

Princess Sumaya University for Technology

King Hussein School for Computing Sciences

**Social Media Analytics for Emergency Response Module**

**An iHELP Component**

**Prepared By:**

Alaa Al-Aghbar

Hajer Qasem

Nermeen Al-Zayed

**Project (1) Supervised By:**

Dr. Sufyan AlMajali

**Project (2) Supervised By:**

Dr. Salam Fraihat

Project Submitted in partial fulfillment for the degree of Bachelor of Science in Computer Science

Second Semester - 2019/2020

## **Declaration of Originality**

This document has been written entirely by the undersigned team members of the project. The source of every quoted text is clearly cited and there is no ambiguity in where the quoted text begins and ends. The source of any illustration, image or table that is not the work of the team members is also clearly cited. We are aware that using non-original text or material or paraphrasing or modifying it without proper citation is a violation of the university’s regulations and is subject to legal actions.

Names and Signatures of team members:

## **Acknowledgments**

We are deeply grateful for the great guidance and help we received from our project supervisors Dr. Sufyan Almajali and Dr. Salam Fraihat, who took real interest in our work and progress and was there to provide all the necessary information for developing a good system.

## **Summary**

Social media outlets have become a common source of data especially in cases when traditional systems like calling 911 fail [1]. Our team will develop a module with software bots that can text-mine logs from social media outlets such as Facebook and Twitter to extract information relevant to affected areas of occurring disaster and parse through it. For instance, if many social media posts are indicating that there are people stranded somewhere, the module should be able to identify their location and send a request to the Intelligent Humanitarian Emergency Logistics Platform (iHELP) on their behalf. The algorithms in this module need to be intelligent enough to decide what is credible and what is not. Therefore, it needs to conduct some analysis to determine the level of statistical confidence in the posts it is reading and whether such posts are truly coming from people in dire or not.

Our module will be developed within the grounds of the iHELP system.

## **List of Abbreviations**

**CRUD​:** Create, Read, Update, and Delete.

**DA**: Donating Agencies.

**DB:** Database.

**ERD:** Entity Relationship Diagram.

**ERT**: Emergency Response Teams.

**FB**: Facebook

**FEMA**: Federal Emergency Management Agency.

**HA**: Humanitarian Agencies.

**NATO:** The North Atlantic Treaty Organization.

**iHELP**: Intelligent Humanitarian Emergency Logistics Platform.

**REQ**: Requirements.

**VM:** Volunteer Management.

## **Table of Contents**

[**Declaration of Originality** 3](#_Toc29159749)

[**Acknowledgments** 4](#_Toc29159750)

[**Summary** 5](#_Toc29159751)

[**List of Abbreviations** 6](#_Toc29159752)

[**Table of Contents** 7](#_Toc29159753)

[**Table of Figures** 9](#_Toc29159754)

[**Table of Tables** 10](#_Toc29159755)

[**Chapter** 1 **Introduction** 11](#_Toc29159756)

[**1.1 Overview** 11](#_Toc29159757)

[**1.2 Problem Statement** 12](#_Toc29159758)

[**1.3 Related Work** 12](#_Toc29159759)

[**1.4 Document Outline** 14](#_Toc29159760)

[**Chapter** 2 **Project Plan** 15](#_Toc29159761)

[**2.1 Project Deliverables** 15](#_Toc29159762)

[**2.2 Project Tasks** 15](#_Toc29159763)

[**Gantt Chart** 17](#_Toc29159764)

[**2.3 Roles and Responsibilities** 18](#_Toc29159765)

[**2.4 Risk Assessment** 18](#_Toc29159766)

[**2.5 Cost Estimation** 19](#_Toc29159767)

[**2.6 Project Management Tools** 20](#_Toc29159768)

[**Chapter 3 Requirements Specification** 21](#_Toc29159769)

[**3.1 Stakeholders** 21](#_Toc29159770)

[**3.2 Platform Requirements** 22](#_Toc29159771)

[**3.3 Functional Requirements** 22](#_Toc29159772)

[**3.4 Non-Functional Requirements** 27](#_Toc29159773)

[**Chapter 4 System Design** 28](#_Toc29159774)

[**Architectural Diagram** 29](#_Toc29159775)

[**Use Case Diagram** 30](#_Toc29159776)

[**Component Diagram** 31](#_Toc29159777)

[**Deployment Diagram** 32](#_Toc29159778)

[**Activity Diagrams** 33](#_Toc29159779)

[**Sequence Diagrams** 35](#_Toc29159780)

[**State Transition Diagram** 38](#_Toc29159781)

[**Database Design** 39](#_Toc29159782)

[**Chapter 5 Implementation** 41](#_Toc29159783)

[**5.1 General Overview & Technologies Used** 41](#_Toc29159784)

[**5.2 Algorithms in Depth** 42](#_Toc29159785)

[**5.2.1 Data Collection & Reading** 42](#_Toc29159786)

[**5.2.2 Tokenization & Data Inspection** 43](#_Toc29159787)

[**5.2.3 Data Preprocessing** 45](#_Toc29159788)

[**5.2.4 Splitting Dataset** 46](#_Toc29159789)

[**5.2.5 Feature Extraction** 46](#_Toc29159790)

[**5.2.6 Classification of Tweets** 46](#_Toc29159791)

[**5.2.7 Evaluation and Results** 47](#_Toc29159792)

[**5.2.8 Report Generation** 48](#_Toc29159793)

[**Chapter 6 Testing** 52](#_Toc29159794)

[**6.1 Comparison Testing** 52](#_Toc29159795)

