Design and Development Document

[Company name] | [Company address]

Samuel Blinkhorne, Peter Feehan, Cameron DAvidson

2023

Contents

[**Introduction and Scope** 2](#_Toc141439755)

[**System Overview** 2](#_Toc141439756)

[**State Diagrams** 3](#_Toc141439757)

[**Storyboards** 4](#_Toc141439758)

[**Activity Diagrams** 4](#_Toc141439759)

[**User Interface** 4](#_Toc141439760)

[**Database Design** 4](#_Toc141439761)

[**Prototypes** 4](#_Toc141439762)

[Initial Research Prototypes 4](#_Toc141439763)

[Cycle 1 4](#_Toc141439764)

[Cycle 2 4](#_Toc141439765)

[Cycle 3 4](#_Toc141439766)

# **Introduction and Scope**

This document records the design and development of the product across its prototype development. It contains diagrams and descriptions that show all aspects of the product including the system interfaces, user interfaces, relationship diagrams and state diagrams.

# **System Overview**

The systems functionality is to allow web users to input their current weather data into a system from a given location and then receive an accurate prediction on the current and future weather in their area for that day. The system will use historic data combined with the user input to make its prediction about what the weather will be. It will then display this information back to the user.

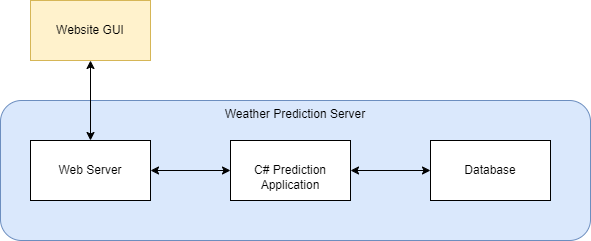
Admins will be able to update the system will historic data to create these predictions.

To achieve this there are three subsystems that are being developed:

Webserver: The role of the webserver is to act as an interface between the internet users and the prediction system. It will allow for a web GUI to be developed allowing for a frontend user interface to make the system user friendly

Prediction Application: This is where the calculation and predictions will be carried out. This subsystem will have an interface with both the webserver and the database to allow for data to be sent and received from both sources for creating predictions and forwarding them to the user

Database: This is the store for all of the historic data and the user information that can be accessed by the prediction app and if needed forwarded to the webserver.



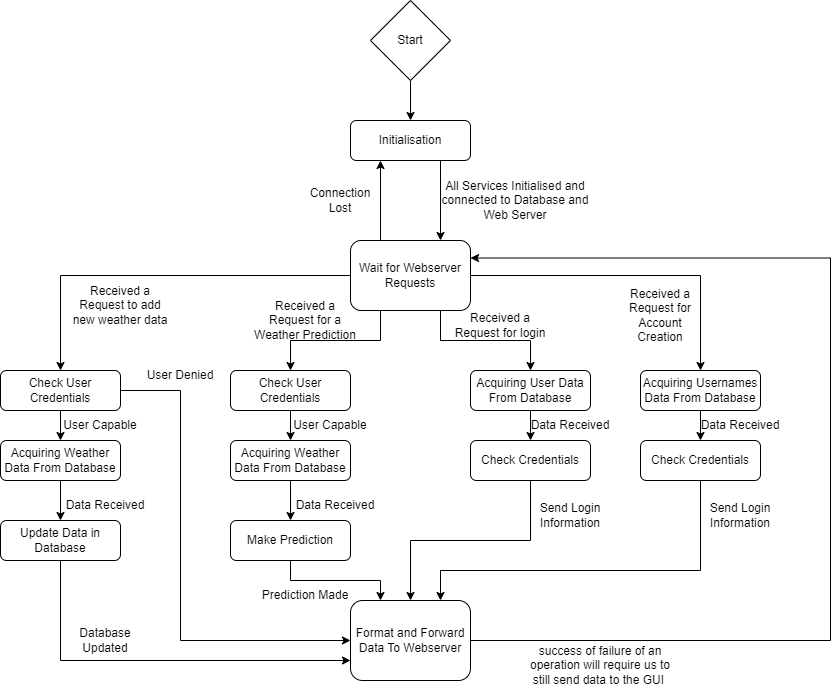
For simplicity and for the demonstration all three subsystems will be run on the same machine which will be known as the ‘Weather Prediction Server’

The Webserver will be hosted using **XXXX**

The Prediction application will be written using C# as this offers the most flexibility in terms of plugins and libraries for connecting to webservers and databases.

The Database will be created using SQLite as there is not a need for a heavy-duty database.

