

# Functional Specification

Project Title: PandemicAnalysis

Student 1 Name: Jaime de Vivero Woods

Student 1 ID: 19447494

Student 2 Name: Alen Tom Joy

Student 2 ID: 18313576

## **Table of contents**

<b>Table of contents</b>	<b>2</b>
<b>1. Introduction</b>	<b>3</b>
1.1 Overview	3
1.3 Glossary	3
<b>2. General Description</b>	<b>4</b>
2.1 Product / System Functions	5
2.2 User Characteristics and Objectives	5
2.3 Operational Scenarios	6
2.4 Constraints	15
<b>3. Functional Requirements</b>	<b>16</b>
3.1. Functional Requirements Summary	16
3.2. User Interface Requirements	17
3.3. Main Page Requirements	18
3.4. Model Page Requirements	21
3.5. Dataset Page Requirements	22
<b>4. System Architecture</b>	<b>24</b>
4.1. Architecture Overview	25
<b>5. High-Level Design</b>	<b>26</b>
<b>6. Preliminary Schedule</b>	<b>29</b>
<b>7. Appendices</b>	<b>32</b>

# 1. Introduction

## 1.1 Overview

PandemicAnalysis will be a web application that will perform NLP (Natural Language Processing) and sentiment analysis on a publically available Twitter dataset about the COVID-19 pandemic. The area this dataset will cover is general chatter where the use of COVID-19 related hashtags are detected. This does not include tweets containing only statistics and figures about the progression of the pandemic. This dataset is planned to be updated on a weekly basis indefinitely until our source stops adding additional data.

In order to perform the sentiment analysis, we will produce a machine learning model using existing classifiers that will automate the process on every tweet in our (currently) 500,000 tweet dataset. Details about this model, and statistics about the accuracy will be provided to the end user, so that it can be incorporated into other sentiment analysis projects based on other COVID-19 Twitter datasets.

The end-user facing aspect of our web application will be comprised of three main elements:

The main page, which will display sentiment analysis statistics on tweets in various ways the user can select.

The model page where users can discover the machine learning algorithms used and their performance/accuracy statistics.

The dataset page where the user can find information about our dataset source, update frequency, the next scheduled server update and estimated downtime etc.

## 1.3 Glossary

Term	Definition
Natural Language Processing (NLP)	A machine learning technology that gives computers the ability to interpret, manipulate, and comprehend human language.
Sentiment Analysis	The use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information.
Logistic Model	A statistical model that models the probability of an event taking place by having the log-odds for the event be a linear combination of one or more independent variables.
Logistic Regression	Estimating the parameters of a logistic model (the coefficients in the linear combination).
Naive Bayes	A classification algorithm of Machine Learning based on Bayes theorem which gives the likelihood of occurrence of an event.
Naive Bayes Classifier	A probabilistic classifier which means that given an input, it predicts the probability of the input being classified for all the classes. It is also called conditional probability.
Bernoulli Naive Bayes	A type of Bayes classifier that is used for discrete data, where features are only in binary form.
SVM (Support Vector Machine)	A non-probabilistic classifier. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other.
TF-IDF (Term Frequency- Inverse Document Frequency)	A numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.
F1 Score	A metric used to measure the performance of a machine learning model or classifier.

(Definitions source: Wikipedia)

## 2. General Description

### 2.1 Product / System Functions

The user facing aspect of the system will be very simple. The web application will consist of a main page, in which a user will be allowed to select a number of different options to view the resulting data after sentiment analysis has been performed.

The main page will consist of a default view, which will display a term frequency graph for the top 20 terms encountered in the dataset, their sentiment scores if applicable and a pie-chart that displays overall sentiment from all the tweets in the dataset. This will be accompanied by a slider with a start and end date setting. This slider will control the timeline used for all the data that will appear on the page and its granularity will be in weeks. By default, the selectors on the slider will be positioned at both ends of the slider i.e., at the first week and final week of the dataset respectively.

The dataset we use to provide our data will be updated on a weekly basis to keep up with the latest information.

There will be additional options on this page available to the user to control exactly what data visualisations are displayed. Most will be disabled by default to improve loading performance.

On the model page, the user will be given all relevant information about the construction of our model and statistics behind its accuracy and performance. The specifics of the information users can acquire are given in the Functional Requirements section.

Finally, on the dataset page, users will be shown miscellaneous information such as the dataset source, hashtags used to generate the data, current server uptime, next scheduled dataset update etc.

## 2.2 User Characteristics and Objectives

The users of PandemicAnalysis have no expected technical expertise or knowledge required in order to use the application and view the visualisations. This is because the application will deliver high learnability and functionality with a clear, simple and intuitive interface. However, in order to understand the information given about our model such as precision scores and algorithms used, a user will need to have some background knowledge in sentiment based machine learning models and text/search information retrieval systems.