[**6.1.1 Model Performance** 52](#_Toc29159796)

[**6.1.2 End to End Test** 53](#_Toc29159797)

[**Chapter 7 Conclusions & Future Work** 54](#_Toc29159798)

[**Image & Video Classification** 54](#_Toc29159799)

[**Using Bigger Dataset** 54](#_Toc29159800)

[**References** 55](#_Toc29159801)

## **Table of Figures**

[Figure 2.1 Gantt Chart 17](file:///C:\Users\Admin\Desktop\Uni\Grad%20Proj%20(1)\GP%20(2)\Social%20Media%20Analytics%20for%20Emergency%20Response%20Module%20-%20An%20iHelp%20Component.docx#_Toc29159847)

[Figure 4.1 Architectural Diagram 29](#_Toc29159855)

[Figure 4.2 Use Case Diagram 30](#_Toc29159856)

[Figure 4.3 Component Diagram 31](#_Toc29159857)

[Figure 4.4 Deployment Diagram 32](#_Toc29159858)

[Figure 4.5(a) Data Analyzation Activity Diagram 33](#_Toc29159859)

[Figure4.5(b) Disaster Status Tracking Activity Diagram 34](#_Toc29159860)

[Figure 4.6(a) Log-In Sequence Diagram 35](#_Toc29159861)

[Figure 4.6(b) User Interactions Sequence Diagram 36](#_Toc29159862)

[Figure 4.6(c) Admin’s Interactions with System 37](#_Toc29159863)

[Figure 4.7 Sequence Diagram 38](#_Toc29159864)

[Figure 4.9 Database Schema 40](#_Toc29159865)

[Figure 5.1 General Overview & Technologies Used 41](#_Toc29159866)

[Figure 5.2 Machine Learning Workflow 42](#_Toc29159867)

[Figure 5.3 Dataset Snapshot 43](#_Toc29159868)

[Figure 5.4 Dataframe 44](#_Toc29159869)

[Figure 5.5 Data Inspection 44](#_Toc29159870)

[Figure 5.3 Power BI Workflow 48](#_Toc29159872)

[Figure 5.4 Dashboard (1/3) 49](#_Toc29159873)

[Figure 5.5 Dashboard (2/3) 50](#_Toc29159874)

[Figure 5.6 Dashboard (3/3) 51](#_Toc29159875)

## **Table of Tables**

[Table 1.1(a). Features Description 13](#_Toc29159807)

[Table 1.1(b). Platform Comparison 14](#_Toc29159808)

[Table 2.1 High-Level Project Tasks 15](#_Toc29159809)

[Table 2.2(a) Project Risks 18](#_Toc29159811)

[Table 2.4(a). Implementation Costs 19](#_Toc29159812)

[Table 2.4(b) Deployment Costs 19](#_Toc29159813)

[Table 2.3 Project Management Tools 20](#_Toc29159814)

[Table 3.1. System Stakeholders 21](#_Toc29159815)

[Table 3.2. Requirements Input Process and Output 23](#_Toc29159816)

[Table 3.3. Non-functional Requirements 27](#_Toc29159817)

[Table 5.1 Model Performance 47](#_Toc29159834)

# **Chapter** 1 **Introduction**

## **1.1 Overview**

Natural and man-made disasters usually lead to humanitarian emergencies and crises. On average, 600 disasters per year occur worldwide, which lead to many deaths and substantial number of people affected or displaced. In recent disasters, text messaging and social media platforms have been used to call for help. Optimizing the assignment of resources to transporters and selecting the best delivery route may mean the difference between life and death. The iHELP system will aid in streamlining information from different sources and optimizing the allocation of resources to tasks and the commodities to transporters as well as optimizing the transportation routes [2].

To address this problem, iHELP was developed – an Intelligent Humanitarian Emergency Logistics Platform with a simulation-optimization engine that matches people’s needs at affected locations with the available resources. iHELP was presented to us and we were very intrigued at the uniqueness of its approach towards disaster relief and the notion it would aid in saving lives. The way iHelp works is it prioritizes combinations of commodities and resources by an algorithm based on fairness, urgency and transporter utilization. iHELP then assigns transporters of various types and modes to routes efficiently to reach their destinations in the shortest possible time [2].

The following modules need to be developed to make iHELP usable in practice:

1. User Interface for Route Animation using Google Maps and Logistics Information Dashboard
2. Data Input Module
3. Social Media Analytics Module
4. Volunteer Management Module.

In our project, we will be mainly focusing on the Social Media Analytics Module where we will text-mine logs from social media outlets such as Facebook and Twitter to extract information relevant to the affected areas and parse through it.

### **1.2 Problem Statement**

When disasters strike, the situation becomes chaotic due to lack of coordination among governments, emergency response units, and humanitarian organizations. Also, not all information may be available from traditional systems such as 911 [1]. Resources and volunteers may be available, but may not be visible to emergency response systems. There is currently no systems that match resources and skills with people’s needs optimally across different organizations and stakeholders. This results in inefficient use of resources and volunteers for rescue operations, and sub-optimal distribution and transportation of relief.

### **1.3 Related Work**

In our search for products of similar nature, we came upon two types of products, with the intention of finding systems that work towards the same goal of disaster control through matching resources and skills with people’s needs optimally as discussed in the overview. These two types are:

1. Large-Scale disaster relief.
2. Applications directed at average users to offer disaster relief.

Countless disaster relief applications have been released due to natural need for such applications to efficiently accomplish disaster control. Facebook has a new feature called “Crisis Response”, this feature is designed to make it easier for people to find out more about recent crisis and access various tools to support recovery summarized in safety check, offering and finding help, fundraising, and information feed.

