University of Dublin

TRINITY COLLEGE

AVERT clinical user interface for patient app data

Angelika Palka
B.A.(Mod.) Computer Science
Final Year Project April 2019
Supervisor: Lucy Hederman

School of Computer Science and Statistics

O'Reilly Institute, Trinity College, Dublin 2, Ireland

Declaration

I hereby declare that this project is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university.

Name: ___Angelika Palka___ **Date:** ___18/04/2019____

Acknowledgements

Firstly and foremost, I would like to express my sincere gratitude and appreciation to my project supervisor, Prof. Lucy Hederman. For the last couple of months, she has provided me with a lot of assistance, guidance and positive attitude.

I would also like to thank all the members of the AVERT group who have helped me and provided useful feedback during the implementation process of this project. Special thanks to Alan Meehan who has helped me and offered his assistance with the technical difficulties I experienced during the implementation process.

I am also thankful to the patientMpower group, especially Martin Holmes and Michal Dabski, with whom I have exchanged numerous emails and who have helped me gain a better understanding of how to work with the patientMpower API.

Abstract

Continuous technological developments in health care have saved countless lives and improved the quality of life for even more. However, even to this day, there are still several conditions for which technology has not yet found a cure, one of those being vasculitis. It is unknown to the doctors what are the exact causes of this condition, one of the potential causes being environmental factors such as pollution, weather, and infections. The AVERT group from Trinity College Dublin have taken up research to determine what environmental factors could lead to a relapse of a specific autoimmune disease called ANCA vasculitis. Their cooperation with patientMpower allowed them to collect patient data and analyze it alongside environmental data.

The AVERT group aims to create a clinical support tool and an intelligent phone application, that will empower patients in managing their own condition. To help patients manage their own condition in the future, the research nurse (who is part of the AVERT team) takes it upon themself to encourage patients to use the mobile application on a daily basis. This will provide more data for researchers to analyze and eventually, the use of the mobile application will become a daily routine for many patients. This project provides the means for the nurse to keep track of all the patients and monitor their engagement with the mobile application, along with their health. The notification system that I have developed allows the research nurse to see which patients need immediate attention and for what reason. Visualizing the patient's data via graphs and charts puts the data in context and thus allows the nurse to recognize and analyze the patterns that may lead to a relapse of the patient's condition.

Table of Contents

List of Figures

List of Scripts

Chapter 1: Introduction

1.1 Motivation

The AVERT project group, with cooperation from patientMpower (pMp), use a mobile application to gather data from patients registered with different health conditions, such as vasculitis, a condition that involves inflammation in the blood vessels [1]. This allows the AVERT researchers to monitor changes and establish patterns that may lead to a flare or relapse of vasculitis. Since there are over 200 patients currently taking part in the AVERT Vasculitis study, researchers need a way to monitor changes of individual patients over time, keep track of their engagement with the mobile application and be notified of patients who have relapsed.

The current approach of monitoring individual patients is challenging, as the pMp user interface provides unprocessed patient's data in a tabular form. At the moment, the research nurse has to select each patient in order to be able to see whether they have been using the patientMpower application, if their health is improving and if there are any worrying signs that might lead to a flare or a relapse. Considering the number of patients that are taking part in the AVERT study, keeping track and analyzing each patient's data in that way becomes a timely process, and with the number of patients growing each year, it would simply become ineffective with time.

By visualizing the patient data and introducing a notification system, the research nurse would be able to see straight away which patients need immediate attention. Thus allowing a more efficient work schedule, better analysis of the data provided and minimising a human error of missing a patient who needs attention. Visualization of data and an intuitive user interface would ensure that any user, not necessarily a technical one, is able to perform data analysis and data pattern recognition.

Since the data from the patientMpower mobile application relates to patients' personal information, data privacy and protection is an important part of this project. Patient's location is a key challenge because if the latitude and longitude coordinates are captured accurately, this would allow other people to get access to the patient's home

address and potentially other private information, thus breaching users' privacy. However, for the AVERT project, patient's location plays an important role. It allows researchers to link the patient to environmental factors that they have been exposed to and analyze whether any of those factors may lead to a relapse of vasculitis.

This project report details the process of building a user interface that captures the engagement of vasculitis patients with the patientMpower mobile application and also allows the researchers to efficiently and effectively track any changes in individual's health and engagement patterns. Further, I will discuss any problems and issues that arise during this process.

1.2 Objective

Patients who participate in the AVERT project use the patientMpower app, to record health and location data. The research nurse needs to be able to easily and effectively review the data in the app developer's database. The aim of this project is to design an appropriate, practical and intuitive staff user interface for patient app data.
The main objectives are:

- **Objective 1:** Find an appropriate web framework that will provide a secure environment for handling real-time patient data.
- **Objective 2:** Gather requirements from the client (the research nurse).
- **Objective 3:** Design an intuitive and easy to use user interface, for patientMpower database, based on the client's requirements.
- **Objective 4:** Query patientMpower database to retrieve useful patient information and present it in a neat and practical way.
- **Objective 5:** Design a notification system that allows researchers to easily identify patients who need their immediate attention.

1.3 Outline

This project is organised as follows:

Chapter 2: Background - Introduces the tools used for the development of this project, based on the requirements gathered from the client. It also shows the context of the project and provides background knowledge, essential for a better understanding of the research and work carried out.

Chapter 3: Design - Describes the process of gathering requirements and designing an appropriate user interface. It also focuses on designing an environment suitable and secure for handling real-time patient data.

Chapter 4: Implementation - Describes in detail, the implementation process and the techniques used to obtain the desired functionality of the user interface, regarding both the security and processing of data. It also explains the process of accessing and using the pMp API, and the problems that arose during that process.

Chapter 5: Evaluation and Discussion - Evaluates the work that has been done based on feedback provided by the client. It also discusses challenges, captures the room for improvement and provides ideas for future work.

Chapter 6: Conclusion - Concludes the work that has been done for this project, in relation to the objectives set out.

Chapter 2: Background

This chapter provides background knowledge on a number of topics that are related to this project. Main topics include an explanation of what an autoimmune disease is and why that is important in the context of the AVERT project. This chapter also introduces the concept of EQ-5D and what it means to be engaged with a mobile application. The final sections of this chapter focus on providing a technical explanation of the tools used.

2.1 Autoimmune disease

Inflammation is our body's natural response to an injury or infection. It is a way of signalling to the immune system to heal and repair damaged cells and defend itself from foreign invaders. The process begins when bacteria enter our body. As a result, small blood vessels become dilated in order to increase the flow of blood. There is an increase in vascular permeability to protein and the movement of fluids to tissue results in swelling. At this point, white blood cells move from blood vessels to tissue, where they destroy foreign invaders and begin the healing process [2].

In some diseases, inflammation can be triggered even if there are no foreign invaders attacking our body. Those diseases are referred to as autoimmune diseases. The normally protective immune system attacks and damages its own tissue. Doctors don't know what causes the immune system to mistake healthy cells as foreign invaders. Yet there are some people who are more likely to get an autoimmune disease than others. It has been found that women are more likely to get an autoimmune disease than men (6.4% vs 2.7%), however, the reasons for that are still unknown [3].

Due to a rising number of incidents of autoimmune disease, researchers suspect a number of factors that could play a major role in triggering the disease, one of those being environmental factors, such as infections, weather, and pollution. Another suspected trigger is diet. People in the Western world have a tendency to eat food that is highly processed and high in fat and sugar. Those types of food are related to inflammation. Hygiene is also a possible trigger of the autoimmune disease. Children nowadays aren't exposed to as many germs as in the past, due to vaccines and

antiseptics, which could possibly lead to immune system overreacting to harmless substances [3].

2.1.1 Vasculitis

Vasculitis is one of the autoimmune diseases that affects a small number of the world's population. In vasculitis, the immune system attacks healthy blood vessels. When blood vessels become inflamed, they become narrow and as a result, make it difficult for blood to get through. The blood vessels can also stretch and weaken to the point that they bulge. This can have a very serious effect, as when the blood vessel bursts, it can cause dangerous bleeding inside the body. Vasculitis can range from a minor problem that just affects the skin, to a more serious illness that causes problems with organs such as heart or kidneys [1].

In most cases, the cause of vasculitis is unknown, however, medication and infections are a possible trigger. It may also be a complication from other immune system diseases such as lupus or rheumatoid arthritis [3]. For most types of vasculitis, treatment can cause the disease to go into remission, which means that it is no longer active and isn't causing any further damage to the organs. However, in most cases, the illness can return or flare.

A relapse or flare is the main challenge in managing vasculitis. Continuous monitoring and communication with a patient can play a critical role in detecting a relapse and minimising the effects it has on a patient [1].

2.2 AVERT Project

The lack of knowledge about triggers associated with autoimmune diseases, in particular ANCA vasculitis, is what motivated the AVERT Project group from Trinity College Dublin to take up research on that particular disease. The research brings together scientists from the disciplines of medicine, computer science, and statistics [6], to determine the triggers associated with a flare of ANCA vasculitis, predict the patient risk and potentially devise ways in which those triggers can be avoided, and tailor treatment to the current risk. This research is of great importance for many patients in

Ireland and worldwide, as it would allow them to have greater control over the disease and improve their overall quality of life.

As mentioned earlier, the triggers associated with a particular autoimmune disease are typically unknown and as a result, relapses of the disease are very likely to occur. The condition can flare repeatedly over time, causing great pain, discomfort and damage to organs of the patient. In ANCA vasculitis, half of the patients with that condition tend to have a flare within 5 years [6] from when it went into remission.

It is believed, but not yet proven, that environmental factors such as pollution, weather, and pathogens play a major role in triggering the autoimmune disease. This is one of the questions that is being addressed in the AVERT research. Their focus is to develop a model that will learn over time about the impact that the environmental factors have on the disease flare. However, the researchers are faced with two problems. First, how to integrate the environmental data with the data collected from the patients? Second, how to encourage patients to continuously record their data? In particular, data about patient's (ideally hour by hour) location is needed to determine the environmental factors that they are exposed to.

This project aims to address the second problem, by allowing a clear overview of patient engagement with a mobile application that allows them to record their data and providing a notification system of the patients who do not use the mobile application on regular basis. It is hoped that with someone monitoring patients on a daily basis and encouraging them to use the mobile application will lead to more patient data collected over time - allowing researchers to achieve their goal. The more data is collected from the patients now, the more accurate real-life analysis of current and future patients will be.

The end goal of the AVERT Project is to develop a clinical support model and mobile application that will allow patients to manage their own condition [6].

2.3 Engagement

Engagement can be defined as “being involved with something” or “the process of encouraging people to be interested in the work of an organisation” [7]. In relation to the research of the AVERT group, engagement between a patient and the pMp mobile application is crucial. Without the help and involvement of patients, this important research would be impossible, due to the lack of patient data. The more patients are willing to record their data, especially location data, and engage with the mobile application, the more accurate the research will potentially be. Having a number of patients continuously recording their data, especially during the period that leads to the flare of their disease, would allow the researchers to perform accurate analysis and find patterns that could lead to determining the triggers that caused the flare.

However, in order to encourage patients to engage more with the mobile application, the person responsible for monitoring them has to show high levels of engagement with this project. By keeping a close eye on patients each day, it would be quite easy to see the engagement levels of patients dropping or increasing, and from that determining whether it has something to do with their health, busy personal life or simply getting bored of using the mobile app.

2.4 EQ5D

In clinical practice nowadays, there is a greater emphasis on using alternative measures to detect and estimate the quality of life of a patient. The use of those measures reflects on an appreciation of the importance of how a patient feels and if they are satisfied with their treatment. By using quality of life measures, physicians are able to capture patient’s perspective on their disease, treatment and their perceived need for health care. It also allows them to get a better understanding of how a disease affects a patient’s life and as a result, provide the patient with a more comprehensive health care [15].

One of the means of measuring the quality of life is with an EQ-5D survey. EQ-5D is an instrument that was developed by a EuroQol Group in the 1980s, to record and measure patient’s health-related quality of life. This instrument can be used in a wide range of diseases and their treatments [4][5] and is available in a wide range of languages and

modes of administration. It was designed for self-completion and therefore it captures the information directly from the patients.

The EQ-5D survey consists of two components, a description of the state of health and evaluation. The first part of the survey consists of five dimensions that measure health status. The five dimensions include: mobility (person's walking ability), self-care (ability to wash or dress yourself), usual activities (work, housework, etc.), pain/discomfort (the level of pain or discomfort), and anxiety/depression (the level of feeling anxious or depressed) [4]. The second part of the survey consists of evaluating your overall health status by using a visual analogue scale (EQ-VAS), which can be used to capture a health status that reflects person's own judgement [5].

Currently, there are three versions of the EQ-5D: EQ-5D-3L, EQ-5D-5L, and EQ-5D-Y.

EQ-5D-3L was introduced in 1990. It's essentially an EQ-5D, where the scale that is used in the health state descriptor consists of three levels: having no problem (1), having some or moderate problems (2), and having extreme problems (3). The patient is asked to select a statement in each dimension that best describes their health status, and the number corresponding to the level selected at each dimension is then combined to form a five digit value that defines patient's health state. There are potentially 243 different health states for this version of EQ-5D [4].

EQ-5D-5L was introduced in 2009 to improve the ceiling effects that resulted from the EQ-5D-3L survey, which was not able to capture small differences of patients in moderate condition. EQ-5D-5L has introduced two extra levels of severity in each dimension. The five levels consist of: no problems (1), slight problems (2), moderate problems (3), severe problems (4), and extreme problems (5). After the patient selects the statement, at each dimension, that best suits their health state, the number corresponding to the selected level is then combined into a five digit value. This new version of EQ-5D can now define up to 3125 different health states [4]. The increase in the number of possible health states also increases the performance of the survey, as it is now able to capture the patient's health state more accurately than before.

EQ-5D-Y was introduced in 2009. It is a 'youth version' of the EQ-5D-3L survey which was designed for self-completion by children and adolescence. It operates on the same basis as the EQ-5D-3L survey (with three levels of severity at each dimension), with the only difference being in the phrases that are used. Those phrases are easier to understand, more clear and relevant to children and younger people. The EQ-5D-Y survey consists of the following five dimensions: mobility, looking after myself, doing usual activities, having pain or discomfort, and feeling worried, sad or unhappy [5].

The AVERT group use the EQ-5D-5L survey due to the increased accuracy in determining the patient's health state and its simplicity. They collect the information from patients via the pMp mobile application and use it in their vasculitis research. Since it is hard to determine, from a five digit value, the effects that each dimension has on patient's health status, this project uses an algorithm (UK version) to convert the five digit value to a decimal value that ranges between -1 and 1. Values such as "0.783" are much more clear than values like "12212", and they allow to capture the increase or decrease in patient's health much more easily, thus allowing the researcher more accurate analysis. The algorithm that allows the computation of the decimal value is different in each country, as it tries to encapsulate the relative importance of different QoL factors on perceived quality of life of people in a given country.

2.5 Web development framework

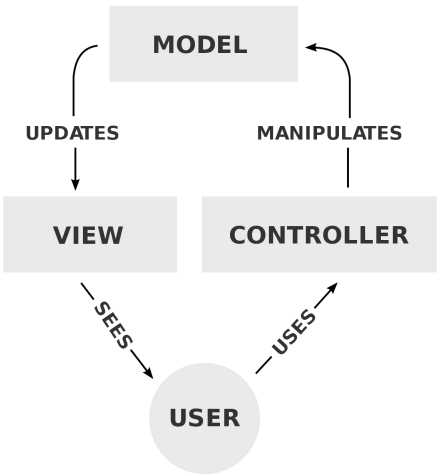
A web development framework is a software tool that is designed for the development and management of web applications, web services, and websites. It has become a vital part of development processes, as it aims to automate common activities performed during web development. Most web frameworks target the development of dynamic websites, however, they also support the development of static websites. Web development frameworks can be divided into two categories, server-side (backend) and client-side (frontend) [16].

Server-side framework deals with the functionality of the website. This type of framework allows the developer to implement simple pages and different types of forms, but also form output data and improve the website's security in case of an attack [16]. All of these built-in features help to speed up web development process.

Server-side frameworks focus on particular and important aspects which account for the proper functionality of the website.

Unlike server-side, client-side frameworks focus mostly on the visual part of the website and have nothing to do with business logic. Their work takes place inside the web browser, where the developer can implement or improve a user interface. Client-side frameworks support the development of animated features, however, their functionality can differ from one framework to another [16].

The architecture of almost all web frameworks is based on the decomposition of several layers, which allows the developer to extend its functionality based on the requirements. A lot of web frameworks follow the Model-View-Controller (MVC) architecture [16],



however, some frameworks alter the functionality of this architecture.

Figure 2.1: Structure of the Model-View-Controller architecture [11]

Figure 2.1 shows the basic structure of MVC architecture. Model contains all data and business logic to manipulate and retrieve that data. View is responsible for the visual representation of the data, such as diagrams or charts. Controller provides logic that binds together models and views, by processing input, calling methods and delivering data to the views [16].

It is important to choose an appropriate framework for the type of website that we are building. There are a number of things that need to be considered when choosing a web development framework, such as:

- The main function of the website.
- Type of features that the website will contain.
- Type of data handled by the website.
- Whether a database is needed.

2.5.1 AngularJS

AngularJS is an open source web framework written in JavaScript. It is one of the most powerful client-side web frameworks [16]. Its main benefit it that it lets the developer turn an HTML-based document into a dynamic web application, by extending HTML's syntax to express web application components in a clear and precise manner. AngularJS allows the developer to write the frontend of their web application in a clean Model-View-Controller way.

This framework was used in the development of popular apps, such as Netflix, PayPal, etc [17], due to the number of advantages that come with using this framework. Developing with AngularJS is an easy process, but has a steep learning curve.

Some of the most compelling advantages of AngularJS include:

- **Two-way data binding:** It is built with Model-View-Controller (MVC) architecture and the framework synchronized the Model and the View. As the data in the Model is updated, the View is updated too [18]. The two-way data binding allows software developers to reduce the development time as no additional code has to be written to synchronize the Model and the View.
- **DOM manipulation:** AngularJS allows developers to assign special behaviour to the Document Object Model (DOM), allowing them to create dynamic and rich content with HTML [18].
- **Server performance:** Due to the fact that AngularJS supports caching and other processes, it reduces load from the server CPU [18]. The server performs extremely well due to the reduced data traffic, which only serves static files and responds to API calls.

