Contents

[Introduction 2](#_Toc160617869)

[Design 2](#_Toc160617870)

[Legalities 2](#_Toc160617871)

[Project Proposal 3](#_Toc160617872)

[UI/UX Designs 3](#_Toc160617873)

[Requirements 5](#_Toc160617874)

[Software Requirements 5](#_Toc160617875)

[Client Side 5](#_Toc160617876)

[Server Side 5](#_Toc160617877)

[Hardware Requirements 6](#_Toc160617878)

[Client Side 6](#_Toc160617879)

[Server Side 6](#_Toc160617880)

[Functional Requirements 6](#_Toc160617881)

[Non-Functional Requirements 8](#_Toc160617882)

[Project 10](#_Toc160617883)

[Project Structure (MVC) 10](#_Toc160617884)

[Project Risks 10](#_Toc160617885)

[Project Security 12](#_Toc160617886)

[Project Methodology 12](#_Toc160617887)

[Business Context 12](#_Toc160617888)

[Uses in a Business Context 12](#_Toc160617889)

[Key Performance Indicators (KPIs) 12](#_Toc160617890)

[User Acceptance Criteria 13](#_Toc160617891)

[Empathy Map 13](#_Toc160617892)

[Data 13](#_Toc160617893)

[Data Map 14](#_Toc160617894)

[Data Requirements 15](#_Toc160617895)

[Testing 17](#_Toc160617896)

[Test and Data Types 17](#_Toc160617897)

[Testing Strategy 18](#_Toc160617898)

[Testing Log 19](#_Toc160617899)

[Feedback Approaches 23](#_Toc160617900)

[Appendix 23](#_Toc160617901)

# Introduction

I have recently been tasked with planning a digital solution for Health Advice Group. They already provide users with:

* advice regarding extreme weather
* information about environmental conditions and allergies
* risk assessments for environments

The digital solution requires that:

* users are provided with a weather forecasting feature.
* users have access to a dashboard for monitoring air quality.
* users can look for advice for dealing with related health matters.

Finally, the client has requested some potential features which could be in the solution:

* health advice based on location.
* accessibility features.
* health tracking tools.

This documentation is designed to outline how I would suggest implementing the solution, the requirements and metrics which would measure the success of the project, and how we will comply with legislation and guidelines.

# Design

In this section I will be discussing several factors which will impact the design choices of our application and how we plan to comply or address these factors.

## Legalities

General Data Protection Regulations (GDPR) / Data Protection Act (DPA) – The GDPR and DPA are pieces of legislation which are designed to safeguard personal data and ensure that the user (data subject) is in control of what data is collected. The legislation highlights that personal data must be relevant and limited, it must be collected fairly, and it must be kept secure. To achieve compliance with these regulations, we have mapped out all the data which we intend to collect and will make use of encryption and individualised accounts to ensure that only the data subject has access to their personal information.

Intellectual Property Act (IPA) – The IPA is a piece of legislation used to protect other people’s work and property from unfair use. It can protect a wide range of property from patterns, to features, to assets, and it enables the owner may take legal action should they be used unfairly or not in accordance with their license. This means that it is fundamental for us to check the license of assets and that they are available for use before we add them to our project.

Equality Act – The Equality Act is a piece of legislation which ensures everyone has equal access to a service. As up to 10% of the internet has accessibility needs, it is very important that we consider how to implement features which allow users with disabilities or impairments to access this service. Should we not implement such features, it is possible that we will be falling liable under this act, as we will not be providing equal service and opportunities to make use of this service and its respective resources. To ensure compliance, we will make use of the Web Content Accessibility Guidelines which are a set of guidelines which help us to make our service more accessible to those who cannot access it conventionally. To achieve compliance with this, we will be making use of the W3C validator, which will look through the HTML, CSS, and JavaScript of any given page and check for compliance with these guidelines.

## Project Proposal

My proposal for Health Advice Group is an easily accessible web application, which allows users to access forecasts for weather and air quality in addition to advice for any health matters related to weather and environmental issues, and management to update and add new pieces of advice.

To enable the provision of weather forecasting, I will be making a webpage which utilises a weather forecasting API. The user’s latitude and longitude will be passed to the API and the current weather and conditions will be returned. This will allow the user to make health decisions, such as whether it is safe for them to go out. Additionally, should I have time, I will also return advice articles which have a similar temperature to the current weather.

For air quality forecasting, I will also be utilising an air pollution API. The user’s latitude and longitude will be passed to the API, and the concentrations for each pollutant in the air will be returned, in addition to an overall air quality index (1-5). I will map each one of these concentrations and using the air quality index for them, I will return the air quality for each of them (i.e. good, poor, very poor) and if I have time, I will map this onto a graph to compare the concentration of each pollutant.

To provide advice, I will create a role system, in which management will be able to add new advice for users to see. Users will be able to search for new advice and be able to save any advice which they can access on a separate page. I have implemented it this way due to the ease of use – it only takes one button to view any relevant advice.