Though any user interested in public opinions based on the pandemic may use our application, it may be of particular interest to healthcare professionals, government officials and data scientists. As an example, our heatmap system will display the sentiment of a group of individuals, the number of tweets made in a selected timeframe for a given country and the major COVID-19 related headlines made during that time for this country. This could give government officials insights into the reactions of people, with regard to major events, like newly announced/removed restrictions, vaccination campaigns, records in death tolls/cases, opening/closing of public services etc. This, along with case statistics, can help officials understand which mitigation measures had the most impact with the least amount of negative reactions.

## 2.3 Operational Scenarios

Use Cases:

<b>USE CASE 1</b>	<b>View overall sentiments in a specific time</b>
<b>Goal in context</b>	<b>User checks overall sentiments</b>
<b>Scope &amp; Level</b>	<b>System, Core.</b>
<b>Preconditions</b>	<b>User is in home page, User has selected a moment in time to view the overall sentiments</b>
<b>Success End Condition</b>	<b>Overall sentiments for the selected time are displayed as a bar chart and a pie chart</b>
<b>Failed End Condition</b>	<b>Overall sentiments are not shown</b>
<b>Primary, Secondary Actors</b>	<b>User System</b>

<b>Trigger</b>	<b>User clicks on view sentiments function in the home page</b>	
<b>DESCRIPTION</b>	<b>Step</b>	<b>Action</b>
	<b>1</b>	<b>User selects a time using the timeline slider in the home page</b>
	<b>2</b>	<b>User clicks on generate overall sentiments function in home page</b>
	<b>3</b>	<b>Sentiments for the time selected are checked</b>
	<b>4</b>	<b>System returns sentiments data as a bar chart and a pie chart</b>
<b>EXTENSIONS</b>	<b>Step</b>	<b>Branching Action</b>
	<b>1a</b>	<b>User does not select a time using the slider. For step 3 sentiments are checked for all time</b>

<b>USE CASE 2</b>	<b>View heatmap of tweets for time selected</b>
<b>Goal in context</b>	<b>User selects to view heatmap</b>

<b>Scope &amp; Level</b>	<b>System, Core.</b>	
<b>Preconditions</b>	<b>User is in home page</b>	
<b>Success End Condition</b>	<b>Heatmap for the number of tweets is displayed</b>	
<b>Failed End Condition</b>	<b>Heatmap is not shown</b>	
<b>Primary, Secondary Actors</b>	<b>User System</b>	
<b>Trigger</b>	<b>User clicks on view heatmap function in home page</b>	
<b>DESCRIPTION</b>	<b>Step</b>	<b>Action</b>
	<b>1</b>	<b>User selects a time using the timeline slider in the home page</b>
	<b>2</b>	<b>User selects heatmap function</b>
	<b>3</b>	<b>User clicks generate button</b>
	<b>4</b>	<b>System checks data on tweet count for given timeline</b>



	5	System returns the data displayed as a heatmap
<b>EXTENSIONS</b>	<b>Step</b>	<b>Branching Action</b>

<b>USE CASE 3</b>	<b>View word clouds for time selected</b>
<b>Goal in context</b>	<b>User view a word cloud for a selected timeline</b>
<b>Scope &amp; Level</b>	<b>System, Core.</b>
<b>Preconditions</b>	<b>User is in home page</b>
<b>Success End Condition</b>	<b>Word cloud is displayed</b>
<b>Failed End Condition</b>	<b>Word cloud is not displayed</b>
<b>Primary,</b>	<b>User</b>

<b>Secondary Actors</b>	<b>System</b>	
<b>Trigger</b>	<b>User clicks on view word cloud function in home page</b>	
<b>DESCRIPTION</b>	<b>Step</b>	<b>Action</b>
	<b>1</b>	<b>User selects a time using the timeline slider in the home page</b>
	<b>2</b>	<b>User selects word cloud function</b>
	<b>3</b>	<b>User clicks generate button</b>
	<b>4</b>	<b>System checks data on keywords for the given timeline</b>
	<b>5</b>	<b>System returns the data displayed as a word cloud</b>
<b>EXTENSIONS</b>	<b>Step</b>	<b>Branching Action</b>

<b>USE CASE 4</b>	<b>View dataset information</b>	
<b>Goal in context</b>	<b>User checks dataset information</b>	
<b>Scope &amp; Level</b>	<b>System, Core.</b>	
<b>Preconditions</b>	<b>User is in home page</b>	
<b>Success End Condition</b>	<b>Dataset information is displayed (dataset details, last update time, next scheduled update)</b>	
<b>Failed End Condition</b>	<b>Dataset information is not displayed</b>	
<b>Primary, Secondary Actors</b>	<b>User System</b>	
<b>Trigger</b>	<b>User clicks on visit dataset function in home page</b>	
<b>DESCRIPTION</b>	<b>Step</b>	<b>Action</b>
	<b>1</b>	<b>User selects view dataset information function in home page</b>

	2	System checks dataset details, last update time, and next schedule update
	3	System returns the data checked in step 2
EXTENSIONS	Step	Branching Action

USE CASE 5	View model accuracy statistics
Goal in context	User checks model accuracy statistics
Scope & Level	System, Core.
Preconditions	User is in model page
Success End Condition	Model accuracy statistics are displayed
Failed End Condition	Model accuracy statistics are not displayed