Along with advantages, this framework comes with a number of drawbacks:

- **Complex:** There are a number of ways to perform the same task with AngularJS [18] and it is hard for a novice developer to tell which way is better for a particular task. AngularJS is not very intuitive and is time-consuming to learn.
- **Not scalable:** AngularJS implementations do not scale very well. As the project grows with time, the developer will have to discard the current implementation and create new versions, using different approaches.

2.5.2 Ruby on Rails

Ruby on Rails is an open-source, server-side web framework written in Ruby, which is a dynamic and object-oriented language that focuses on simplicity and productivity [19]. This framework uses Ruby to dynamically assemble HTML, CSS and JavaScript files. It helps developers build websites by abstracting and simplifying common and repetitive tasks.

Ruby on Rails operates using Model-View-Controller architecture. It powers well-known websites such as the original version of Twitter, Basecamp, etc [19].

Some of the major advantages of Ruby on Rails include:

- **Time efficiency:** The process of developing website functionality with Ruby on Rails is fast, due to the availability of open source code and a set of existing protocols.
- **Availability of gems:** A ‘gem’ is a software package that contains a library, used to modify the functionality of this framework. Some gems contain utility that will help to automate tasks [19] and as a result, speed up the development process.
- **Testing:** The framework was developed with a focus on testing and comes with a built-in testing framework [19].
- **Scalability:** Ruby on Rails is known for its predictability. Due to the structural nature of Ruby programming language, modifying or adding new features to an existing website is simple. This is particularly valuable for long-term projects.

However, there is also a number of drawbacks that come with this framework:

- **Slow:** The boot time of the framework is quite slow [20], especially for larger projects. As a result, Ruby on Rails has a much slower runtime speed than the majority of other web frameworks.
- **High cost of bad decision:** At the early stage of the development process, the software developer has to make a choice about the type of architecture to be used. The structural dependencies are hard to fix [20] (due to components being highly dependent on each other) and can have a negative effect on the website's performance.
- **Lack of documentation:** It can be extremely hard for novice developers to find good documentation for some 'gems' [19]. Instead of documentation, developers often have to rely on the understanding of the code.

2.5.3 Django

Django is an extremely popular and fully featured client-side web framework written in Python. It is free and open source. Django aims to ease the creation of complex, database-driven websites, by emphasizing the reusability of components. It has a powerful template engine and its own markup language with many tools. Templates are files with HTML code that's used to render the data [21].

Django uses Model-View-Template (MVT) architecture [21], which is a slightly modified version of Model-View-Controller (MVC) architecture, in which the View works more like the Controller and Template acts as the View. The MVT architecture allows the developers to change the visual part of a web app and business logic part separately, without them affecting one another. It powers popular web applications such as Instagram, Pinterest, NASA, etc [21].

There are many advantages to using this web framework. Django is:

- **Quick:** Django has been developed to allow software developers to make an application as fast as possible. It speeds up the development process, as it has its own Object Relation Mapping (ORM) layer for handling database access, sessions, and multi-language support [22].

- **Written in Python:** It is one of the web frameworks that is written in Python programming language. It is commonly known that Python is one of the top-notch programming languages due to its ease of learning, design, and flexibility. The syntax structure of Python makes it simpler for software developers to create a website with meaningful and viable code. Django inherits all of Python's benefits, like great support for external libraries [22].
- **Secure:** Django has a number of built-in security protocols, against attacks such as clickjacking, SQL injection, cross-site request forgery, and cross-site scripting [22]. It is regularly updated with security patches and has a long term support version. To efficiently deal with the username and passwords, Django's user validation framework is the key.
- **Scalable:** Software engineers can easily redo, scale or expand the web structure by improving individual templates. As the project grows over time, previously created templates can be reused.

Like every web framework, Django also comes with a number of disadvantages:

- **It's monolithic:** The framework directs you into given patterns, which is a major disadvantage for developers who want to pick their own structure, architecture and configuration designs. Some internal modules such as ORM and forms are hard to replace [23].
- **Doesn't have the ability to manage multiple requests at once:** Django does not allow individual procedures to deal with a number of requests at the same time [23]. It is up to the developers to design a system that will control various requests proficiently and simultaneously.

2.6 Data Visualization

Data visualization is a graphical representation of data. As mentioned in Chapter 3, the human brain can process visual representation of data 60,000 times faster than any other type of data. By using visual elements such as graphs or charts, the data is communicated more clearly and efficiently, when compared to the data in a spreadsheet. Effective visualizations help to understand data and allow complex data to be more accessible, understandable and usable.

Data visualization tools are essential for analyzing large amounts of information. The factors we need to consider while choosing a visualization tool are the type of data that is being visualized and the desired functionality of the graph.

2.6.1 D3.js

Data-Driven Documents (D3) is an open-source JavaScript library that allows users to create custom and interactive data visualizations in the web browser. It takes advantage of SVG, HTML and CSS standards and provides great flexibility in creating dynamic visualizations. D3.js allows the software developer to create projects where data is bound to the Document Object Model (DOM) elements [25].

The major advantages of using D3.js are:

- **Variety of features:** Data manipulation can create non-standard visualizations for large amounts of data, thus creating unique and interesting graphs.
- **Interactive:** Charts are rendered with Scalable Vector Graphics (SVG) [26], which allows the user to interact with every element on the chart.
- **Large number of examples:** There is a large number of various examples online that show how to use this library.

However, there are a number of drawbacks that come with using this library:

- **Complex:** The syntax of D3.js is complex, as it involves creating each element of the chart.
- **Time-consuming:** In order to create a graph, the software developer has to write a lot of code.

2.6.2 Highcharts

Highcharts is a JavaScript library that allows the user to add interactive charts to their website. It is open-source for non-commercial use [27]. It also provides a wide range of chart types and even allows the user to combine together multiple charts. Highcharts uses simple syntax, making it easy to be used by beginners.

Some of the advantages of the Highcharts library include:

- **Variety of charts:** This library currently provides 20 different types of charts and also makes it easy to combine together different types of charts [27].
- **Customizable:** The charts are easy to customize in relation to both their visual appearance and functionality [27].

However, the major disadvantages of this library are:

- **Lack of good documentation:** Highcharts documentation is not self-explanatory and can be confusing for novice users. It lacks in details of the explanation of customized graphs.
- **Slow:** The rendering of charts can be quite slow, especially with large amounts of data.

2.6.3 Chart.js

Chart.js is an HTML based JavaScript library for creating animated, interactive and customizable charts. It uses HTML canvas element to draw the charts on the web page and allows the user to easily visualize mixed types of graphs [28]. This library is highly customizable and little effort is needed to produce great charts.

The advantages of using Chart.js include:

- **Responsive charts:** All charts are fully responsive, which means that they will adapt to the available space [28].
- **Clear documentation:** Documentation provides a clear and detailed explanation on how to use each feature. It gives a list of all parameters that are accepted by each graph and provides a list of attributes that can be used to change the appearance of the graph. It makes the use of basic as well as advanced features very easy.
- **Time efficient:** Charts are easy to implement and do not take many lines of code.

However, Chart.js has a major drawback:

- **Limited features:** Currently consists of only 6 types of charts [28] and doesn't have the flexibility offered by other charting tools.

2.7 Chapter Summary

This chapter introduces the concept of an autoimmune disease such as vasculitis. Next, it explains the aim of the AVERT project and the research that is taking place to determine the triggers of vasculitis. This chapter also explains the concept of engagement and how vital it is for the AVERT project, that patients engage with the pMp mobile app. Then we move onto the explanation of what EQ5D survey is. Lastly, we are provided with the explanations of what web development framework and data visualization are and the factors that need to be considered when choosing each tool. Next chapter will explain the process of designing this website.

Chapter 3: Design

This chapter will explain in detail the process of gathering requirements from the client and using them to design an initial set of prototypes of the user interface. It will also explain the necessary changes that had to be made to the initial design, the reasons for them and how the design ensures security to the patient data.

3.1 Design process

A website development process is an outline of steps needed to be taken, from start to finish, in order to complete a website design project [9]. It transforms high-level tasks to smaller subtasks by dividing and categorising the work that is to be done. It is a way of organizing developer's thoughts, avoiding repetitive tasks and speeding up the project's timeline.

Figure 3.1 shows the process and steps involved in the design of this project.

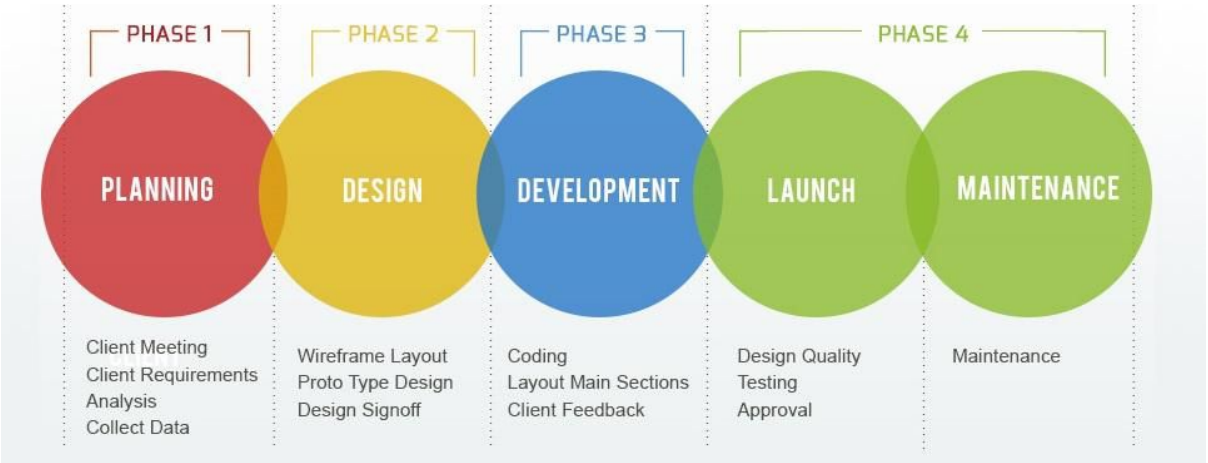


Figure 3.1: Design process [10]

The four different phases highlight all the actions that are needed in order to develop a high-quality website. Concentrating on one phase at a time allows the developer to carry out their work in an efficient and effective manner. After each of the first three phases, it is a good habit to go back and review the work that was done and check if there are any corrections necessary, before moving on to the next phase.

This design process will ensure that this project meets all of the objectives outlined in Chapter 1, that it works correctly in different web browsers and that it meets usability standards. Designing for usability occurs in all stages of the process, and it is not a single and isolated activity. Keeping in mind who the client is, is a key to a well-designed website.

3.1.1 Planning

This is the first and, by far, the most important stage of the design process, because what's decided and mapped here, sets the stage for the whole project. This stage provides the most interaction between a developer and a client. It is important at the very beginning of the process, to get a better understanding of the client, their industry, and what is the ultimate goal of the website. By the end of this phase, the client should provide the developer with enough information, to allow them to create a simple website view with some basic functionalities.

As mentioned above, this phase consists of a lot of interaction between a developer and a client. The first action taken for this project was to set up a meeting with the client. The research nurse (client), provided basic information about her role in the AVERT group and their vasculitis research. The continuous monitoring of patients and encouraging them to use the pMp mobile application to record their health and location data is the client's major responsibility. Thus she needs a tool what will give her an overview of the engagement levels and health status of each patient, in a time efficient manner.

Since there currently exists a tool that displays all the information about each patient, the research nurse demonstrated its functionality. It turned out that the major disadvantage of the current tool is the inability to see straight away which patients need her immediate attention. In order to see the patient's information, if they are recording their health and location data and if they have entered any worrying symptoms, the research nurse has to go through each patient individually. With the number of patients increasing each year, this approach would become very inefficient and very time consuming within the next few years.

Another major disadvantage of the currently used tool is the data representation format. For each patient, the data is represented in a tabular form. Human beings process visual representation of data much more quickly than any other type of data. To put it in perspective, the human brain can process images 60,000 times faster than any text [8]. From this, a conclusion can be drawn that representing patient data in a tabular format is not the most effective way, especially if a person has to capture engagement levels for a large number of patients.

The goals that the client specified for this website are as follows:

- To have access to patient data in real-time.
- To be able to clearly see which patients need attention.
- To visualize the data and capture patient engagement levels.

Having access to patient data in real time is, of course, the most important requirement that was set. The research nurse needs to be able to see at all times the data that patients enter and when they have entered it, so there can be no delay in updating any patient data on the website. The delays could cause misunderstanding between a patient and the nurse, in cases where a patient was involved with the mobile application, but that activity was not updated on the website fast enough.

Having to monitor each patient individually is a major inconvenience; a dashboard view would allow the research nurse to get an overview of the overall patient engagement over a specified amount of time. This is the next requirement for this project. The features requested for the dashboard were as follows:

- Ability to search individual patient based on their ID number.
- Notification table that allows a clear overview of patients who do not engage with the pMp mobile application.
- Summary reports (in the form of graphs) which indicate overall patient engagement.

The last requirement established by the research nurse concerns how the patient's engagement should be measured. The most useful patient information that is to be displayed in the form of graphs includes:

- The frequency of opening the pMp mobile application.

- The frequency of recording of symptoms.
- The frequency of completing EQ5D.
- The number of location readings per day/week.
- The number of recordings per day/week.

3.1.2 Prototype design

This is a second phase of the design process and most important from the client's perspective. Prototypes allow the client (especially a non-technical one) to be more involved with the project, as it is a language that both client and designer have in common. One of the major advantages of using prototypes before starting a project is the fact that it allows the client to see what the product will look like early on in the project lifecycle. It helps the designer and client to come to an agreement about what is in and out of the scope of the project.

Prototypes allow the designer to find implementation issues early on in the project. There is a number of prototyping methods and each of those methods can be categorised either as low or high fidelity prototyping. Low fidelity prototyping is a quick and easy way to translate a high-level concept into a visualized object [29]. This type of prototyping usually involves simple paper sketches and is a good way to test the functionality of the product rather than visual appearance. High fidelity prototypes are an interactive mockup that resembles both the appearance and functionality of the end product [29].

As part of the design phase, I created a high fidelity prototype to show my client the basic functionality and appearance of the website. The prototype is interactive and provides the user with an experience of using a real website. I adopted the patientMpower logo and chosen an appropriate theme that is consistent across the whole prototype. In terms of functionality, all of the components were filled with appropriate details and different pages are linked together.

All of the requirements set out by the research nurse are part of the prototype. One of the most important features to the nurse is a dashboard overview of engagement levels

of all patients and an alerts table containing ID numbers of patients who haven't been using the pMp mobile application.

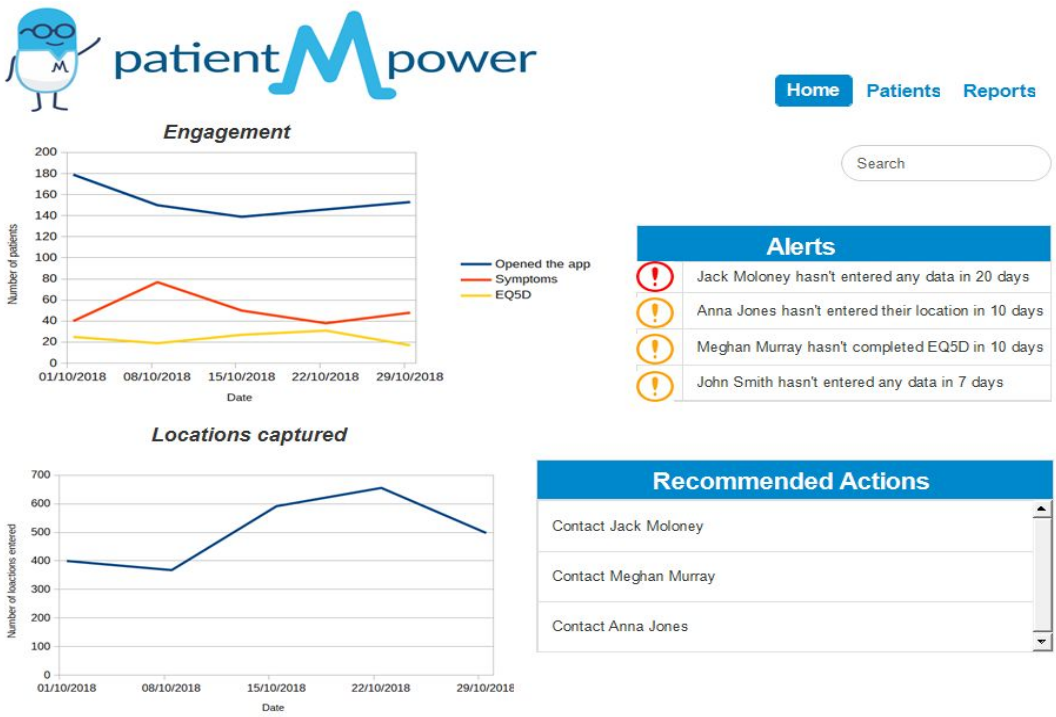


Figure 3.2: Initial design of the dashboard page

Figure 3.2 shows the initial idea of what the dashboard would look like and what type of features it would contain. This is the first page that the research nurse will see when she opens the website and signs in to their account. The nurse can immediately see on the left-hand side a graph that shows engagement levels of patients based on the opening of the app, recording of symptoms and completion of EQ5D. The number of patients using the mobile app is computed on a weekly basis and the data is displayed over a five-week period. Since the success of the AVERT project is highly dependent on the number of locations captured by vasculitis patients, there is a separate graph dedicated to displaying the number of patients capturing their location each week.

On the right-hand side of the dashboard, there is an alerts table that consists of a list of patients who have not been engaging with the pMp mobile app. The coloured alert symbol (red and orange) is assigned to each alert entry based on the period of inactivity of a patient. Colour coded alerts will allow the research nurse to focus her attention on patients with the longest period of inactivity (red alert) first, before moving onto other patients. An alert consists of a patient's name, a meaningful message regarding the data

that lacks in patient's engagement, and a number of days since the patient has last engaged in recording of that data.
One of the features of the alert table is the ability to click on the patient's name and be quickly redirected to that patient's profile.

The recommended actions table is an additional feature that will help the nurse to determine what action should be taken regarding the patient. Patients whose names appear in the alerts table are automatically added to the recommended actions table. However, patients who have recently shown a high level of engagement are also included in the recommended actions table.