To administer personal health advice, I will be incorporating a location system into the forecasting features which gets the user’s location to return relevant temperatures and air quality. This allows the user to make more accurate decisions and means that the user does not need to manually enter coordinates or location data.

In order to make a personal health tracking tool, I plan to add a system similar to a diary, in which users will be able to log the number of calories, steps, and water that a user has had. This allows a user to easily log and keep track of metrics related to their activity, hydration, and eating habits.

## UI/UX Designs

It is important that the website is designed with ease of use and accessibility in mind. This is because the audience which uses Health Advice Group’s current solutions are not likely to be technical, and hence they may be frustrated or seek alternative solutions if our website does not incorporate these principals. Hence, it is fundamental that the website is accessible to users of any skill level or experience with the internet.

We will base the UI off any corporate branding, colour schemes, or resources which Health Advice Group send us. However, as of right now, we have not received any resources from Health Advice Group tutoring. In the meantime, we will either use placeholder assets for theming the website, or we will use a generic light or dark theme. I have attached a design below which considers all possibilities.

A screenshot of a computer

Description automatically generated



# Requirements

It is worth noting that for the application to be fully functional, a list of requirements and prerequisites need to be met.

## Software Requirements

For the application to work, both the user and service which hosts our solution will need to have some software installed.

### Client Side

We have tried to remove as much software requirements as we possibly could from the client side to make the website as accessible as possible. The only piece of software the user needs is an up-to-date web browser which supports pulling from APIs and modern JavaScript / CSS.

### Server Side

These are the required pieces of software which must be installed on the computer which is hosting our solution. Without this software, the solution cannot be functional.

|  |  |  |
| --- | --- | --- |
| Software Name | Software Type | Reasoning |
| Bootstrap | CSS Framework | Bootstrap is a framework which generates hundreds of existing CSS classes, reducing the amount of CSS that we must write ourselves. |
| jQuery | JavaScript Library | jQuery is a JavaScript library designed to make the way that JavaScript can be written more concise. This will improve the readability, scalability, and performance. |
| ASP.NET MVC | Backend Framework | ASP.NET MVC is a backend tool which is used to generate HTML webpages using C# queries and code, retrieve and manipulate data from a database, and authorise current users and their permissions. |
| Microsoft SQL | Query Language | Microsoft SQL is a Query Language which is used to store, retrieve, and update data. By default, ASP.NET MVC uses Microsoft SQL and I will be using it as they are both well integrated. |

## Hardware Requirements

Whilst there are no specific hardware requirements, the host machine must be able to run all the software requirements, mainly ASP.NET MVC, which comes bundled with the majority of the other software requirements (excluding APIs).

### Client Side

Whilst there are no explicit hardware requirements as long as the user can run a web browser, it is recommended that users have at least 4 gigabytes of ram and a display device which can support at least 720p to support video content and make text more distinguishable.

### Server Side

I would recommend a machine with a minimum of 8GB RAM and a i3 5th gen or higher processor to ensure that the solution can handle a moderate number of users and can easily run the required software. This could be scaled back or forward as required should Health Advice Group opt for a hosting provider with virtualisation support.

## Functional Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement Number | Description | Priority | Reasoning |
| 1 | Everyone has access to an accounts system. | HIGH | For data protection purposes, it is important that each user signs up with their individual information and do not have access to other’s information without their consent. |
| 1.1 | Individuals can sign up for their own account using their name, email, and a password. | HIGH | For similar data protection reasons, the user must be able to make an account on their own behalf. |
| 1.2 | Individuals can log in using the email and password that they used to register. | HIGH | To avoid confusion and improve user experience, users will be able to use the same password to log in as they used to register. |
| 1.3 | Individuals can reset their password by themselves. | MEDIUM | To reduce the need for staff intervention, users will have an email with a password reset link sent to them. |
| 1.4 | Users will have a role of management or user | HIGH | As the use case will vary differently between trustees and users, it is important that there is a role system to differentiate between them. |
| 2. | Management role will have features relevant to their role. | HIGH | It is important that management have access to appropriate features so databases which hold advice can be interacted with without our intervention. |
| 2.1 | Management will have the ability to add new advice for users to see. | HIGH | This will make it easy for advice to be quickly added without the intervention of our staff. This improves user experience and means that appropriate information can be published quicker. |
| 2.2 | Management will have the ability to see, update or delete existing advice. | HIGH | For similar reason, this will allow staff to quickly correct or remove inappropriate or inaccurate advice, reducing misinformation. |
| 2.3 | Management will be able to see a dashboard containing information related to the solution. | LOW | Should we get time, we may be able to create a dashboard which will display user’s interactions with advice and the website in general. This can allow management to make decisions about the solution and the advice they offer. |
| 3. | User role will have features relevant to their role. | HIGH | For security reasons, users should not have the ability to update advice or create new advice. This is to reduce misinformation and inappropriate content being spread. |
| 3.1 | Users will be able to see a forecast of the current weather at their location. | HIGH | This is one of the main factors which influence environmental conditions, hence it is important users can see this quickly to make health decisions. |
| 3.2 | Users will be able to see a forecast of the current weather at a location they enter. | MEDIUM | In some situations, users may be headed to another location and hence may need to see the temperature elsewhere to decide or to inform family. |
| 3.3 | Health advice for conditions within a certain range of the current temperature will be outputted on the forecast page. | MEDIUM | Whilst users can see the advice by accessing its respective page, displaying advice with the weather may quickly remind the user of any conditions they may have overlooked. |
| 3.4 | Users will be able to see a forecast of the air pollution at their current location | HIGH | The air pollution can affect conditions such as asthma, hence it is important that users can also access this quickly to make health decisions. |
| 3.5 | Users will be able to see a forecast of the air pollution at a location they enter. | MEDIUM | In some situations, users may be headed to another location and hence may need to see the air quality elsewhere to decide or to inform family. |
| 3.6 | Users will be able to see and access details about advice for health conditions. | HIGH | As the charity offers advice about health conditions, it is important that users can access this remotely to reduce travel and business at branches. |
| 3.7 | Users can search for condition advice by the name or temperature | MEDIUM | This will make it easier to navigate the sight and find relevant advice. |
| 3.8 | Users can save advice and access all their advice on a page | MEDIUM | This will reduce time spent trying to find advice, which will improve usability and accessibility. |
| 3.9 | Users can access a health tracker and input their calories, steps, and water | HIGH | This will allow users to make better health choices. |
| 3.10 | Users can see what they inputted into the tracker on previous days. | MEDIUM | This will allow users to see their health choices and progress over a longer period. |
|  |  |  |  |
|  |  |  |  |

