|  |
| --- |
| H446 Programming Project |
| F1 Fantasy League |
| Lord Williams School |

|  |
| --- |
| Thomas Draper  7-7-2021 |

Table of Contents

[Analysis 2](#_Toc80105964)

[Problem and Explanation: 2](#_Toc80105965)

[Stakeholders 2](#_Toc80105966)

[Interview with stakeholders 2](#_Toc80105967)

[Research 2](#_Toc80105968)

[API 2](#_Toc80105969)

[GUI 4](#_Toc80105970)

[Client Server Networking 4](#_Toc80105971)

[Comparison of Existing Programs 9](#_Toc80105972)

[Features and Limitations 10](#_Toc80105973)

[Design 10](#_Toc80105974)

[Server Login 10](#_Toc80105975)

Analysis

Problem and Explanation

Stakeholders

Interview with Stakeholders

Research

Comparison of existing solutions

Features and limitations

Design

Loading Screen

Main Menu

Client Server

Classes

Algorithms

Web Scraping

Data Analysis

Development

Evaluation and Testing

Testing

Stakeholder feedback

Final evaluation

# Analysis

## Problem and Explanation:

For my project I am planning on making a F1 Fantasy League Game similar in which Fantasy Football works, but for Formula 1. It works by assigning each player a price based of their performance in the sport, so the user creates a team and the better the team of drivers perform in real life, the more points that team receives. So that you can compare your teams points with your friends and others. I am planning on having a chat built in so that users can talk to each other about their teams or the sport.

## Stakeholders

Due to the scope of my project my target audience is any enjoyer of Formula 1, and the chat will have a profanity filter option so that any ages can use the program.

I have selected a group of F1 fans of ages 16-17 to represent my target audience so that the program matches what they would expect and want out of a F1 Fantasy League, this will help get information on what they would like and dislike when I present prototypes of the program, this will help me stick to what the target audience would want.

## Interview with stakeholders

(Add Interview)

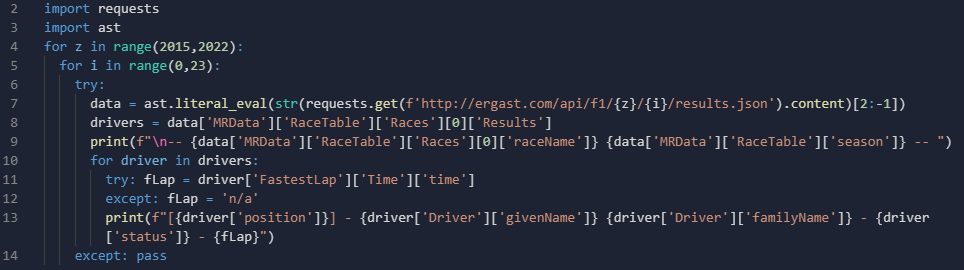
## Research

### API

When thinking of the project I knew I had to find a way to get all the data needed, such as race winners fastest lap and lots of other data to help compile prices and points system. When searching for a way to get this data I came across a website called [Ergast Developer API](http://ergast.com/mrd/), This is a free API that can provide years of data and is updated after new races.

I thought I would try and use the API to make sure it was easy to navigate and easily worked with python (so I got It to retrieve all year’s race results since 2015):

Code:



Defines what years you want to search and how many rounds each year.

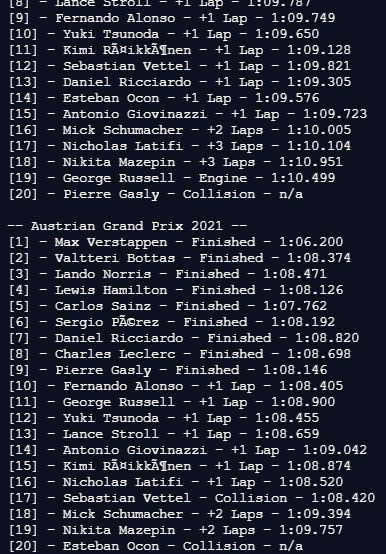
Sends an API get request to receive the json data which gets converted into a dictionary data structure.

Shows the position, name, end reason and fastest lap time of each driver.

Shows which race the driver data is coming from.

Console output:

As there was a lot of data outputted this is just a snipped of the output near the end.



Shows each divers position at the end of each race.

Displays the drives name and their reason for ending the race.

It also displays each drivers fastest lap, this will help provide extra points if the data is analysed.

