



TASK 3 PART B - EVALUATION

Health Advice Group

Evaluation - Effectiveness

Proposal

Firstly, it should be said that the client has asked for a weather-based application which ensures accurate and thorough information that is specifically tailored to the customer and their needs. This includes weather health effects, weather conditions as well as the possibility for allergies, warnings for other risks such as allergen and give advice on how to manage the extreme weather conditions.

The requirements included in the proposal should also be stated as a reminder of what the aims of the digital solution or rather application were, these include:

Functional requirements

1. Information/data on how to handle severe weather conditions.
2. Email and password user registration.
3. Personalised health advice based on risk assessment forms.
4. Personalised health advice on provided health conditions for a given location and time.
5. Weather forecast widgets displayed on an overview dashboard.
6. Air quality widgets displayed on an overview dashboard.
7. Links to relevant information based on weather conditions viewable on a weather forecast widget on an overview dashboard.
8. Screen reader accessibility.
9. Information on environmental health conditions.
10. Tracking trends with weather conditions relevant to a user's health conditions.

Non-Functional requirements

1. Average API request speed of below 1 second.
2. Average page rendering speeds below 5 seconds.
3. Software must be able to be accessed on any device with internet access and a browser.
4. The software must be compatible with multiple browsers.
5. The software must be available to access 24 hours per day.
6. Screen readers must be able to read all text and describe all images used within the software.
7. Colour contrast for headings and images must follow WCAG AAA guidelines.
8. Colour contrast for content and text in the application must follow WCAG AA guidelines.
9. The software must be written in English and contain no grammatical or spelling errors.
10. Navigation between pages must be available in a header bar for large screens as well as a sidebar for small screens.

Assets

When it comes to assets used, they each had to be considered carefully due to no budget being present. Taking that into consideration, there were tools available for free such as Figma, Tkinter-Designer and Flaticon which allowed for a smooth development without the concern of paying for content.

Why were they selected?

These tools were considered and used mainly due to the aspect of them being free to use as well as having content made with such tools be published without difficulties concerning legality. Other content was rejected either due to being a service that includes monetisation in some way for example, a membership or not being as useful when compared to other tools available such as Figma and Flaticon. Tkinter-Designer was an importance choice and heavily contributes to the project as it essentially allows for designs on Figma to be converted into python code which greatly decreases the amount of time to make the application. Normally Figma converts its designs into HTML, CSS and JavaScript which is used for making websites, however, due to the digital solution required to be an application it was

important to find a work around, after careful consideration this is how Tkinter-Designer was chosen.

Figma was selected as it was a free service which allowed for plenty of options and plugins to be used whereas other tools lack this feature. Figma offers many ways to change designs and makes it easy to place content accurately as well as create separate windows that can link to each other. Figma was also chosen as it behaves similarly to how a GUI would be made in python for example.

Flaticon was decided upon after the designs on Figma were produced, as the designs were colourful and light, it was decided that icons on Flaticon were to be used as the icons used complimented the designs of Figma. Flaticon was also easy to use, only requiring an icon to be downloaded and then used.

The 100+ colour palettes by Aleksandra Sherr has been used to provide inspiration of what the Figma designs would look like as it would take a significant amount of time to think of a unique colour scheme, as the time available was limited it was decided that using an existing colour scheme would be a sufficient option. Aleksandra Sherr has been credited for her work in the application repo available on GitHub. There were other colour palettes available but they did not go in-depth or were not ideal which led to them being rejected.

Lastly, the visual python GUI kit has been used for inspiration on how the layout of the application should be set as it would also take a significant amount of time to make a unique layout from scratch. This has purely been chosen to save time and take inspiration from, this work has also been credited in the repo available via a md file. It should be noted that other GUI kits were viewed, however, most were confusing and may intimidate users if implemented which ultimately led to the kits being rejected.

When it comes to the sources of information used, it was kept formal (official) meaning that information was sourced from the UK government as well as other trusted sources. Other sites were rejected as it was unsure if they could be trusted sources which would jeopardize the application and client's integrity and/or reputation.

The validity and reliability of the sources of information used

As mentioned previously, information used has come from the UK government (Met Office) as well as other trusted sources that are known give accurate weather readings and other relevant information such as health reports.

The effectiveness of such sources is great as it removes the time needed to validate if the sources used are reliable or not as they come from an official source or are in some way connected. It should be noted that the Met Office Agency includes the expertise of physicists and mathematicians as they develop weather models and are well equipped for example, having access to an environmental supercomputer.

Another important note that was considered was their position or reputation and what they stand to lose if their source of information was proven to be inaccurate when compared to other sources available. The cost would be tremendous as the Met Office Agency is an official source. This results in information being reliable as well as valid.

