|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| https://lh4.googleusercontent.com/proxy/YA9Xoqs7jhpeuwrEjwhdi_EVSCDwUdpr72V-2YHZ2lz2y1FaqityK8c8RlZRTvUDEw3Y2TekyGNi07wcREil5Ez3ii80dA-DE8G6HAQjEmJVz8W32Wy2uaDAWwuZs6uPZtJp2zrUJ_Qps2T1CUmSpuPR8dk2XA=w128-h144-k-no | | | **Document Ref.:** | | |  | |
| **Version No.:** | | |  | |
| **Date:** | | | 21st March 2019 | |
| **Copy No.:** | | |  | |
| **Project Name:** | Detection of Events and Emerging Themes from Social Media Streams | | | | | | |
| **Project Code:** | PW19COP01 | | | | | | |
| **Status:** | **Current** | | | | | | |
| **Document Type:** | **Uncontrolled** | | | | | | |
| **Detection of Events and Emerging Themes from Social Media Streams** | | | | | | | |
| **Detecting events of national importance, disease outbreaks, and other emergencies and displaying the same on a user’s Twitter feed.** | | | | | | | |
| **Prepared By:** | | | | **Reviewed By:** | | | |
| **Name** | | **Date** | | **Name** | | | **Date** |
| Vaishnavi Rao B USN: 01FB15ECS334 | | 21/03/2019 | |  | | |  |
|  | | |  |
| Varsha R USN: 01FB15ECS337 | | 21/03/2019 | | **Approved By:** | | | |
| **Name** | | | **Date** |
|  | |  | |  | | |  |
|  | | |  |
| **Distribution List** | | | | | | | |
| **Project Representative(s)** | | | | | **Guide Representative(s)** | | |
| 1. Vaishnavi Rao B 2. Varsha R | | | | | 1. Prof. C. O. Prakash | | |

**TABLE OF CONTENTS**

[**Definitions, Acronyms and Abbreviations 2**](#_3as4poj)

[**References 2**](#_1pxezwc)

[**Change History 2**](#_49x2ik5)

[**1.0**](#_2p2csry) **Introduction 2**

[**1.1**](#_147n2zr) **Overview 2**

[**1.2**](#_3o7alnk) **Scope 2**

[**2.0**](#_23ckvvd) **Design Constraints, Assumptions and Dependencies 2**

[**3.0**](#_ihv636) **Design Description 2**

[**3.1**](#_32hioqz) **Master Class Diagram 2**

[**3.2**](#_1hmsyys) **Module 1 2**

[*3.2.1*](#_41mghml) *Description 2*

[*3.2.2*](#_2grqrue) *Use Case Diagram 2*

[*3.2.3*](#_vx1227) *Class Diagram 2*

[**3.2.3.1**](#_44sinio) **Class Description 1 2**

[**3.2.3.2**](#_2jxsxqh) **Sequence Diagram 2**

[**4.0**](#_3fwokq0) **ER Diagrams 2**

[**5.0**](#_1v1yuxt) **User Interface Diagrams 2**

[**6.0**](#_4f1mdlm) **Report Layouts 2**

[**7.0**](#_2u6wntf) **External Interfaces 2**

[**8.0**](#_19c6y18) **Packaging and Deployment Diagrams 2**

[**9.0**](#_3tbugp1) **Help 2**

[**10.0**](#_28h4qwu) **Alternate Design Approach 2**

[**11.0**](#_nmf14n) **Reusability Considerations 2**

[**12.0**](#_37m2jsg) **Traceability Matrix 2**

**References**

[1] Catching the Long-Tail: Extracting Local News Events from Twitter; Puneet Agarwal, Rajgopal Vaithiyanathan, Saurabh Sharma and Gautam Shroff

[2] Topic Extraction from News Archive Using TF\*PDF Algorithm; Khoo Khyou Bun Mitsuru Ishizuka

[3] A Survey of Techniques for Event Detection in Twitter; Farzindar Atefeh and Wael Khreich

[4] Sentiment-Based Event Detection in Twitter; Georgios Paltoglou

[5] Emerging Event Detection in Social Networks with Location Sensitivity; Unankard Sayan, Xue Li, and Mohamed A Sharaf

[6] Predicting Crowd Behaviour with Big Public Data; N. Kallus

[7] Event Detection in Social Media Data; Wenwen Dou, Xiaoyu Wang, William Ribarsky, Michelle Zhou

[8] Tweet Analysis for Real-Time Event Detection and Earthquake Reporting System Development Ekta, Paahuni Khandelwal, Priya Bundela, Richa Dewan

**Change History**

This section describes the details of changes that have resulted in the current High-Level Design document.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Date** | **Document Version No.** | **Change Description** | **Reason for Change** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# **Introduction**

