SDV602 Project Milestone 1

Mark Christison

Nelson Marlborough Institute of Technology, New Zealand

**Contents**

[**Contents** 2](#_Toc79994188)

[Description of Application 2](#_Toc79994189)

[Scientific Motivation 2](#_Toc79994190)

[Storyboards 2](#_Toc79994191)

[Login 2](#_Toc79994192)

[Data 2](#_Toc79994193)

[Chat 2](#_Toc79994194)

[Settings 2](#_Toc79994195)

[Test Scripts 2](#_Toc79994196)

[Coding Practices 2](#_Toc79994197)

[References 2](#_Toc79994198)

# Description of Application

## Scientific Motivation

There has never been a more important time in history to pay attention to data and numbers, specifically in the case of the Covid 19 pandemic.

Watching the number count, eagerly awaiting the mid-day pressers held by Jacinda and Dr. Bloomfield to give us the daily update on the COVID-19 situation became normal in New Zealand in 2020, and again in 2021. All we wanted to know was anything about the situation and more importantly the numbers of cases.

Being able to view these numbers allows for scientists to extrapolate and build models to allow for predicting where services will be needed. These models and numbers can also be used by businesses to plan and prepare for lockdown situations. The numbers overall became something of common place discussion amongst people as well.

Having ways to display this information, to show trends and make accurate predictions is critical to not only scientists but everyone in society.

The application that I will be building will take data from Center for System Science and Engineering at Johns Hopkins University, specifically their repository on GitHub (CSSEGISandData - Overview, n.d.) which is kept up to date with daily reporting of numbers from around the world.

The application will be built with Python and the framework for creating graphical interfaces called PySimpleGUI. The benefit of using this framework is that it provides methods and functions to interact with 4 different python interface builders using one set of combined methods. The 4 underlying interfaces that Py tkinter

## Login

As a requirement of the application, a login feature has been requested. A user will need to first register an account with at minimum a username and password. To access the data explorer, the user will have to enter the username and password. This username will be displayed on the chat window whenever the user chats.

Data Explorer

The requirements specify a minimum of 3 data explorer screens. These screens will display the data to the user in graphs or other visualizations. The graphs will be able to be displayed differently to the user based on different settings that they could adjust on the window such as dates, or country to view.

Chat Window

Having a chat system is also specified as a requirement of the application. Users of the application will be able to chat with one another in real time, with the app using API calls to get and send messages to a remote source in the final version of the application.

# Storyboards

## Main Screen

This is the screen that is presented to the user on opening the application.



1. A common element that all screens have is the top menu bar which allows navigation throughout the different screens that the application has
2. On the home screen there is 2 buttons that prompt the user to engage and interact with the application. This button prompts the user to login
3. This button prompts the user to create an account if they do not have one already

## Login



1. On the login screen the user enters their username
2. The user can enter their password
3. When the user clicks ok, the application checks where the users input matches a valid user in the database, if so, they can proceed to the data explorer screens. Otherwise, they are presented an error message
4. The user can return to the previous menu without logging in by clicking this button

## Register Account



1. The user can enter a desired Username
2. The user enters a desired password
3. On clicking this, the application checks the database to ensure that the username is unique, if so, it creates the account with the provided details, else it displays an error.
4. The user can revert the account creation and return to the main screen

Data Explorer Screens

As identified in the requirements for the application, the data explorer screens are for this milestone to only include a graph as a place holder. I cannot see why all the data explorer screens cannot be the same but show data from a different source. As such, until later versions of the application are required, I have replicated the data explorer screens. The interactions are listed below.

The data explorer screen is indicative only and is subject to change. It may include different ways to manipulate the data or different buttons based on the data that will be displayed. Included are sample buttons as was to manipulate the data on the screen.

DES 1



1. The Graph, which as shown is a placeholder. This displays the data that will be displayed to the user.
2. A Country selector. This allows a user to scroll through a list of countries and select 1 to display on the screen.
3. This button allows the user to pick a start date for the graph. Clicking it displays a date picker.
4. This button allows the user to pick an end date for the graph. Again, clicking it displays a date picker.
5. This button updates the graph by fetching the remote data and updating the graph with the selected values the user inputs.
6. This button resets the graph to default settings.
7. This is the navigation bar, on all major screens in the application.

DES 2



DES 3



Chat Window



1. This field is where the user can type their message that they want to send to the other users
2. The send button is pressed to send the message. This will also be mapped to the enter key.
3. All user messages are displayed in this element. The username they created will be the name displayed before their message
4. The navigation bar that allows the user navigation to other windows in the application.

# References

*CSSEGISandData - Overview*. (n.d.). GitHub. Retrieved August 27, 2021, from https://github.com/CSSEGISandData

‌