Legal and ethical implications of the assets and content selected

Legal and ethical implications are important to take note of and have been considered when choosing assets. A concern of the assets used would be if any policies change that drastically change how content can be used and/or published which would compromise the application and could potentially become a major issue. During this point in time however, there are no concerns when it comes legality and ethics as all the tools used are open-source and creators as well as teams have been credited for their work, following the code of conduct and licensing agreements.

Lastly, a future legal and ethical concern would be if the application were to release in other countries. The law of each country would have to be adhered to which could raise potential issues if not followed and/or conflicts with how assets and content are used.

How well the digital solution planned and produced meets:

Functional and non-functional requirements of the solution

As mentioned previously, both functional and non-functional sections include 10 requirements, being a total of 20 requirements. Due to the sheer number of requirements having to be fulfilled, it was difficult to fulfil all of them which unfortunately led to some requirements having to be left uncompleted.

Although there are requirements missing in the prototype, the build also includes many important requirements such as information providing users ways of how to handle severe weather conditions, email and password registration and weather forecast widgets displayed on an overview dashboard.

The requirements implemented have been followed effectively as many factors were considered before their implementation such as gathering official sources for information on how to handle severe weather conditions. Email and registration were implemented in the design before then going on to be developed on the homepage. Widgets were also considered when choosing Flaticon assets and appear neatly on the dashboard window. All requirements whether functional or non-functional are considered carefully before implementation to find the best way of how to use them efficiently.

However, some functional requirements that were not implemented include:

- Personalised health advice based on risk assessment forms. - It has not been solved on how to implement this yet.
- Personalised health advice on provided health conditions for a given location and time. - Did not have the opportunity to set up yet.
- Air quality widgets displayed on an overview dashboard. - Air quality information was planned to be implemented but due to time constraints, it did not happen.
- Screen reader accessibility - Is not a priority thus it was not focused on.
- Tracking trends with weather conditions relevant to a user's health conditions. - Do not know how to implement this requirement yet.

As there included many requirements, the focus of the prototype was to fulfil most of the functional requirements as most non-functional requirements could either be implemented during deployment or be added in future updates made. Not all the non-functional requirements were missing, some of the fulfilled requirements include the application being available on any device with internet access and a browser, compatibility with browsers, software being written in English and having no grammatical errors and lastly navigation between windows for large and small screens fulfilled through a sidebar as well as buttons leading to the same windows.

The non-functional requirements missed include:

- Average API request speed of below 1 second. - This would be considered before deployment.
- Average page rendering speeds below 5 seconds. - Would be considered before deployment.
- The software must be available to access 24 hours per day. - Would be the last requirement to consider before deployment.
- Screen readers must be able to read all text and describe all images used within the software. - Would be considered in a future update.
- Colour contrast for headings and image must follow WCAG AAA guidelines - Would be considered before deployment of the application.
- Colour contrast for content and text in the application must follow WCAG AA guidelines. - Same for above.

These requirements were not implemented as ideally, they would be considered before deployment of the application. A build of the application needed to be completed first for it to then be tested. Even though a lot of requirements were not met, they can all be easily implemented due to many factors such as plan made in the proposal which gives an idea of how each requirement were to be implemented as well as the designs/algorithms shown depicting how the system would behave. Lastly, the code was kept clean and well maintained for this reason, making it much easier for new code to be implemented due to comments and how files and code link to each other.

Key performance indicators (KPIs)

It was important to include KPIs as they play a significant role in how work can be tracked and maintained efficiently. An example of an important KPI would be code churn as code churn measures how often a piece of code changes throughout the development cycle which occurs frequently. Code churn, in the project was used to measure the stability of code and how it could be kept to a minimum to avoid bloats of code. Code churn was used to monitor changes of specific files such as the home page and dashboard file as they were the most active at the time.

User acceptance criteria

As an integral part of the project, most of the user acceptance criteria have carefully considered throughout the project as to provide a more effective digital solution. Unfortunately, not all the criteria were implemented such as users being able to see air quality forecasts and personalised health advice based on location as there was no time left for it to be implemented. Below will be explained how the acceptance criteria has been followed:

A user must be able to access the weather forecast to be informed on health decisions. This has been implemented by expanding the weather forecast widgets of the dashboard to include individual days hourly forecasts as well as links to relevant reports explaining health information on the current weather conditions. A widget that displays the next week's weather forecast on a dashboard has yet to be implemented due to time constraints.

A user should be able to get advice on how to manage their health related to weather/environment conditions. This has been adhered to by adding links to relevant resources where weather or environmental conditions on the dashboard are of concern. Links to reports have also been included that provide accurate information and can be access via the application by users.

How the prototype could be developed further

Taking the feedback given into consideration, it is believed that adding an air quality page, inputs for temperatures and visual representations of data should be added to further improve the user experience follow the requirements given as well as the user acceptance criteria. There are still requirements that should be implemented however, due to the plans and designs considered before development as well as the clean and maintained code of the application will make it easy to implement changes.

The conventions of the application will also be closely monitored in order to make sure that the application gets the most out of them; increasing efficiency.