# **State Diagrams**

****Below is a state diagram developed to show the states and transitions for the Prediction Application. This diagram demonstrates its responsibilities as it handles requests from the webserver. From the diagram it can be seen there are four main types of requests from the webserver. A request to change database data, a request for a weather prediction, a request to log in and a request to create an account.

# **Storyboards**

# **Activity Diagrams**

# **User Interface**

The first decisions we had to make regarding the design of the GUI was to agree on the general layout of the GUI. This included how we would want our pages to be navigated to, whether we want to use/create any images, and what colour scheme we want to implement. Very early on we knew we had to consider mobile users due to the client’s requirements, this meant we would need to make our GUI mobile friendly, to do this the page had to be margined at the sides and centred which would allow smaller screens to still see the content with little inconvenience.

Once these were agreed on, a first draft of the GUI was created. This was then presented to the rest of the team where we would discuss any adjustments we want to implement. An example of this would be the layout of the navigation bar. Originally, all the links on the bar started from the left, however, it was decided that the login/signup link should be separate and located on the far right of the page, so it is easier to see and get to quickly.

# **Database Design**

The Database is designed using SQLite

Below are the relationship diagrams for the tables in the software database.

There are two main tables in the software that are used. These are the User Data and then Historic Entry Data. This simple database allows us to keep track of our user details and allow us to generate historic data.

By users having a unique ID it allows us to associate Historic Data with particular users.

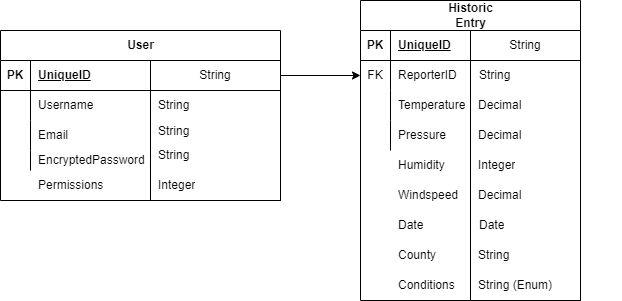
The User data has a username field for users to use as a sign in

Email field for account recovery and association to a specific user

Password for securing a user’s account

Permissions for determining whether a user is an admin or not

The historic data entry has all the relevant weather data including Temperature, Pressure, Humidity and Windspeed. It also includes the date, county and a Unique Id.



With this an interface has been developed to allow for data to be sent to the database via the application in a standardised format below is a table of the data set that is being sent to the database and the format the data should be in when the database ‘AddData…’ functions are called

|  |  |  |
| --- | --- | --- |
| **Data Set** | **Data To Include** | **Format** |
| User Data | Username, Email, Password, Permissions |  |
| Weather Data | Temperature, Pressure, Humidity, Windspeed, Date, County, Condition |  |
|  |  |  |

