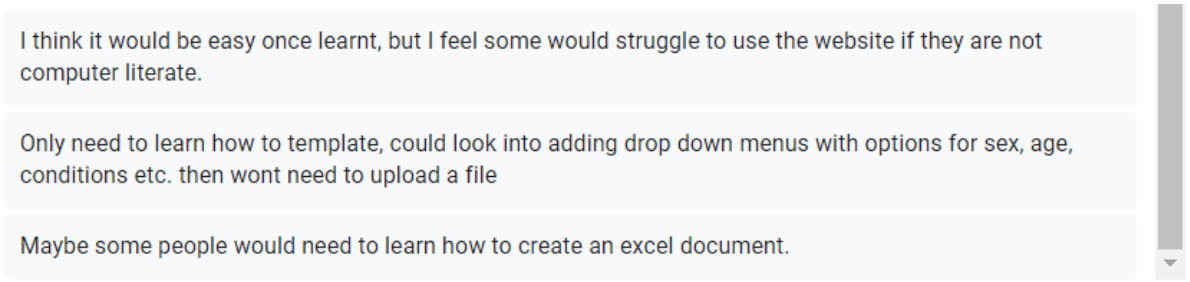
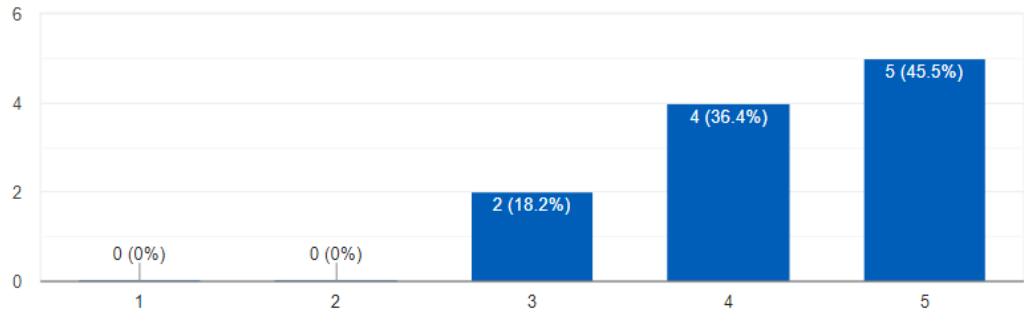


Figure 76: Mid Development System Usability Scale – Survey Results

b) Post-development

If you were a medical worker trying to distribute vaccinations, would you like to use this website frequently?

11 responses



Please Explain Your Answer

9 responses

I think it would be a useful tool

Having a page for results which everyone who uses the website can see the results of might be an issue for patient confidentiality.

Gives you an idea as to potentially high risk patients

Easily accessible information clearly presented

Invaluable in prioritising vaccines - allows the professional to be objective and speeds up the decision making process. Justification can easily be provided if required.
The information is set out simply and is clear.

When you've done it once it is quite easy to do, but the results page is static and doesn't allow sorting or filtering to identify those at highest risk.

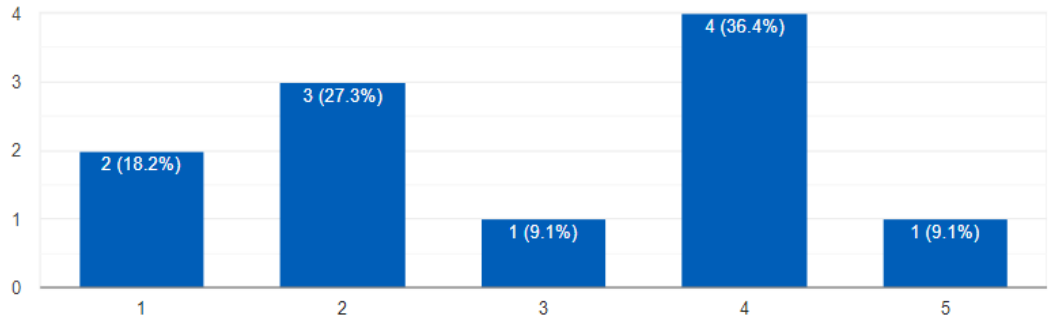
It seems to be an easy, useful tool to quickly analyse patient files and organising them by priority.

good easy to use system

Website helps identify who is highest priority for vaccine

I found the website unnecessarily complex

11 responses



Please Explain Your Answer

10 responses

Was quite simple

The how to section was very long

Not as intuitive as I would like

Once used the operations and entry/upload methods were straightforward

Essential to follow the instructions but once you've input the data once, this will be much easier. The amount of data required in small so inputting it is straightforward. It was easy to navigate around the site and find the information I needed.

