SDV602 Assignment 2

Ollie Moss

Contents

[Introduction 1](#_Toc182859237)

[Requirements 1](#_Toc182859238)

[Functional 1](#_Toc182859239)

[Non-functional 1](#_Toc182859240)

[Story Boards 2](#_Toc182859241)

[Test Scripts 5](#_Toc182859242)

[Coding Practices 5](#_Toc182859243)

[Python as a Programming Language 5](#_Toc182859244)

# Introduction

The proposed application is to provide a way to analyse, view and discuss data sets in a scientific and technical context. To achieve this the application will be separated into 3 Data Explorer Screens (DES). A DES will consist of a graph to visualize the data source, various settings surrounding the data will be provided to manipulate the graph in a purely cosmetically way e.g. pan, zoom, columns, title, etc. Each DES will also provide a text chat in which the current viewers of the DES will be able to communicate through. The final application will be provided with a login and signup functionality.

# Requirements

## Functional

* The system will display 3 data explorer screens
* The system will allow users to login
* The system will allow users to signup
* The system will allow communication between users via the chat system
* The system will display a graph based on the selected dataset
* The system will allow navigation between data explorer screens
* The system will allow all 3 data explorer screens to be viewed at the same time using command line arguments

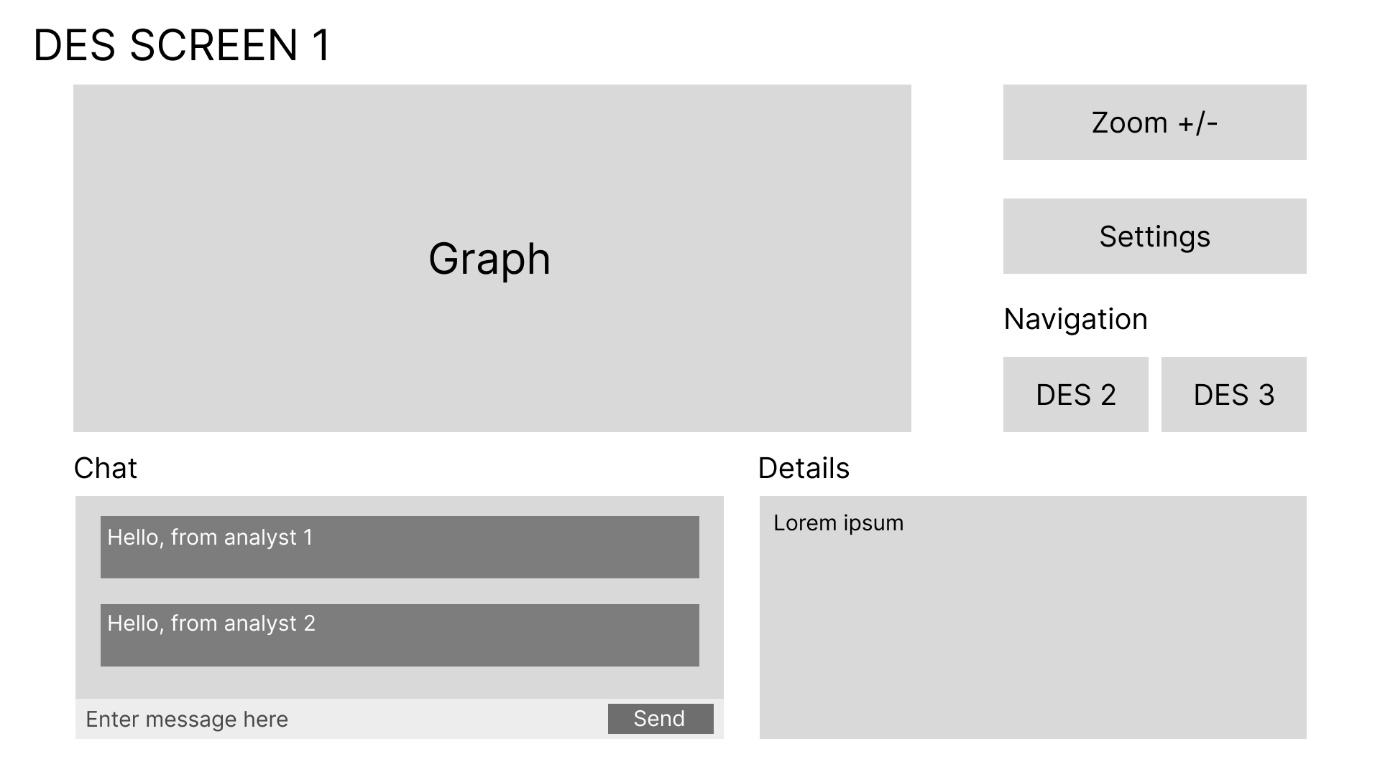
## Non-functional

* The system will use a weather data of Nelson ranging from 1/1/2023-13/10/2024
* The system will provide a good user experience
* The system will follow good coding practices and formatting
* The system will contain useful comments describing each section’s purpose
* The system will be made in a modular manner, allowing for scalability

# Story Boards

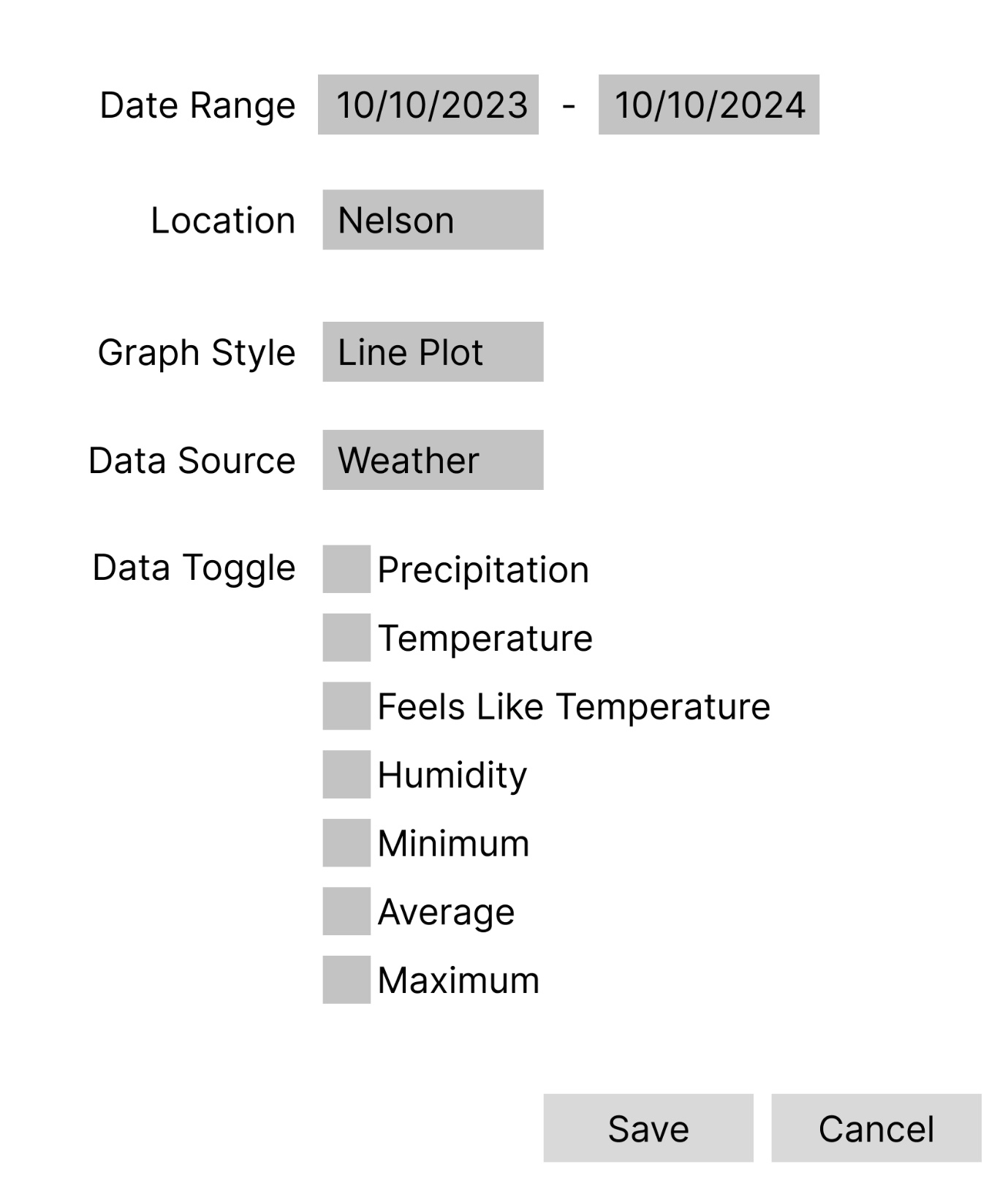
The story boards provided will describe the full functionality of the application including the flow between screens, and the full functionality of each screen.