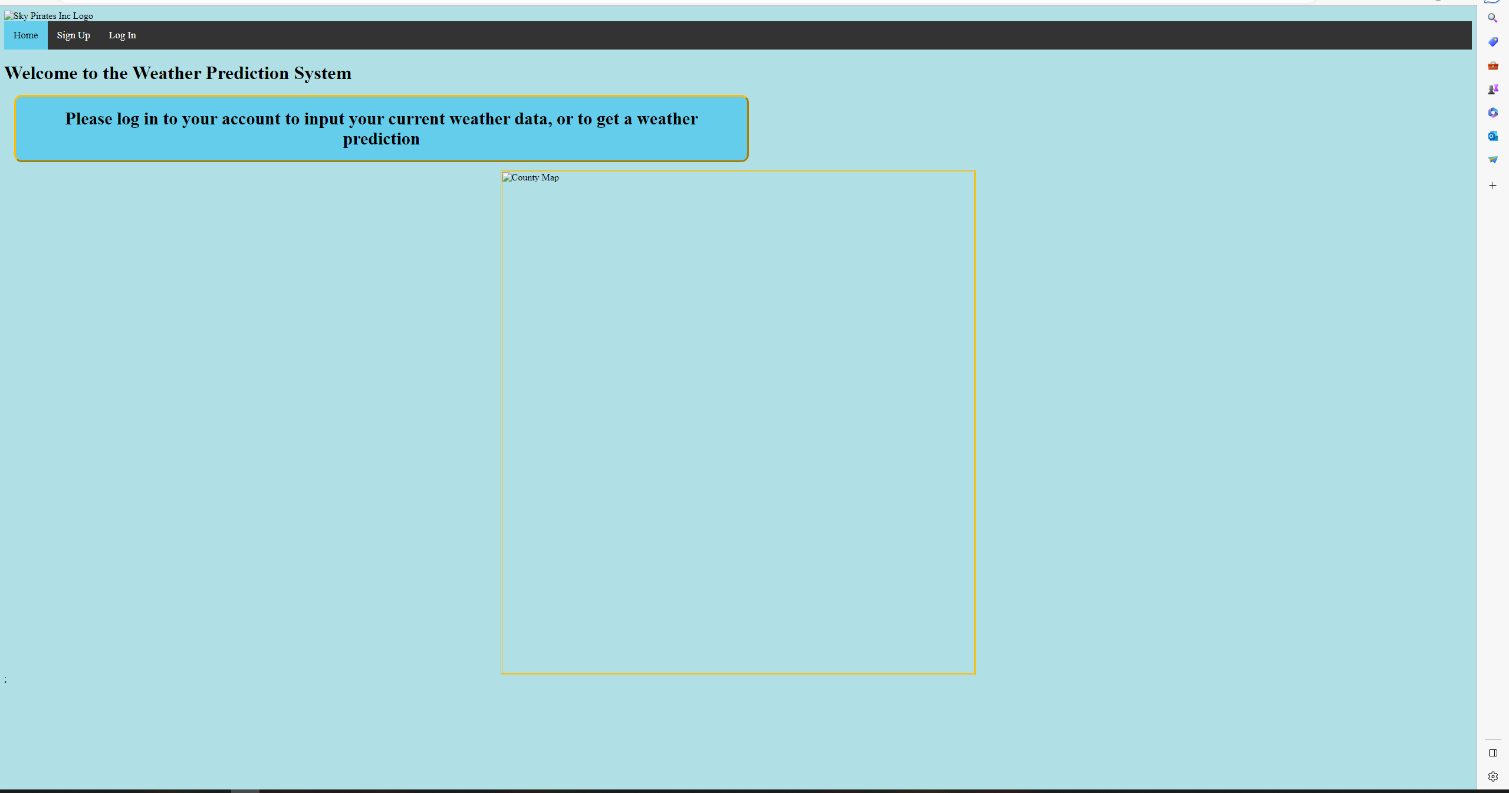
# **Appendix - Prototypes**

## Initial Research Prototypes

This first of the initial prototypes that had to be developed was an initial GUI layout. This was done using a 2-way approach. The first was developing a design drawing with low fidelity to create an outline of the initial parts of the GUI we were looking for. The second part was to develop this with the HTML along side of it. The reasoning for this is the team has a low level of expertise with HTML so this allows us to gain expertise from an early phase of development which will make prototyping faster in later cycles.