<b>Primary, Secondary Actors</b>	<b>User  System</b>	
<b>Trigger</b>	<b>User clicks on view model accuracy statistics function</b>	
<b>DESCRIPTION</b>	<b>Step</b>	<b>Action</b>
	<b>1</b>	<b>User clicks on view model accuracy statistics function in model page</b>
	<b>2</b>	<b>System checks model accuracy statistics</b>
	<b>3</b>	<b>System returns the data checked in step 2</b>
<b>EXTENSIONS</b>	<b>Step</b>	<b>Branching Action</b>

<b>USE CASE 6</b>	<b>View model accuracy graphs</b>
<b>Goal in context</b>	<b>User checks model accuracy graphs</b>

<b>Scope &amp; Level</b>	<b>System, Core.</b>	
<b>Preconditions</b>	<b>User is in model page</b>	
<b>Success End Condition</b>	<b>Model accuracy graphs are displayed</b>	
<b>Failed End Condition</b>	<b>Model accuracy graphs are not displayed</b>	
<b>Primary, Secondary Actors</b>	<b>User System</b>	
<b>Trigger</b>	<b>User clicks on view model accuracy graphs function</b>	
<b>DESCRIPTION</b>	<b>Step</b>	<b>Action</b>
	<b>1</b>	<b>User clicks on view model accuracy graphs function in model page</b>
	<b>2</b>	<b>System checks model accuracy graphs</b>
	<b>3</b>	<b>System returns the data checked in step 2 as a graphical confusion matrix and a ROC curve</b>

EXTENSIONS	Step	Branching Action

## 2.4 Constraints

Lists general constraints placed upon the design team, including speed requirements, industry protocols, hardware platforms, and so forth.

There are several constraints and technical issues we face with the development of our system.

### Performance constraints

#### *Databases:*

A key area that will need focus is system performance. With databases that can range up to 500,000 tweets in size, special consideration is required for the type of database we will use and the queries we will perform. The end product should minimise UI loading times and maximise responsiveness.

#### *Server:*

We plan to build and use our own dedicated Linux server instead of relying on cloud hosting providers to maximise our control over the system. The budget will constrain us to a 6 core/12 thread system with 16GB of memory and 500GB of SSD storage. If this proves insufficient, we will look to cloud hosted servers as a backup option. This server will use a 300 megabit connection.

#### *Internet Connection:*

Because of the vast amounts of data that a user may choose to access, a reliable internet connection with a speed of at least 40 megabits/s is likely to be required for a smooth experience.

**Time Constraints:***Model:*

The vast majority of development time is likely to be used for creation and evaluation of our machine learning model. We hope to mitigate this slightly by parallelizing UI development, database structuring and work on the model. This can be done thanks to our dataset being pre-tagged, which means the tweet dataset will be ready to add to the database once preprocessing is done.

Preprocessing itself will be a time consuming task requiring:

1. Hydration of tweets through the Twitter API
2. Removal of null values
3. Removal of punctuation
4. Stop-word removal
5. Conversion to lowercase
6. Cleaning up columns

The most time consuming of all these tasks is tweet hydration. As of 2020, Twitter requires all publicly exposed tweet datasets to use only the 64-bit Tweet ID to represent the contents of an entire tweet. Tweet hydration is the process of converting the 64-bit tweet IDs to the original tweet, using Twitter's API. This API imposes various limitations on developers, the main one being a rate limit of 900 tweets per request with 100 possible requests in every 15 minute interval. There is no way to increase the rate limit other than by paying a large fee.

In essence, this means hydrating the entire dataset of about 500,000 tweets alone takes about 1.5 hours.

*Update Automation:*

As mentioned previously, we plan to update the dataset on a weekly basis to collect the latest tweets being sent. We are also required to do this in order to comply with Twitter's offline data handling policies. These policies state that any tweet content stored offline must be updated regularly to represent the current state of the tweet as it exists on Twitter. This means that hydration of the entire dataset including the update set is required, which will once again take 1.5 hours and increase overtime as the dataset grows in size.

**3. Functional Requirements****3.1. Functional Requirements Summary**



**User Interface:** Simple and informative, easy to navigate, minimum number of pages

**Main page:** User may scroll through data overview, user may adjust tweet timeline, user may generate word-cloud, user may generate country heatmap, navigate to ML model page, navigate to dataset details page.

*Optional (no requirements table given):* More user accessible representations of data such as scatter plots, retweet frequency charts, individual tweet retrieval for shorter timespans (< 500 tweets)

**Model page:** User may view: model used, model precision, recall, MAP and F1-score, confusion matrix, ROC curve, user may visit: main page, dataset page.

**Dataset page:** User may view: dataset details, last update time, next scheduled update , user may visit: main page, model page.