* 1. ***Overview***

The micro-blogging platform, Twitter, has opened people up to an entirely new world of information-gathering and news-sharing, including but not limited to detailing everyday events and activities of the users, the “Tweeple”. Twitter, then, becomes an exhaustive source of data for analysis as well as the first place people are increasingly turning to for daily alerts or updates, in the face of news organisations and other traditional forms of media people looked to earlier. A major challenge facing event detection using social media streams is to extract useful real-world events from the mundane and polluted text (abbreviated words, spelling and grammatical errors and text written in mixed languages). [3] Event detection requires the automatic answering of what, when, where, and by whom. After reporting on the most recent efforts in the area, it is clear that no method addressed all of these questions. [5] Investigating how to model the social streams together with other data sources, like news streams to better detect and represent events was another complication faced earlier. [6] A considerable effort is still required to achieve efficient, scalable and reliable systems for event detection, summarization, tracking and association. [7]

* 1. ***Scope***

The project will depend on the efficient functioning of Twitter minus any crashes on any

given day to be able to conduct real time analysis smoothly in the long run. A proper, working internet connection is required that keeps the end user connected with their surroundings in order to access Twitter easily. Additional limitations include false positives generated by the model and disinformation provided by people on Twitter. Twitter users are assumed to be sensors, who make observations, which are their actual tweets to detect a target event. The tweets are given a time values and a location as well. The detection of these events and other emerging topics is restricted to the Indian subcontinent.

# **Design Constraints, Assumptions and Dependencies**

TECHNICAL CONSTRAINTS:

Programming Language: The programming language of choice for the bulk of the project is Python, given its immense popularity in non-enterprise-related applications. It offers an easier and faster way to build high-performing algorithms, along with an extensive collection of specialized libraries available as well.

Operating Systems and Platforms Supported: The vision is to build a fully-functional project that works smoothly across platforms, and are currently implementing the project on a standard Windows device.

Frameworks: The front-end (user interface) is based on a standard MVC framework.

DEPENDENCIES:

A major portion of the project design and implementation is dependent on the availability of data in large numbers, and subsequently, necessary servers to be able to stream data without any hindrance.

# **Design Description**

### 3.1 **Use Case Diagram**

### 

|  |  |
| --- | --- |
| **Use Case Item** | **Description** |
| User | The users who will use our system |
| Twitter | Twitter.com API |
| Classifier | The algorithm |

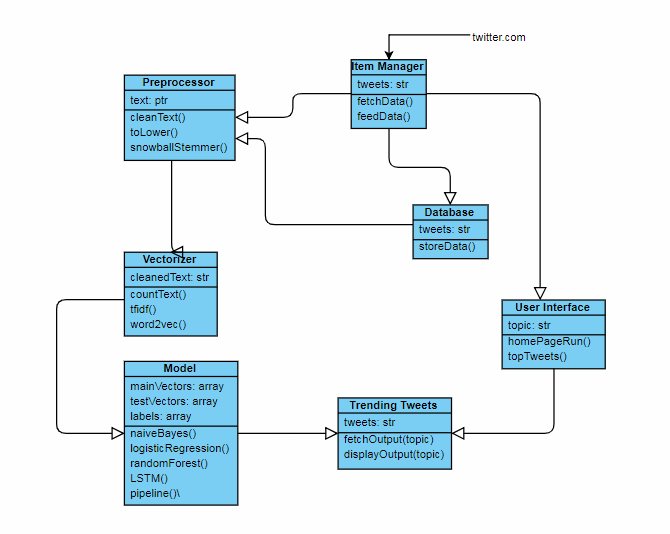
### **Class Diagram**

Here, a description of each class in this class diagram will be given. A diagram of the entire system will be given at a high level and then broken down into sub levels. Classes maybe repeated across class diagrams, to show the interfaces with other classes. The detailed explanation of each class with its methods will be covered in the low-level design document.