## Non-Functional Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement Number | Description | Priority | Reasoning |
| 1. | Accessibility | HIGH | About 10% of people have a disability, hence it is important that we have features to accommodate such conditions. |
| 1.1 | Theme switcher | MEDIUM | Having a theme switcher can help to reduce eye strain when viewing in light or dark environments. |
| 1.2 | High contrast | MEDIUM | High contrast is one of the simplest ways to add accessibility to a website. By adding a bright background colour to important items such as navigation links, people with visual impairments can quickly see what parts are important. |
| 1.3 | Additional accessibility options | LOW | Due to the harsh time constraints, it is unlikely that we will be able to add more accessibility features for now. However, we will design the application with third-party accessibility extensions in consideration. |
| 2. | Security | HIGH | Due to data protection laws and to ensure the integrity of the system, it is important that we keep |
| 2.1 | Passwords and sensitive information are encrypted or hashed accordingly | HIGH | Due to data protection laws, appropriate measure such as one way encryption (hashing) must be taken to ensure sensitive information cannot be accessed if intercepted. |
| 2.2 | Inputs are safe from cross site scripting attacks | HIGH | To stop users being able to input malicious scripts which may lead to sensitive data being disclosed or making parts of the website inaccessible, it is important to validate against any inputs. |
| 2.3 | Inputs are safe from SQL injections | HIGH | It is important that we validate against and limit any “search” options which interact with a database. This is to stop users from performing malicious queries to obtain sensitive data or to cause strain on the database. |
| 3. | Scalability | MEDIUM | It is important that the solution is scalable in case Health Advice Group wants us to add to the solution in the future. |
| 3.1 | Modular approach | MEDIUM | It is somewhat important that we take a modular approach whenever possible in case we need to add upon, update, or remove a part of the system. |
| 4. | Maintainability | HIGH | It is important that should someone else need to edit the code in the solution, it is easy to understand. |
| 4.1 | Comments | HIGH | In order to improve maintainability, it is important we annotate complex sections or sections which need extra context. |

# Project

## Project Structure (MVC)

The solution will use a Model View Controller (MVC) structure, which splits the objectives of the application into 3 distinct categories.

The model refers to the database object which stores data, and it specifies how the data is stored. It is structured in a similar fashion to a C# class and constraints and conditions can be added using data annotations.

The view refers to the web page and content which the user sees. It uses razor views, which allows the usage of C# code and queries to generate HTML conditionally. This allows us to return different content to different users.

The controller refers to the server which responds to client requests and passes data between models and views. It also sends views back to the user and is responsible for authentication and authorisation before allowing access to certain features.

By separating these concerns, we ensure that no additional data or features are exposed to users, which increases the confidentiality and integrity of the solution. Additionally, by separating these concerns, it makes it difficult for the user to manipulate values or features, which helps to make the solution more secure and ensures users cannot modify data, information, or features.

## Project Risks

Cybersecurity – one of the biggest risks is cybersecurity risks. This is where a malicious actor can gain access to restricted parts of the system which or data it stores by any means.

