Contents

**No table of contents entries found.**

# 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.

## Project Proposal

## UI/UX Designs

## Legalities

# Requirements

## Software Requirements

|  |  |  |
| --- | --- | --- |
| Software Name | Software Type | Reasoning |
| HTML / Razor | Markup Language | HTML is needed to insert basic content such as text, buttons, and links into web pages, which the solution will be using. |
| CSS | Styling Language | CSS is needed to edit or manipulate the layout or appearance of basic HTML elements on a webpage. As HTML is used, CSS is also important. |
| JavaScript | Programming Language | JavaScript is the language which runs any basic logic on the web page and is used to retrieve data and generate new or edit existing HTML elements and CSS properties. |
| 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. |
| OpenWeatherMap Current Weather | API | The current weather API is a large set of data for the current temperature at a certain longitude and latitude. This will allow users to easily see the weather at any time. |
| OpenWeatherMap Air Pollution | API | The air pollution API is a detailed list of the concentration of all the pollutant chemicals in the air at a certain longitude and latitude. This will allow users to see if the air quality is healthy or unhealthy and allow users to take action based on the result. |
| OpenWeatherMap Geocoding | API | Whilst you can get the Latitude and Longitude of a user from their location, users may opt to see a different location. As the APIs expect a Latitude and Longitude, this API can convert a readable address to a Latitude and Longitude to be used in the other features. |
| 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).

## 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 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

## Project Risks

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

* Interception of details
* Access to database
* SQL injection
* Keyloggers and spyware

Scope / Constraints

* Incorrect displays / bugs
* Incomplete features

Uptime Issues

* Dependence on third party services

Human Risks

* Phishing
* Insider threats
* Weak passwords

# Business Context

## Key Performance Indicators (KPIs)

|  |  |  |  |
| --- | --- | --- | --- |
| 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. |

## User Acceptance Criteria

* The layout of the solution must be clear and concise.
* The website should allow users to access resources related to conditions through the “advice” system.
* The website should provide features which allow users to inform their health decisions.
* The website should allow users to save advice and resources which help them to inform health decisions.
* The website should allow management to add new advice and update existing advice.

# Data

## Data Map

## Data Requirements

# Testing

## Testing Strategy

## Testing Log

## Feedback Approaches