Too much explanation, the user needs to know the outcome or results but not loads of info about how the results are calculated.

Vital info is missing from the instructions, e.g. deleting lines 2-6 before uploading the file, so everyone's upload would fail first time unless they had never uploaded a file before and went to the step-by-step instructions.

The website is straight-forward to use.

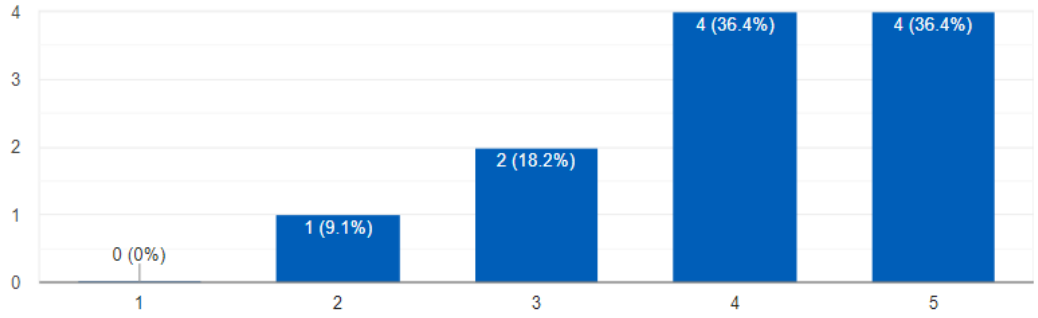
Simple format easy to use template

It was easy to navigate on pc but wasn't able to use on mobile device

guidance features were clear

I thought the website was easy to use

11 responses



Please Explain Your Answer

9 responses

I struggled a bit at first, but the instructions and how to guides are helpful and after figuring it out is easy enough to use.

The process was a bit difficult when trying to complete the information on numbers whilst having to swap pages to read the instructions on the site. Once you've completed the process a few times it wouldn't be a problem

It looks nice however even following the instruction I had no idea what was going on

Yes - once fathomed necessity for CSV file format

Easy enough after familiarisation

Once you've navigated your way around the site, it's easy to use. Initially there seemed to be a lot of information that seemed overwhelming but soon became clear with a bit of practice. I had trouble inputting my phone number as it kept trying to convert it to a formulae. I might have read the instructions wrong but was frustrating.

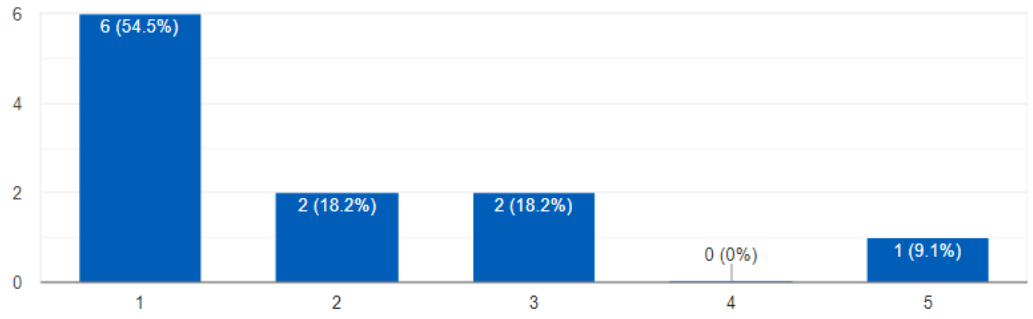
Uploading a file, looking at the results and deleting where appropriate is very easy (once you've done it once and failed!)

Things were labelled clearly and each step had instructions available if they were needed.

nice bold and clear

I think that I would need the support of a technical person to be able to use this website

11 responses



Please Explain Your Answer

8 responses

At first I think I might need help but again the instructions clear this up.

Once I'd realised I had to save the template into numbers and then save it as a csv file it was quite simple

Not necessary, once I re-read the instructions!