### 3.2. User Interface Requirements

<b>Title</b>	<b>User Interface R1</b>
<b>Name</b>	Simple and informative
<b>Description</b>	The application as a whole should be laid out in a simple and informative manner.
<b>Stakeholders</b>	User
<b>Criticality</b>	Essential
<b>Technical Issues</b>	
<b>Dependencies</b>	
<b>Comments</b>	

<b>Title</b>	<b>User Interface R2</b>
<b>Name</b>	Easy to navigate
<b>Description</b>	Each view of the data must be labelled clearly and a short description of what they illustrate is required.
<b>Stakeholders</b>	User
<b>Criticality</b>	Essential

<b>Technical Issues</b>	
<b>Dependencies</b>	
<b>Comments</b>	

<b>Title</b>	<b>User Interface R3</b>
<b>Name</b>	Minimum number of pages
<b>Description</b>	Preference should be given to page scrolling instead of adding subpages to each primary page.
<b>Stakeholders</b>	User
<b>Criticality</b>	Essential
<b>Technical Issues</b>	
<b>Dependencies</b>	
<b>Comments</b>	This is because most of the data on a primary page will be linked or related in some way.

### 3.3. Main Page Requirements

<b>Title</b>	<b>Main Page R1</b>
<b>Name</b>	User may scroll through data overview
<b>Description</b>	The user is presented with an overview of the data on the main page of the web application. The tweet timeline is preset to its widest position with adjustments ranging in months.
<b>Stakeholders</b>	User
<b>Criticality</b>	Essential

<b>Technical Issues</b>	The slider will have to be updated dynamically as the dataset increases over time.
<b>Dependencies</b>	<b>User Interface: ( R3, R1), Main Page: R2</b>
<b>Comments</b>	The user's unmodified view should have a pie chart containing the overall sentiment (positive/negative) for the period of time. There should also be a bar graph listing the top 10 most common terms over the whole period (not counting stop words). Can be used in conjunction with all other main page functions

<b>Title</b>	<b>Main Page R2</b>
<b>Name</b>	User may adjust tweet timeline
<b>Description</b>	The user can drag date sliders to filter the data between a desired period of time
<b>Stakeholders</b>	User
<b>Criticality</b>	Essential
<b>Technical Issues</b>	This feature is likely to require complex manipulation of database views
<b>Dependencies</b>	<b>User Interface: R3, R1</b>
<b>Comments</b>	Use of this function by the user would essentially change the results of every other data analysis tool on this page. Every other tool will be dependant on it

<b>Title</b>	<b>Main Page R3</b>
<b>Name</b>	User may generate words clouds
<b>Description</b>	The user will be given the option to generate word clouds for the positive and negative sentiments respectively for the given timeline
<b>Stakeholders</b>	User
<b>Criticality</b>	Essential
<b>Technical Issues</b>	
<b>Dependencies</b>	<b>User Interface: ( R3, R1), Main Page: R2</b>
<b>Comments</b>	Can be used in conjunction with all other main page functions

<b>Title</b>	<b>Main Page R4</b>
<b>Name</b>	User may generate country heatmap
<b>Description</b>	The user may generate a heatmap for the chosen timeline that displays tweet frequency for valid English speaking countries and major covid headlines from that period for each country
<b>Stakeholders</b>	User
<b>Criticality</b>	Desirable
<b>Technical Issues</b>	Automating retrieval of Covid-19 headlines for the given time periods
<b>Dependencies</b>	<b>User Interface: ( R3, R1), Main Page: R2</b>
<b>Comments</b>	Can be used in conjunction with all other main page functions

<b>Title</b>	<b>Main Page R5</b>
<b>Name</b>	User may navigate to model page
<b>Description</b>	The user may use the navigation bar near the top of the web page to navigate to the model page
<b>Stakeholders</b>	User
<b>Criticality</b>	Essential
<b>Technical Issues</b>	
<b>Dependencies</b>	<b>User Interface R2</b>
<b>Comments</b>	

<b>Title</b>	<b>Main Page R6</b>
<b>Name</b>	User may navigate to dataset details page
<b>Description</b>	The user may use the navigation bar near the top of the web page to navigate to the dataset details page
<b>Stakeholders</b>	User

<b>Criticality</b>	Essential
<b>Technical Issues</b>	
<b>Dependencies</b>	User Interface R2
<b>Comments</b>	

### 3.4. Model Page Requirements

<b>Title</b>	<b>Model Page R1</b>
<b>Name</b>	User may view model accuracy statistics
<b>Description</b>	The user may view statistics about the model used such as model type, model precision, model recall precision, model F1-score and model mean average precision
<b>Stakeholders</b>	User
<b>Criticality</b>	Essential
<b>Technical Issues</b>	
<b>Dependencies</b>	User Interface: R1, R3
<b>Comments</b>	

<b>Title</b>	<b>Model Page R2</b>
<b>Name</b>	User may view model accuracy graphs
<b>Description</b>	The user may view the model's graphical confusion matrix and ROC curve
<b>Stakeholders</b>	User
<b>Criticality</b>	Desirable
<b>Technical Issues</b>	
<b>Dependencies</b>	User Interface R1, R3
<b>Comments</b>	

<b>Title</b>	<b>Model Page R3</b>
<b>Name</b>	User may navigate to the main page
<b>Description</b>	The user may use the navigation bar near the top of the web page to navigate to the main page
<b>Stakeholders</b>	User
<b>Criticality</b>	Essential
<b>Technical Issues</b>	
<b>Dependencies</b>	<b>User Interface R2</b>
<b>Comments</b>	