For Example

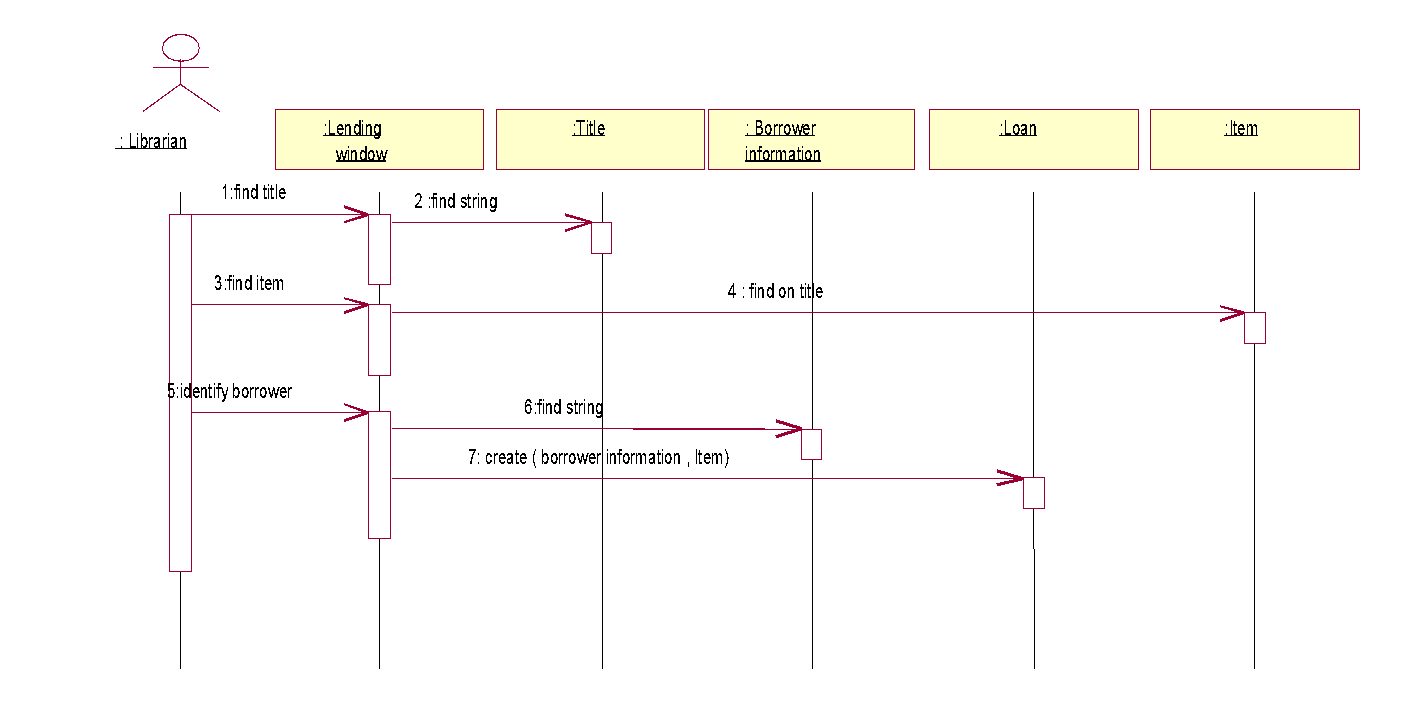
|  |  |  |
| --- | --- | --- |
| Class | Properties | Methods |
| Preprocessing | Using the nltk library, we will preprocess the tweets by removing the stop words, reduce characters to lower case and tokenize | A simple preprocessing function that will be run on every tweet |
| Vectorization | Tweets cannot be utilized directly in string format. Hence tweets will need to be vectorized to a format the algorithm will understand | Direct modules to vectorize texts are used |
| Model  ModelTraining  ModelTesting  EvaluationMetrics | Model Parameters and the subclasses of this class provide more model training and testing related parameters. | Train the model and test the model. Also the subclasses have methods that calculate the error of the methods |
| Item Manager (In this context refers to a Live incoming tweet) | This runs a code and fetches the top tweets of the day from that day 00:00 | Fetch the top tweets of the day |
| Database | This basically has the entire training data as well as the new incoming testing data | Store the incoming and old tweets in a database |
| User | Every user is associated with an unique id and is displayed the home page when our website is opened | Display home page with trending tweets and drop down menu |
| Trending tweets | After user makes a choice, turn to another page, display tweets of the topic chosen. if no tweets for the topic are available, display trending tweets of the day | Displayed trending tweets by running the best performing algorithm on the live twitter data |

#### ***Class Description 1***



#### ***Sequence Diagram***

The Sequence diagram for each module will be presented here.