**Data Explorer Screen:**



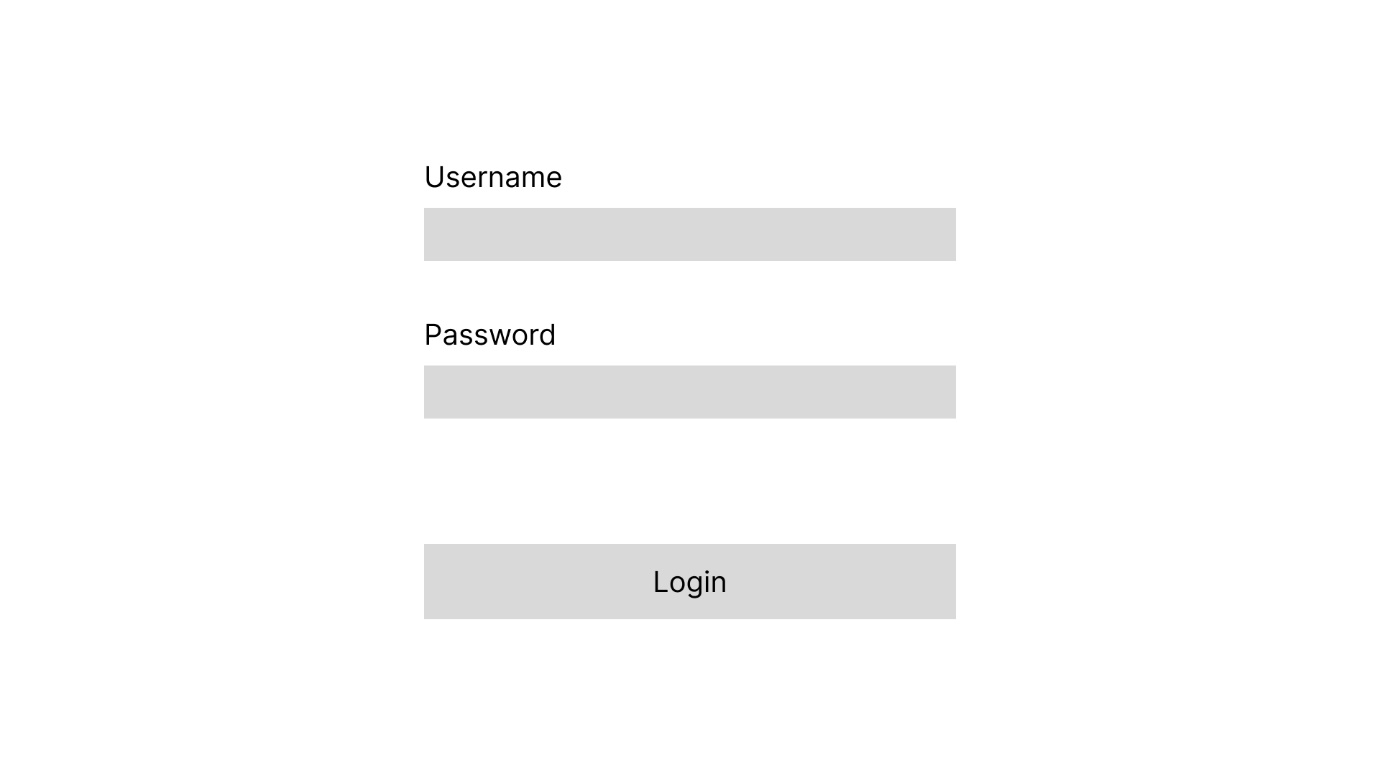
The data explorer screen is where the majority of the time spent while using the application. This screen allows navigation to the other data explorer screens, provides the chat system, and extra details surrounding the dataset selected. This also allows for navigation to the settings menu where the graph can be manipulated. A user will use the chat to discuss the graph with other users and then modify the settings by pressing the settings button, then eventually they may navigate to another data explorer screen in which they will repeat the process until they eventually would like to exit the application using the window exit button.