* Interception of details – malicious actors may be able to monitor packets sent over the network, which may contain personal or login details. To counter this, encryption will be used so any data which is intercepted cannot be revealed or used. Additionally, users will not be able to send any data until a secure connection with the server has been established.
* Access to database – should malicious actors be able to gain access to our database, they may be able to access sensitive information. This may lead to a data breach, which the software house or Health Advice Group could be held liable for. Hence, it is important that our database is hosted by a provider who prioritises security and that all necessary data is encrypted, to minimise the amount of data which could be disclosed.
* SQL injection – malicious actors may attempt to run SQL queries to gain access to data or to cause strain on the database by performing resource intensive queries. To address this, it is important that we minimise the number of inputs which are directly linked to the database (by using ASP.NET’s entity framework) and that any inputs which are directly linked undergo intensive validation before the query is ran.

Scope / Constraints – as of right now, the development window is a relatively small amount of time (approximately 30 hours), and it only has one staff member dedicated to the entire development process. This could cause some issues with the quality of the prototype.

* Incorrect displays / bugs – as there has been no allotted window or staff member for testing the prototype, I must also complete this during the development process. This means the amount of test cases I can perform are extremely limited. This means that in some cases, something may be displayed incorrectly or there may be a fault within a certain feature, which will need to be fixed after the solution is deployed. To identify these features as quickly as possible, I will be incorporating a “report bug” feature into my solution which allows users to easily report a fault or incorrect display.
* Incomplete features – due to the harsh time constraints and other commitments in the development process such as testing and documentation, I may not have sufficient time to implement all the features which Health Advice Group have requested. To address this, I have split the requirements into “high”, “medium”, and “low” priorities based on their contribution to the solution’s functionality. High requirements will be prioritised first in the development of the application, followed by the medium and the low requirements.

Uptime Issues – as Health Advice Group have not provided us with any data, resources, or specific hosting which they want us to use, we will be making use of third-party solutions.

* Dependence on third-party APIs - as we do not have any in-house solutions for collecting data regarding the weather or air pollution, we will have to make use of third-party APIs and solutions. This means that we cannot control nor guarantee the uptime of these services, and hence some features of the application may not work correctly should these services go down.
* Hosting – during deployment, it is likely that Health Advice Group will have to use a third-party hosting provider. This also means that Health Advice Group will not be in control of the hardware or conditions which the solution is hosted on and hence the uptime of the solution is entirely dependent on a third-party. This means that should the third-party experience a problem or outage, the solution may not be accessible. To address this, it is highly recommended that Health Advice Group opt for a backup provider or system and ensure that both hosts prioritise uptime to minimise disruption.

Human Risks

* Phishing – employees or trustees from both the software house and the Health Advice Group may compromise login details should they get an email, call, or text from someone who pretends to act as us or another entity. This may give a malicious actor elevated access to our system, which could result in the loss or modification of data. To address this, it is important that both of our groups train employees on how to avoid these threats.
* Weak passwords – should an account on the system use a weak or easily guessable password, it may be vulnerable to brute force attacks. This means that a malicious actor may be able to gain access to the system if they are able to guess enough passwords. To limit this, we have implemented constraints and validation on the password field to force users to use more complex passwords, through the mandated use of numbers, symbols, lowercase, and uppercase letters.
* Keyloggers and spyware – should malware be installed on either the software houses or Health Advice Group’s computers, they may be able to gain access to credentials which may compromise the system. A malicious actor may be able to gain access to a management account, or even an account on the hosting solution which we use, which could result in the solution being deleted, tampered with, or unusable. To address this, both the software house and Health Advice Group must use up to date anti-virus software, run routine scans, and train their staff on how to avoid installing such a program.

## Project Security

Whilst our strategy for mitigating risks mentions security, the measures that we will use to secure our platform will be discussed in finer detail here. So far, we plan to use the following:

* Encryption / hashing – sensitive data which is sent and stored on the server will be encrypted or hashed, which is a form of encryption which is irreversible, meaning that the original value can not be retrieved. This means that if a database was compromised, user accounts could still not be compromised. This will utilise the SHA256 algorithm as it is one of the more secure hashing algorithms as every value added to the input will generate a new string of jumbled characters (hash), rather than appending to the existing string.
* Role based authorisation – should users want to update pieces of information which are shown on the website (i.e. articles, blogs), they will need a management or administrative role to make changes. Otherwise, they will only be able to view information. This improves the integrity of the data presented.
* Content-login lock – users will only be able to access content after they log in. This is to improve data security and to remove access from this content should a user be using this content or our platform inappropriately.
* Account lockout – should a user incorrectly input a password for an account 3 times, their account will be locked out and they will not be able to log in, even if they input the correct password afterwards. This will lock the user out for 5 minutes by default. This will help to prevent the chance of brute force attacks, as they will often have to enter thousands or millions of passwords before obtaining the correct password.

## Project Methodology

The project methodology is an influential part of the project as it will dictate the order in which we develop the solution’s features. To decide on an effective methodology, I had to consider both the number of staff / resources and the amount of time we had to build this prototype.