Only needs brief training but instructions could perhaps be presented in step by step bullet-point form

Initial training in how to use it would be useful but not essential. It could save time and prevent people getting frustrated and help to alleviate concerns and answer questions. It would be a shame to put people off for the sake of a few minutes training.

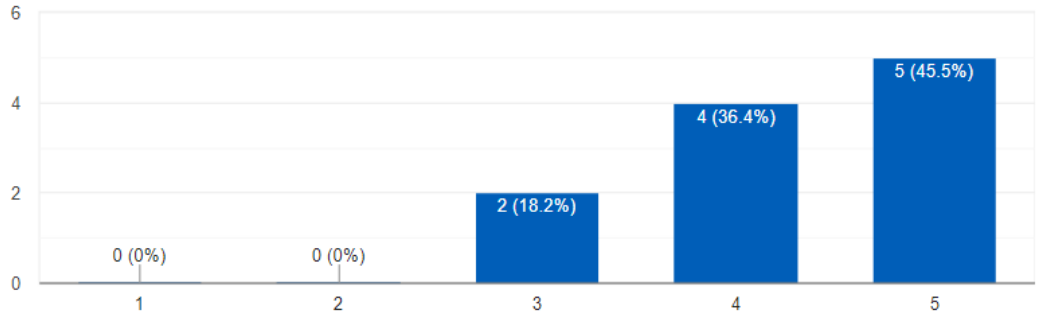
It was very easy to use.

It is a simple design that I think anyone would be able to use.

help guides very useful for this

I found the various functions in this website were well-integrated

11 responses



Please Explain Your Answer

7 responses

All seemed to flow well

The website works well but if you could input data straight into the website then I think it would be more efficient

Seems to link well together

The functions were simply presented and seemed to follow on from each other well. It was easy to see which tab you needed if you wanted to move functions.

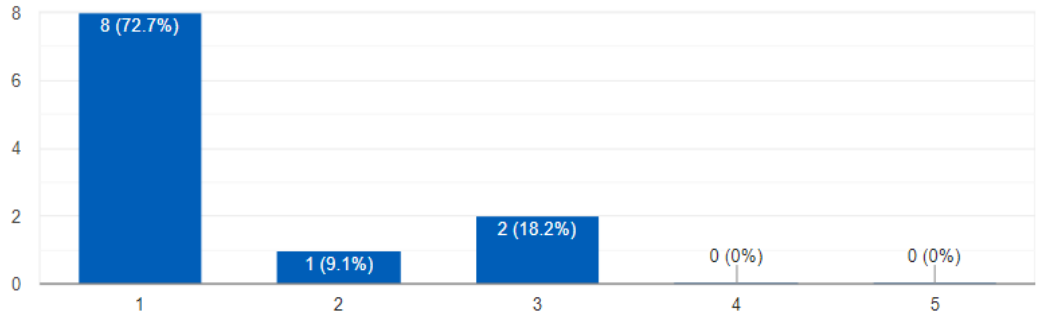
Don't know what that means...

Extra Information being hidden away in drop down segments is a useful way of giving information without cluttering the screen. Each section of the website continued the same visuals which helped to create a flow throughout the website.

all flowed well

I thought there was too much inconsistency in this website

11 responses



Please Explain Your Answer

7 responses

All looked consistent

Couldn't see any - all functions/results seem to respond consistently

No inconsistency evident -planning and organisation seems to be well thought out.

Didn't see any inconsistency

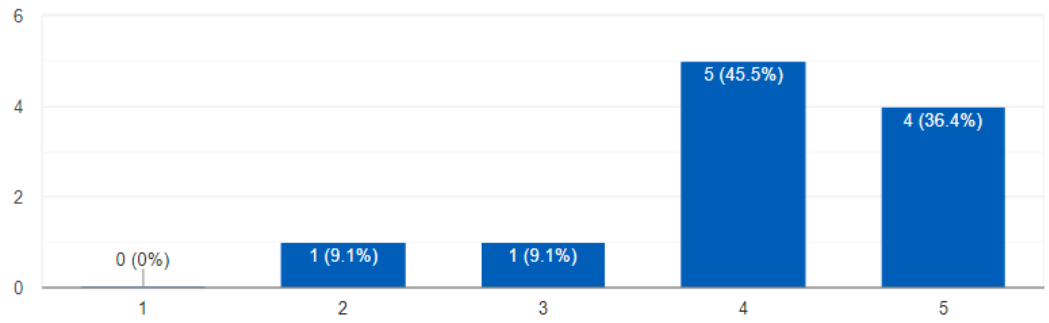
I wouldn't specifically call it inconsistency but spelling of certain words were sometimes correct and sometimes not so correct. I could not spot any inconsistencies in facts or content though.