The last feature on the dashboard is a search bar that will allow the nurse to enter the patient's ID number and be quickly redirected to that patient's profile.

Visualizing patient's health and engagement data was another requirement set out by the research nurse. This requirement should be adapted to both, data collected by an individual patient and cumulative data of all patients.

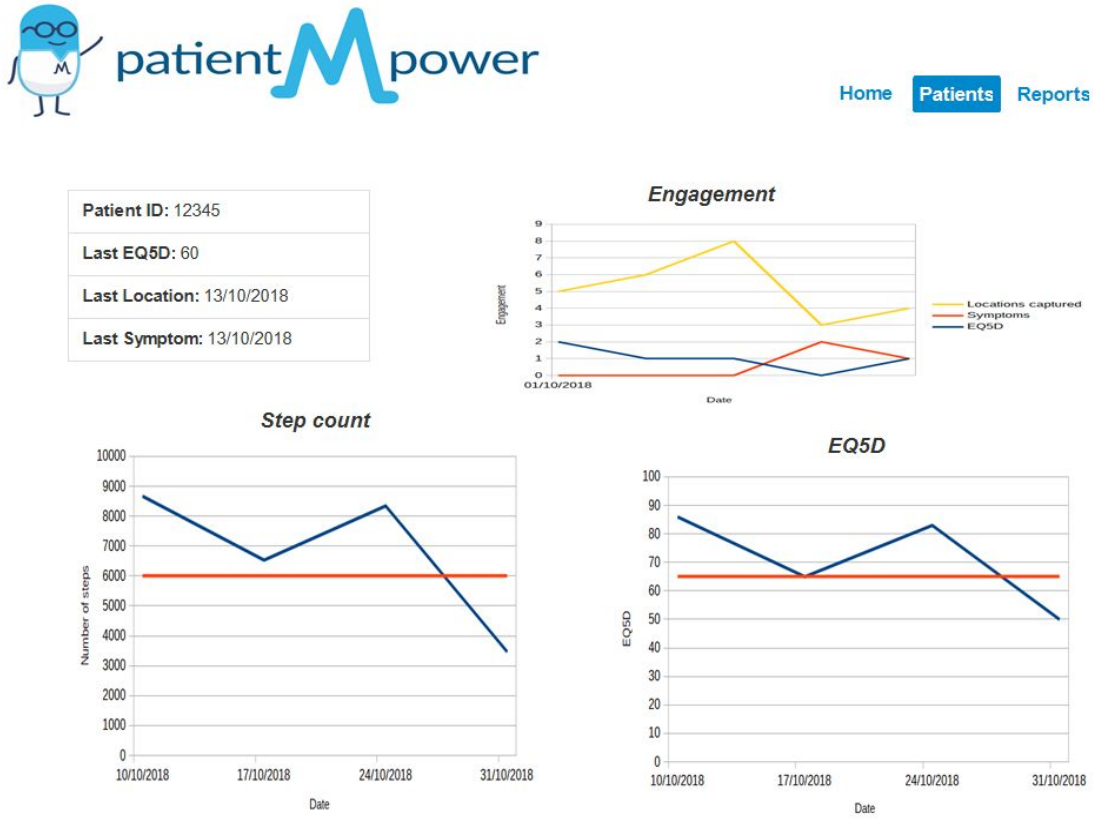


Figure 3.3: Initial design of the individual patient profile

Figure 3.3 shows the initial design of what a patient's profile will look like. This page can be accessed from the dashboard (by searching the patient's ID number or by clicking on the patient's name in the alerts table) or from patients page (by clicking on the patient's ID number). The top left-hand side displays information on when the patient has last captured their health and location data. In this way the research nurse can see exactly what was the last date a patient was engaging with the mobile application, and in the case of EQ5D data, the latest value is displayed to allow the nurse to monitor patient's quality of life.

Patient engagement is captured based on the patient's weekly recordings of location, symptoms, and EQ5D. Those three types of data are the most important to the nurse as they provide a good overview of the patient's health status and ensure that patients are capturing their locations regularly. For each of the data types, the number of recordings is calculated on a weekly basis and displayed on the graph.

Step count and EQ5D are both charts that visualize patient's health data. The red line indicates an average that is computed based on patient's health goals, whereas the blue line indicates a trend of weekly recorded values.

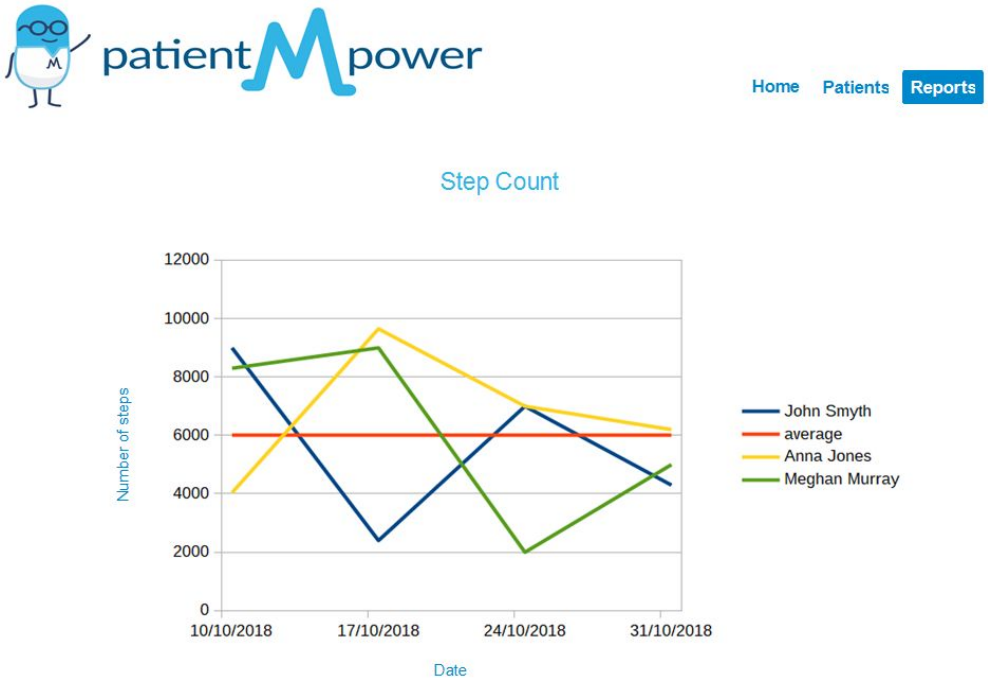


Figure 3.4: Initial design of the summary report

Figure 3.4 is an example of a summary graph that accumulates data of all patients. The step count values are counted over a one week period and are displayed in contrast with the red line which indicates a weekly step count of an average person. Names on the right-hand side of the graph are clickable and will redirect the research nurse to the profile of the corresponding patient.

The entire set of initial user interface prototypes can be found in Appendix A.

3.1.3 Final design

The second phase of the design process isn't complete until the client signs off on the prototypes. Good communication between the client and the designer is key during this phase. The client and the designer have to agree on the appearance and functionality of the final product and clarify any misunderstandings. The prototypes might have to be modified a number of times before the product design resembles the client's vision.
Due to the nature of the work of my client, they were not able to meet with me for prolonged periods of time - so we could not regularly go over the initial set of prototypes. To avoid slowing down the progress of this project, I composed an email instead, that explained in detail the functionality and contained screenshots of each prototype. The chief clinician and one of the AVERT medical staff were also copied in on the email, in order to get as much feedback as possible. Both the chief clinician and the research nurse provided me with a lot of useful feedback on the prototypes regarding the functionality of the website. The research nurse added the following requirements to the scope of this project:

- Ability to filter the alerts table and cross off notifications of patients she has already contacted.
- Add an RKD filter to the summary reports.
- Ability to maintain a list of RKD patients.
- Forgotten password option.

RKD filter allows the research nurse to focus on patients registered with rare kidney disease (RKD). Those patients are very important in AVERT research, so it is important to have the ability to distinguish and compare data between different types of patients.

All of the additional requirements listed above have been incorporated into the final design of the user interface.

After gaining a better understanding of the project and feedback provided by the client, it became apparent to me that the patientMpower logo is not appropriate to use on this website. Instead, the logo was replaced with the AVERT logo, which as a result changed the colour scheme of the entire project.

Changes had to be made to a number of features on the dashboard page. Also, some additional features were added in order for the research nurse to get a better overview of the engagement levels of the patients. The ability to filter the alerts table had to be incorporated into the final design.

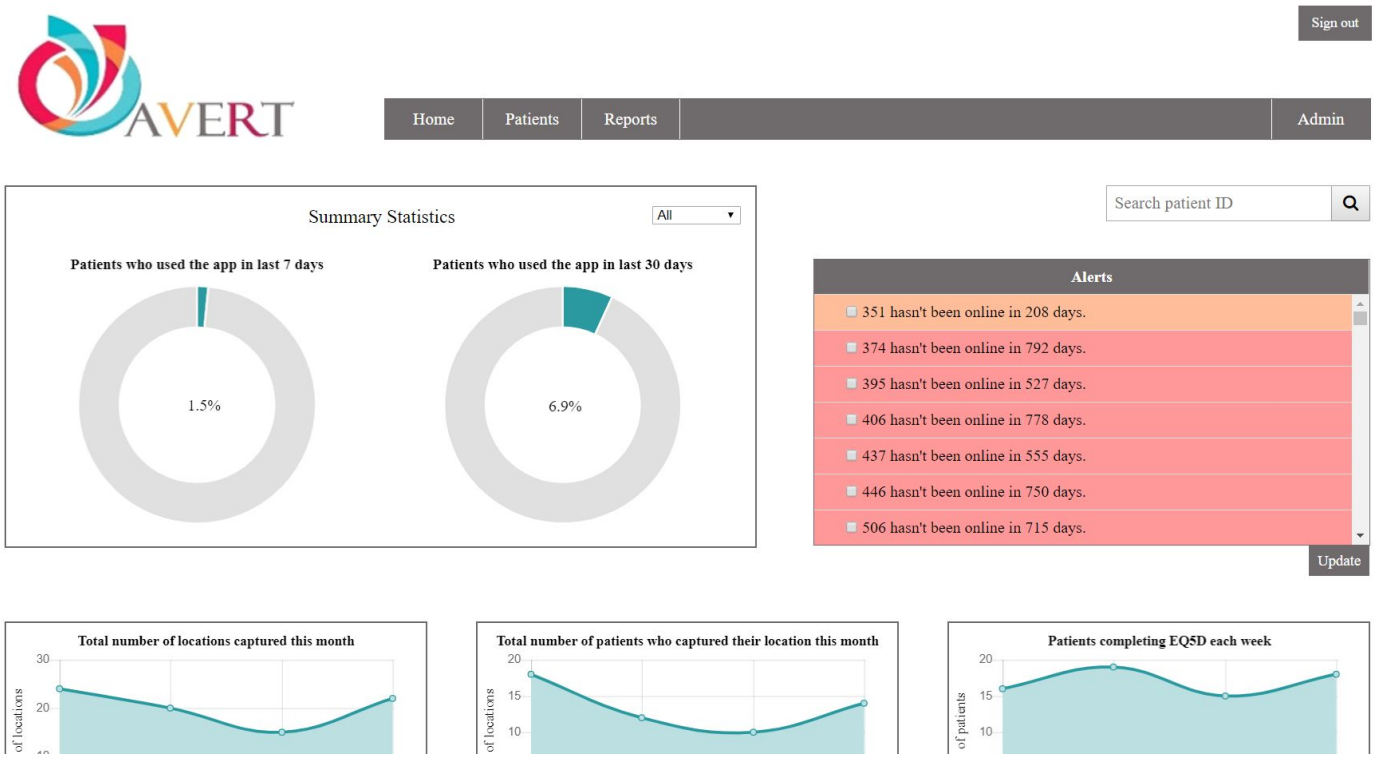


Figure 3.5: Final design of the dashboard page

The first thing that the research nurse will see signed in is the dashboard page, which is shown in Figure 3.5. A new feature that has been added to the design is the summary statistics section located on the left-hand side of the page. This section consists of two graphs that provide a clear overview of the percentage of patients using the pMp mobile application in the last 7 days and the last 30 days. Once the user hovers the mouse over different sections of the graph (gray or green), they will be able to see the exact number of patients who do and do not use the pMp mobile application. Both of those graphs have the ability to be filtered by the following parameters: all patients, active patients only and RKD patients only.

The alerts table has been adapted to meet the additional functionality. Previous alert symbols have been replaced with colour coded rows. There are three types of alerts:

- **Yellow** for patients who haven't shown engagement in less than 100 days.
- **Orange** for patients who haven't shown engagement between 100 and 300 days.
- **Red** for patients who haven't shown engagement in more than 300 days.

Due to data privacy issues that will be discussed later on in this chapter, the user interface does not contain any personal information and patients' names have been replaced with ID numbers. Also, each entry in the table contains a checkbox that will allow the user to select patients that have been already contacted and temporarily delete them from the table by clicking on the 'update' button. If the patient has been marked as 'inactive' by the admin, the alerts for that patient will no longer appear in the alerts table.

The need for the recommended actions table has been eliminated. Initially, the alerts table was supposed to show patients that are not engaging with the mobile application, and also patients who have been doing very well and use the mobile app on a regular basis. The research nurse's view has changed on this matter and decided that only the patients who lack engagement should be displayed in the alerts table.

Additional graphs have been added to the bottom of the page in order to give a better overview on the number of patients capturing their location each week and also on the

number of locations that are captured each week by patients. The EQ5D graph allows the user to see whether or not patients complete the survey on a regular basis.

Due to the additional requirement of maintaining a list of patients who are RKD registered, an admin page had to be implemented. This list can only be updated by the

admin (research nurse) and no other user.

Figure 3.6: Final design of the admin page

Figure 3.6 shows Django's default admin page. A button has been added to the navigation bar in order to allow the admin to have quick access to this page. This page can only be accessed if the user has used admin's credentials (email and password of research nurse) when logging in. Admin has an ability to add/remove users, add/remove ID numbers in inactive patients list, and add/remove ID numbers in the RKD patients list. The process of adding new users is very simple, the admin has to input a unique email address and set an initial password (which can be changed by the user by following the forgotten password link). An admin can also edit user accounts and add additional information such as user's full name, date of birth and organisation.

'Inactive patients' is a list of IDs of patients who have signed up to the pMp mobile application, however, they don't intend to use the app anymore. Those patients will remain in the pMp database, and to allow them to be removed from the alerts table, the

research nurse will need to keep track of those 'inactive' patients. The ID numbers of those patients will be stored in the website's database. If at any point, a patient decides that they would like to keep using the app, the research nurse can remove their ID number from this list and once again the alerts table will contain entries for that patient. A similar process occurs with the list of 'RKD patients', where the research nurse will input ID numbers of patients that are RKD registered.

In the feedback I have received from the initial set of prototypes, the research nurse provided a set of guidelines regarding patient-specific graphs. The individual patient profile had to be adjusted to meet those guidelines. Also, more information had to be added to allow the nurse to get a better overview of the patient.

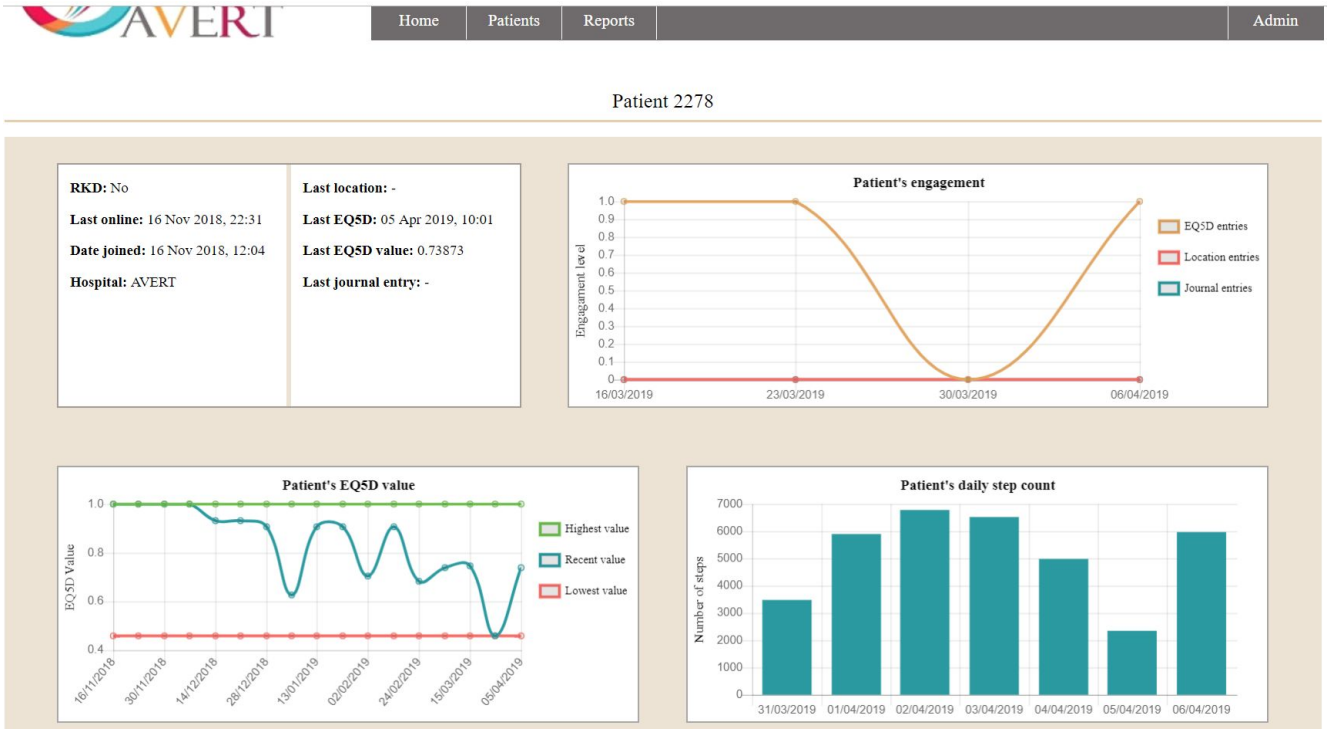


Figure 3.7: Final design of the individual patient profile

Figure 3.7 shows the screen that a user will see once they open (or are redirected to) a patient's profile. Patient's information had to be expanded upon to contain patient's RKD status, hospital information, date of joining pMp mobile application and a date patient was last seen online.