Facebook works towards connecting people to tackle the disaster relief problem. The way it finds help is through categorizing offered areas of help such as transportation, food or supplies. Individuals in need can scroll through & select a particular category, Facebook then redirects them to a page where they can browse through posts from other individuals in the affected area who are offering help. If no posts are found, then they have the option of making a request. “Crisis Response” also offers the option of browsing map view [3].

RedCross Organization offers multiple disaster relief applications that enable the user to receive step-by-step instructions to help them know what-to do in case of emergencies such as flood, hurricane, tornado… etc. It offers notifications about distant friends and family in danger with ability to receive warning alerts based on their location [4].

FEMA: coordination of the federal government’s role in the areas of domestic disasters preparation, prevention and effects mitigation, response and recovery [5].

Table 1.1(a) describes the features used in comparison and Table1.1(b) summarizes differences between platforms.

Table 1.1(a). Features Description

|  |  |  |
| --- | --- | --- |
| Feature ID | Feature Name | Description |
| F1 | Text-mining | Collect text from social media logs. |
| F2 | Interactive Maps (heat maps) | Maps to reveal the geospatial distribution of emergency help requests, and the geospatial distribution of emergency infrastructure damages, and a brief description of each specific emergency. |
| F3 | Distinguishing real from fake posts | Implementing intelligent system. |
| F4 | Volunteer Management | Environment to sufficiently manage volunteers. |
| F5 | Classification of the nature & effect of the damage | Implementing intelligent system. |
| F6 | Real-time alerts | Provides needed alerts in timely manner. |
| F7 | Safety-check | Mark individuals as safe. |

Table 1.1(b). Platform Comparison

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| System | Audience | | | | Features | | | | | | |
| Volunteers | ERT | Users | DA | F1 | F2 | F3 | F4 | F5 | F6 | F7 |
| FB | X | x | X | x | x | x | x |  |  | x | x |
| FEMA |  | x | X |  |  |  |  |  |  | x | x |
| RedCross |  |  |  |  |  |  |  |  |  | x | x |

Our application will differ in that it is far more advanced in collecting relevant information from social media logs, text-mining social media, identifying relevant messages (e.g., people seeking different types of help, roads closed, buildings collapsing, areas flooded, power is out, etc.), extract reported locations and geocodes, classify the nature of the damage and help needed, quantify the effect of the damage and amount of help needed, identify what is real and what is not. It also saves the information in a cloud-based database to be accessed by other modules of the system [2].

### **1.4 Document Outline**

Document starts with introductory chapter describing idea behind project and problems it aims to resolve, discussing why it is important to do so, and providing intuitive research of pre-existing comparable systems.

Second chapter discusses project plan which consists of system deliverables, tasks organization using Gnatt chart, roles and responsibilities of team members, and risks accompanying project deployment and implementation along with ways of handling them. Chapter closes with project overall cost estimation, and what tools were utilized to organize the project plan.

Third chapter discusses the system requirements. It begins with identifying the stakeholders of the system, followed by platform requirements, and lastly closes with the functional and non-functional requirements.

Fourth chapter projects the system design, it is shown through Architectural, Use Case, Component, Deployment, Activity, Sequence, State Transition, Entity Relationship (ER), and Database Schema diagrams. Lastly, document closes with collection of references.

# **Chapter** 2 **Project Plan**

### **2.1 Project Deliverables**

The project will provide the following deliverables:

* Web-Based Solution (server side, client side, and database).
* Documentation file, code documentation, user manual.
* Resource Files (diagram designs, figure pictures, etc.).

### **2.2 Project Tasks**

Our project tasks conclude as table 2.1:

Table 2.1 High-Level Project Tasks

|  |  |  |  |
| --- | --- | --- | --- |
| **Task ID** | **Task Name** | **Task Description** | **Time Duration** |
| 1 | Attending Presentation By Prof. Gaith Al Rabadi | Understand the nature of iHELP | 1 day  16 Mar |
| 2 | Client Meeting | Discussing project primitives with Dr. Sufyan | 1 day  20 Mar |
| 3 | Concept Comprehension | Studying the provided documents by Dr. Sufyan describing earlier work done on iHELP | 6 days  22 Mar -  27 Mar |
| 4 | NATO’s Rapid Deployment Logistics Code Read-Through | Discussing the provided code for better and deeper understanding of what we are required to do | 2 days  26 Mar -  27 Mar |
| 5 | Research | Collecting insight about social media analytics | 9 days  1 Apr - 9 Apr |
| 6 | Client Meeting | Discussing system’s non-functional requirements | 1 day  11 Apr |
| 7 | Related Work | Reviewing existing systems of similar nature in order to draw comparisons | 10 days  12 Apr -  21 Apr |
| 8 | Client Meeting | Discussing similarities and iHELP advantages over other comparable systems  Discussing basic view of system functional requirements | 1 day  24 Apr |
| 9 | System Requirements | Composing system requirements through research, related work and meeting stakeholders | 10 days  25 Apr – 4 May |
| 10 | Related Work Modifications | Revisiting Related Work after consulting supervisor | 3 days  26 Apr -  28 Apr |
| 11 | Client Meeting | Discussing basic version of system functional requirements | 1 day  29 Apr |
| 12 | System Requirements (second iteration) | Editing requirements to better fit client’s feedback | 16 days  30 Apr -  15 May |
| 13 | Client Meeting | Interviewing our client; Dr.Gaith in order to collect his input and any insight he can provide about our work so far | 1 day  13 May |
| 14 | System Design | Interface design and system design diagrams and choices. | 11 days  12 May -  22 May |
| 15 | Basic Version for the Documentation | Organizing the chapters and arranging all the pieces together | 5 days  17 May -  21 May |
| 16 | Client Meeting | Discussing created system design diagrams and finalizing graduation project (1) document | 7 days  16 May – 22 May |
| 17 | Implementation | Implementing the system | 15 Oct – 31 Nov |

### **Gantt Chart**

Figure 2.1 shows the tasks of the project on a Gantt chart.