Everything linked easily

I felt it was clear the website had been developed by the same person and did not notice any significant inconsistencies

I would imagine that most people would learn to use this website very quickly

11 responses



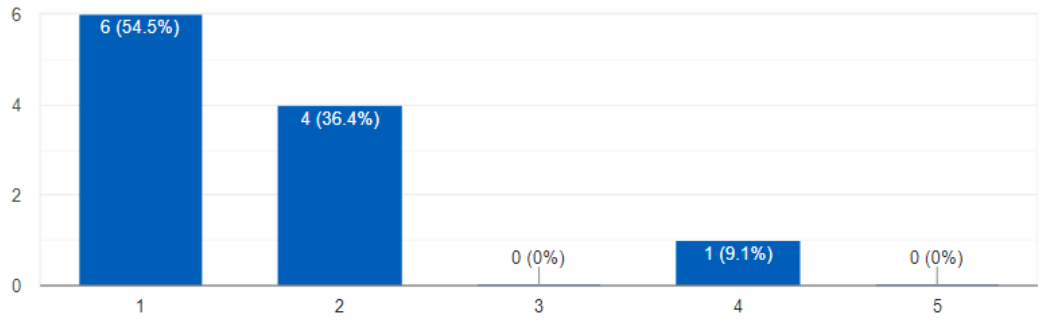
Please Explain Your Answer

9 responses

- It seemed quite simple to use when you've collated all the information required
- I think it would take a while to learn what is required of the template
- As per previous answer - not as intuitive as I would like.
- Professionals would learn to use it quickly and with constant use it should be very quick
- Once you've used it once, it's straightforward. Analysis of the results was a useful feature.
- It is very easy to use
- It's very simple and comes with explanations.
- very informative and easy to pick up
- I would think by the second/third go round use of the website would be second nature

I found the website very cumbersome to use

11 responses



Please Explain Your Answer

6 responses

It was simple to use

Simple to use - the algorithms actually do all the work

It seemed very straightforward - once you've worked out how to input the data the algorithm does the work for you.

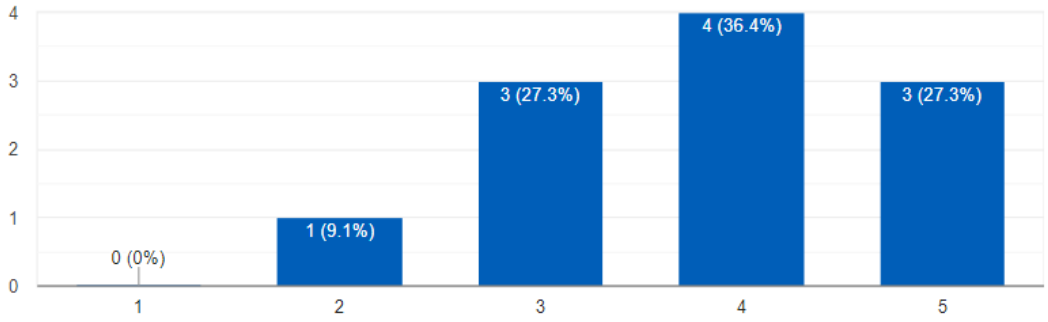
I found it cumbersome to use the first time as vital information wasn't provided in an obvious place. Also, there is no sort or filter function so you have to look through the results manually to identify the patients at highest risk.

It has the right amount of different sections to stop the user becoming overwhelmed with the amount of information on any one page.

results quickly obtained

I felt very confident using the website

11 responses



Please Explain Your Answer

7 responses

Simple to use

After a while

Once familiar with the instructions it was easy to enter data and easy to read and analyse

Initially I found it difficult to input the data -maybe short bullet point instructions might be useful. Having done it once, it was very easy and I felt confident I was doing it properly.

Once I'd deleted rows 2-6 it was very easy to use.

Directions were clearly labelled with extra information if help is needed.

Not at first - perhaps include some confirmation notices along the way

Figure 77: Post Development System Usability Scale - Survey Results