For the first part of the assessment, I have been asked to create a storyboard, depicting each stage of the application and what each element will do. I have also been asked to give a brief run down of the application, such as what libraries will be used and which data source I am going to use.

To begin, I will discuss the different Python GUI libraries that are available for this type of application, and I will explain further why I have decided to use the library I chose.

There are many Python GUI languages out there to choose from. A few of the most popular ones being PySimpleGUI, PyQt5, TKinter and PyForms. I have decided to use TKinter as it is one of the oldest Python GUI libraries and offers a lot of documentation on how it works. It also works very well with MatPlotLib, the library used to turn data from a CSV file into an easy-to-read plot depicting different information. I have also used PySimpleGUI in the past, and recently I found out that PySimpleGUI is built on TKinter, so I figured it would be more beneficial to learn the original language as I would be using TKinter functionality on PySimpleGUI anyway.

For the data source I have chosen to use a CSV file containing information about crimes in Atlanta. I have chosen this CSV file as it displays a lot of information such as crimes, the area they were committed, it also contains data from 2009-2017. This gives me a lot of data to work with when creating my 3 DES. I have decided to create 3 separate CSV files from the original file as the original file is very big and takes a long time to load, although this can be changed depending on the feedback from my tutor Todd.

## Practical

Data cleaning this CSV was very easy, it was well formatted, and I did not need to replace any blank fields. The hardest part was deciding what plots I wanted to create with the data. I decided to create 3 plots: The highest crime areas, total crimes per year and the most common crimes. I wanted to create 3 different types of plots, but I was not able to find a plot that fit this use case. The most useful were pie charts and horizontal bar charts in my opinion.

Creating the GUI was very easy as I used the official TKinter documentation; <https://docs.python.org/3/library/tkinter.html>

This allowed me to find out how to create GUI elements, make multi-level GUI applications and most importantly, how to add MatPlotLib plots to TKinter widgets; <https://matplotlib.org/3.1.0/gallery/user_interfaces/embedding_in_tk_sgskip.html>

The next objective was to link each of the windows together but only to open when the user specifies, they want to open it. For this I used TKinter’s option menu and an array containing the 3 different plots, I then created an if/else statement to select each one.

Lastly, I had to make sure each window opened in a specific location each time so the user didn’t have to find each screen once they opened it. For that, I used the built in TKinter attribute named geometry, this allows the creator to specify the dimensions of the window, as well as setting where in the screen it will open. <https://www.geeksforgeeks.org/python-geometry-method-in-tkinter/>

## What’s next?

The next objective is to create the log-in screen. I will be using SQLite as it is built into Python. This will create a database file in the application directory. Another option is to create a remote database and connect to it in our application, this would allow the user data to update remotely allowing it to be live updating. I will need to talk to Todd about these options.

I will also start on the live-chat aspect of the application; I have briefly looked into it, and it seems like I will need to use the socket and threading libraries available from Python. Using these libraries will allow the chat system to be live updating.

I will also start creating the functions that will handle the user inputs, the reason why I have chosen to use 3 separate screens instead of updating one widget is so I can create separate functions for each window.