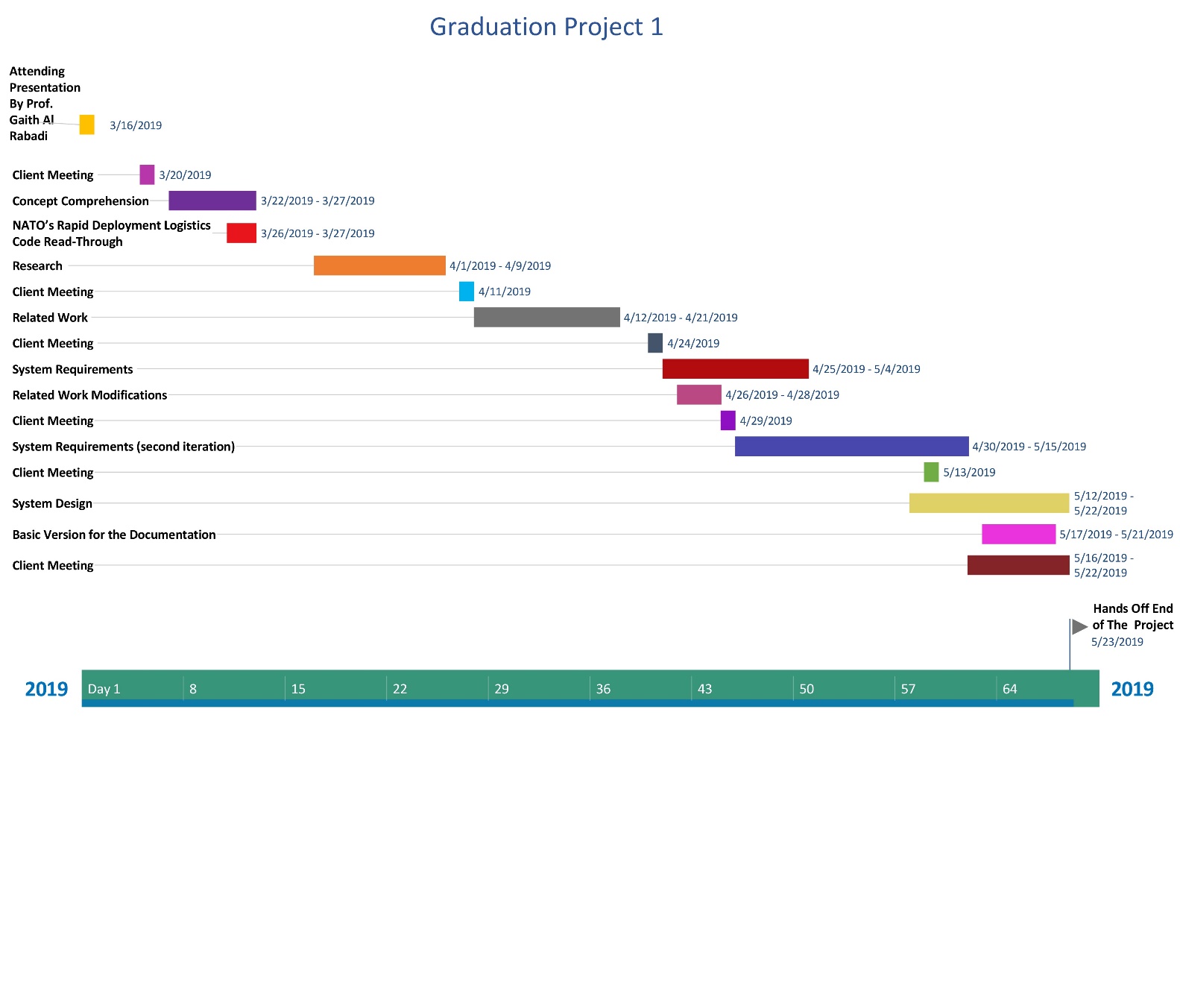


Figure 2.1 Gantt Chart

### **2.3 Roles and Responsibilities**

All the tasks of the project have been carried out by the 3 project members, simultaneously and equally.

### **2.4 Risk Assessment**

Table 2.2 defines the list of project risks and how to handle them:

Table 2.2(a) Project Risks

|  |  |  |
| --- | --- | --- |
| ID | Risk | Countermeasure |
| 1 | Solution inapplicable or unsatisfactory by stakeholders. | * Research for and validate another solution |
| 2 | Time frame not enough. | * Consult an expert |
| 3 | No free online resources available. | * Books * Paid subscriptions to online courses |
| 4 | Some tasks not doable by tools choice of implementation | * Resort to other implementation choices |
| 5 | Availability of a good dataset | * Apply data engineering |

### **2.5 Cost Estimation**

The system is a web-based solution. A fixed monthly cost for renting the web servers hosting the system applies. You can find the implementing and deployment stages cost details in tables 2.4(a) and 2.4(b):