**Settings Screen:**



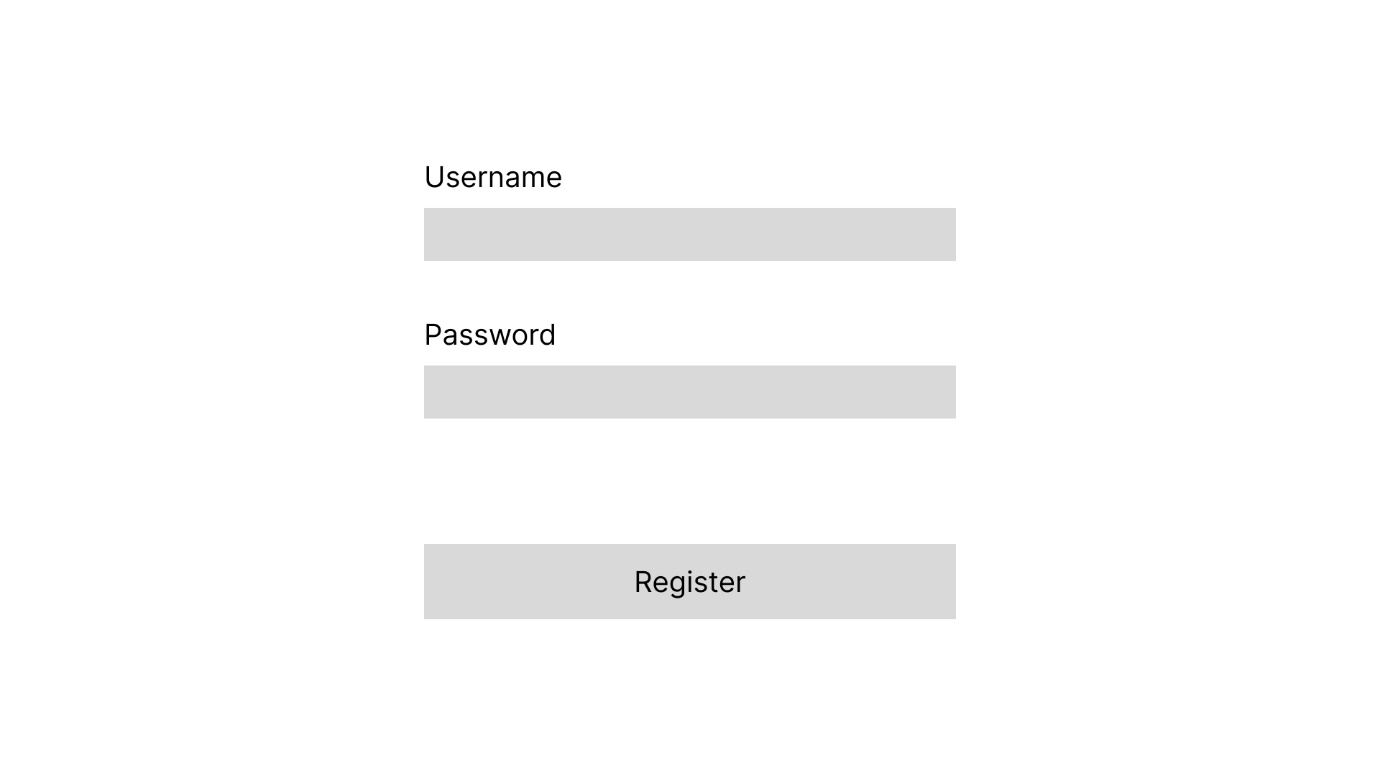
The settings screen allows for manipulation of the graph on a data explorer screen. Users will modify the existing settings to match their desired outcome, if they believe they are happy with the changes they have made they will press save otherwise they will press cancel.