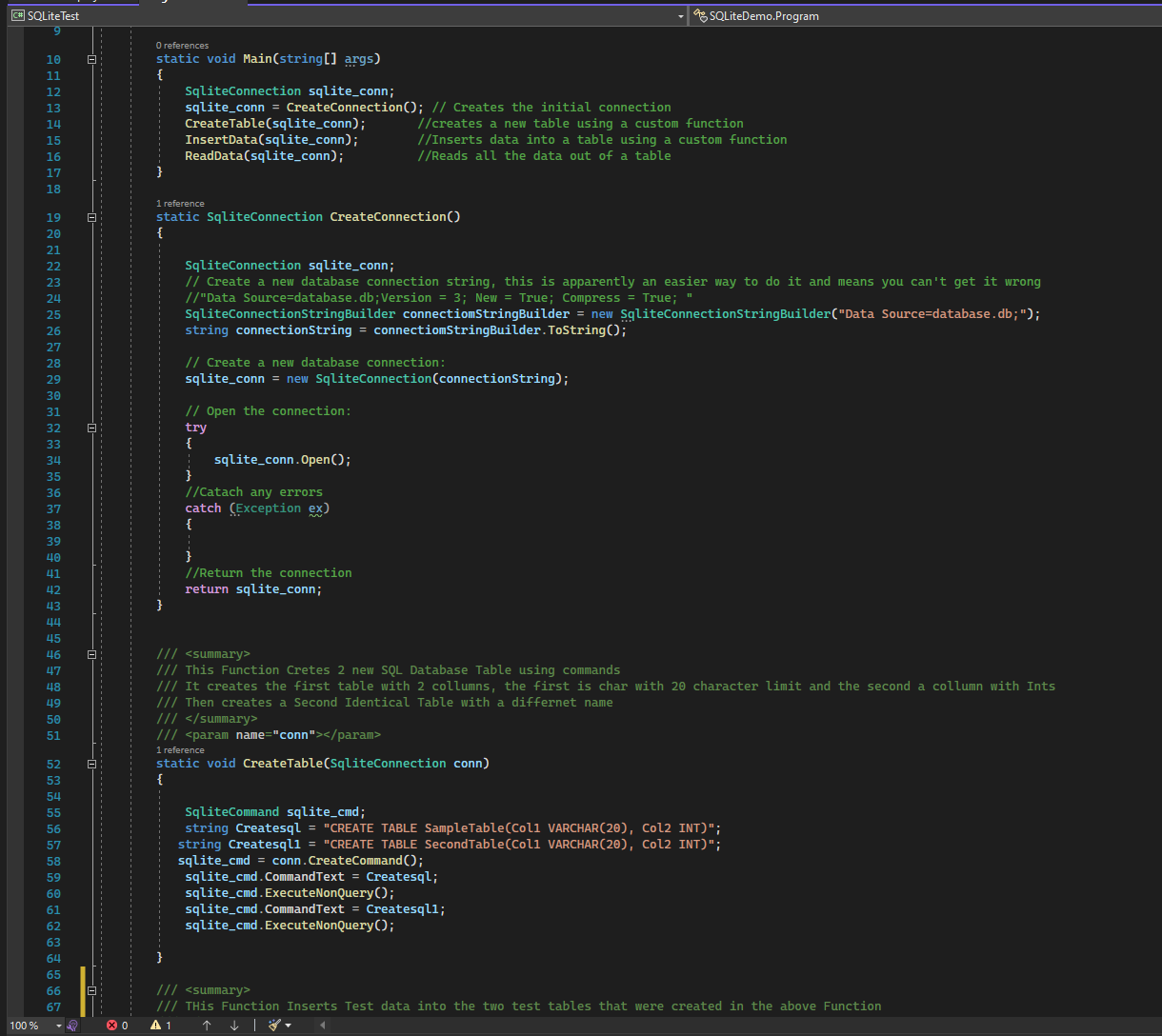


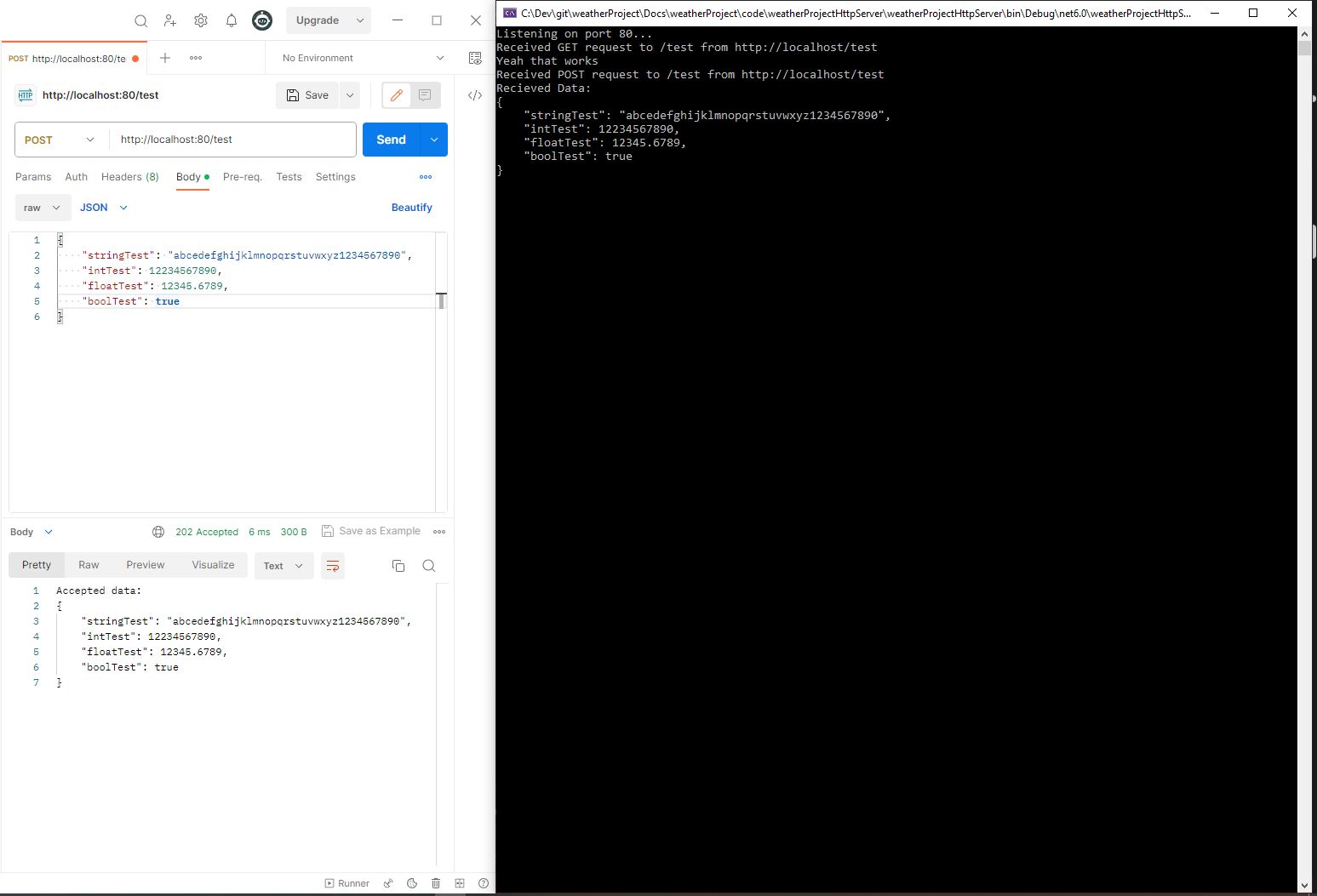