Table 2.4(a). Implementation Costs

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Item Name | Item Description | Monthly Cost |
| 1 | bluehost VPS hosting | Website hosting environment that allows resources such as RAM and CPU to be dedicated to the system. | $30. |

Table 2.4(b) Deployment Costs

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Item Name | Item Description | Monthly Cost |
| 1 | bluehost VPS hosting | Website hosting environment that allows resources such as RAM and CPU to be dedicated to the system. | $30. |

### **2.6 Project Management Tools**

Table 2.3 defines the list of tools used to manage and track the project:

Table 2.3 Project Management Tools

|  |  |
| --- | --- |
| Tool | Usage |
| Google Docs + Microsoft Office Word | Documentation |
| Microsoft Power Point | Gantt Chart |
| Creately | Sequence Diagram + State Diagram + PERT Chart |
| ERDPlus | Entity Relationship Diagram |
| Visual Paradigm | Activity Diagram + Component Diagram |
| LucidChart | Use Case Diagram |

# **Chapter 3 Requirements Specification**

### **3.1 Stakeholders**

The system is affected by four entities, Emergency services, Donating agencies, Volunteers and Citizens.

Table 3.1 depicts the importance of each stakeholder’s role and their relationship to the system.

Table 3.1. System Stakeholders

|  |  |  |
| --- | --- | --- |
| Stakeholder No. | Name | Interest |
| 1 | Emergency services (police, fire-station, hospital...) | Collecting information and sending professional help (doctors, nurses, medications) throughout nations reaching the affected areas. |
| 2 | Donating agencies (transporters, items, food) | Donating any necessary resources available (such as items, equipment, food) and send them to the affected areas. |
| 3 | Volunteers | Volunteers want to be given efficient tasks where they are in fact needed and they want to be called upon within the limits of their free times according to each volunteer’s given schedule. They want to placed among groups that fit their skills and kind of help they can provide. |
| 4 | Citizens | Citizens want to receive aid as soon as possible; they want their calls for help to be heard and registered and acted upon. |
| 5 | Government | Government are interested in being able to know the places of danger to take appropriate action. |

### **3.2 Platform Requirements**

Web based systems are divided into 2 subsystems, the server and client subsystems. For each subsystem, a minimum and a recommended (whenever applicable) hardware and software specifications have been discussed in tables 3.2(a) and 3.2(b):

System Server Side Platform Requirements:

* Web Server
* AI libraries
* MYSQL Server
* Google Maps Services
* Python 2.7, 3.7

System Client Side Platform Requirements:

* Web browser that supports HTML5, CSS3, and Javascript.

### **3.3 Functional Requirements**

1. The system must allow admin to update their password.
2. The system must allow admin to perform all CRUD operations to the volunteers, ERT & HA & DA accounts.
3. The system must allow HA to perform CRUD operations to their inventories and items.
4. The system must allow HA to perform CRUD operations to their transportation fleet.
5. The system should allow volunteers to sign in.
6. The system must allow volunteers to update their schedules and locations.
7. The system should allow volunteers to enter their skillset and preferred workgroup.
8. The system must allow volunteers to check-in once they are on site.
9. The system might allow volunteers to send feedback in order to enhance VM process.
10. The system must allow volunteers to view interactive maps that reflect the status of the disaster.
11. The system might allow ERT to check-in once they are on site.
12. The system might allow ERT to send updates.
13. The system must collect information from social media logs.
14. The system must decide whether collected social media posts are genuine or fake (with highest accuracy possible).
15. The system must automatically classify the nature of disaster and help needed.
16. The system must notify volunteers/humanitarian agencies of any imminent disasters.
17. The system must extract reported locations and geocodes.
18. The system must automatically quantify the nature of disaster.
19. The system might create interactive maps (heat maps) to reveal the geospatial distribution of emergency help requests, and the geospatial distribution of emergency infrastructure damages, and a brief description of each specific emergency.
20. The system must save the information in a cloud-based database to be accessed by other modules and to be presented on the output dashboard.

Table 3.2 shows the inputs, outputs, process, constraints, and class of each requirement.