As of right now, one singular developer will be creating the prototype, in a 30-hour window. Due to these harsh time constraints, I will be using an adapted waterfall model. In this version, I will be focusing on the high priority requirements for each module, and then the medium priority requirements for each module, and then the low medium priority requirements for each module, as demonstrated by the Gantt Chart below.

A screenshot of a computer

Description automatically generated

# Business Context

## Uses in a Business Context

## Key Performance Indicators (KPIs)

The key performance indicators are a vital part of the project. This is because they will dictate and measure the success of the proposed project. I have written these indicators with the proposed format of the solution and Health Advice Group’s existing solution in mind. The name, measurement, and reasoning of each KPI I have chosen can be seen listed below.

|  |  |  |  |
| --- | --- | --- | --- |
| KPI name | KPI metric | KPI Description | Reasoning |
| Uptime | Percentage / 100 | The percentage of the time that the solution is accessible to the user. | The uptime is a necessary metric to prioritise as users are likely to seek alternative resources or solutions if ours is not available for long periods of time. |
| Average load time | Milliseconds | How long it takes to receive a view or response from the server. | The load time must be low to ensure user retention, this is because users may also seek alternatives if it takes a long time to load necessary resources. |
| Advice interactions | Total number | The number of times that a user clicks on the “details” page of an advice object. | The number of interactions with a piece of advice can be used to determine how common certain issues or conditions are, allowing management to post more relevant content and reach a wider audience. |
| Number of logins | Number / {time frame} | The number of users which log in to our solution in a certain period of time. | Number of logins can be a way to directly measure how many users our solution has reached and interested. |

It is worth noting that some of the KPIs such as number of logins and average load time cannot be accurately tracked during the development phase. This is because the solution will be hosted locally whereas it would be hosted elsewhere in the world by a third-party service provider should the solution be deployed. This means that we will not be testing for these KPIs during the development phase to avoid inaccuracies or confusion.

## User Acceptance Criteria

The app has 2 separate use cases dependent on whether the user simply uses the charity or if they are involved in the management of the charity. There are also criteria which must be met regardless of which role the user has, hence, I have decided to split the criteria into 3 different sections accordingly.

These criteria are important to ensure that all users of the app can use it without limitation, to meet their needs.

|  |  |  |
| --- | --- | --- |
| Common | Users | Trustees / Management |
| The layout of the solution must be clear and concise. | The website must provide features which let users inform their health decisions. | The website should allow management to add new advice. |
| The website should allow for everyone to register for an account and be able to log in. | The website must allow users to save advice and resources which help them to inform health decisions. | The website should allow management to update, edit, and delete existing advice. |

## Empathy Map

# Data

In order to comply with data protection laws, it is explicit that we state what data we are collecting and storing from this prototype. For the benefit of Health Advice Group, I have listed any annotations which may be used in sections regarding data below:

|  |  |  |
| --- | --- | --- |
| Abbreviation | Purpose | Example |
| PK / Primary Key | The main property or value which can be used to uniquely identify an entry in a table. | AdviceID: 5  UserID: 43fa119e-e98f-4d5e-9541-335954d8cebd |
| FK / Foreign Key | An identifier which is used as the primary key in another table. | AdviceID: 5  UserID: 43fa119e-e98f-4d5e-9541-335954d8cebd |
| CHAR NUMBER / Characters | A string of characters which is the length specified by NUMBER | Abcde: CHAR 5  ABCDEFGH CHAR 8 |
| VARCHAR NUMBER / Variable Characters | A string of characters which can be any length up to the length specified by NUMBER | Dave: VARCHAR 10  ABCDEFGHIJ: VARCHAR 10 |
| INT / Integer | A whole number which can be used to perform calculations or ordering | 1234  5432875309123290  2 |
| BOOL / Boolean | A data type which can be in one of 2 states, true or false. | True  False |
| DATE / Date | A data format designed to indicate the date of an entry or event | 27/01/2024  01/27/2024  2024/01/27 |

## Data Map

In order to comply with Data Protection laws, we need to state how we collect data and what data is collected. To easily visualise what data we collect, how it will be stored, and how data is linked across the service, I have attached a visualisation below.

The first most value refers to the “primary key” of the table and it is used to identify any entry in a table. Additionally, I have made use of arrows wherever necessary to indicate data which is used in another table, and where it derives from.

