Jakub Zurakowski, Michael Sibbald and Alan Cahill

C00205431, C00206817, C00174533

4th Year Data Science Functional Specification

Investigating Baltimore Arrest Rates

Institute of Technology, Carlow

Table of Contents

[Objectives 1](#_Toc527484568)

[Architecture 1](#_Toc527484569)

[Functionality 1](#_Toc527484570)

[Functionality Description 1](#_Toc527484571)

[User Interface Description 2](#_Toc527484572)

[External Interfaces 2](#_Toc527484573)

[Potential risks / issues 2](#_Toc527484574)

[Gantt chart / timeline 3](#_Toc527484575)

[Reference/research documents 3](#_Toc527484576)

# Objectives

The application will be for the use of the Baltimore Police Department [1]. The application will provide the police department with predictions of future crime via a secure login. These predictions will include data such as, where the crime will happen, when it will happen, who will commit the crime and what type of crime are they most likely to commit. The objective of this is to help police reduce crime. It is hoped that by analysing the predictions of crime the police will be able to make more informed decisions on where to send patrols at certain times. The predictions might also help the police in profiling the likely criminal offenders.

The application will also provide a service to the public allowing them to see a heat map of which areas they should avoid due to high levels of crime as well as provide statistics on the crime and arrests committed throughout Baltimore, visible to the public. The objective of the map is to improve public safety and to allow potential victims to avoid dangerous areas. The objective of the statistics is to show the arrest rates and other statistics to show the progress that the police department is making.

# Architecture

The application will use Python 3.6 Flask [2] for the backend side of things. The data will be stored in a JSON file for development and in MySQL [3] database for production.

The application will use yet not decided machine learning technique for data processing and making predictions.

The frontend of the application will be created using Jinja2 [4] templating language in conjunction with HTML, CSS and JavaScript. The frontend will require some external libraries including jQuery and the Google Maps API library. [5]

If given time, the frontend application may be created using Angular listening to the backend Flask RESTful API instead.

# Functionality

## Functionality Description

The overall functionality of the application is to perform analysis on Baltimore arrest data that is provided on their website. The first step is the downloading of the data and putting it into a Database, or JSON for local use. The data will then be read into a python program.

This application will manipulate the data and perform various analysis using TensorFlow [6]. The results of these analysis’ will then be turned into graphs or visualised in other ways. The other methods of visualisation include displaying a heat map of arrests overlaying a map of Baltimore using the Google Maps API. The program will then proceed with applying machine learning techniques which will perform predictions of crime and arrests.

There will be an authentication process for the police department which will allow the police department to see the predictions made. The details of the created users will be saved in an SQL database using SQL statements. When the user is logging in the details given will be compared to the details that are in the SQL database. Only certain parts of the application will be available without authentication. The application will need to focus on security, making sure no confidential information can be accessed from outside.

## User Interface Description

Upon arrival at the web application the user will see the Home page. This page will consist of a description of the project. At the top of the application, on every page, there will be a navigation-bar. This navigation-bar will contain links to home page, graphs page and the heatmap of Baltimore. At the very right-hand side of the navigation-bar there will be a link to the login page. If the user is already logged in there will instead be a drop-down menu titled with the user’s username allowing the user to logout. A logged in user will also have a further link on the toolbar allowing them access to the Predictions page. Clicking this link brings the user to the Predictions page.

The Heat Map link will bring the user to the heat map page which will contain the heat map of Baltimore using Google Maps API.

## External Interfaces

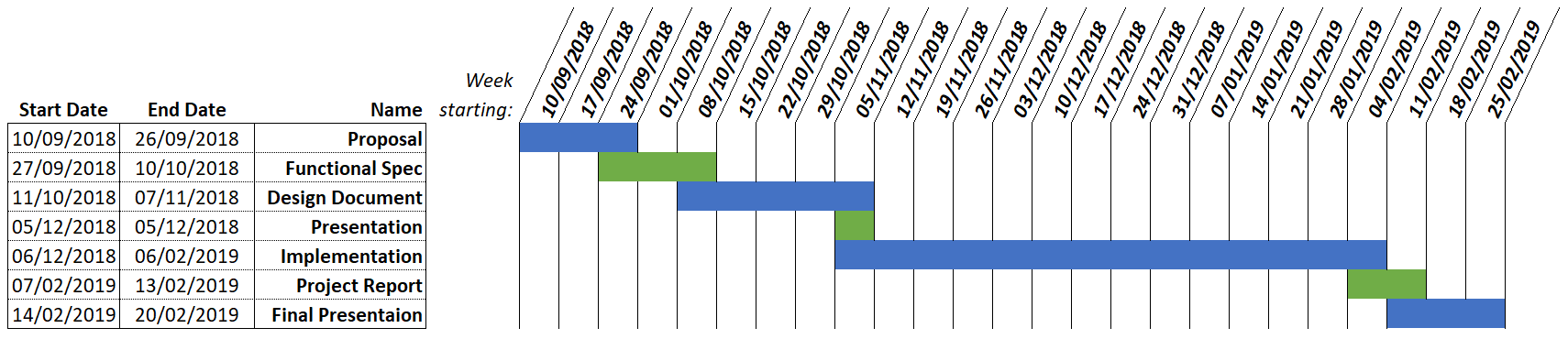
The application will make use of Google Maps API library to display a heatmap of Baltimore indicating the ‘hot zones’ of criminal activity.

# Potential risks / issues

The first potential risk which has already been solved was the potential use of criminals to use the application to organize their criminal activities to avoid the Police patrols. This was solved by adding in an authentication process to view the predictions and recommendation of patrols. This of course raised a second issue of hacking. If a criminal could gain access they could again use the application for criminal activities.

Another issue is the cleanliness of the data. There are some entries that have null values. This could cause some potential problems when manipulating and analysing the data. The main data column that is missing data is the arrests location, these locations however are present in the majority of the entries in the data set.

# Gantt chart / timeline



# Reference/research documents

**[1]** Home | Baltimore Police Department. 2018. *Home | Baltimore Police Department*. [ONLINE] Available at: <https://www.baltimorepolice.org/>. [Accessed 16 October 2018].

**[2]** Python.org. 2018. *Welcome to Python.org*. [ONLINE] Available at: <https://www.python.org/>. [Accessed 16 October 2018].

**[3]** MySQL. 2018. *MySQL*. [ONLINE] Available at: <https://www.mysql.com/>. [Accessed 16 October 2018].

**[4]** Welcome to Jinja2 — Jinja2 Documentation (2.10). 2018. *Welcome to Jinja2 — Jinja2 Documentation (2.10)*. [ONLINE] Available at: <http://jinja.pocoo.org/docs/2.10/>. [Accessed 16 October 2018].

**[5]** Google Maps. 2018. *Google Maps* . [ONLINE] Available at: <https://www.google.com/maps>. [Accessed 16 October 2018].

**[6]** TensorFlow. 2018. *TensorFlow*. [ONLINE] Available at: <https://www.tensorflow.org/>. [Accessed 16 October 2018].