|  |
| --- |
| **Team 6**  **+1**  **4216** |
| Assignment 4216: Due: 5th April 2022  Cohort Members:   * Cameron Marsh * Ethan Ibrahim * Laura J. Phillips * Luke Citrine * Luke Curren * Niamh Walsh * Shane Mendes |
|  |

# Contents

|  |  |  |
| --- | --- | --- |
| Section | Detail | Page No. |
| 1 | Design Stages |  |
|  | Project Synopsis |  |
| 2 | Team Building |  |
| 3 | Project Management Approach |  |
| 4 | Data Acquisition |  |
| 4.1 | Concept Stage |  |
| 4.2 | Logical Stage |  |
| 4.3 | Implementation Stage |  |
| 4.4 | Project Closure |  |
| 5 | Learning |  |
| 5.1 | Conclusion |  |

# Project Synopsis

The project required us to build a successful, supportive team and collaboratively design, build and implement a data analysis tool that could output meaningful information, displayed in graphs.

We decided to use standard project management techniques to manage the project. It was agreed that each team member would be responsible for leading the team for two weeks, perform a programming and administrative function and provide ad-hoc support to each other. This was to purposely ensure all team members obtain a fulfilling and successful learning experience and result in a successful project delivery.

# Design Stages

Data Acquisition: As a team, we considered several data such as, climate change and criminal activity data, both were rejected due to incomplete records or low volume datasets. Lastly we evaluated weather temperature records and decided that it was suitable with over 13,000 quality records that covered a time period of 1900-2013.

# Project Readiness

# Team Building

|  |  |
| --- | --- |
| Table  Description automatically generated | As a team we needed to appreciate our individual skill strengths and weaknesses and the table below illustrates our starting position.  Although the scores recorded were not captured scientifically the exercise did help us understand the potential roles each of us could naturally play. |

# Project Management

Within the GitHub repository we created project documents that we organised into folders. These assets were instrumental in helping us to manage the project effectively.

Graphical user interface, application

Description automatically generated

For example, this is the file we created for planning our tasks.

Graphical user interface, application, email

Description automatically generated

This is an example of the notes we captured during one of the team meetings.

Graphical user interface, text, application, Word

Description automatically generated

Below is an example of a Milestone Plan.

Table

Description automatically generated

# Communications

|  |  |
| --- | --- |
| Text  Description automatically generated | The first thing we did as a team was set up a WhatsApp group so everyone is aware of progress, issues or support requests first hand.  We then agreed to set up regular face to face meetings at 12:00pm, room 1.73 before our tutorial class would start at 1pm. We then continued the meeting at 14:30, this proved to be invaluable as we were able to manage the workload, discuss any issues and effectively support each other as.  Each week we recorded the minutes of the meetings and uploaded them onto GitHub. |

# Concept

# Design Decisions

Diagram

Description automatically generated

*Fig. 1: Original data overview.*

|  |  |
| --- | --- |
| Reduce the number of records available for analysis. | 13,000 global weather records were too many for us to process, so we chose to trim down the volume. |
| Simplify data content | By filtering UK records out from the global data we obtained a more manageable data set of around 1300 records. (see fig. 1 above). |
| Data split | This data was then allocated evenly amongst all seven team members. |

The data we chose was UK weather temperature data from the years 1900-2013, made up of average temperature measurements from the first of the month throughout each of the 113 years. It covers both land and ocean temperatures, with twelve records for each year in this period. Originally, the data was in three tables: Global Temperature, Country Land Temperature, and City Temperature (see fig. 2 below).

*Diagram

Description automatically generated*

*Fig. 2:* an *entity relationship diagram for the original three tables, showing the fields, relationships, and linking keys for each.*

# Design Decision

We decided to combine all of these into one table for ease of access and data cleanliness. When first analysing the data for eligibility, we split the data into equal chunks of eighteen years, which made it easier to analyse. As such, we decided to take a similar approach when drawing up our list of requirements for the software design.

It was our intention that the user should be able to utilise the historical data throughout the period to be able to identify changes between and patterns in the temperatures, by allowing the user to search and filter different seasons and months for years. We needed the data to be easy to manipulate, reflecting this in our chosen design: the data will be explored by the user mixing and matching various filters, including month, season, and year. The landing page will have options for all three, though within each the user will be able to add the other filters should they wish to do so. The end result will be a dynamic graph, either a pie chart or a bar chart depending on which is most appropriate, that changes with the data the user chooses.

# Software Requirements

Our requirements for the software:

* Allow the user to filter data by:
  + Month
  + Year
  + Season
* Produce graphs that changes with the data the user filters
  + Pie charts
  + Bar charts
* Be intuitive and easy to use
* Read from the .CSV file to obtain the data needed
* Allow the user to draw conclusions as to temperature patterns in the data
* EXT: write to a .txt file to produce a downloadable file of filtered data.

[more software requirements once we have decided upon a structure for our code]

# Algorithm Justification

We have decided to utilise the Matplotlib library to create our graphs, as it would allow us to do so dynamically by simply changing the values of the variables assigned, rather than needing to recode the graph each time a new filter is added. It is our hope that by making use of dynamic graphs, the user can easier draw conclusions from our data, thereby achieving point 5 in our software requirements while achieving point 2.

We have also decided to utilise semi-identical functions for each filter; as we intend to import the data into a 2d array, functions would streamline the code, allowing for a simpler and more efficient program. Each search function – one for seasons, one for months, and one for year – would be almost identical, simply swapping out variables as needed.

[more algorithm discussion once we have decided upon a structure for our code, include pictures of pseudocode/flowcharts]

# Logical Stage

# Testing

[insert discussion of testing, including testing tables, test descriptions, results, and actions taken. Include screenshots of test code, the works.]

# Implementation Stage

# Description of implementation

[insert discussion of implemented features, how they match up to our software requirements, and our design. Include screenshots of code, flowcharts, the works.]

# Evaluation

[insert evaluation here.]

# Project Closure

# Learning

During the project the team’s dynamics changed, many of us were surprised with the skills that we thought we had, those that we actually had and were able to develop as a result of the assignment.

# Conclusion

As a team we felt as though we performed with maturity, respecting and supporting one another we were pleased with the outcome.

# Appendix A

PowerPoint Presentation Slide-pack. [To be added]