This provided a good insight as it showed that the free API I found, provided all the data I would need and would work perfectly for getting the data needed, that can be processed and stored on the server.

### GUI

I looked at many GUI modules for python or using a web-based interface using Django. I looked at using Tkinter as I was taught the basics of this GUI module, so I could easily learn the enough to make my project, but I decided against using Tkinter as it had a very old and outdated looking design and code was hard to follow. So instead I looked at PyQt6 as this looked much better, good support for widgets and it also had build in software which helps mould the GUI to whatever I need.

I decided to make a simple word processing program to test the viability of using this as my primary GUI tool.

#### QTDesigner

(Add photo of code and explanation)

#### Intergration

(Add photo of code and explanation)

### Client Server Networking

I needed a way to send data to the clients so that the data on the drivers and their team can be transferred from the server to ensure no cheating can occur and also so that a functional chat group could be active and the only way to do this was through a client-server network

#### Program Login

So that users fantasy team data can be accessed from any device wherever, we have to have some kind of login system so that players can save their data to a server, and it be sent back when they login using there unique credentials.

I decided to test if I would be able to make this system, to I made a simple client server login page that asks for a name and pin to access.

**Server.py:**

The server code sets up the connection and waits for an input, to which is processes the result and returns whether it’s a valid login or not.

The server.py as a test only has a 2-dimensional list to store the correct name and pin, this is not secure or efficient for my project but as I was testing whether I could create a system like this using client-server connection it was enough to store the users name and pin to be searched through.

Code:



Send back accepted or denied to the client.

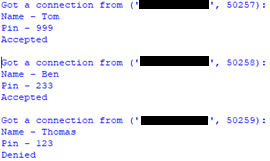
Searches logins for any matches with the data they sent

Defines Host IP and port

Simple client name and pin storage

Accepts connection from client

Console output:



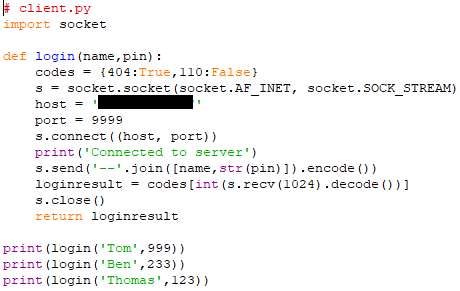
Shows the inputs and the result from the server.

This was the only input to be incorrect and the code spotted this and denied access.

**Client.py:**

Connects with the server and send the inputs it receives, then waits for a response from the server. After which it processes the response and outputs a Boolean True/False response.

Code:

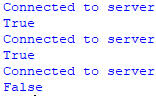


Defines host IP and port that the client needs to connect to.

Sends the name and pin provided, they had to be sent in one string. Then receives the result processes it and returns True or False

The provided test set to see if the program works correctly.

Console output:



The client shows that the last one was denied and the rest where accepted.

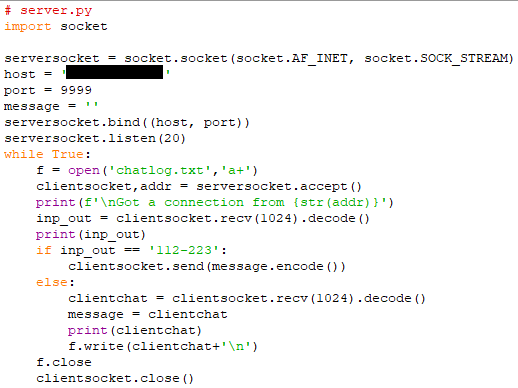
#### Chatroom

I decided to add a chatroom so that the users can talk to each other and give more reasons to use my f1 fantasy league rather than alternatives.

I decided to make a mock-up of a quick client-server chatroom to show the basic aspects.

**Server.py:**

The server has to receive each chat message from the clients and redistribute that message to all the clients so they receive what was sent by one user.



Opens a text file that will log all messages received.

Receive code ‘send’ if the client wants to send a message.

If client wants to receive messages, then the sends the last message sent.

Adds to chatlog and updates message variable based of message what’s received.

Defines host IP and port that the client needs to connect to.

**Client.py:**

The client has to send the server each message that the user inputs, and also receive and display any other messages sent.



A subroutine to send messages to the server, which connects ,sends a command then sends the string to the server.