The EQ5D graph has been improved and displays a trend of values compared to the patient's lowest (red line) value recorded and highest (green line) value recorded. This

allows the user to put the patient's current EQ5D value into perspective and get a better overview of their quality of life. According to the standard adopted by the research nurse, patients should complete the EQ5D survey at least once a month. Initially, it was thought that the 'health' component of EQ5D should be visualized on the graph, however, after careful consideration, the value that is visualized now is a result from the computation of values from all five dimensions of the survey. The graph ranges from -1 to 1 and displays all of the patient's EQ5D entries.

The daily step graph has also been updated to display the patient's daily step counts over a seven day period. The style of the graph has been adjusted to a more suitable bar representation.

Finally, the last requirement that had to be incorporated into the website's design is the ability to filter summary reports based on patient type. The research nurse is interested in monitoring all patients, however, she prioritises patients that are RKD registered and needs to be able to have an independent overview of how those patients are doing.

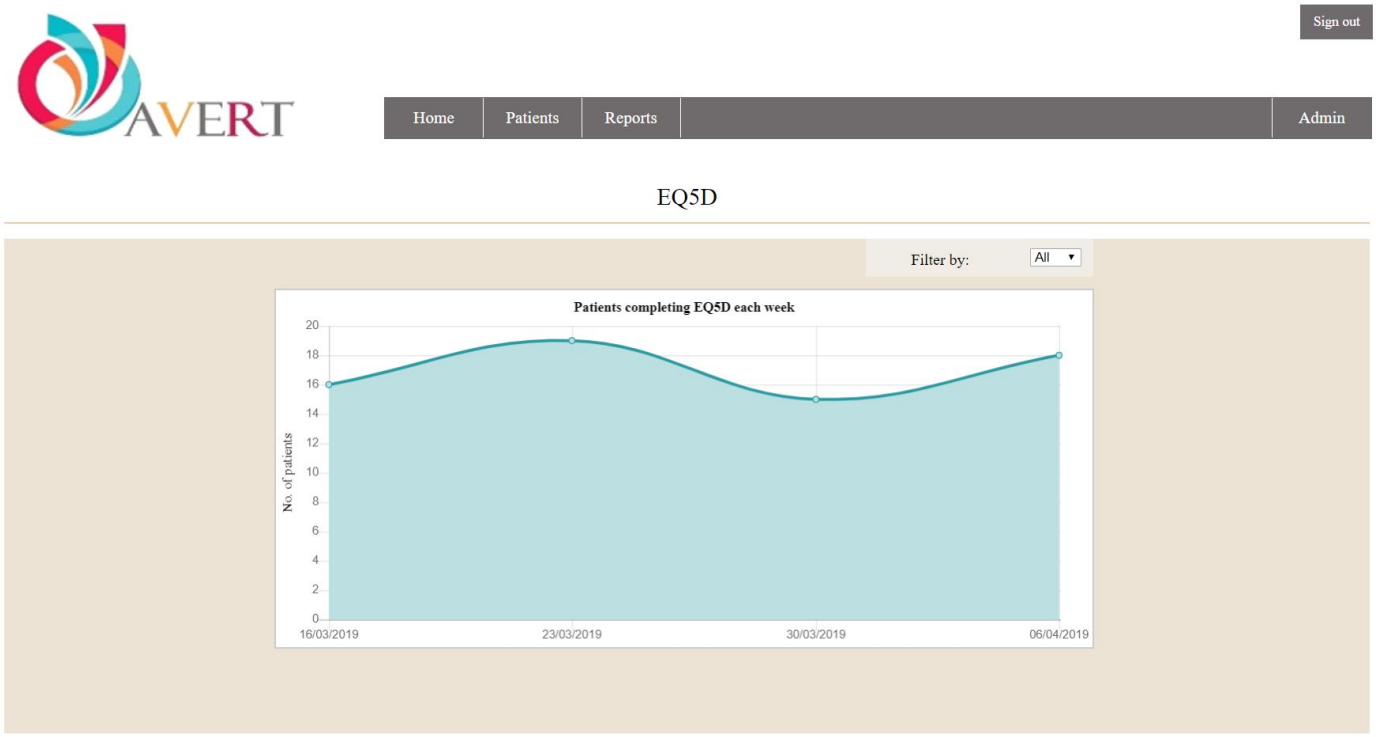


Figure 3.8: Final design of the EQ5D summary report

Figure 3.8 shows a modified EQ5D summary report. Previously the graph visualized information for all patients, however, the filter option that is located just above the graph, allows the user to navigate between a graph for all and RKD registered patients.

The entire set of the final design of the user interface can be found in Appendix B.
In order to complete the design phase, I had a meeting with the chief clinician who has accepted the final design of the user interface. There were no additional changes that had to be made and the project was able to move to the third phase of the design process.

3.2 Data security

Digital technology is now a big part of our everyday life. It brings multiple advantages, especially in the medical field, but it also has a number of major disadvantage. In recent years there is an increasing amount of cyber attacks, during which a lot of personal information is leaked. In the medical field, all of the information that is digitalized makes the job of medical staff much more efficient as all the information is easily accessible. With great convenience regarding digital storing of data, comes great responsibility of ensuring that all of the data is secured.

Data security refers to the process of protecting data from unauthorized access. There are a number of ways in which we can secure the data, using hardware and software technologies. Sensitive patient data, such as patient's medical history and personal information, should only be accessed on computers with an antivirus or firewall that would stop and notify the user about any malicious attack [24].

In order to ensure that only authorized users have access to patient data, there should be a secure connection established between the website I have designed and patientMpower production server. The website should be hosted on a secure server where the URL begins with https, rather than with http. Having a secure connection means that patient data that is sent from the pMp production server to this website, will be encrypted and if anyone tries to intercept that data, what they receive will be meaningless. The encrypted patient data can only be decrypted by using a key that is only known to the two computers - the production server and this website.

I have chosen to implement this website using the Django web framework, due to the number of built-in security protocols that Django offers. One of the most important security protocols that are used is the cross-site forgery request (CSFR) protection. CSRF attacks allow a malicious user to execute actions using the credentials of another user without them knowing about it. CSRF attacks can have severe consequences such as leak or theft of data. Once the attacker is able to access the website using another user's credentials, they would have access to all of the data. Django has built-in protection against the CSRF attacks, as it checks the secret (which is user specific) in each POST request. An extra secure connection can be formed by combining CSRF protection with using a secure connection (https), as Django will check that the http referer header is set to a URL at the same origin on top of detecting a CSRF attack. For a website containing any sort of patient information, it is crucial to protect it against this type of attack.

Another default security protocol that is used by Django is the cross-site scripting (XSS) protection. During an XSS attack, the attacker injects client-side script into a browser of a different user. This can be achieved in two ways, firstly it can be done by storing malicious scripts in the database from where they can be retrieved and displayed to another user. The second way this can be achieved is by getting a user to click on a link that will cause the attacker's JavaScript to be executed by the user's browser. Django's template system protects the user against the majority of XSS attacks (especially at fields populated by user input such as login credentials), by escaping specific characters that are considered "dangerous" in HTML. Those characters include '<', '>' and '&', and are particularly dangerous to use as they can alter intended functionality of the script.

Despite the number of built-in Django security protocols that will protect the user from malicious attacks, a security system needs to be designed to ensure that only authorized users can access the website and its features. The following security features are included in the final design of this website:

- Unique user accounts
- Email rather than username verification
- Automatic user log out after 10 minutes of inactivity

- Easily accessible logout button
- Required login to access any page within the website

3.2.1 Data Privacy

The pMp mobile application is used by vasculitis patients to record their health and location data. Both health and location data is considered Personally Identifiable Information (PII), which means that the data recorded can be used on its own to identify or contact a specific individual [24]. The health data recorded by patients is also considered as Sensitive Personal Information (SPI) and should be handled with extra security and kept separately from any other personal data [24].

Developers of the pMp application store all of the patient's health and location data on the same server without using any encryption. This is a very insecure way of storing the data, that can have severe consequences. In case of data leak or data theft, each patient is easily identifiable by their full name, date of birth and gender, can be contacted by using their email address and also can be tracked as the location coordinates and corresponding timestamps are also not encrypted.

Data privacy governs how the data is collected, shared and used. This project has no influence on how the patient data is collected or shared, however, it can ensure that all of the patient data is used in an appropriate and secure manner. Since SPI should be kept separately from other personal information, patient's name, date of birth, gender, email address, and location coordinates will not be part of the user interface. On top of that, all of the data will be pseudonymised by replacing identifying information such as the patient's name, with an artificial identifier such as the patient's ID number. In the case of data leak occuring, this will ensure that the identity of all patients is kept anonymous.

The research nurse is currently using a spreadsheet that stores all of the patient's personal and contact information. That spreadsheet is encrypted and stored on their local computer, so only she has access to it. For the purpose of this project, that spreadsheet will be used by the nurser to create a link between an anonymous patient (by using their patient ID) and their personal information. This ensures that only the research nurse knows patients' real identities and is the only person who can get in

contact with them. This greatly reduces the chance of any patient being identified in the case of a data leak.

As part of this project, I had to design a database to store ID numbers of patients who are registered as rear kidney disease (RKD) patients and those who do not wish to use the pMp mobile application anymore and want to be considered inactive. Considering that the only information stored in the database is patient's ID number, this also does not raise any data privacy issues as it cannot be connected to any sensitive personal information, without the nurse's spreadsheet.

3.3 Chapter summary

This chapter explains the concept of a design process, its importance and how it was adopted by this project. Next, the planning phase is introduced with an explanation of how requirements were gathered from the client. We then move onto the design phase, where the requirements from the previous phase are transformed into an initial set of prototypes. The final design explains the necessary modifications that had to be made based on the feedback provided by the client. Finally, the security aspect of this project and data privacy issues are explained. Next chapter will introduce techniques used to implement the final design of the user interface and the implementation issues that arose during that process.

Chapter 4: Implementation

This chapter will provide an explanation on how to access and query a server using an API. It will also provide an explanation on how to use tools to visualize data obtained by the pMp API. Next, the chapter will provide a detailed explanation of the database that was designed as part of this project and describe the process of implementing a user interface, along with a discussion of issues that arose during that implementation process.

4.1 patientMpower API

An Application Programming Interface (API) is a set of methods that allow applications to communicate with each other. An API is not a database or a server [12], but rather it is part of the server that receives requests and sends responses. A good API makes it easier for the software developer to create an application, as it provides all the building blocks, which are then put together by the software developer.

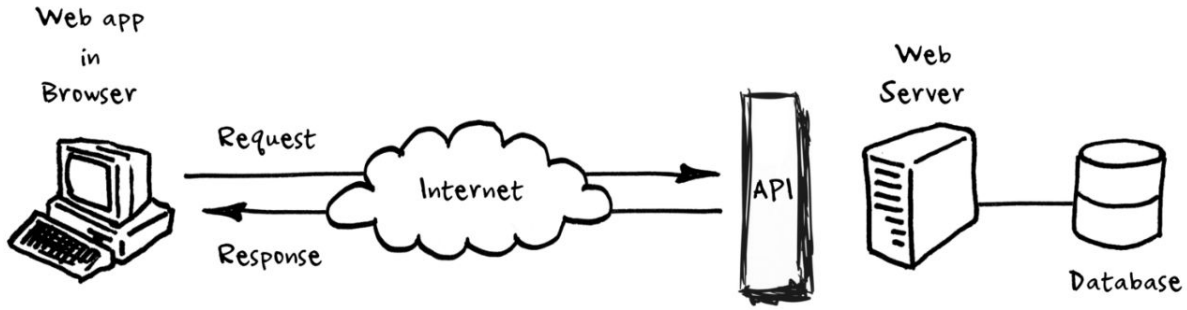


Figure 4.1: How an API works [32]

Figure 4.1 demonstrates the process of how a web-based API works. The user makes a request to access some data on a server, the server accesses the database, and returns the data to the user in a form of a response. Requests to write or retrieve data are generally done by sending a HTTP request to the server, and the server responds to them by sending a HTTP response. Those responses are typically served as JavaScript Object Notation (JSON) or Extensible Markup Language (XML).

Every company has a different API. Before starting the development phase, I had a meeting with patientMpower regarding their API. During that meeting, a pMp software

developer informed me that their API is built using Django REST (Representational State Transfer) framework and explained the type of endpoints it contains. I was also assured that the API documentation is self-explanatory and demonstrates how to correctly access and query the API.

From looking at the API documentation I was able to see clearly the type of data stored in each endpoint. This gave me an overview of the endpoints I needed access to for the development of this project. The documentation contained an explanation on how to access an endpoint, however, it did not explain how to query it.

4.1.1 Accessing

In order to have access to the API, pMp set up an account with my credentials and granted me access to the endpoints I needed for the development of this project. The documentation explained that the API uses a token authentication system to ensure that only authorized users can access it. If the user's authentication is successful, the server will respond with code 200, which means that their HTTP request was successful. However, the pMp API can also respond with code 400 for an invalid HTTP request, 401 for an unauthenticated HTTP request and 500 for server error.

A 'token' is a piece of information that has no meaning on its own, but when combined with a correct tokenization system, it can be used to verify the user's identity. Token-based authentication ensures that each request to the server has a single token, which is verified by the server and only then the server responds to the request [13].

pMp uses JSON Web Tokens in their API authentication system. JSON tokens consist of three parts: header, payload, and signature. Both the header and payload are encoded and then algorithmically signed to protect against any form of manipulation [13].

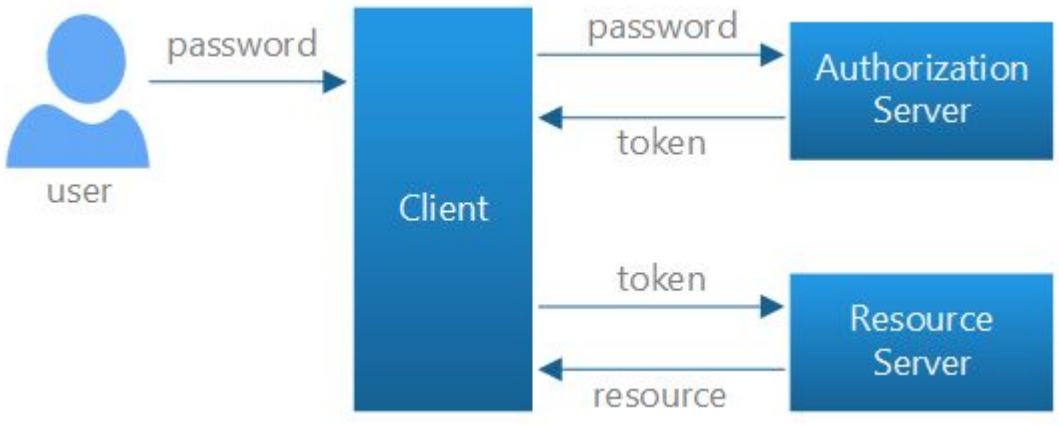


Figure 4.2: Token-based authentication mechanism [14]

Figure 4.2 shows the token authentication mechanism used by pMp. Firstly, the user needs to input their credentials (email address and password), which are then passed to the authorization server. The authorization server verifies those credentials and if they are valid, it will return a signed token to the user. Script 4.1 shows the Python script used to obtain an authentication token from the pMp authorization server, where

```
Token_request = requests.post('https://patientmpower.com/api/auth/token-auth/',data =
{'username':'my_email_address','password':'my_password'}).json()
```

variables ‘my_email_address’ and ‘my_password’ contain user credentials.

Script 4.1: Obtaining JSON Token

Once the token is obtained, it can be directly passed to the production server. The token will be validated by the server and only if valid, the user will be granted access to the data. I incorporated the token, into the header which is used to access every pMp endpoint. Script 4.2 shows the structure of the header, where variable ‘token_request’ is the response from script 4.1 and contains a unique token. ‘Accept-Content-Type’ ensures that the data returned from the production server is in JSON format.

```
header = {'Authorization':'Token token  request',
```

Script 4.2: Header structure

4.1.2 Querying

In order to get data back from the production server, a request has to be sent to the server. The request should consist of the endpoint we are trying to access and also a

```
patient_data = requests.get('https://patientmpower.com/api/patient/', headers
```

header (see script 4.2) with our authentication token.

Script 4.3: Sending a request to the server

Script 4.3 demonstrates how to use Python, in order to get a response from the production server. 'patient_data' is an object that contains all data stored in the 'patient' endpoint. The response from the API comes in a JSON format which is shown in Figure 4.1. The patient object can be treated as a multidimensional array.


```
[{'id': 550, 'user': {'id': 628, 'email': 'martin+mixedpanels@patientmpower.com', 'last_login': '2018-07-10T13:04:10.745
131+01:00', 'is_superuser': False, 'first_name': 'Martin', 'last_name': '', 'is_staff': False, 'is_active': True, 'date_
joined': '2018-07-10T13:04:10.447129+01:00', 'groups': [], 'user_permissions': []}, 'conditions': [{'id': 549, 'name': '
vasculitis', 'type': {'id': 6, 'consents': [{'id': 5, 'user_text': '<div style="background:#f3f5fb; border:1px solid #cc
cccc; padding:0px 5px">\r\n<p><span style="color:#30446a"><span style="font-size:18px">By using this app, you will be co
ntributing your health data to our Digital Biobank<sup>TM</sup>. Your data will be </span></span><span style="color
:#30446a"><span style="font-size:18px">aggregated (combined with a group of other users data) and de-identified (id
entifiable information will be removed). Examples of how your data may be used: </span></span></p>\r\n\r\n<ul>
\r\n\t<li><span style="color:#30446a"><span style="font-size:18px">To help researchers better understand health conditio
ns</span></span></li>\r\n\t<li><span style="color:#30446a"><span style="font-size:18px">To help companies develop new tr
eatments</span></span></li>\r\n\t<li><span style="color:#30446a"><span style="font-size:18px">To help improve your exper
ience of this app </span></span></li>\r\n</ul>\r\n\r\n<p><span style="color:#30446a"><span style="font-size:18px">W
e place great importance on protecting your data and we employ strict security measures. You can read more about ou
r Privacy Policy and Terms of Use below.</span></span></p>\r\n\r\n<p><span style="font-size:18px"><strong><a href="https
://info.patientmpower.com/privacy-policy/">Privacy Policy &gt;</a>      </strong></span><span style=
"font-size:18px"><strong><a href="https://info.patientmpower.com/terms-of-use/">Terms of Use  &gt;</a></strong></sp
an></p>\r\n</div>', 'name': 'App Consent', 'name_en_uk': 'App Consent', 'name_en_us': None, 'name_es': None, 'name_de':
None, 'name_it': None, 'name_cs': None, 'text_en_uk': '<div style="background:#f3f5fb; border:1px solid #cccccc; padding
:0px 5px">\r\n<p><span style="color:#30446a"><span style="font-size:18px">By using this app, you will be contributing yo
ur health data to our Digital Biobank<sup>TM</sup>. Your data will be </span></span><span style="color:#30446a"><sp
an style="font-size:18px">aggregated (combined with a group of other users data) and de-identified (identifiable in
```


Figure 4.1: Part of the patient JSON object

In order to access only the data we are looking for, the patient object was parsed using Python. Script 4.4 demonstrates how to obtain a list of patient ID numbers from the patient object. The code iterates through each item (patient entry) in the patient object and adds the value of the 'id' variable into the patient_id array. This approach can be

```
for item in patient_data:
        patient_id.append(item['id'])
```

used to access any type of data within any endpoint.

