**Team meeting 2 – Connor’s input**

|  |  |  |
| --- | --- | --- |
| Requirements | | |
|  | **Individual requirement** | **Notes** |
| **‘Big Picture’ Requirements** | Project can work on any operating system | This project should be usable on any OS that has a web browser with JavaScript enabled |
| Project can work on any modern web browser (Chrome, Firefox, Safari etc) | This is essential for cross operability |
| Project can load external data provided by the user | This can be in the form of SQL import  Or by a file upload from the user |
| Project can be collaboratively worked on by all members of the group | Utilising the github repo will ensure this is the case.  Branches should be used for any changes and then remerged with the main branch when complete. |
| Project must have version control | Achieved through Github and essential for redundancy should any files be corrupted. |
| User can interact with the software without it crashing | This will likely be achieved through good testing and strong coding practises |
| User can upload their own data by a file | File types to be dicussed but at a minimum CSV? |
| User can use pre-built data | The system will have an option to give the user functionality through pre-loaded data to ensure it can always display something |
| Software will have accessibility built in from the start | Utilsiing the information on <https://www.w3schools.com/accessibility/> |
| Project will display information graphically to the user | Graphs, charts, heatmaps etc  Possibly add multiple pages with different graphics on each |
| Project will give the user the ability to export the graphics produced. | Recommend going for a JPEG and then further formats can be explored later.  Possibily integrate email, SMS etc as well |
|  | Project uses HTML and CSS for frontend |  |
|  | Project uses JavaScript for the backend |  |
|  | Project will have an appropriate distribution system | HTML file?  Web hosting?  AWS? |
|  |  |  |
| **Error handling requirements** | Software will manage blank, incorrect or incomplete data entries within the users uploaded data | To ensure the program doesn’t crash |
| Software will manage the user uploading the incorrect file type | To ensure the program doesn’t crash |
|  |  |  |
| **Design Requirements** | Project uses OOP principals | Where possible Object Orientated Programming principals should be used to enhance reusability of code. |
| Project uses a standardised style guide | HTML/CSS from Google’s style guide  <https://google.github.io/styleguide/htmlcssguide.html>  JavaScript from Google’s style guide  <https://google.github.io/styleguide/jsguide.html> |
| Project uses comments throughout | Comments help other developers maintain and utlise the code produced |
| Project is designed with Pseudocode | Pseudocode is important for any functions or data processing as it means the project can expand to use the new HTML <py-script> tags in the future with little hassle converting JavaScript to Python. |
| Project utilises UML for design | Reference the FreeCodeCamp course if needed: <https://www.youtube.com/watch?v=WnMQ8HlmeXc>  UML is a good tool to show the overall data flow |
| Project will utilise node.js and other JavaScript package managers to add functionality to the software. | This will speed up development time and add functionality to the software |
|  |  |  |
| **Security Requirements** | Software will prevent SQL Injections for any user input if the decision is made to implement SQL functionality | Wider project management criteria |
|  | If web hosted ensure appropriate authentication occurs | HTTPS or other secure web serving |