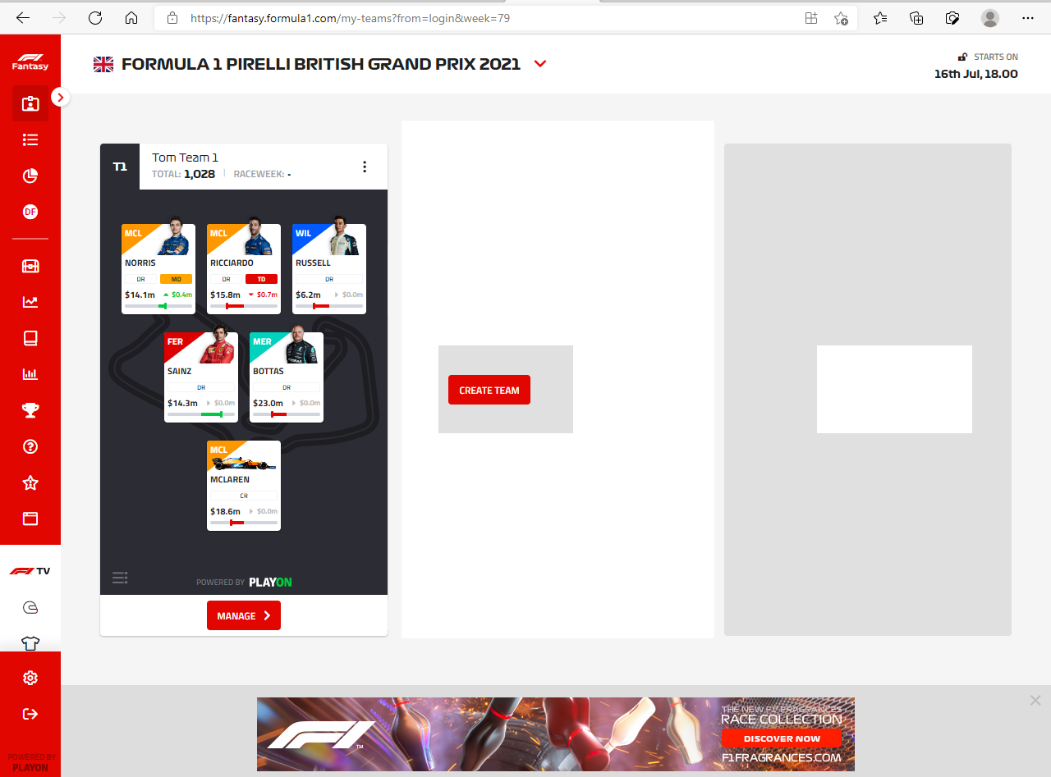
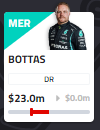
To run the message subroutine constantly.

A subroutine which sends out a receive code so that the server sends back the last message it has received, which the client checks if it is a new message before displaying it.

This runs both the message\_constant and receive\_messages subroutines at the same time using multiprocessing.

## Comparison of Existing Programs

Currently there is a similar [Official Formula 1 Fantasy](https://fantasy.formula1.com/) web application that is similar to what I would like to achieve, I hope to compete with this program, but it is unlikely as their program has a large corporation behind them, so I cannot create a just as fleshed out project, but I will try.



Shows how the price/value of that driver has changed recently, to help people make good decisions about their drivers

Shows each drivers name what team they are in and a photo of them, to help anyone that may be confused.

Shows your created teams,

Shows next race to be held.

The Official Formula1 Fantasy program has lots of good features but does not provide a chatroom to be able to talk with other people about the sport or about their program, this is what my project will include to bring to community together.

When browsing the website, I liked the recent price/value change as this helps people make informed decisions to add their drivers, so I have decided to try and add this to my project to show the change in price over 2 races.

I also saw that they have lots of animated graphics that I will not be replicate, as I do not have any experience in that field and it would take too much time to incorporate.

## Features and Limitations

I would like my project to incorporate:

* A client-server chatroom.
* Server-side web scraping/API usage to get the data needed.
* A price system that analyses the data from races, to give each driver a price.
* Shows what the next race is and when it occurs.
* A good-looking GUI to show all the data needed and provide a way around the program.
* A login system so that user’s data can be transferred to different machines.
* Basic encryption of data that is sent between the server and clients.

Some of the limitations I would face are:

* Lots of animated graphics
* High security encryption
* Learning how to create a GUI that supports what I need.

# Design

## Server Login

For my login page I would like to have a name and pin login system so that the name can be used later for the chatroom.