A diagram of a computer

Description automatically generated

## Data Requirements

Mapped below are all the variables which our database will store, an example of a variable, why it is needed in the database and any validation which must be applied to it. This is intended to explain why each piece of data is needed for the app to work and how it must be implemented in regards to security and limitations.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Data Type | Format | Constraints | Purpose | Example |
| User ID | VARCHAR | N/A | N/A | Used to uniquely identify any user regardless of similar First Names, Second Names, emails, etc. | 43fa119e-e98f-4d5e-9541-335954d8cebd |
| First Name | VARCHAR | N/A | Minimum length 2  Maximum length 20 | Used to greet the user upon logging in | Jamie |
| Last Name | VARCHAR | N/A | Minimum length 2  Maximum length 20 | Used to greet the user upon logging in | Jamie |
| Email / Username | VARCHAR | email@site.com | Contains @  Contains . | Used to register and log in to an account and can be used to reset / confirm account | [example@gmail.com](mailto:example@gmail.com) |
| HashedPassword | VARCHAR | N/A | N/A | Used to store user passwords in a “scrambled” format which cannot be unencrypted | AQAAAAEAACcQAAAAEFCqbaXbSmBXZO3xTNYNTJiD7VmX1rJ3L8uzOhhU4SluohAPQf1uf1Pg7TuFQdQMRA== |
| EmailConfirmed | BOOL | N/A | N/A | Used to check if the user has clicked the activation link on their account. Cannot sign in if it is False. | True |
| LockoutEnabled | BOOL | N/A | N/A | Used to check if the user can sign into their account. Cannot sign in if it is True. | False |
| LockoutEnd | DATE | YYYY/MM/DD | N/A | If LockoutEnabled is True, this is the date it will return to false. | 2024/01/29 |
| TrackID | INT | N/A | N/A | Used to uniquely identify a health diary entry submitted by a user. | 45 |
| Date | DATE | YYYY/MM/DD | N/A | Used to identify when a user made an entry to the health tracker. | 2024/01/27 |
| Calories | INT | N/A | N/A | One of the main metrics used for users to track their health and habits. | 1980 |
| Steps | INT | N/A | N/A | A metric used to track the activity of the user | 19000 |
| Water | INT | N/A | N/A | The cups of water a user has had. Metric used to track if the user is drinking healthily. | 13 |
| AdviceID | INT | N/A | N/A | Used to uniquely identify any advice added. | 12 |
| Title | VARCHAR 100 | N/A | Minimum length 5  Maximum length 100 | This is used to quickly indicate what a piece of advice is about. | Hypothermia |
| Temperature | INT | N/A | Minimum number -90 (degrees)  Maximum number 60 (degrees) | The temperature that a specific condition starts to occur at, used to offer advice when forecast is near this value. | -2C |
| Description | VARCHAR 1000 | N/A | N/A | A long piece of text used for advice and reporting a bug. | Lorem ipsum sit dolor amit conneticut adiscping elit |
| Interactions | INT | N/A | N/A | The number of times the “details” button has been clicked on a piece of advice. | 456 |
| BugID | INT | N/A | N/A | Used to uniquely identify any bug reported. | 12 |
| Feature | VARCHAR 50 | N/A | Minimum length 2  Maximum length 50 | The feature the user was accessing when they encountered a bug. | Weather Forecast |
| Browser | VARCHAR 50 | N/A | Minimum length 2  Maximum length 50 | The browser the user is using when they encounter the bug | Firefox |
| Device | VARCHAR 50 | N/A | Minimum length 2  Maximum length 50 | The device the user is using when they encounter the bug | Samsung A52s |
| OpenTicket | BOOL | N/A | N/A | If a bug has been fixed. | True |

# Testing

For the application to function consistently and as expected, it is important that some of the development window is dedicated to testing. This section will discuss how we intend to test the solution; however, it is worth discussing that this section is not for testing the solution.

## Test and Data Types

In order for the web application to be secure, we must complete multiple types of tests and insert multiple categories of data into inputs, to make sure that they are accepted and rejected as appropriate. This section will detail each type of test and their expected outcome.

* Value Testing – value testing involves entering specific valid and invalid data into inputs to see if they are correctly rejected (not added to the database) or accepted (added to the database). Value testing often requires multiple types of data to ensure that any security / validation is working correctly, this may include:
  + Erroneous Data – data which does not meet the correct format or input type, this should be rejected and prompt the user to use the right format.
  + Invalid Boundary Data – data which is one value or condition away from being accepted, such as 1 length or 1 number off a minimum value, this should be rejected and prompt the user to re-enter.
  + Valid Boundary Data – data which is one value or condition away from being rejected, such as being minimum length or a minimum value, this should be accepted and added to the database.
  + Valid Data – data which should be accepted and added to the database.
* Accessibility Testing – accessibility testing involves checking elements on the page to ensure that they are accessible to all users. I will be utilising an accessibility checker called “WAVE accessibility checker” to improve the efficiency and consistency of the testing process. WAVE checks elements of the page, for example:
  + Font size – it is important that the text is big enough for people to read, regardless of any visual impairments.
  + Images – it is important that every image has alternative text so that users who cannot access images can still know what is supposed to be there.
  + Nesting of elements – it is important that elements are nested properly to make it easy for users to alternate between and focus on elements using a keyboard or alternate devices if they cannot use a mouse.
* Functionality Testing – functionality testing involves checking that features work correctly and are displayed correctly with valid data. It can also include checking the accuracy of data pulled from an API compared to other services or providers (also known as accuracy testing).