**Login Screen:**



The login screen allows for users to login with existing login credentials. Users will enter their details and press the login button. If the login credentials are valid then the user will be logged in and forwarded to a data explorer screen, otherwise the user will be prompted to enter valid login credentials.

**Register Screen:**



The register screen allows for users to signup, this allows them to make use of the login screen. Users will enter their details and press register, if the username is not already in use an account will be created otherwise the user will be prompted to choose a new username.

# Test Scripts

The provided tests scripts will run the three required DESs, the proposed functionality of the test script is as follows:

* Accept correct input and displays correct information
* Display placeholder for the graph
* Provide navigation to other DES screens, this includes a top-level command interface.

Functionality not provided in the test script is a command line argument for multiple windows at the same time.

# Coding Practices

Comments will follow the pep8 style guide

Code formatting will follow the pep8 style guide

Naming conventions will be clear and follow the pep8 style guide

# Notes

Due to limited time the final hand in is missing some functionality and requirements:

* Comments
* Data source from online database
* Data source per DES
* Settings features

# Python as a Programming Language

Python is a widely used programming languages often used by beginners due to is simple syntax, readability, and versatility. Although used by beginners, it is a very powerful language, that is capable in a wide range of tasks, including data science, artificial intelligence, and more. Python has a variety of strengths and weaknesses, ranging from performance to developer experience, which will be discussed further as the history, philosophy and environment in which Python is used are expanded on.

Python started its development in the year 1989 by Guido Van Rossum and later released in 1991. Initially starting as a hobby project to keep Guido busy over the Christmas, described as the successor to the ABC programming language, became a widely used and depended on programming language. The main philosophy behind Python is code readability and a more enhanced developer experience. This can be seen through the simplistic nature of the syntax and human-like keywords.

Programming languages will often have specialised tools built to support them improving the developer experience, allowing for a more streamlined workflow while still providing high quality results. In cases such as the .NET environment for C# and other programming languages, Microsoft offers Visual Studio, an IDE which features deeply integrated features such as static error checking and auto complete. Various IDEs are built specialised for Python offering optimized auto complete, formatters, and error checking, a popular example of this is PyCharm. Furthermore, the pip package manager is designed to simplify management of external libraries and packages, by streamlining it into a CLI. To further increase developer experience when dealing with external libraries, the Python virtual environment was created. It is designed to isolate and snapshot the version of all packages and Python version within a project. This ensures that any given project should run on any machine given its virtual environment. Although there are specialized tools to make the experience using more simplistic, using a “one fits all” IDE such as Visual Studio Code or in my case Neovim, a smooth developer experience can be had, with simple feature such as syntax highlight, error checking and auto complete.

It features a mix of both strong and dynamically typed language as types matter when performing operations, but variables are not restricted to one static type. Python provides a wide range of capabilities to suit many different programming paradigms such as object oriented, functional, procedural, etc. Through the flexibility and versatility of Python, it provides a lot of strengths when developing applications. Popular use cases for Python include server-side systems such as backends, data analytics, and machine learning. Due to there being a very well-established collection of libraries that are optimized for these such use cases Python has become very popular within these fields and due to Python supporting modules, C or C++ code can be directly imported and used within a Python script. This opens the gateway to vast performance increases, which improves Python greatly as its main weakness is its performance in highly stressful tasks. This is due to the interpreted nature of Python when compared to compiled languages such as C or C++ and is also a byproduct of the various abstractions required to make the syntax, and readability of Python so good.

In conclusion python is a very versatile, and powerful programming language, praised for its simplicity, readability and productivity centred philosophy to development. Python offers a wide range of libraries and a well-developed ecosystem making it suitable for fields such as web development, machine learning and data science. While having limitations Python still thrives in the industry in 2024 and will surely be relevant for years to come as further improvements and discoveries are made within the community.