<b>Title</b>	<b>Model Page R4</b>
<b>Name</b>	User may navigate to the dataset page
<b>Description</b>	The user may use the navigation bar near the top of the web page to navigate to the dataset page
<b>Stakeholders</b>	User
<b>Criticality</b>	Essential
<b>Technical Issues</b>	
<b>Dependencies</b>	<b>User Interface R2</b>
<b>Comments</b>	

### 3.5. Dataset Page Requirements

<b>Title</b>	<b>Dataset Page R1</b>
<b>Name</b>	User may view dataset details
<b>Description</b>	The user may the background details on the dataset such as source, licence, tweet keywords and hashtags used for dataset generation etc.

<b>Stakeholders</b>	User
<b>Criticality</b>	Essential
<b>Technical Issues</b>	
<b>Dependencies</b>	User Interface R1, R3
<b>Comments</b>	

<b>Title</b>	<b>Dataset Page R2</b>
<b>Name</b>	User may view last dataset update time
<b>Description</b>	The user may view the latest time the dataset was updated on the server
<b>Stakeholders</b>	User
<b>Criticality</b>	Essential
<b>Technical Issues</b>	
<b>Dependencies</b>	User Interface R1, R3
<b>Comments</b>	

<b>Title</b>	<b>Dataset Page R3</b>
<b>Name</b>	User may view next scheduled update time
<b>Description</b>	The user may view the time at which the next dataset update is scheduled
<b>Stakeholders</b>	User
<b>Criticality</b>	Essential
<b>Technical Issues</b>	
<b>Dependencies</b>	User Interface R1, R3

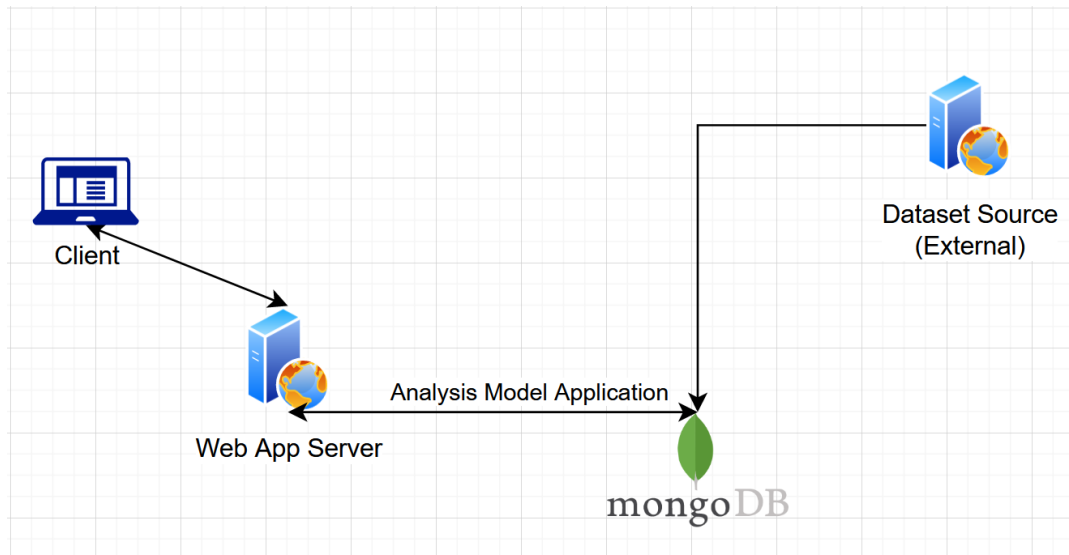
<b>Comments</b>	During this time, the web application will be unavailable to users
-----------------	--

<b>Title</b>	<b>Dataset Page R4</b>
<b>Name</b>	User may navigate to the main page
<b>Description</b>	The user may use the navigation bar near the top of the web page to navigate to the main page
<b>Stakeholders</b>	User
<b>Criticality</b>	Essential
<b>Technical Issues</b>	
<b>Dependencies</b>	<b>User Interface R2</b>
<b>Comments</b>	

<b>Title</b>	<b>Dataset Page R5</b>
<b>Name</b>	User may navigate to the model page
<b>Description</b>	The user may use the navigation bar near the top of the web page to navigate to the model page
<b>Stakeholders</b>	User
<b>Criticality</b>	Essential
<b>Technical Issues</b>	
<b>Dependencies</b>	<b>User Interface R2</b>
<b>Comments</b>	

## 4. System Architecture





## 4.1. Architecture Overview

### Backend

The backend will consist of a Django based Linux web server that is attached to a NOSQL MongoDB database containing the tweet dataset. The database will also contain the news article headlines that will be scraped during the dataset update process. We plan to use either PyMongo or Djongo to connect the database and server.

The source for our dataset resides at [1]

<https://ieee-dataport.org/open-access/coronavirus-covid-19-geo-tagged-tweets-dataset> .