Script 4.4: Querying a JSON object using Python

4.2 User interface

User Interface (UI) is a visual part of a computer application which allows the user to interact with computers. There are three types of a user interface: command line, menus and graphical user interface (GUI) [30]. Command line user interface requires the user to be familiar with program-specific instructions, whereas menus provide the user with a list of instructions to choose from. However, in both cases the user needs to be familiar with the program's instructions, thus those types of UIs are frequently used by software developers (technical users). A graphical user interface allows the user to give commands to the computer, by clicking on icons or buttons. It is an appropriate UI choice for this project, given the nature of my client's work.

In Chapter 3, I have discussed various prototype designs of the user interface, and the implementation process is based on the final design of prototypes. I have decided to use the Django web framework for the development of this website and Chart.js library for the implementation of the graphs.

Despite the number of disadvantages that are mentioned in Chapter 2, the major advantage that favoured Django is its security protocols. Due to the nature of the data that will be processed by the website, security is the number one priority. Django has built-in protocols that will protect the website from clickjacking, SQL injection, cross-site request forgery, and cross-site scripting attacks. It also has a number of built-in functionalities that are used in the development of this website.

Chart.js is a data visualization tool I have chosen to use for the implementation of the graphs. Even though the library does not offer a large variety of graphs, it has all the functionality that is necessary for the visualization of patient data. On top of that, the library is easy to use and provides clear documentation of each graph, with a detailed explanation of all its attributes. For the purpose of this website, the graphs do not need to be over complicated, but instead should be simple and make it clear to the user if patients have been using the pMp mobile application.
The first step of the development phase is to set up a HTML template for all pages that should be included in this website. Django's template engine allows the developer to create templates that can be reused and linked to other templates, thus eliminating the

need to write duplicate code. I designed a template of a simple header that contains the AVERT logo, navigation bar with multiple tabs and a sign out button. This header is linked to every web page by using Django's markup language.

4.2.1 Dashboard page

This is the first page that the user sees after they have signed in to their account. The dashboard page consists of six different features that were carefully designed in the previous phase. Those features include a search bar, an alerts table, summary statistics charts, two location related charts, and an EQ5D related chart.

The search bar feature allows the user to search a patient by their ID number. It contains autocomplete functionality, which shows a list of possible ID values based on the user's input. For example, if the user inputs '1' into the search bar, a list of ID numbers that begin with '1' will appear at the bottom of the search bar. This functionality has been implemented for the sole purpose of reminding the user about their possible input options. Once the user inputs a value or selects one from the list, the Python script will check whether a list of patient IDs obtained from the pMp server contains that value. If so, the user will be redirected to the corresponding patient profile. However, if the value entered into the search bar is not in the ID list, an error message will appear at the top of the screen, notifying the user that the value they have entered does not correspond to any patient ID.

The alerts table is one of the features requested by the client. It displays ID numbers of patients that fit the following criteria:

- Patient hasn't been online in the last 30 days.
- Patient hasn't completed EQ5D in the last 30 days but has completed it at least twice before.
- Patient hasn't captured their location in the last 21 days but has captured at least ten locations before.

Based on the criteria that fit the patient, an appropriate alert message will appear beside their ID number, along with a number of days they lacked in engagement. The aim of checking each patient against these criteria is to avoid creating alerts for patients who have signed up to use the mobile app but never shown much engagement or interest in

using the app. The table should only contain alerts for patients whose engagement levels have dropped recently, as those patients need to be contacted by the research nurse.

Each row in the alerts table is colour coded according to the severity of patient's lack of engagement with the pMp mobile application. There are three levels of severity: yellow for patients who did not show engagement in under 100 days, orange for patients who did not show engagement in between 100 to 300 days, and red for patients who did not show engagement in over 300 days. It is JavaScript that assigns a colour to each row, based on the time period for which the patient lacked in engagement. The colour coded rows are there to help the research nurse to distinguish between which patients need her immediate attention.

The alerts table also has the functionality to delete selected rows for a period of time. By selecting a checkbox of an appropriate alert and clicking the 'Update' button, the research nurse can temporarily delete an alert entry. This functionality is particularly useful for when the nurse has contacted a number of patients and wants to see what patients are remaining. Patients have a choice of whether they want to use the mobile app. The research nurse needs to have the ability to mark patients who have given up on using the app, to prevent the creation of unnecessary alerts. This can be achieved by adding the patient's ID number to a list of 'inactive patients' on the admin page.

Summary statistics charts display the percentage of patients who have used the app in the last 7 and the last 30 days in the form of a donut chart. The Python script checks the 'last online' attribute of each patient and compares it to today's date. If the date is within the last 7 or 30 days, the appropriate chart is updated. Both of those charts can be filtered based on the type of patient the user wants to view. The three filter options are all, active only (patients that are in the 'inactive patients' list are excluded) and RKD only (patients who are not in the 'RKD patients' list are excluded). The user can also hover on the different parts of the chart to see the exact number of patients who have and have not used the app.

The three charts at the bottom of the page, two location and one EQ5D, are a preview of some of the charts in the summary reports tab. Since location data is so important in the AVERT project, it is important for the user to see straight away if the patients are

capturing their locations. The user also needs to see whether patients have been completing the EQ5D survey that estimates their quality of life, as it is an important aspect in monitoring the patient's wellbeing.

4.2.2 Patients page

The patients page can be accessed by selecting a patients tab in the navigation bar. It consists of a table that shows the most important patient information to the user. Each row in the table corresponds to a single patient.

There are seven pieces of information that the user sees. The first three cells in each row correspond to the patient's ID number, their RKD status (yes/no), and the date they last opened the app. The user can click on each ID number and will be redirected to the corresponding patient's profile. The last four cells correspond to the patient's EQ5D information. It is not enough to simply show the latest EQ5D value, as that value needs to be put into context. This is achieved by displaying the patient's lowest, highest, current and previous EQ5D values, as it allows the user to see whether the patient's quality of life is improving or not.

I use the API to query the pMp server to obtain the above data for each patient. That data is then displayed in the appropriate row cell, using Django's markup language.

4.2.3 Individual patient profiles

An individual patient profile can be accessed from the dashboard or patients page. It consists of four different features, a section with patient information and three graphs that show engagement levels and health status of the patient.

Due to the data privacy issues discussed in section 3.2, patient profiles do not contain any sensitive personal information. The data displayed is split into two sections, information related to the patient and information regarding the patient's engagement levels. The user can see the dates of when the patient has last recorded their location, completed EQ5D or recorded a journal entry. In the case of EQ5D, its last value is also displayed.

It is very important for the user to clearly see patient's engagement levels. The engagement graph located on the right-hand side of the screen (see Appendix B.4), shows the number of location, EQ5D and journal entries a patient has recorded each week. A Python script checks today's date and separately counts the number of entries for each data type, over a period of four weeks. I have chosen to use a line graph to display this information, as it allows the user to clearly see and analyze the trends of all three types of data at once. In conjunction with the EQ5D graph, the user can analyze whether the patient's quality of life affects the amount of data they record.

Although the primary purpose of this project is to allow the user to monitor patient's engagement levels, it is important to also monitor their wellbeing. The EQ5D graph shows the trend of EQ5D values ever recorded by the patient. Since patients are only expected to complete the EQ5D survey once a month, it would be insufficient to display information only for a certain time period. The values recorded by the patient fluctuate between their lowest and highest value ever recorded. This gives a bit of a context, as to what is the patient's current quality of life status. The x-axis displays dates that correspond to each EQ5D value, however, due to the functionality of Chart.js, the x-axis scale is adjusted in proportion to the graph. This means that if a patient has completed a lot of EQ5D surveys, the result from each survey will be graphed, but not every date will be displayed, thus preventing the graph from being clustered. If the user hovers their mouse over each EQ5D entry, they will be able to see the exact value (to five decimal places) and its corresponding date.

The daily steps graph is another feature that monitors the patient's wellbeing. It displays patient's daily step count from the last seven days in the form of a bar chart. A Python script checks today's date and counts back six days. It then checks whether the patient has recorded their step count within that period and if so, the value is displayed on the graph. The user can hover over each graph entry and view the exact step count without having to estimate it based on y-axis values.

4.2.4 Summary reports

Currently, the website contains a summary report graph for each of the three data types: location, EQ5D, and journaling. Each of those graphs can be accessed by selecting an

appropriate option in the reports tab. All three graphs work on the same basis, where the Python script checks the current week of the year based on today's date and counts back three weeks. It then checks whether patients have recorded their location, EQ5D or journal data within that period and separately counts the number of times each type of data was recorded. For the location summary report, an additional count of patients who captured their location within that period is performed. This count is displayed as a separate graph. Each summary report displays data in the form of a line graph over a four week period.

All summary report graphs can be filtered by patient type. The user has the ability to filter each graph by all or RKD patients (data of patients who are not in the 'RKD patients' list is excluded).

4.2.5 Forgotten password

Since it is the job of the research nurse to set up accounts for all users, the user needs to be able to change their initial password. In order to do that, the user needs to follow the 'Forgot your password?' link located on the login page. The process of changing the password consists of four steps.

Once the user clicks on the forgotten password link, they will be redirected to a different page where they are asked to enter their email address (the same as their username). Django will then check whether that email address is in its database and if so it will redirect the user to a new page where they are informed that an email with further instructions has been sent to them. This email is sent from the admin's email address and will also appear in the admin's 'sent' folder. The user needs to open their email and click on the link provided to reset their password. The link has an expiry time, so if the user doesn't use it within a certain time period, they will have to request a new one. Otherwise, the user will be redirected to a page where they are asked to enter a new password and confirm their new password. There is a set of guidelines displayed (see Appendix B.12) to help the user select a strong password. Once the user is happy with the password they have chosen and click on the confirmation button, they will be redirected to the page that allows them to proceed and login to their account with new credentials.

The above mechanism for resetting user's password is one of Django's built-in functionalities. It saves the software developer from writing code to create that functionality. I altered the visual appearance of Django's default templates by connecting their functionality to my own templates. This can only be achieved due to the Model-View-Template (MVT) architecture of the Django framework.

4.2.6 Admin page

The admin page is also one of Django's built-in functionalities and can be accessed from anywhere on the website via the admin tab on the navigation bar. By default, it contains the ability to add new users and provides admin authentication, so only a user with admin credentials can access this page.

As mentioned earlier, the research nurse (admin) needs to have the ability to maintain 'inactive patients' and 'RKD patients' lists. To incorporate this functionality into the admin page, I have implemented inactive patients and RKD patients models. The admin can add or remove patient ID numbers in both of those lists. The process of creating a new patient entry is quite simple. It consists of one mandatory field, which is the patient's ID number.

The admin has the ability to remove or add new users. This process is quite simple as the only required fields are the username (email address), password and password confirmation. However, by default, the admin also has the ability to add user's name, date of birth and organisation to their account information.

4.3 Django database

Django has many built-in features and its database functionality plays an important role in this project. Implementing a database can be complex and time-consuming, especially for novice software developers. Django's Object Relational Mapping (ORM) function takes care of most of database related work and it encapsulates database tables through models. By default, Django automatically creates an SQLite database for every project. It also has the flexibility to use other relational databases such as MySQL or Oracle, by a simple settings configuration [30].

For the purpose of this project, the research nurse (admin) is the only user who interacts with the database. The admin needs to be able to add new users and maintain lists of 'inactive patients' and 'RKD patients'. Django creates tables in the database through the use of models. By default, it creates a user model with attributes that are shown in figure 4.3. Every time the admin creates a new user, the database will be updated and a new row entry with all the attributes will be added to the table.

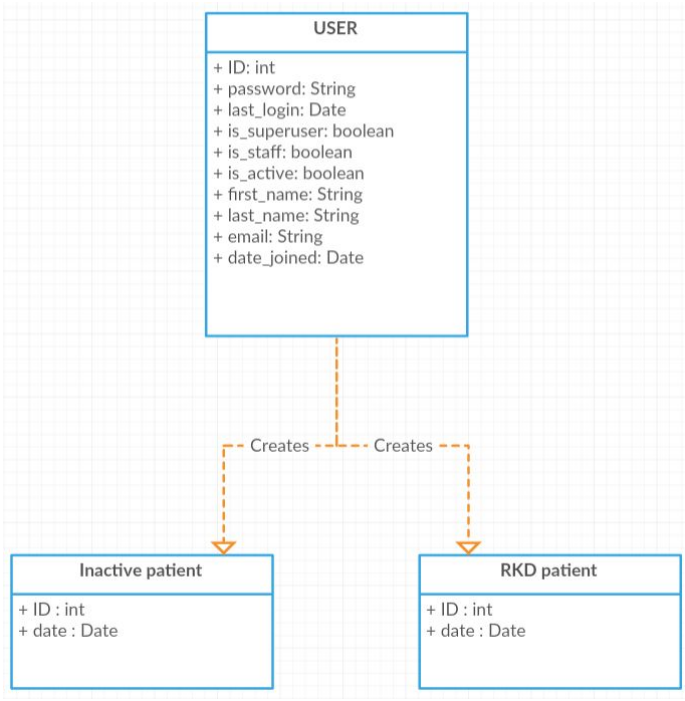


Figure 4.3: Class diagram of models

It is the admin's responsibility to maintain 'inactive patients' and 'RKD patients' lists. These lists essentially contain ID numbers of patients who do not wish to use the pMp mobile app anymore (inactive patients) and ID numbers of patients who are RKD registered (RKD patients). Both of those lists are essential for the correct functionality of the alerts table, summary statistics charts, and summary report graphs.

I have implemented two separate models to cater for this functionality. Each model consists of an ID attribute that only accepts integers as an input and a date on which the ID number was added to the list. The ID number has to be input manually by the admin, however, the date is automatically added. After the models are implemented, a

migration creates corresponding tables in the database. Each time the admin adds a new ID number, a new row entry is added to the table in the database.

4.4 Implementation issues

Every project is faced with an issue at some stage. Issues that occur during the planning or design phase are easier to predict than those that occur during the development phase. Having a good project plan certainly decreases the chances of running into some major issues throughout the project, however, even the best plan doesn't cover all possible scenarios.

Like with every project, I was faced with some issues too. At the early stage in my project, I had noticed that the API documentation provided to me by patientMpower is out of date. The documentation provided an explanation of all endpoints and the relationship between them, however, many concepts are not documented properly and the documentation lacks detail. As a result, I had to look for online resources to clarify certain concepts, such as gaining access to the API. The documentation does not provide the exact method of accessing the pMp API, and this concept was verified by looking at online resources and exchanging emails with the pMp software developer. Another downfall of the documentation is the fact that it does not contain any explanation on how to query the server and retrieve data from an endpoint. Once again, I had to use online resources, which were very limited, to gain knowledge on how to retrieve data from a server using a third party Django API.

As explained earlier, Django uses models to create tables in the database. It creates a default user model with the attributes shown in figure 4.3. To improve the security of the authentication system of this website, I wanted to change username login to email login, as usernames tend to be forgotten. There are numerous ways to achieve this functionality, one of which is creating a new custom user model. Despite a number of different attempts, I was not able to implement a custom user model, as it caused various errors Django was not able to overcome. However, I did overcome this issue by manipulating the functionality of the authentication system.

Most of the implementation issues I was faced with, occurred towards the end of the development phase. I designed the website and implemented its functionality based on the test server provided by pMp. Towards the end of the implementation process, I connected the website to the pMp production server and I was immediately faced with a number of problems.

Firstly, being previously not informed about this, I discovered that from 2019 onwards, patients' location data was no longer stored on the production server. This resulted in limited functionality of the website, as the summary report graph and patient engagement graph are both dependent on the location data. Since the website needs to support real-life patient data, I immediately contacted pMp about this issue. The technicalities needed to use location data on the website are implemented, and all that is needed is the link to where the location data is stored. Due to the limited time I had left to work on this project and the lack of confirmation from patientMpower as to where the location data is stored, currently, the website is not connected to any location data from 2019 onwards.

Secondly, the production server contains inconsistent data, which limits the functionality of the alerts table, summary statistics charts, and patient's profile information. All those features are in some way dependent on the 'last_online' value of the patient object. It turns out that this value is not updated in the production server when the patient is using the pMp mobile app. In most cases, the 'last online' date is the same as 'date joined'. The alerts table uses the 'last online' value to determine whether a patient has used the mobile app in the last 30 days. Due to the 'last online' and 'date joined' dates being the same in many cases, the alerts table contained entries for almost all patients, except for those whose account was created in the last 30 days. Similarly, the summary statistics charts currently only capture activity of patients whose account has been created in the last 30 days, due to the 'last online' value not being updated. The patient information that is displayed on the patient's profile contains patient's 'last online' date, however, since the value is not updated that information is not valid for most patients. I have contacted patientMpower about this issue and currently, they are in the process of fixing it.

Lastly, the production server does not contain information about step count for any patient, as all fields are set to 'null'. As part of the requirements gathered from the client, I have implemented a graph that would display a daily step count for each patient over a seven day period. The functionality of this graph is currently suspended as there is no data available for visualization. I and the client, have addressed pMp with this issue, however, I did not receive a reply from them. This could be an issue with the pMp mobile app not recording this data or it could be a similar situation as with location data where it is simply stored somewhere else. The technical aspect of implementing the correct functionality of this graph is in place and all that is needed is the link to the step count data.