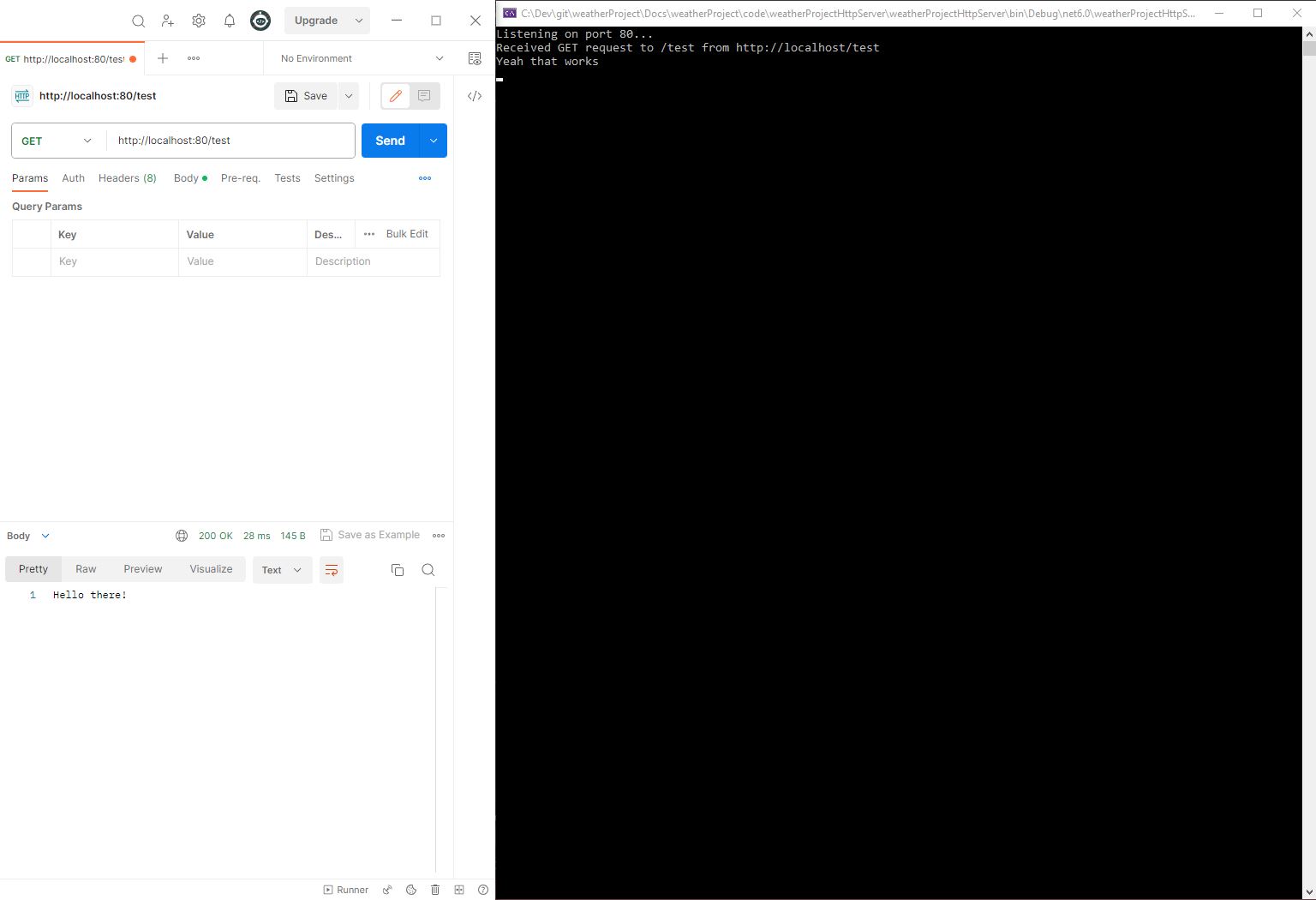
Screenshot of Initial HTML Code