This dataset is currently updated on a daily basis with new tweets. We plan to update our local database on a weekly basis to maintain a good balance between uptime and recency. News headlines will also be updated in a similar timeframe.

The update process will be automated through scripts and CI/CD tools . We plan to use Gitlab's CI/CD solutions. We will make use of Gitlab Runners hosted on the same physical server as our web application to automate updates. The Runner will ideally start a script that logs on to our dataset source, retrieves the link for the latest dataset and downloads it locally. The hydration process will then be triggered again for the entire updated dataset and then analysed using our existing model. The resulting data will be saved in a new database. At this point, the web app will be taken offline and connected to the new database, minimising downtime.

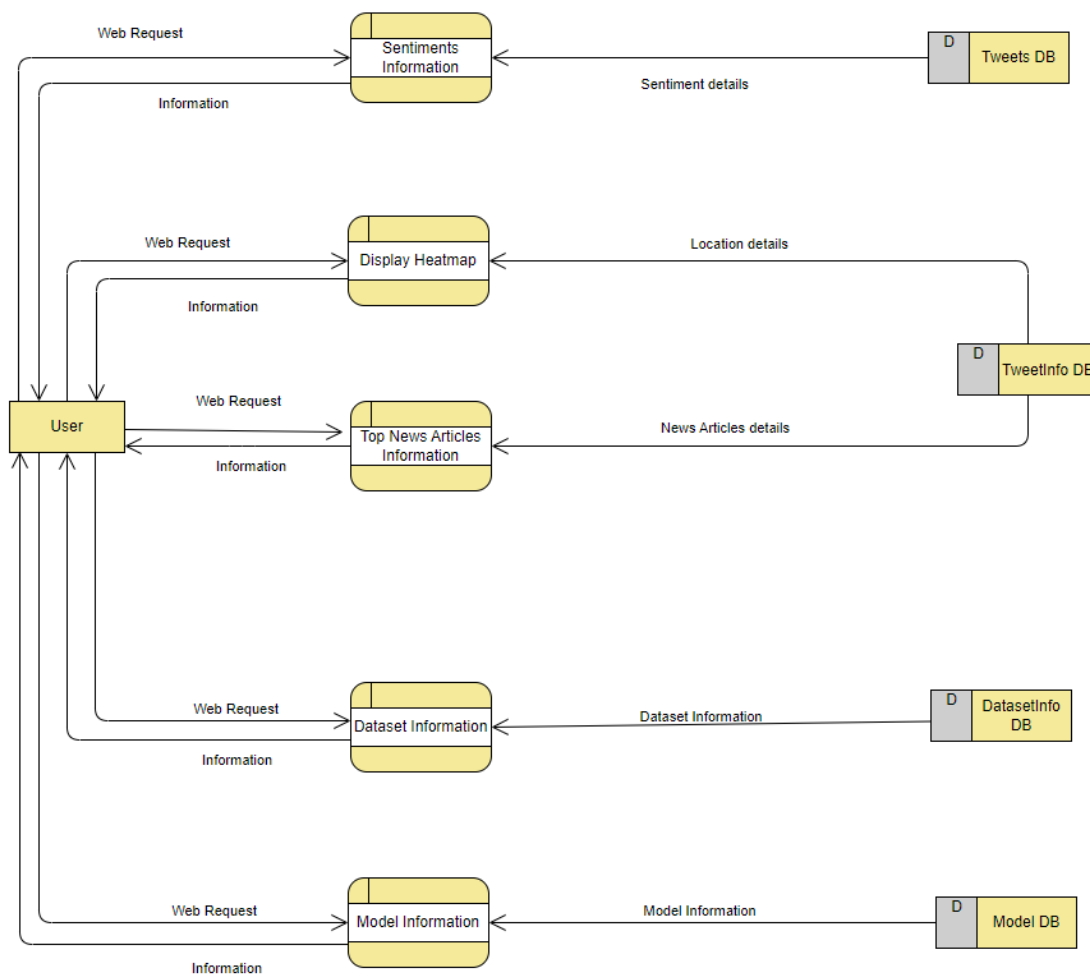
The ML model generation process will only be carried out once on the initial tweet dataset. We will generate the model using well established algorithms and techniques used in machine learning. The classifiers we plan to evaluate for use with our model are: SVM, Bernoulli Naive Bayes and logistic regression.

After the model has been generated, subsequent dataset updates will have sentiment analysis carried out by this same model.