4.5 Chapter summary

This chapter explains what an API is and the steps needed to access and query patientMpower server by the means of an API. Next, it discusses the process of implementing each feature of the user interface, along with explaining the role of Django database. Lastly, we moved onto the issues that arose during the implementation and how they affected the process. Next chapter will provide a detailed evaluation of the user interface, talk about the challenges and limitations of this project as well as the future work.

Chapter 5: Evaluation and Discussion

This chapter provides an evaluation of this project, in relation to the feedback provided by the client and the golden rules that should be followed during a web design process. It also contains a discussion of the challenges and limitations that arose during the implementation process. The chapter finishes with a discussion of possible future work, to improve and add other functionality to this project.

5.1 Evaluation

Website design is a very subjective area and it is impossible to define a set of metrics to say what definitely makes a good website. However, there are some golden rules that must be followed, which include key elements such as visual appearance, accessibility, and usability [31]. The design process plan, that was presented and discussed in Chapter 3, ensured that all of those elements are addressed in the design of this website.

The main objective of this website is to allow the user to get a clear overview of how vasculitis patients are engaging with the patientMpower mobile application. In order to capture that patient engagement, a number of graphs have been implemented to show how many locations and health data is captured cumulatively and also individually by vasculitis patients. The website should also highlight to the user those patients whose engagement levels have dropped recently. The feature that supports this functionality is the alerts table, which displays ID numbers of patients who have not recorded a particular type of data in a specified time period.

This website has been designed to be used by clinical staff. However since it focuses only on vasculitis patients, its main users will be the clinical staff involved in the AVERT project, such as the chief clinician and the research nurse. At the end of the development phase, separate meetings took place with the chief clinician and the research nurse. The purpose of those meetings was to get independent feedback regarding the website's functionality from both future users.

5.1.1 Visual appearance

In terms of the visual appearance of this website, both chief clinician and the research nurse have agreed that the design and colour scheme complement each other and create a neat look. This is a very important aspect in terms of web design, as people have a tendency to use websites more often if they find them visually appealing. Since it is the nurse's responsibility to monitor the engagement of vasculitis patients, it is very important that they find this website appealing.

The major components of the website's design that make it visually appealing to the user are structure and consistency. All the features of this website are neatly aligned on the screen and sufficiently spaced apart to make it easier to click on each feature. The text font is consistent across the whole website. I have chosen to use 'New Times Roman' font, as it improves the readability of the displayed data. The size of the text font varies, as the headlines are significantly larger than normal text. Larger headlines allow the user to quickly find the piece of information they are looking for, as they attract the user's attention.

Colour scheme is another aspect that plays an important role in making a website appealing to the user. I believe that in order to create a neat look, the use of colours has to be well balanced and consistent across the website. The colour scheme I have chosen for my design comes from the AVERT logo (see Figure 3.5). Most of the website background is white, and appropriate colours are used to highlight website features. This creates a nice contrast and makes all the features stand out to the user.

5.1.2 Accessibility

One of the key attributes of this website is its user-focused navigation system. During a meeting with the research nurse, she pointed out that the good navigation system is kept up throughout the whole design. The user knows at all times where they are on the website and it is easy to go back between different pages. The navigation ensures that if the user knows what they want, they can go ahead and do it with ease.

The main pages that the research nurse will be interested in, once she signs in and sees engagement levels of patients, are the individual patient profiles and summary reports.

The nurse can quickly and easily navigate to those pages as they are accessible from the home page in a number of ways. Individual patient profiles can be accessed by clicking on an ID number in the alerts table or by searching a patient ID in the search bar. Navigation bar located at the top of each web page also has the ability to bring the user to the individual patient profiles (patients tab) and summary reports (reports tab).

Another important aspect is the accessibility of the admin page. The authentication system in place allows only the research nurse (admin) to access the admin page. Due to this, the link to that page is incorporated in the navigation bar and is one click away from anywhere on the website. If a different user tries to access the admin page, they will be asked to provide admin credentials or go back to the home page.

5.1.3 Usability

Usability includes proper functionality of the website such as user-friendly, security and speed. During the meeting with the research nurse, she found the website very intuitive and user-friendly. While hovering the mouse over clickable features, the colour of those features changes, thus signalling to the user that if they click on the feature some action will be performed. Autocompletion of values in the search bar also adds to the user's experience. It gives the user a list of possible values based on their current input and prevents the user from forgetting the value they are looking for or searching a value that doesn't exist in the database. Both of those features are not part of the requirements but have been implemented for the sole purpose of making the website more user-friendly.

The research nurse also pointed out that all of the security protocols in place make the website safe to process patient data. Django's built-in protocols discussed in section 3.2, protect the website from malicious attacks and authentication system in place ensures that only authorized users are permitted to access the website. Additional authentication for access of the admin page ensures that no one but the admin has the ability to add or delete users and maintain inactive patients and RKD patients lists.

In terms of speed, the website is able to handle patient information in real-time, which means that when the pMp production server is updated with new patient information,

the information on the website will update almost immediately. This ensures that the research nurse has access to real-time patient data at all times.

Overall, both future users have expressed their satisfaction with the functionality and visual appearance of the website. All the requirements have been incorporated into the design and the website does everything that the client expects it to do. On top of that, it ensures security in terms of user authentication and processing of data.

5.2 Challenges and limitations

As mentioned in section 4.5, there have been a number of issues with using the pMp production server. This project was implemented using a test server provided by patientMpower. When the implementation process was finished, I connected the website to the pMp production server and I was immediately faced with two major problems.

The first problem being, lack of step count and location data from 2019. This limited the functionality of several website features. The second problem is the inconsistency of patient data, which also limited the functionality of several website features (see section 4.4 for more detail).

Due to the nature of my client's work, hosting meetings at times that suits both parties was also a challenge. As mentioned in section 3.1.3, the client was not able to meet face to face to evaluate the initial set of prototypes and an alternative arrangement had to be made. Despite this, the objective of that meeting was reached, the screenshots of the high fidelity prototypes and their explanation, provided enough information to the client to evaluated and provide me with a lot of useful feedback.

Despite those challenges, the remaining aspects of the website work as intended. After the work is carried out by pMp to fix the issues with their production server, the functionality of the website will be correct.

5.3 Future work

Although this project has met all of the objectives set out in Chapter 1, there are a number of ways in which it could be improved.

Data privacy has played an important role in this project, and a number of potential functionalities had to be overlooked in order to ensure that no data privacy issues are raised.
Currently, the research nurse, who is responsible for monitoring vasculitis patients, has a spreadsheet that links an anonymous patient (by using their ID number) to their personal information. Since the spreadsheet is encrypted and stored on the nurse's computer, no one but the nurse has access to it. Due to the fact that this project is prioritising data security, there is no link to the patient's personal information on the website, and the nurse has to continue to use the spreadsheet. The efficiency of the nurse's work would increase if she was able to have both, patients' engagement and health data, linked to their personal information. This would avoid the hassle of looking up patient ID in the spreadsheet to obtain the patient's name and email address.

A possible approach to this would involve uploading the nurse's spreadsheet to a secure server (separate from the website server), to store patients' personal data. A secure connection would have to be established between the two servers, and the personal data would have to be encrypted. By allowing an authorized user to upload patients' personal data to the website for the duration of their session, all of the patient data would be temporarily in one place, eliminating the need to go back and forth between the website and spreadsheet.

As another possible extension to the project, if the patient's personal details were linked to this system, there could be a feature to automatically send emails to patients with low engagement levels, encouraging them to engage more with the pMp mobile application. This would decrease the nurse's workload and the work that was previously done manually would now be automated. The research nurse would be left with the responsibility to do spot checks to ensure that emails are sent to correct patients and keep updating the database of patients registered as RKD or inactive.

At the moment, the graphs that show patient's engagement levels and health are only showing information as instructed by the client, mostly going back only one month from the current date. However, it would be a useful feature to be able to scroll back on the graphs to see a trend for each patient over a couple of months rather than just over one month. This feature would be particularly useful for patients with a flare, as it would allow better analysis of the patient's data during the period that leads to the flare. Future work on this project should involve modification of engagement and health graphs, to display information over a longer period of time.

Before this project will have a real-world usage, it firsts needs to be hosted on a secure server, rather than on a local machine. There have been some issues as to where the website should be hosted due to the sensitive information that it is linked to. The future work on this project will have to involve finding a secure server to host the website and performing regular maintenance.

5.4 Chapter summary

This chapter demonstrates the evaluation process by highlighting how the website appears to the user and the actions taken to make the design appealing to the clinical staff. Next, it discusses the challenges I was faced with after the implementation process was finished and the website was connected to the production server. It also discusses the potential work that can be done in the future to improve its functionality. Hosting the website on a secure server and connecting it to the location data are the necessary works that have to be done before this website will have a real-world usage. Additional features such as extending graphs and linking information from the nurse's spreadsheet in a secure manner would add to the user experience.

Chapter 6: Conclusion

This chapter provides the conclusion of this project in relation to the objectives set out in Chapter 1.

6.1 Meeting the Objectives

This project is a fully developed website which includes a dashboard and alerts table for a clear overview of patient engagement and visualizes patient data recorded by the pMp mobile application. The website will be used by a research nurse who is responsible for monitoring vasculitis patients and encouraging them to record their health and location data. That data will be analyzed by the AVERT researchers in order to discover triggers responsible for the flare of ANCA vasculitis.

The objectives that have been listed in Chapter 1 have all been achieved. These objectives are:

- **Objective 1:** Find an appropriate web framework that will provide a secure environment for handling real-time patient data.

  Chapter 2 introduces web frameworks most commonly used for web development and lists both advantages and disadvantages of each web framework. After careful a consideration of each web framework, I have decided to use Django for the development of this project.

  There are a number of reasons for choosing the Django web framework. The main reason being, its built-in security protocols, and scalability. Since the website handles sensitive patient data (health and location data), having strong security is mandatory. With all the built-in protocols that Django offers, this reduced my workload and I was able to use the extra time on other aspects of the website. As the number of patients grows over time and a need arises for additional functionalities, the website can be easily scaled. Python syntax, carefully commented code and the use of templates, allow easy addition or extension of features.

The disadvantages of Django that are highlighted in Chapter 2, do not affect the functionality of this project in any way, thus making Django the ideal framework to use.

- **Objective 2:** Gather requirements from the client (the research nurse).

  Chapter 3 explains the process of designing this website. The meeting that took place between me and the research nurse regarding the gathering of the requirements, was the first and most important step of the design process. It gave me an idea early on in the project, as to what the basic functionality of the website should be.

- **Objective 3:** Design an intuitive and easy to use user interface, for patientMpower database, based on the client's requirements.

  Chapter 3 explains the process of designing a user interface. After collecting requirements from the client, I designed the initial set of prototypes. The creation of prototypes and their evaluation by the client provided me with very important feedback. Some necessary changes had to be made regarding both, the functionality and visual appearance of the website. The last meeting that took place with the research nurse provided positive feedback regarding the intuitive nature of the website. The nurse was able to navigate the website with ease, she found all the features easily accessible and the visual appearance was appealing.

  Due to a lack of deep technical knowledge of my client, intuitive and easy to use user interface was the priority. The research nurse was overall very satisfied with the end product.

- **Objective 4:** Query patientMpower database to retrieve useful patient information and present it in a neat and practical way.

  Chapter 4 describes the process of accessing and querying patientMpower server. Despite poor API documentation, I have successfully retrieved patient data from the pMp production server. The patient data is retrieved from different

endpoints by the View component of Django's architecture and is passed to the Template component, where it can be visualized.

Chapter 4 also provides a description of how patient data is visualized. After careful consideration of the advantages and disadvantages of each library, I have chosen to use Chart.js to graph the data. Despite the limited number of graphs, this library offers excellent functionality. Due to the limited time I had to work on this project, ease of use and clear documentation are compelling advantages that played an important role in my choice of using Chart.js. The library produces clear and responsive graphs that are ideal for analyzing trends in patient's health and location data.

- **Objective 5:** Design a notification system that allows researchers to easily identify patients who need their immediate attention.

  The notification system needs to be easily accessible, thus making the dashboard page its ideal location. Chapter 4 describes the process of implementing the user interface and explains the reasons behind different functionalities of the alerts table. An alert is created for every patient who lacks in engagement with the patientMpower mobile app. Each alert is also colour coded to help the research nurse distinguish between patients who need immediate attention. The research nurse also has the ability to delete alert entries after she has contacted corresponding patients.

  The alerts table provides a clear overview of patients who haven't been using the pMp mobile app and allows the research nurse to identify with ease, which patients need to be contacted.

6.2 Final Thoughts

To conclude, this project has been a success. It includes all the functionality desired by the client and also ensures the security of the patient data. Once it is hosted on a secure server, the website will be used by the research nurse to help with the AVERT project.

Bibliography

[1] nhlbi.nih.gov, (2017). Vasculitis. [online] Available at: https://www.nhlbi.nih.gov/health-topics/vasculitis.

[2] Eustice, C (2019). What is Inflammation?. [online] verywellhealth.com. Available at: https://www.verywellhealth.com/what-is-inflammation-187934.

[3] Sampson, S (2019). Autoimmune Diseases: Types, Symptoms, Causes and More. [online] healthline.com. Available at: https://www.healthline.com/health/autoimmune-disorders.

[4] Devlin, J. and Brooks, R. (2017). EQ-5D and the EuroQol Group: Past, Present and Future. Appl Health Econ Health Policy, 15(2), pp. 127-137.

[5] EQ-5D homepage (2019). Available at: https://euroqol.org/eq-5d-instruments/.

[6] AVERT home page, https://www.tcd.ie/medicine/thkc/avert/index.php/.

[7] Engagement. (n.d.) In Cambridge Dictionary. Available at: https://dictionary.cambridge.org/dictionary/english/engagement.

[8] Eisenberg, H. (2014). Humans Process Visual Data Better. [online] t-sciences.com. Available at: http://www.t-sciences.com/news/humans-process-visual-data-better.

[9] Bowlby, S. (2014). 6 Phases of the Web Site Design and Development Process. [online] idesignstudios.com. Available at: https://www.idesignstudios.com/web-design/phases-web-design-development-process/.

[10] Website Design Phases Phases Of The Web Site Design And Development Process Entero Download. [image] Available at: http://make-shop.info/website-design-phases/website-design-phases-website-design-stages-6-essential-phases-for-a-flawless-web-design-free/.

[11] Frey, R. (2010). The model, view, and controller (MVC) pattern relative to the user. [image] Available at: https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller.

[12] Eising, P. (2017). What exactly IS an API?. [online] medium.com. Available at: https://medium.com/@perrysetgo/what-exactly-is-an-api-69f36968a41f.

[13] (n.d.). Token Based Authentication Made Easy. [online] auth0.com. Available at: https://auth0.com/learn/token-based-authentication-made-easy/.

[14] Wasson, M. (2014). Local Login Credential Flow. [image] Available at: https://docs.microsoft.com/en-us/aspnet/web-api/overview/security/individual-accounts-in-web-api.

[15] Carr, J. (2001). Are quality of life measures patient centred?. BMJ, 322(7298), pp:1357–1360.

[16] Ryabtsev, A. (n.d.). Web Frameworks: How to Get Started. [online] djangostars.com. Available at: https://djangostars.com/blog/what-is-a-web-framework/.

[17] Maurya, S. (2016). Top 10 Apps and Websites Developed with AngularJS. [online] mobiloitte.com. Available at: https://www.mobiloitte.com/blog/top-10-apps-websites-developed-angularjs/.

[18] Rajput, M. (2016). The pros and cons of choosing AngularJS. [online] jaxenter.com. Available at: https://jaxenter.com/the-pros-and-cons-of-choosing-angularjs-124850.html.

[19] Yevtushenko, A. and Yalanska, M. (n.d.). Ruby on Rails Development: Benefits and Pitfalls. [online] tubikstudio.com. Available at: https://tubikstudio.com/ruby-on-rails-development-benefits-and-pitfalls/.

[20] Morgunov, A. (n.d.). Pros and Cons of Ruby on Rails. [online] sloboda-studio.com. Available at: https://sloboda-studio.com/blog/pros-and-cons-of-ruby-on-rails/.

[21] Zublenko, E. and Beklemyshleva, A. (n.d.). “Why Django is the Best Web Framework for Your Project”. [online] steelkiwi.com. Available at: https://steelkiwi.com/blog/why-django-best-web-framework-your-project/.

[22] Hansen, S. (2017). Advantages and Disadvantages of Django. [online] hackernoon.com. Available at: https://hackernoon.com/advantages-and-disadvantages-of-django-499b1e20a2c5.

[23] Advantages and Disadvantages of Django to Consider Before Use (2018). [online] probytes.net. Available at: https://www.probytes.net/blog/advantages-disadvantages-django/.

[24] Irvin, L. (2018). The GDPR: What is sensitive personal data?. [online] itgovernance.eu. Available at: https://www.itgovernance.eu/blog/en/the-gdpr-what-is-sensitive-personal-data.

[25] Skau, D. (2013). Why D3.js is So Great for Data Visualization. [online] visual.ly. Available at: https://visual.ly/blog/why-d3-js-is-so-great-for-data-visualization/.

[26] Iwanska, K. (2017). Data Visualisation With D3.js In Your Application. [online] netguru.com. Available at: https://www.netguru.com/blog/data-visualisation-with-d3.js-in-your-application.

[27] Highcharts REVIEW (2019). [online] reviews.financesonline.com. Available at: https://reviews.financesonline.com/p/highcharts/.

[28] D3.js or Chart.js: Detailed comparison (2017). [online] navyuginfo.com. Available at:
https://navyuginfo.com/d3-js-chartjs-use/.

[29] (n.d.). Prototyping. [online] usability.gov. Available at: https://www.usability.gov/how-to-and-tools/methods/prototyping.html.