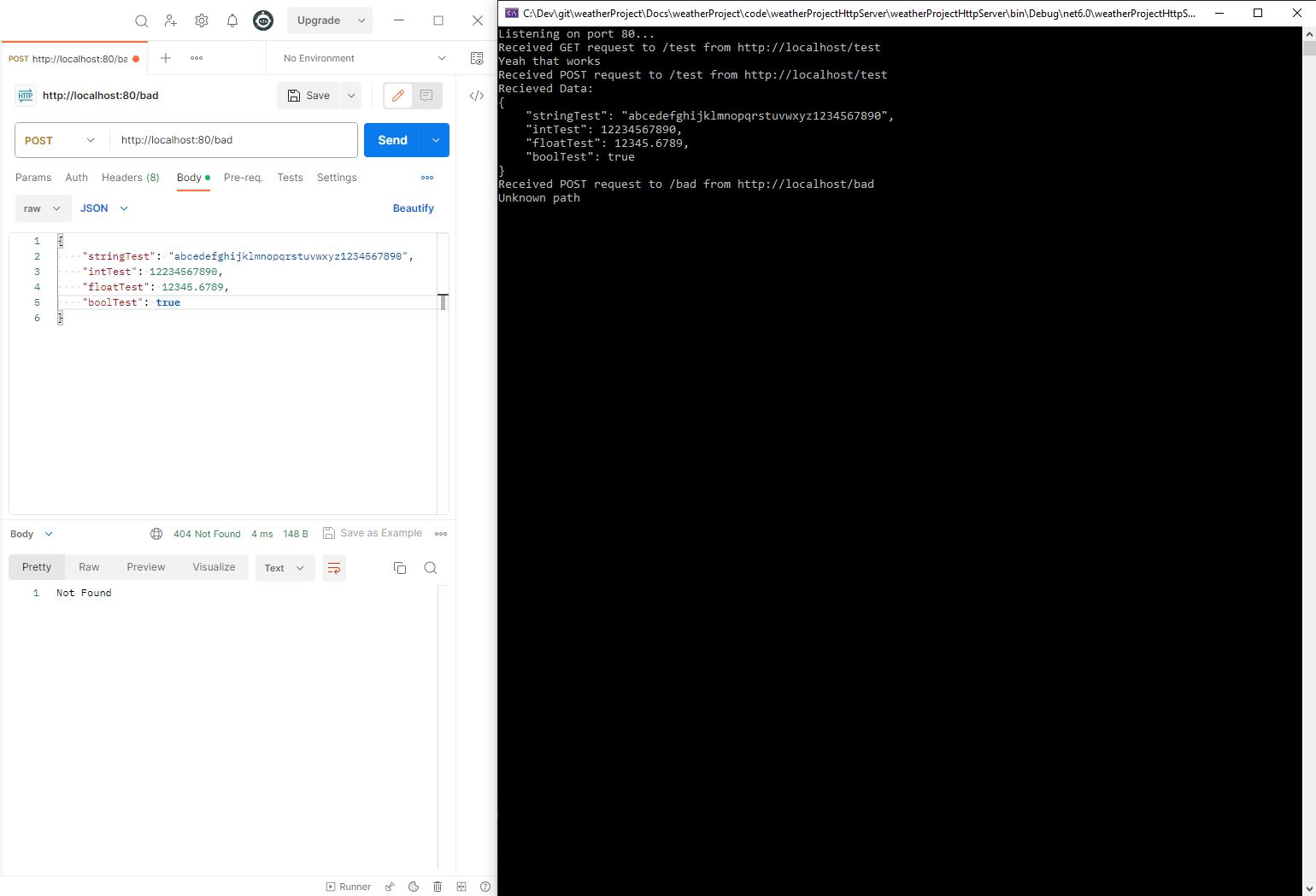


Screenshot of Initial HTML Code Output

The next Initial Prototype was allowing SQLite to be used in C# this was created with the use of online sources to get the interface working correctly. So, then it could be understood by the team, commented and then tailored and implemented into the prototype.