Table 3.2. Requirements Input Process and Output

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| REQ. No. | Input | Process | Output | Constraints | Class (Importance) |
| R1 | Admin ID, Old Password, New Password | Match old password to database and update new password | User Profile | New Password length does not exceed 254 characters | Essential |
| R2 | Volunteer ID, Volunteer Name  ERT ID, ERT Name  HA ID, HA Name  DA ID, DA Name | Add volunteer to DB  Add ERT to DB  Add HA to DB  Add DA to DB | Updated Volunteer List  Updated ERT List  Updated HA List  Updated DA List | - | Essential |
| Volunteer ID  ERT ID  HA ID  DA ID | Remove volunteer from DB  Remove ERT from DB  Remove HA from DB  Remove DA from DB | Updated Volunteer List  Updated ERT List  Updated HA List  Updated DA List | - | Essential |
| R3 | Inventory ID, Inventory Name | Add inventory to DB | Inventory List | - | Essential |
| Inventory ID | Load inventory statistics to view | Statistics Page  Dashboard |
| Inventory ID, New Inventory Name | Update inventory in DB | Inventory Dashboard |
| Inventory ID | Remove inventory from DB | Inventory Dashboard |
| Inventory Item ID, Inventory Item Name | Add inventory item to DB | Inventory Dashboard |
| Inventory Item ID | Load inventory item statistics to view | Inventory Items Statistics Page |
| Inventory Item ID, New Inventory Item Name | Update inventory items in DB | Inventory Items Dashboard |
| Inventory Item ID | Remove inventory Item from DB | Inventory Items Dashboard |
| R4 | Transporter ID | Add vehicle to transportation DB | Transporters Dashboard | Vehicle must have reasonable capacity | Essential |
| Transporter ID | Load transporter object to view | Transporter Dashboard |
| Transporter ID, Transporter New Name | Update transporter in DB | Transporter Dashboard |
| Transporter ID | Remove transporter from DB | Transporter Dashboard |
| R5 | Volunteer Email, Volunteer Password | Match inputs to information on DB | iHelp Dashboard | Email must be valid | Essential |
| R6 | Volunteer Location, Available Days | Link location & available days to volunteer profile on DB | Volunteer Profile | Location must be in area where volunteers resides | Essential |
| R7 | Volunteer Chosen Skillset & Workgroup | Link skillset & workgroup to volunteer profile on DB | Volunteer Profile | - | Recommended |
| R8 | Volunteer ID & Location | Validate entered location with affected area location | Pinpoint Volunteer Location on Map | Volunteer must have their location turned on | Essential |
| R9 | Feedback | Send feedback to admin | Message assuring volunteer their feedback was received | - | Recommended |
| R10 | Location of Affected Area | Load map to view | Animation using Google Maps | Volunteer must have their location turned on | Essential |
| R11 | ERT ID & Location | Keep track of ERT via animation map | Pinpoint ERT Location on Map | ERT must have their location known | Recommended |
| R12 | Real-time Communications | Send feedback to admin | - | Communications must arrive on time | Recommended |
| R13 | - | Collect related data | Collected social media Data | - | Essential |
| R14 | Collected social media Data | Distinguish genuine posts from fakes | Class (genuine or fake) for every post or social media item collected | Accuracy Level | Essential |
| R15 | Collected Data | Classify nature of disasters and the help needed and store in cloud-based DB | Datasheet | Accuracy Level | Essential |
| R16 | Emergency Alert Recipients  Emergency Alert Type  Emergency Alert Body | Send alert notification to recipients (volunteers, HA, DA, ERT, govt. etc..) | - | - | Essential |
| R17 | Collected Data | Extract reported locations and geocodes | Save extracted locations and geocodes | - | Essential |
| R18 | Collected Data | Quantify nature of disasters and the help needed and store in cloud-based DB | Datasheet | Accuracy Level | Essential |
| R19 | Extracted locations and geocodes | Create interactive maps (heat maps) | - | - | Recommended |
| R20 | Collected Data | Save the data in a cloud-based database | - | - | Essential |

### **3.4 Non-Functional Requirements**

Table 3.4 lists the non-functional requirements with a brief description of each requirement.

Table 3.3. Non-functional Requirements

|  |  |
| --- | --- |
| **Category** | **Requirement** |
| Usability | The system should be easy to handle and use. |
| Reliability | The system should provide trusted information from different resources and link them to the responsible bodies.  The system should be able to sustain its capabilities regardless of heavy traffic. |
| Portability | The system should allow users connected to internet to access the platform regardless of their devices’ types. |
| Security | The system must maintain the confidentiality, integrity, and availability of the information and system functions. |
| Performance | The system should allow the users to check on the disaster status in 3 seconds. |
| Scalability | The system design should scale well regardless of the number of uses involved. |

# **Chapter 4 System Design**

This chapter will cover the design diagrams of the system, a list of the following diagrams is presented:

1. Architectural Diagram.

2. Use Case Diagram.

3. Component Diagram.

4. Deployment Diagram.

5. Activity Diagrams.

6. Sequence Diagrams.

7. State Transition Diagram.

8. Entity Relationship (ER) Diagram.

9. Database Schema.

\*All diagrams are available as a soft copy in high resolution.

### **Architectural Diagram**

Figure 4.1 shows the architectural diagram of the system.