## Testing Strategy

In order to ensure consistent testing, it is important that we understand what components need to be tested and what must be completed before this can be done. As of right now, Health Advice Group have not provided a concrete window for testing, hence it will needed to be included in the development process. Due to prerequisites and dependencies, it is not clear as to when these tests can be completed, hence the date of test will remain as “undecided” until we have more information or until the development process starts.

|  |  |  |  |
| --- | --- | --- | --- |
| Date of test | Component to be tested | Type of test to be carried out | Prerequisites and dependencies |
| Undecided | Register system | Range testing | Users database must be set up. Relevant validation systems must be implemented. |
| Undecided | Register system | Value testing | Users database must be set up. Relevant validation systems must be implemented. |
| Undecided | Login system | Value testing | Users database must be set up and user must be registered. |
| Undecided | APIs | Accuracy testing | APIs must be set up and pull data. Alternative sources must be available to compare values. |
| Undecided | Advice system | Range testing | Advice system and display must be set up. |
| Undecided | Bug report system | Range testing | Bug system and database must be set up. |
| Undecided | Bug report system | Value testing | Advice system and display must be set up. |

## Testing Log

Whilst the testing strategy identifies the main components we need to test, it does not explain how we plan to test these components and what we are checking for. For Health Advice Group’s benefit, I have attached a copy of the test plan, which will be used in the development phase of the prototype, below.