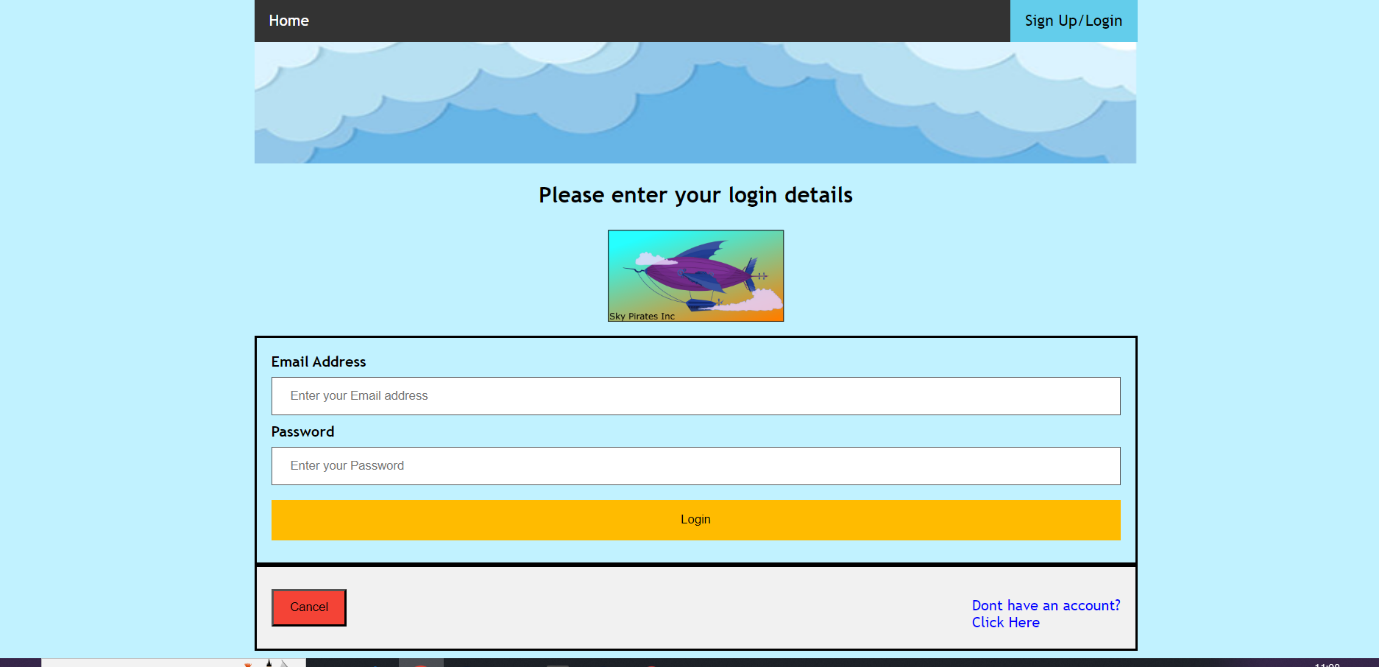
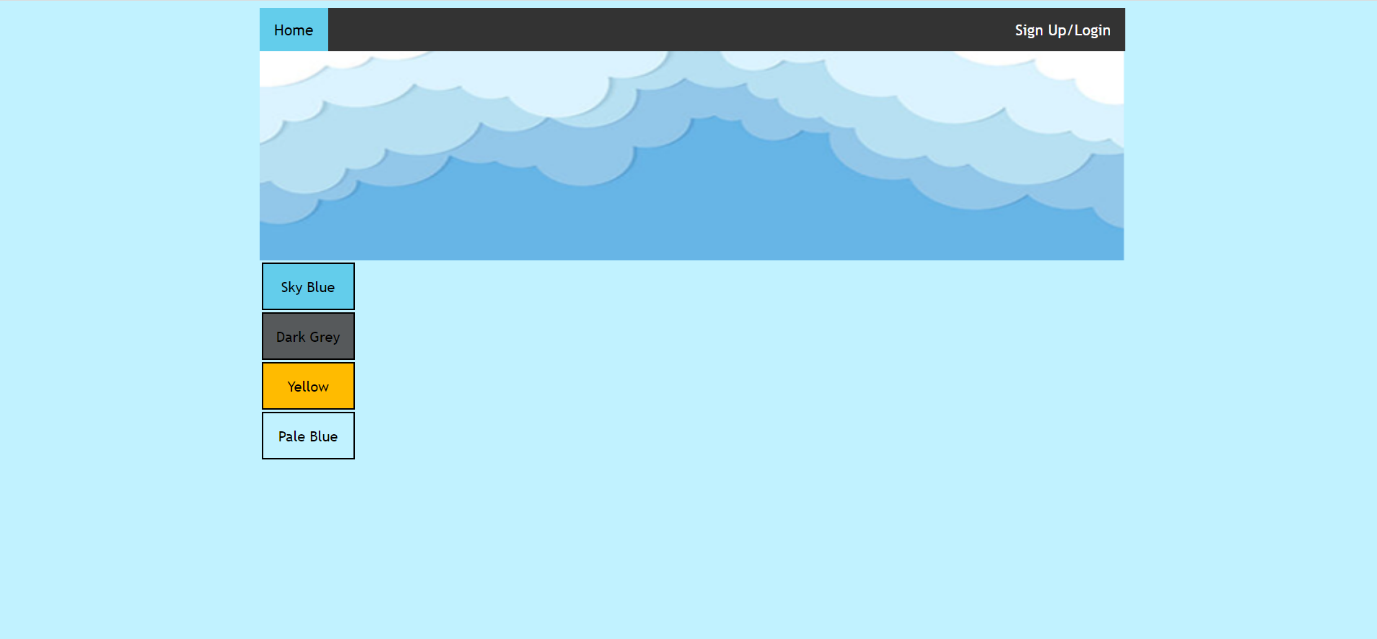
The third of our initial prototypes was making sure that we could set up a webserver, this was done using XXXX tools and creating some C# code to allow for it to be created programmatically. It was then tested that posts and gets could be done correctly through this interface. **PostMan**