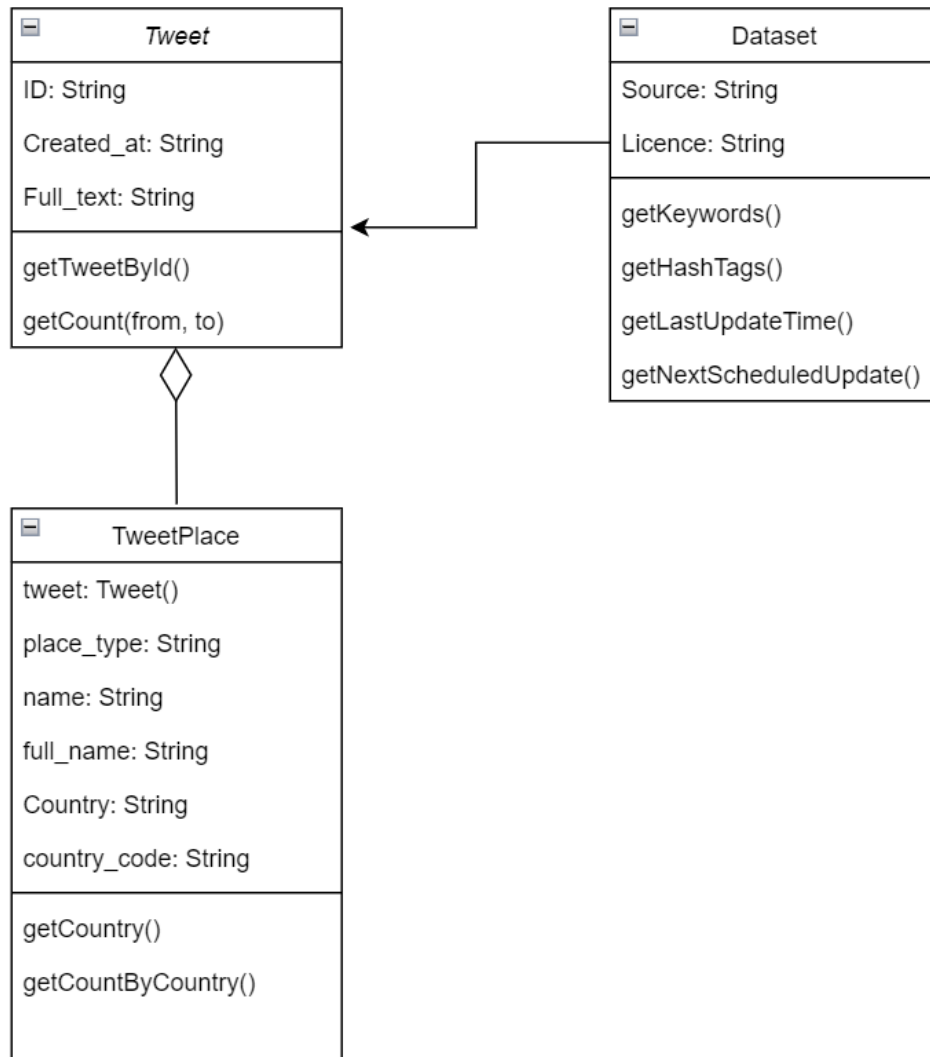
## Frontend

The client-side UI will be built using Django, CSS and Javascript. Consequently, the frontend will be based on a Model-View-Template (MVT) architecture.