[30] (n.d.). Set up a database for a Django project. [online] webforefront.com. Available at:

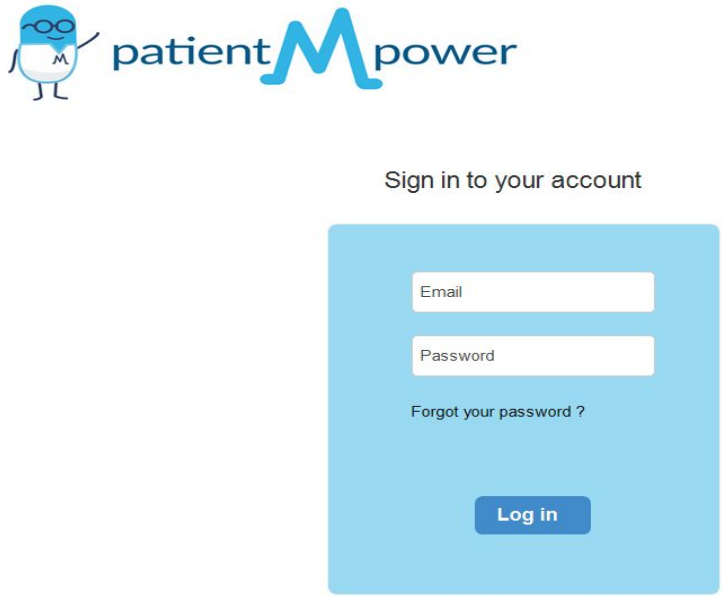
https://www.webforefront.com/django/setupdjangodatabase.html.

[31] Aum (2017). How To Evaluate Quality Of Your Web Design?. [online] nirmal.com.au. Available at: https://www.nirmal.com.au/how-to-evaluate-quality-of-your-web-design/.

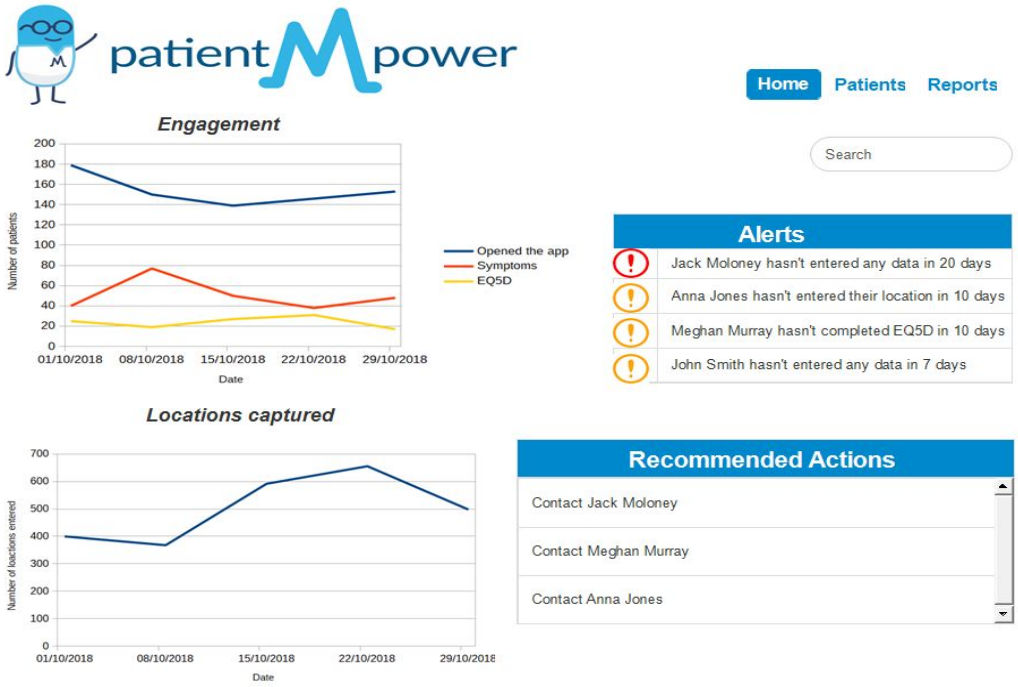
[32] Eising, P. (2017). An API is not a database. It is an access point to an app that can access a database. [image] Available at: https://medium.com/@perrysetgo/what-exactly-is-an-api-69f36968a41f.

Appendix A: Initial User Interface Prototypes

A.1 Login page



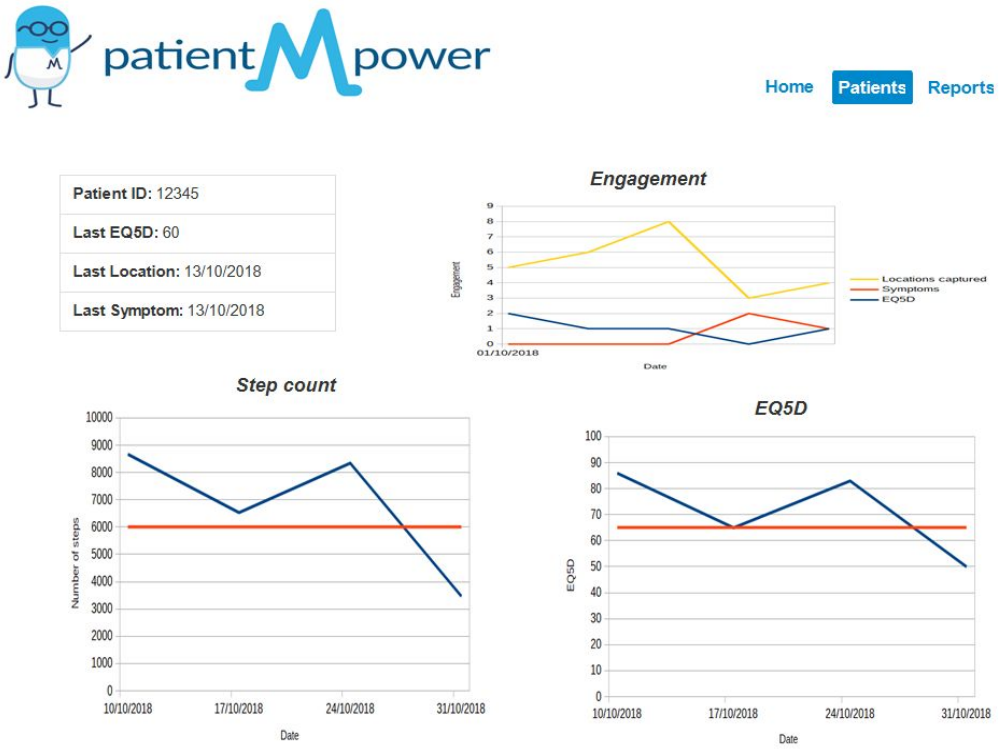
A.2 Home page



Home Patients Reports

Patient	Last opened the app	Change in EQ5D
pateint_ID	03/11/18	10
pateint_ID	03/11/18	15
pateint_ID	02/11/18	4
pateint_ID	30/10/18	7
pateint_ID	30/10/18	5
pateint_ID	27/10/18	20
pateint_ID	27/10/18	34
pateint_ID	27/10/18	12
pateint_ID	23/10/18	8
pateint_ID	15/10/18	4
pateint_ID	11/10/18	9
pateint_ID	03/10/18	17