For Example:

# **ER Diagrams**

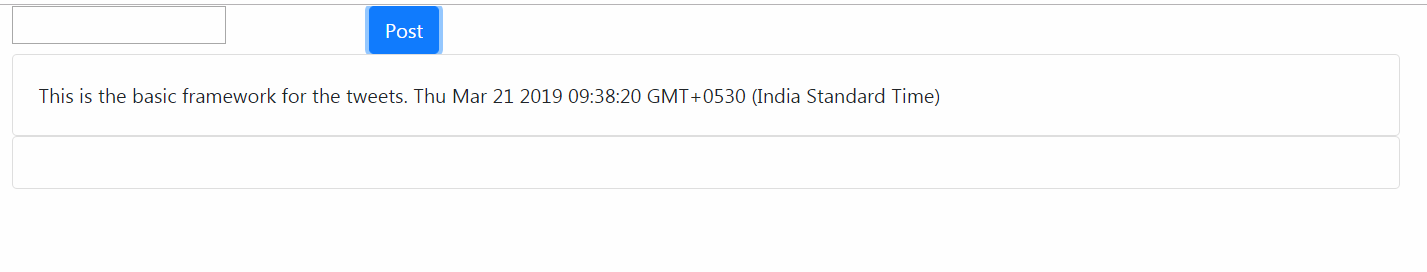
This section will include the ER Diagram. The following table shall be filled for details of the entities and their data elements / attributes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Entity** | **Name** | **Definition** | **Type** |
| **ENTITIES** | | | | |
|  | twitter\_feed |  | Refers to the user’s Twitter feed. |  |
|  | twitter\_user |  | Refers to the Twitter user themselves. |  |
| 3. | twitter\_follower |  | Refers to the Twitter follower(s). |  |
| 4. | twitter\_favourite |  | Refers to the followers who engage most actively with a specific user. |  |
| **#** | **Attribute** | **Name** | **Definition** | **Type (size)** |
| **DATA ELEMENTS** | | | | |
|  | user\_id  user\_name  tweet\_text  like\_count  repost\_count  create\_date |  |  | int str str int int date |
|  | user\_id name  gender  location  link  email |  |  | int str str str str str |
| 3. | user\_id follower\_id  follower\_name |  |  | int int str |
| 4. | user\_id tweet\_id  tweet\_text create\_date  hashtag\_used  retweet\_count  from\_user\_id |  |  | int int str date str int int |

This section shall describe the data / function used in each module / function.

# **User Interface Diagrams** The UI as it currently stands, looks like this:



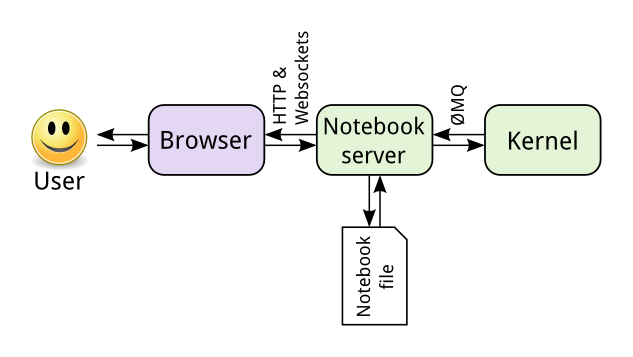


# **Report Layouts**

There is no reports requirements for this product as such.

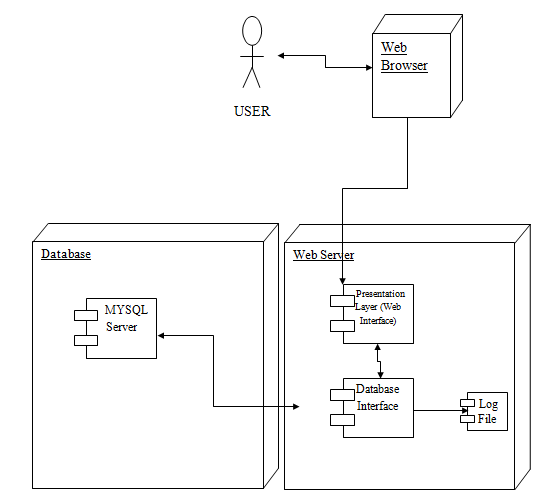
# **External Interfaces**

Gives an overall diagram as to how the system with known interfaces will work. However, the description of the interface may or may not be covered in this document, depending on whether it is within the scope of the offshore development.



# **Packaging and Deployment Diagrams**

Following is an abstract package and deployment diagram for the system that shows the most important modules of the system and how they interact with each other.



# **Help**

Since the product is a user friendly webpage, we do not find the necessity to provide an explicit documentation or user guide. Basic introduction at the start on the home page that guide him through the page such that he can be familiar with all the features of the product should be enough.

# **Reusability Considerations**

This project involves training machine learning models for twitter data analysis, these models are created once and reused, multiple times. We’re using a simple Jupyter notebook interface where the algorithms can be run again and again and the training data is stored in a simple csv format and not using any database management systems. The live testing data from Twitter.com however is stored in a mySQL database.

# **Traceability Matrix**

|  |  |
| --- | --- |
| **CRS Reference Section No.**  **and Name.** | **DESIGN / HLD Reference Section No.**  **and Name.** |
|  |  |
|  |  |
|  |  |