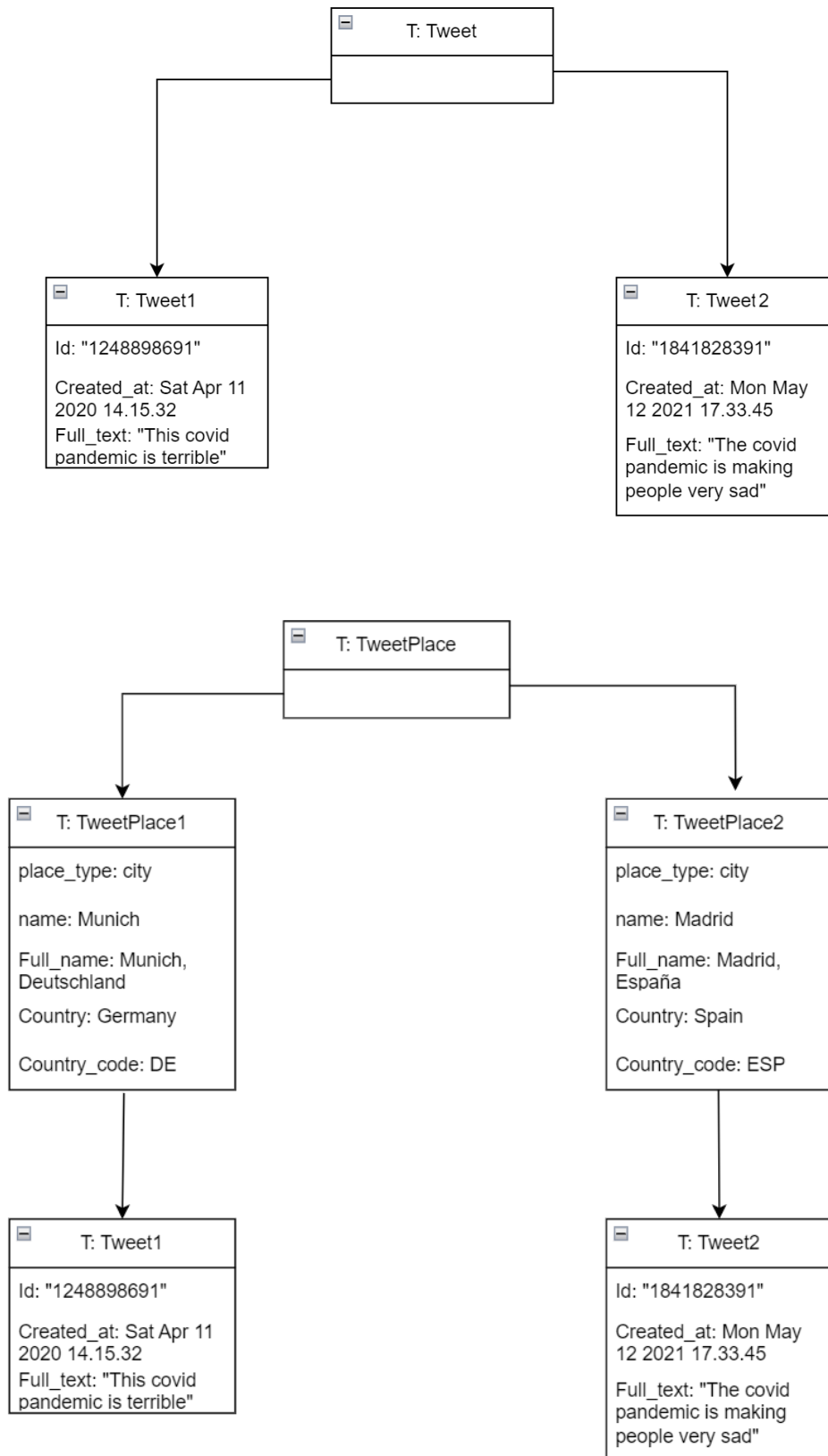
## 5. High-Level Design




Class Diagram:



Object Diagram:

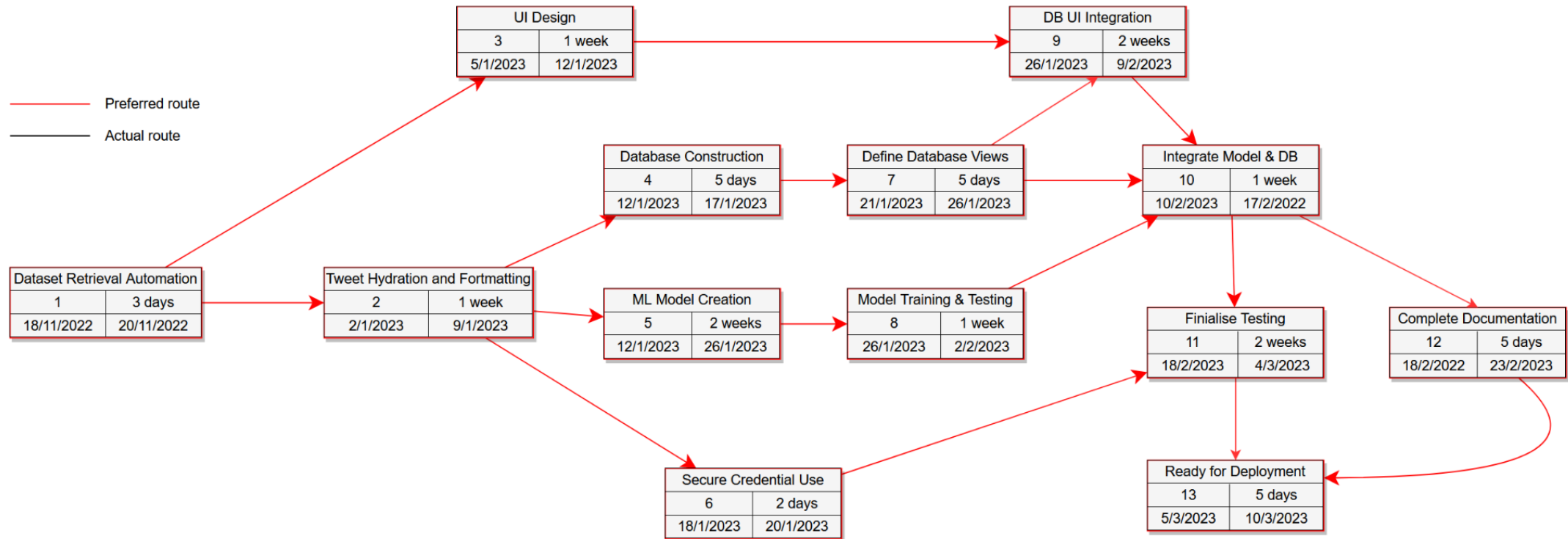


 <i>D: Dataset</i>
DOI: <a href="https://dx.doi.org/10.21227/fpsb-jz61">https://dx.doi.org/10.21227/fpsb-jz61</a> License: Creative Commons <a href="https://creativecommons.org/licenses/by/4.0/">https://creativecommons.org/licenses/by/4.0/</a>

## 6. Preliminary Schedule

Please find the PERT chart attached below on pg.30.





## 7. Appendices

Dataset Source:

[1] Rabindra Lamsal. (2020). Coronavirus (COVID-19) Geo-tagged Tweets Dataset. IEEE Dataport. <https://dx.doi.org/10.21227/fpsb-jz61>