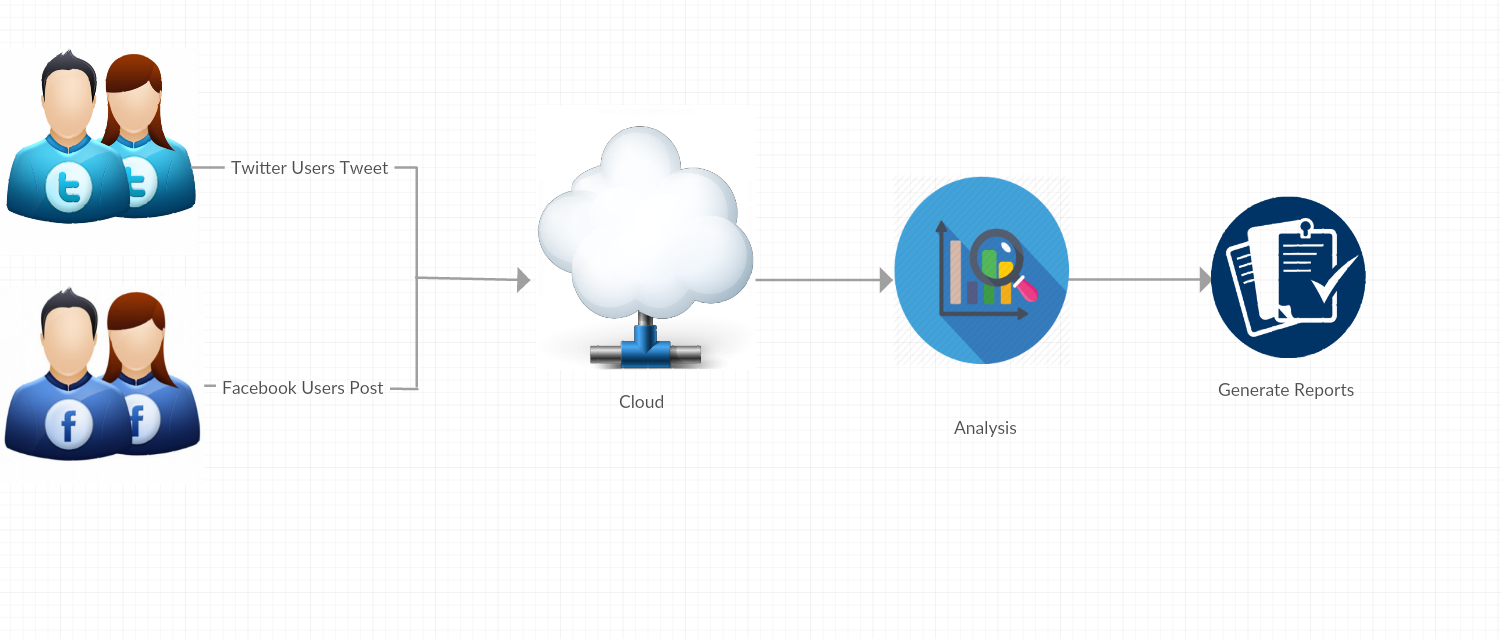


Figure 4.1 Architectural Diagram

### **Use Case Diagram**

Figure 4.1 shows the use case diagram of the system.

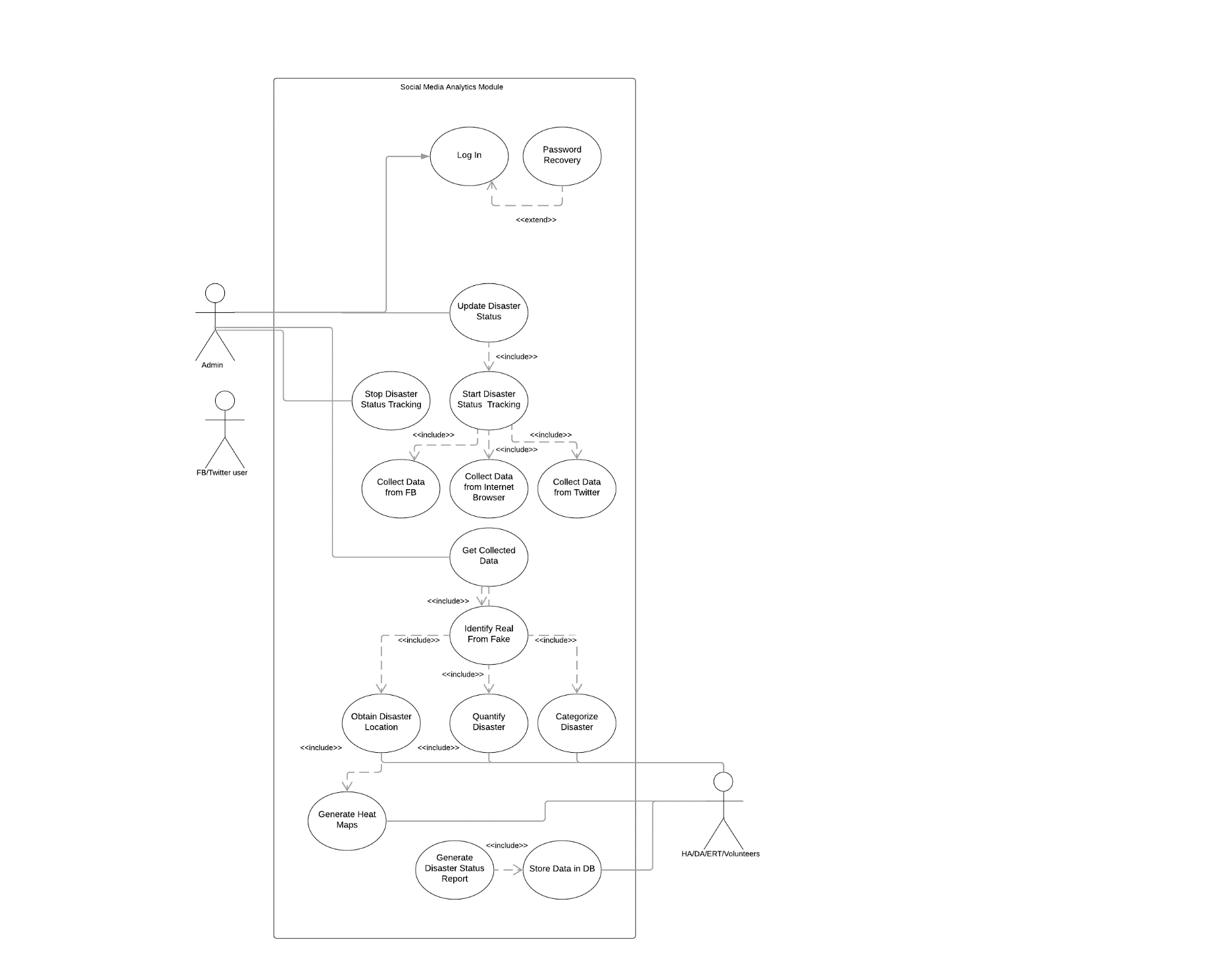


Figure 4.2 Use Case Diagram

### **Component Diagram**

Figure 4.3 shows the component diagram of the system.