|  |  |  |  |
| --- | --- | --- | --- |
| Description | Data | Valid? | Reason |
| Value test for email in register action | jamieh | N | No @ symbol – email is invalid |
| Value test for email in register action | jamieh@gmail | N | No top-level domain, for example, .com – email is invalid. |
| Value test for email in register action | jamieh@gmail.com | Y | N/A |
| Range test for first name in register action | J | N | The first name is 1 character, which is less than the minimum length of 2 characters. |
| Range test for first name in register action | Ja | Y | The first name is 2 characters, which is the minimum length of 2 characters. |
| Range test for first name in register action | JamieLoremIpsumDolor | Y | The first name is 20 characters, which is the maximum length of 20 characters. |
| Range test for first name in register action | JamieLoremIpsumDolore | N | The first name is 21 characters, which is more than the maximum of 20 characters. |
| Range test for last name in register action | H | N | The last name is 1 character, which is less than the minimum length of 2 characters. |
| Range test for last name in register action | Ho | Y | The last name is 2 characters, which is the minimum length of 2 characters. |
| Range test for last name in register action | HodgsonLoremIpsumDol | Y | The last name is 20 characters, which is the maximum length of 20 characters. |
| Range test for last name in register action | HodgsonLoremIpsumDolo | N | The last name is 21 characters, which is more than the maximum of 20 characters. |
| Range test for password in register action | pass | N | The password is 4 characters, which is less than the minimum length of 6 characters. |
| Value test for password in register action | password | N | The password doesn’t contain special characters, uppercase or numbers |
| Value test for password in register action | Passw0rd | N | The password doesn’t contain special characters |
| Value and range test for password in register action | P@ssw0rd | Y | The password is more than minimum length and has a lowercase, uppercase, numbers, and symbols. |
| Value test for confirm password in register action | P@SSw0rd! | N | Does not match original valid password |
| Value test for confirm password in register action | P@ssw0rd | Y | The password matches the valid password which we inputted in the first password field. |
| Presence test for email address in log in action |  | N | The email address is blank – a user cannot have a blank email address, hence it is not valid. |
| Value test for email address in log in action | hodgsonj@gmail.com | N | This email address does not match the email we signed up with. Assuming that no other user has registered with this email, it is not valid. |
| Value test for email address in log in action | jamieh@gmail.com | Y | This email address matches the email we signed up with. Assuming the registration was successful and the email was confirmed, this email would be valid. |
| Presence test for password in log in action |  | N | Due to earlier constraints and validation which we have defined, a password cannot be blank, hence this is invalid. |
|  |  |  |  |
| Value test for password in log in action | P4ssw@rd | N | This does not match the password we originally signed up with. Assuming that password has not changed, this is invalid. |
| Value test for email address in log in action | P@ssw0rd | Y | This matches the original password which we signed up. Assuming that this hasn’t changed, the password would be valid. |
| Accuracy test for weather API | Latitude: 52.040623  Longitude: -0.759417 | N/A | We will need to test if the weather API is accurate for our location by comparing against other reliable services and sources before we begin to use it. |
| Accuracy test for air pollution API | Latitude: 52.040623  Longitude: -0.759417 | N/A | We will need to test if the air pollution API is accurate for our location by comparing against other reliable services and sources before we begin to use it. |
| Accuracy test for geocoding API | City Name:  Milton Keynes  State:  England  Country Code: UK | N/A | We will need to test if the geocoding API is accurately converting our location by comparing and making use of mapping tools and other sources or tools.. |
| Range test for calories in track health action | -1 | N | Users cannot enter negative calories as there is no such thing. |
| Range test for calories in track health action. | 0 | N | The calories system will add to the calories which have already been logged, meaning adding 0 calories will not do anything. |
| Range test for calories in track health action | 1 | Y | As the number is positive, the number of calories is valid. This would then be added onto the calories total for the day, |
| Range test for steps in track health action | -1 | N | Users cannot enter negative steps as there is no such thing. |
| Range test for calories in track health action. | 0 | N | The steps system will add the input to the steps which have already been logged, meaning adding 0 steps will not do anything. |
| Range test for steps in track health action | 1 | Y | As the number is positive, the number of calories is valid. This would then be added onto the steps total for the day, |
| Functionality check for water in track health action | 1 | Y | This test simply ensures functionality as clicking the “add water” button will simply increment it by 1. |
| Range test for title in add advice action | Lore | N | The title is 4 characters, which is less than the minimum length of 5 characters. |
| Range test for title in add advice action | Lorem | Y | The title is 5 characters, which is the minimum length of 5 characters. |
| Range test for title in add advice action. | Lorem ipsum dolor sit amet, consectetur adipiscing elit. In tempus lorem sit amet pharetra tincidunt | Y | The title is 100 characters, which is the maximum length of 100 characters. |
| Range test for title in add advice action. | Lorem ipsum dolor sit amet, consectetur adipiscing elit. In tempus lorem sit amet pharetra tincidunt. | N | The title is 101 characters, which is more than the maximum length of 100 characters. |
| Range test for temperature in add advice section | -91C | N | The minimum temperature which can be entered is -90C as the coldest temperature recorded is roughly this. |
| Range test for temperature in add advice section | -90C | Y | The temperature is -90C, which is the minimum temperature. |
| Range test for temperature in add advice section | 60C | Y | The temperature is 60C, which is the maximum temperature. |
| Range test for temperature in add advice section | 61C | N | The maximum temperature which can be entered is 60C as the hottest temperature recorded is roughly this. |
| Range test for description in add advice section | Lore | N | This text is 4 characters, which is below the minimum length of 5 characters. |
| Range test for description in add advice section | Lorem | Y | This text is 5 characters, which is the minimum length of 5 characters. |
| Range test for description in report bug section | Lore | N | This text is 4 characters, which is below the minimum length of 5 characters. |
| Range test for description in report bug section | Lorem | Y | This text is 5 characters, which is the minimum length of 5 characters. |
| Range test for feature in report bug action | L | N | The feature is 1 character, which is less than the minimum length of 2 characters. |
| Range test for feature in report bug action | Lo | Y | The feature is 2 characters, which is the minimum length of 2 characters. |
| Range test for feature in report bug action | Lorem ipsum dolor sit amet, consectetur adipiscing | Y | The feature is 50 characters, which is the maximum length of 50 characters. |
| Range test for feature in report bug action | Lorem ipsum dolor sit amet, consectetur adipiscing. | N | The feature is 51 characters, which is more than the maximum length of 50 characters. |
| Range test for browser in report bug action | L | N | The browser is 1 character, which is less than the minimum length of 2 characters. |
| Range test for browser in report bug action | Lo | Y | The browser is 2 characters, which is the minimum length of 2 characters. |
| Range test for browser in report bug action | Lorem ipsum dolor sit amet, consectetur adipiscing | Y | The browser is 50 characters, which is the maximum length of 50 characters. |
| Range test for browser in report bug action | Lorem ipsum dolor sit amet, consectetur adipiscing. | N | The browser is 51 characters, which is more than the maximum length of 50 characters. |
| Range test for device in report bug action | L | N | The device is 1 character, which is less than the minimum length of 2 characters. |
| Range device for feature in report bug action | Lo | Y | The device is 2 characters, which is the minimum length of 2 characters. |
| Range test for device in report bug action | Lorem ipsum dolor sit amet, consectetur adipiscing | Y | The device is 50 characters, which is the maximum length of 50 characters. |
| Range test for device in report bug action | Lorem ipsum dolor sit amet, consectetur adipiscing. | N | The device is 51 characters, which is more than the maximum length of 50 characters. |

## Feedback Approaches

Whilst methods such as range, functionality, and value testing can help determine whether a website is functional, it cannot help to measure factors such as usability, practicality, design, and security, hence it is important to incorporate user testing for feedback and improvements into our strategy.

During and after the development process, I will be employing the use of white box testing. This means that users with an understanding for how the system or similar systems work will be testing the prototype. In order to carry out white box testing effectively, I will be employing methods listed below:

* Video observation – I am intending to record videos of the prototype, in which I will showcase the functionality of a feature and then the code / logic behind it. After showcasing each appropriate feature, I will ask for feedback regarding metrics such as the ease of implementation, security, and code readability. This will likely be in the format of a Microsoft Form.

After the development process, I will also be employing the use of black box testing. This will involve users with limited or no understanding behind how the system works to test and provide feedback on the prototype. In order to carry out black box testing effectively, I will utilise the following methods:

* Design survey – I intend to capture feedback from a non-technical audience. This will be by a Microsoft Form and will ask questions regarding the intuitiveness, cleanliness, and layout of the solution by providing snippets such as short videos or screenshots.

# Appendix

Source 1 research document