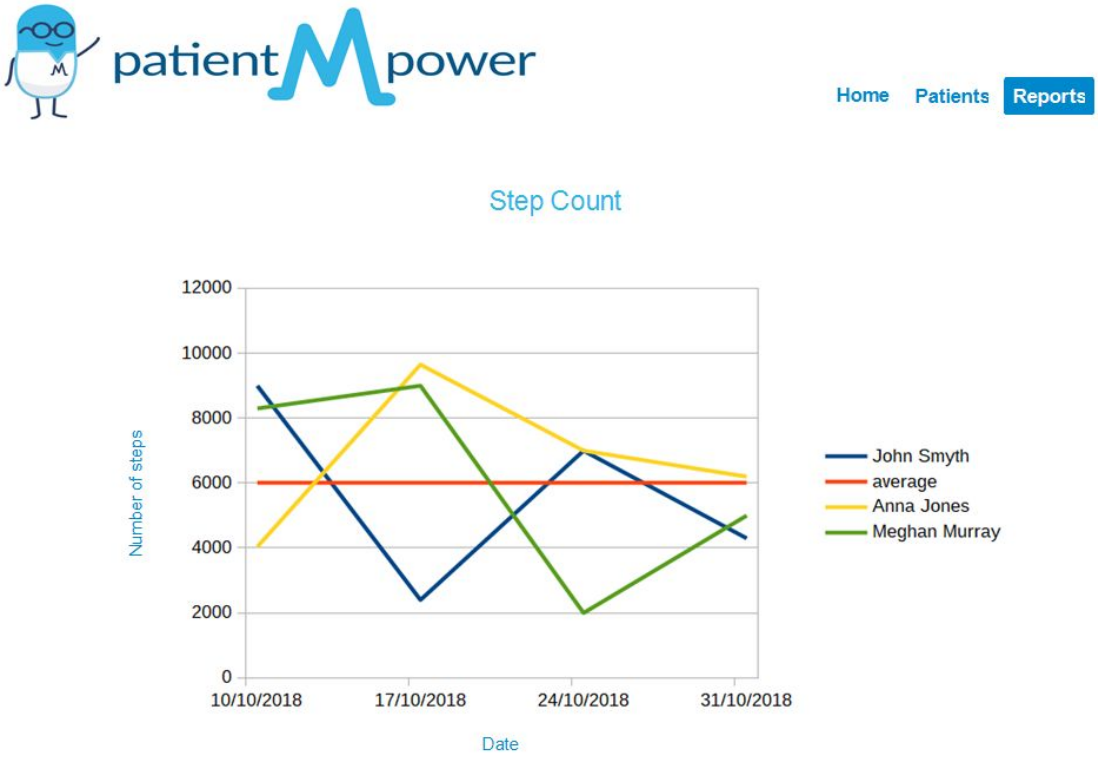
A.3 Patients page



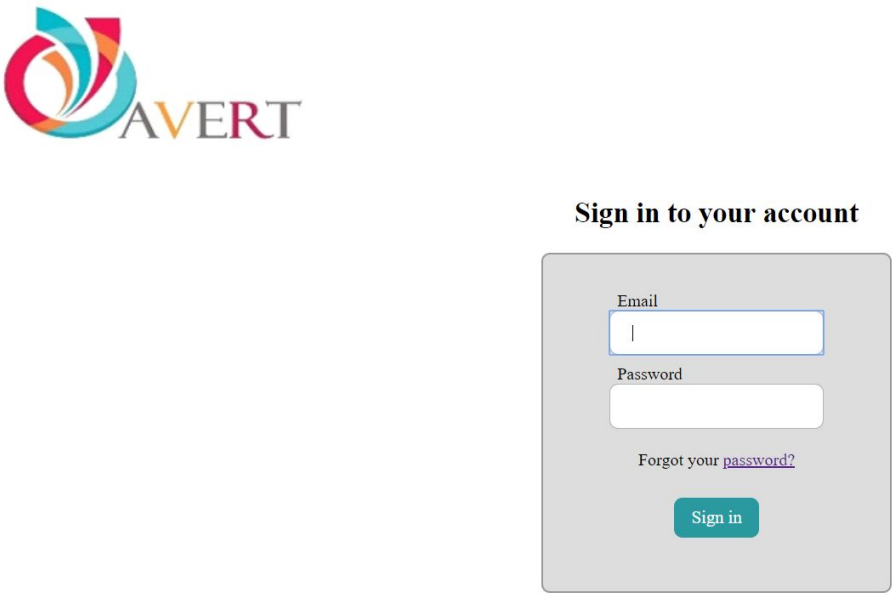
A.4 Individual patient page

A.5 Reports page

A.6 Individual report page

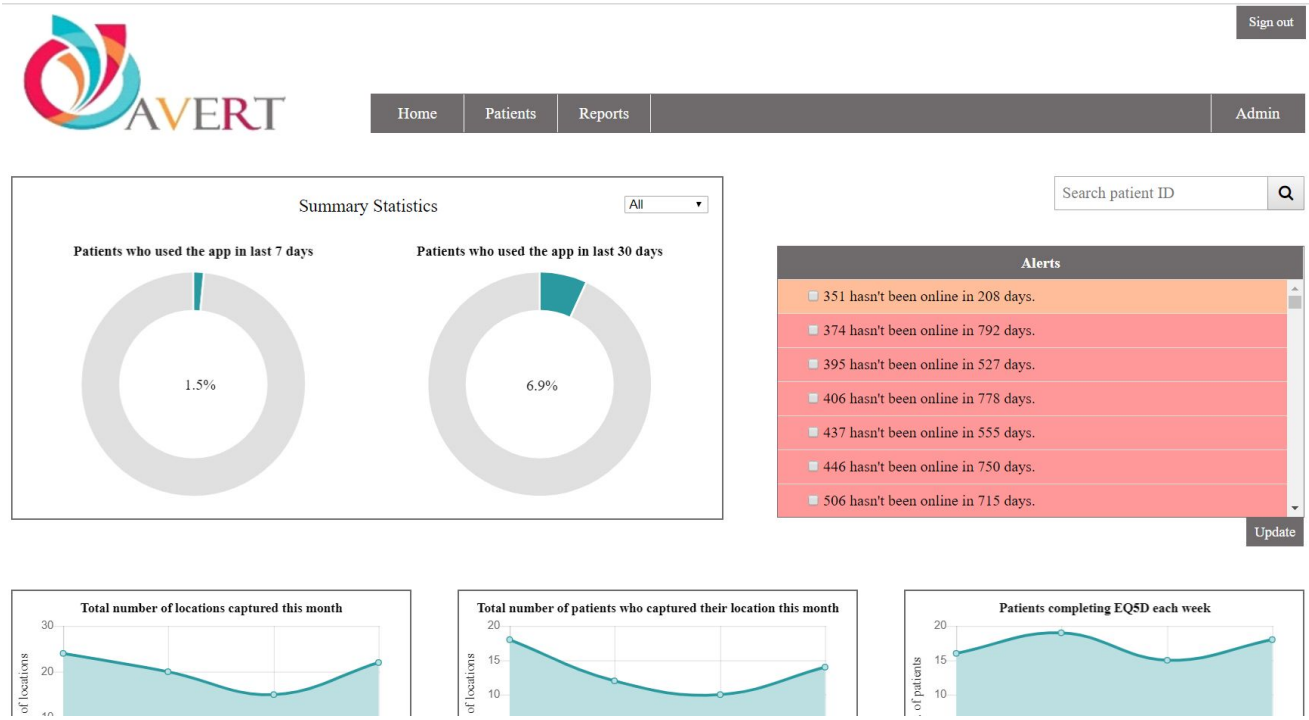


Appendix B: *Final Design of User Interface*

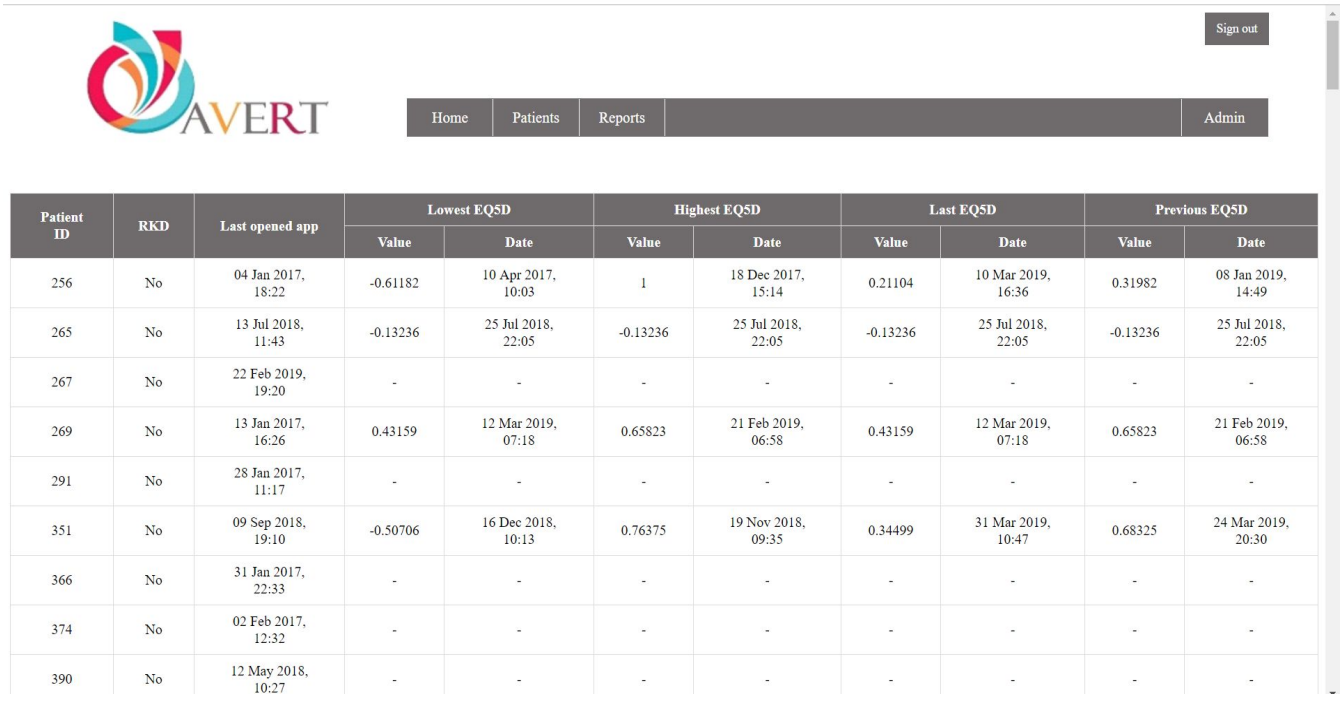


B.1 Login page

B.2 Dashboard page

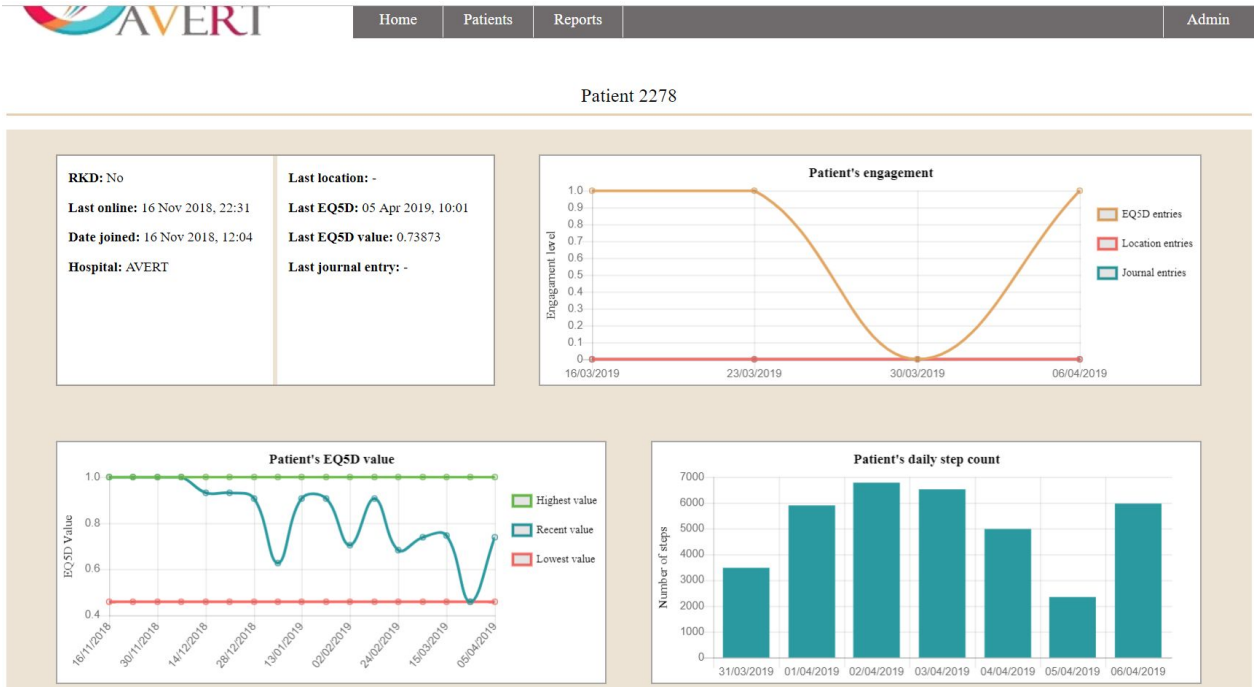


B.3 Patients page

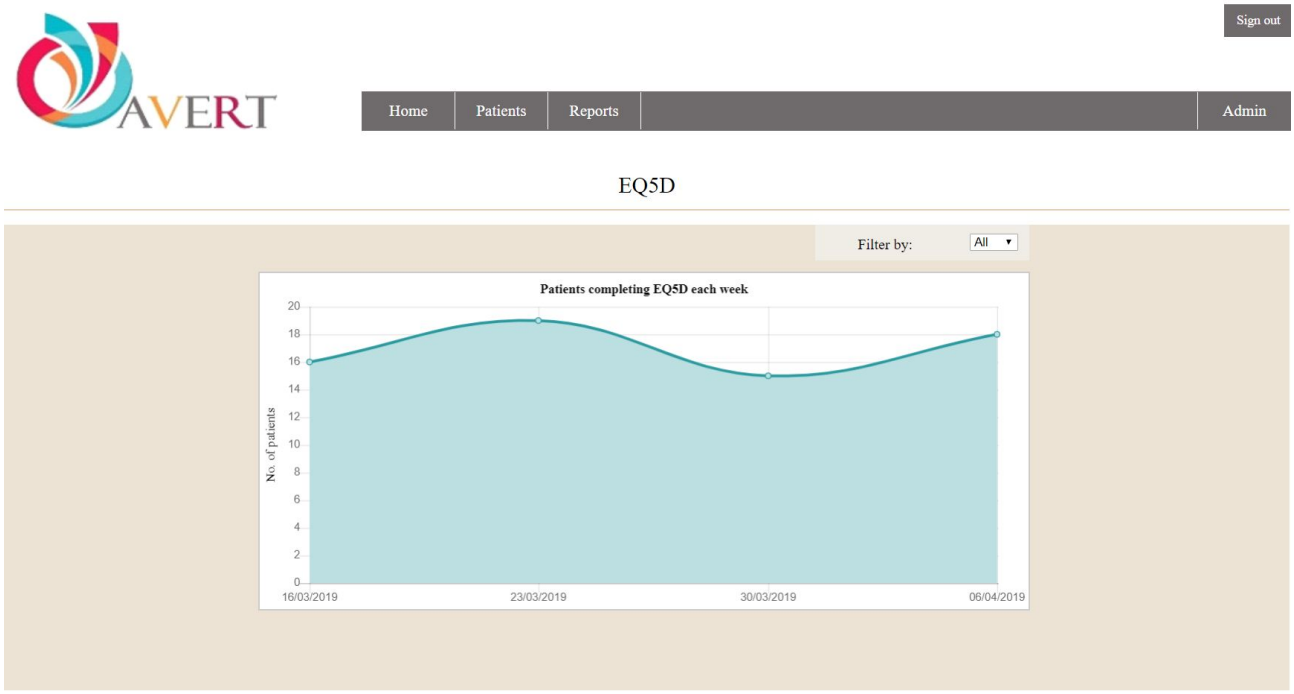


Patient ID	RKD	Last opened app	Lowest EQ5D		Highest EQ5D		Last EQ5D		Previous EQ5D	
			Value	Date	Value	Date	Value	Date	Value	Date
256	No	04 Jan 2017, 18:22	-0.61182	10 Apr 2017, 10:03	1	18 Dec 2017, 15:14	0.21104	10 Mar 2019, 16:36	0.31982	08 Jan 2019, 14:49
265	No	13 Jul 2018, 11:43	-0.13236	25 Jul 2018, 22:05	-0.13236	25 Jul 2018, 22:05	-0.13236	25 Jul 2018, 22:05	-0.13236	25 Jul 2018, 22:05
267	No	22 Feb 2019, 19:20	-	-	-	-	-	-	-	-
269	No	13 Jan 2017, 16:26	0.43159	12 Mar 2019, 07:18	0.65823	21 Feb 2019, 06:58	0.43159	12 Mar 2019, 07:18	0.65823	21 Feb 2019, 06:58
291	No	28 Jan 2017, 11:17	-	-	-	-	-	-	-	-
351	No	09 Sep 2018, 19:10	-0.50706	16 Dec 2018, 10:13	0.76375	19 Nov 2018, 09:35	0.34499	31 Mar 2019, 10:47	0.68325	24 Mar 2019, 20:30
366	No	31 Jan 2017, 22:33	-	-	-	-	-	-	-	-
374	No	02 Feb 2017, 12:32	-	-	-	-	-	-	-	-
390	No	12 May 2018, 10:27	-	-	-	-	-	-	-	-

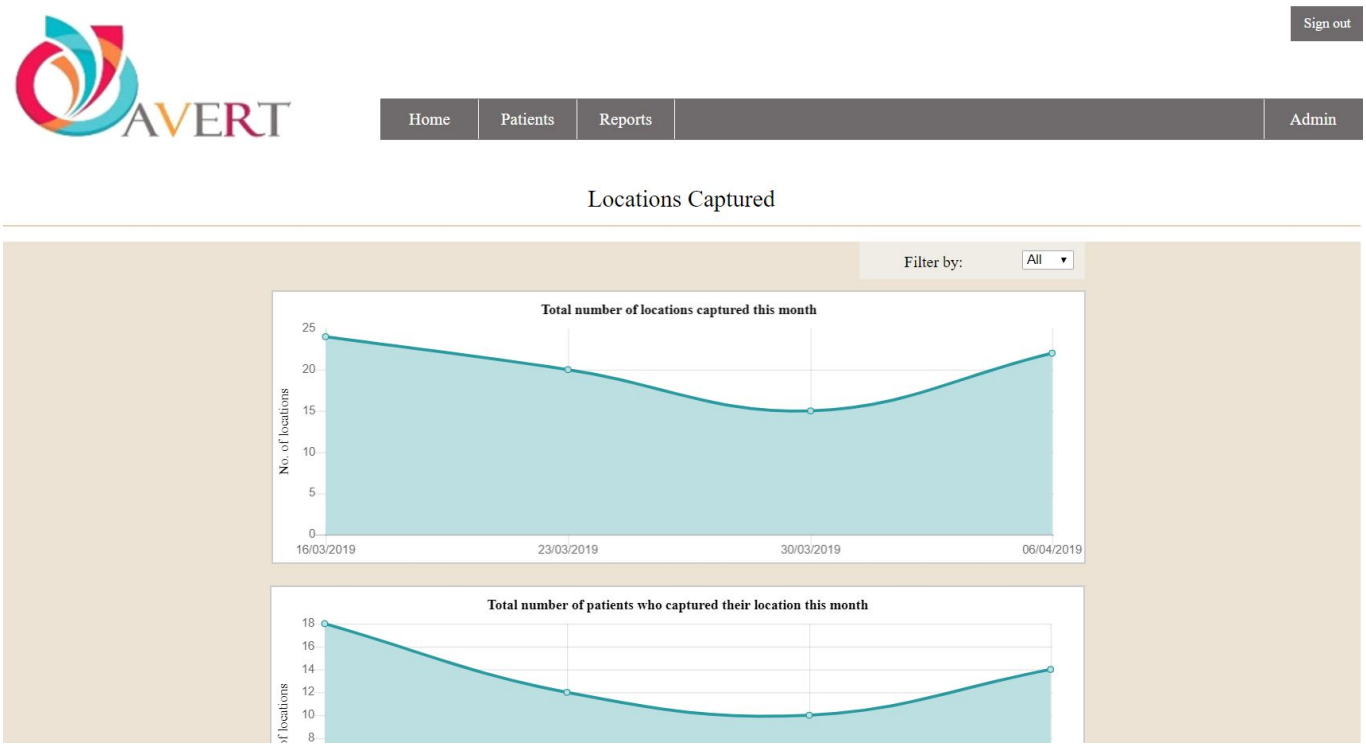
B.4 Individual patient page



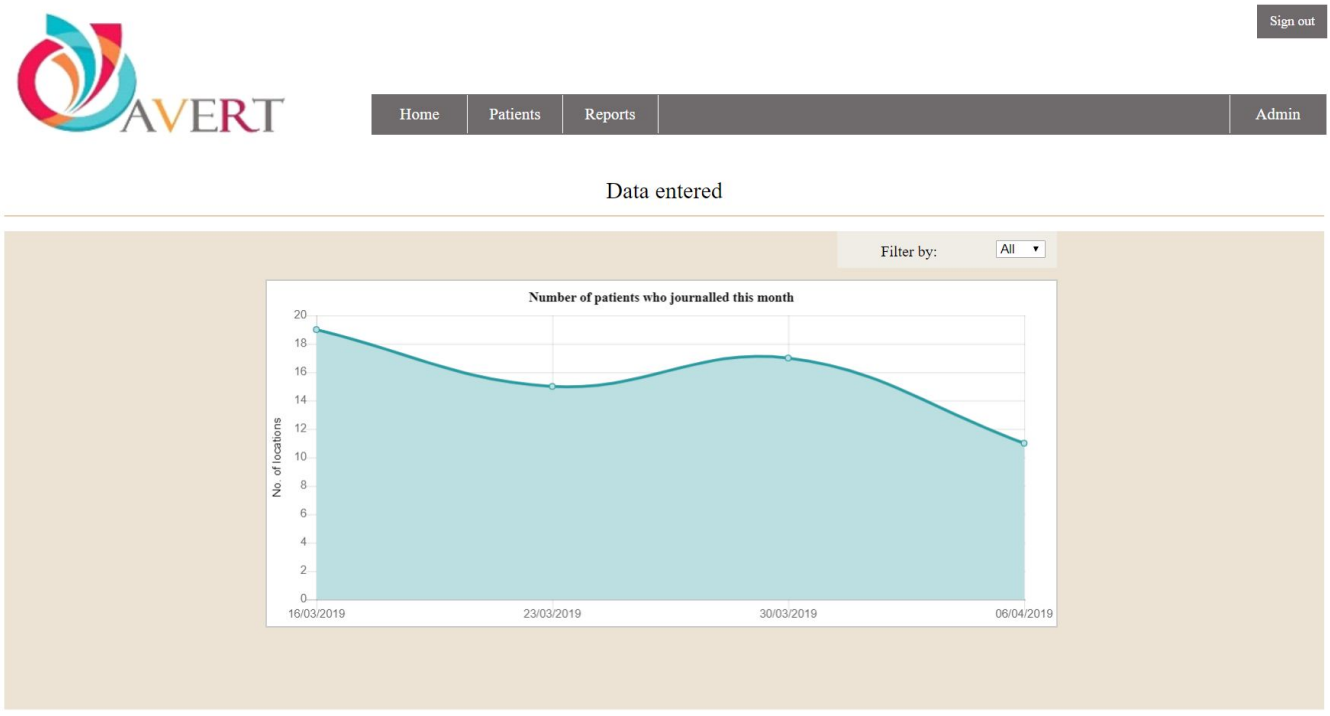
B.5 EQ5D report



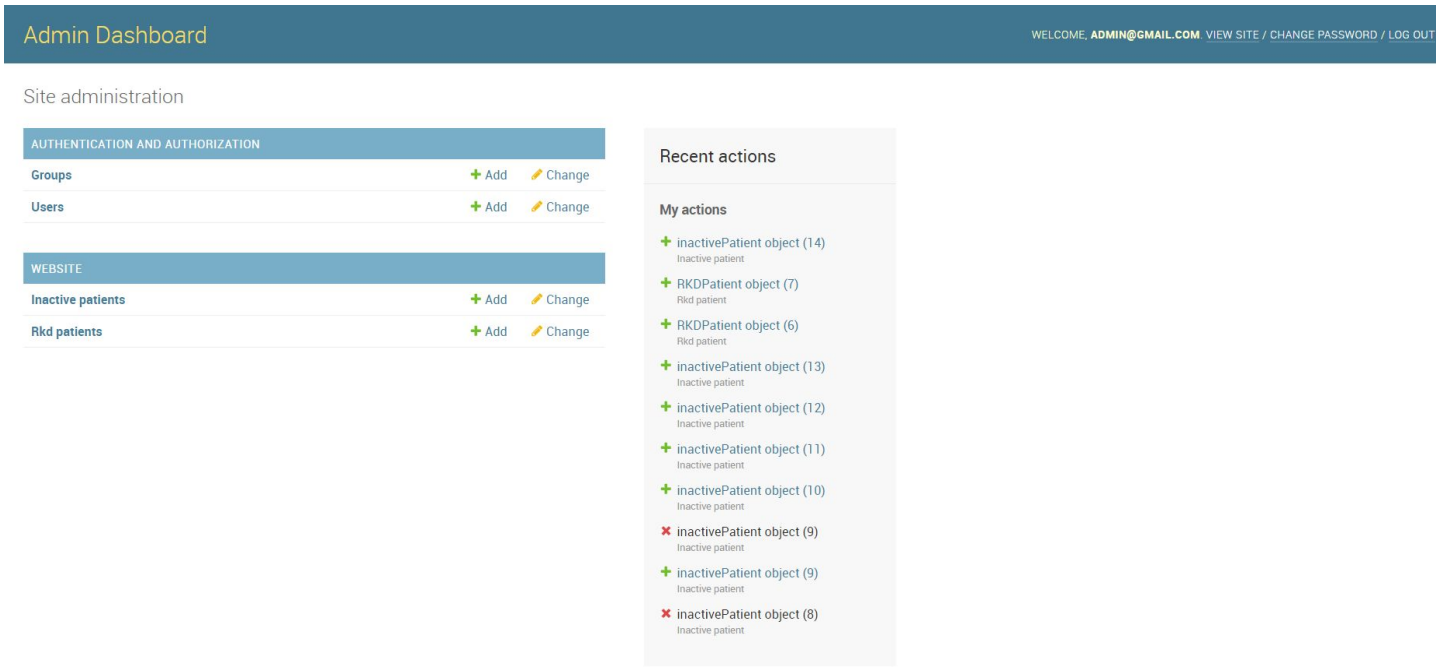
B.6 Location report



B.7 Journalling report

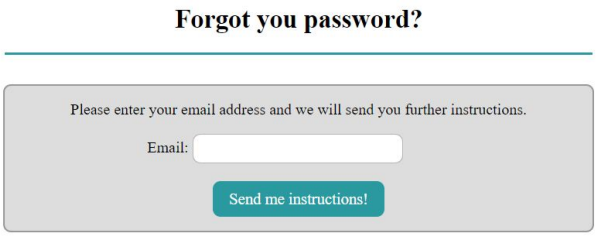


B.8 Admin page

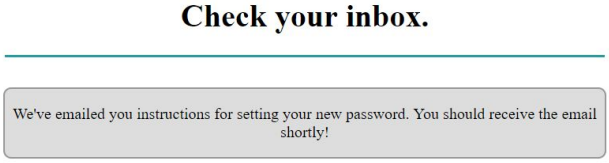


B.9 Logout page

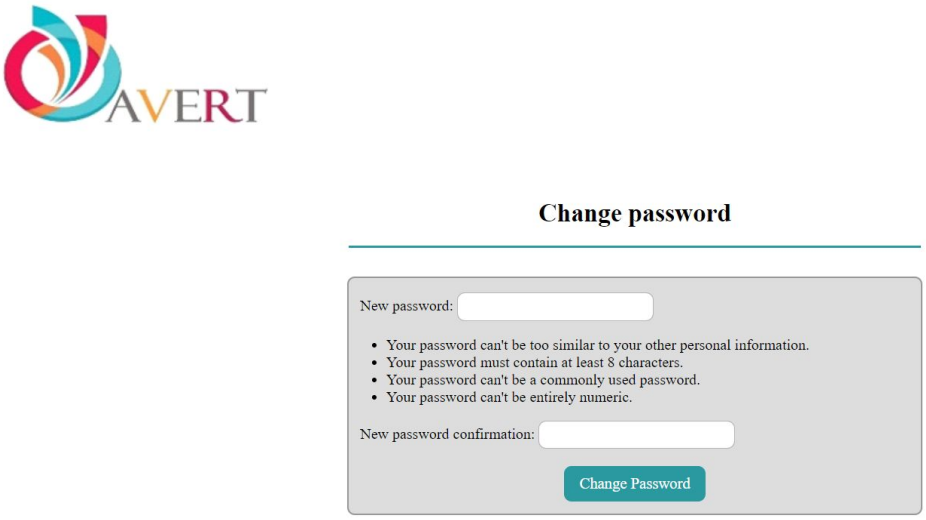
B.10 Reset password



B.11 Password reset done



B.12 Reset password confirm



B.13 Reset password complete