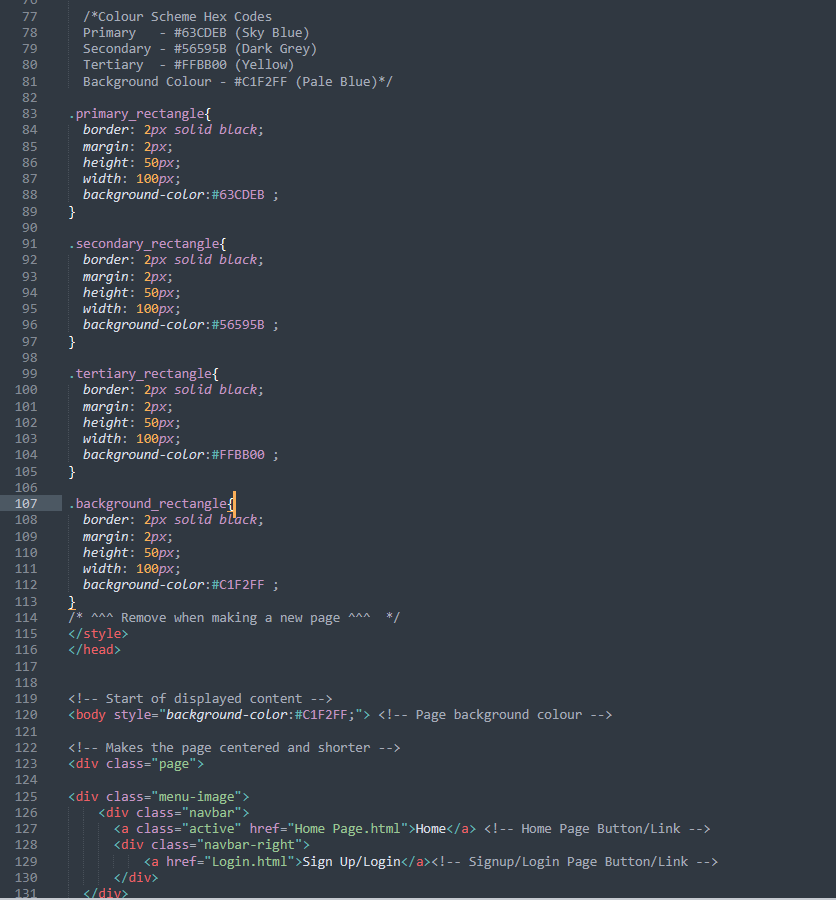


## Cycle 1

For Cycle 1 there are a number of developments that were made. The first was the finalisation of the initial design of the GUI. This allowed us to create a skeleton page, so that new pages could be developed in future cycles easier. It had our base task bar at the top of the page, our general theme and the colours we would be using for the project.

To add on to this the pages that were developed for this page had their user input boxes created and the pages linked together. So a feel for how the GUI functioned was developed.





## Cycle 2

## Cycle 3