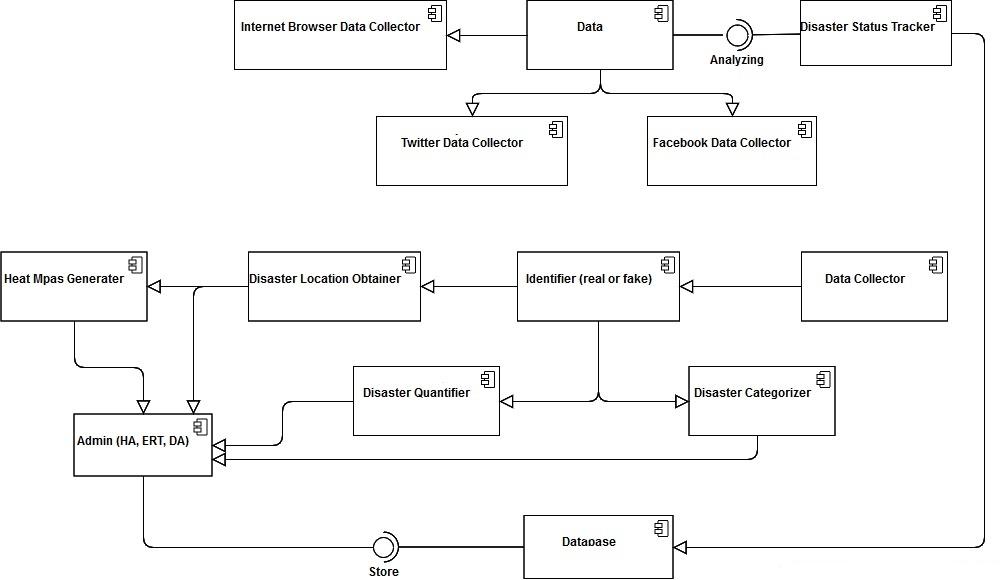
****

Figure 4.3 Component Diagram

### **Deployment Diagram**

Figure 4.4 shows the deployment diagram of the system.

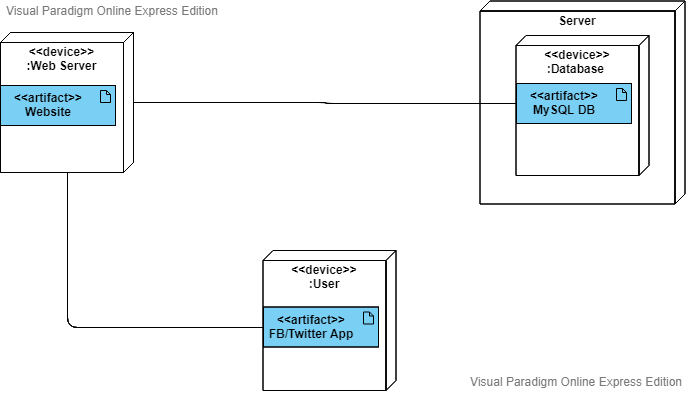


Figure 4.4 Deployment Diagram

### **Activity Diagrams**

Figure 4.5(a) shows the data analyzation activity diagram of the system.

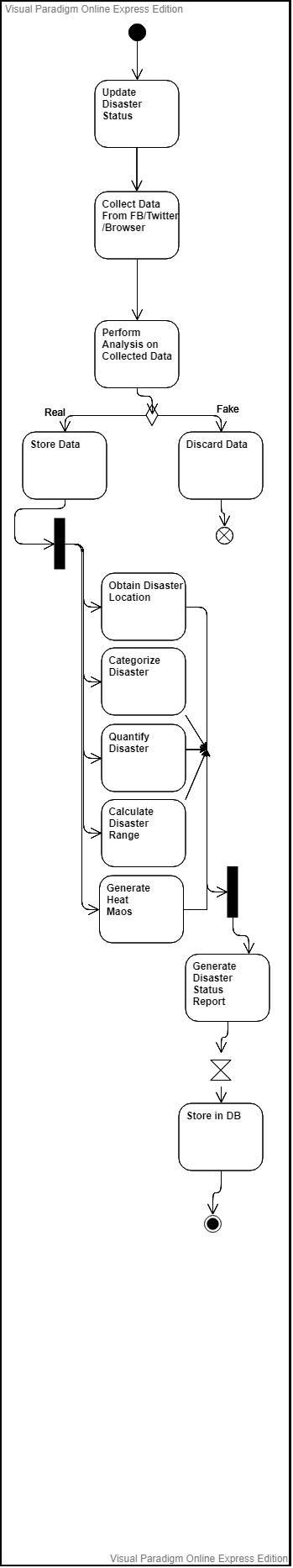
****

Figure 4.5(a) Data Analyzation Activity Diagram

Figure 4.5(b) shows the disaster status tracking activity diagram of the system.

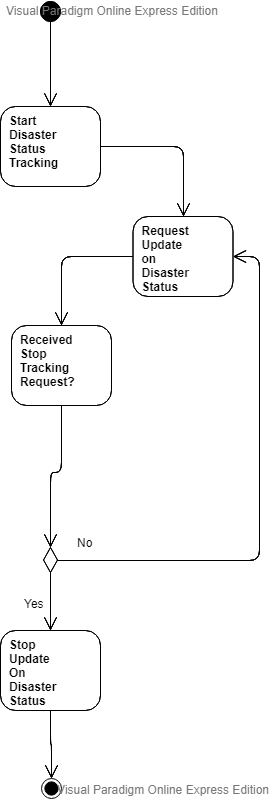


Figure4.5(b) Disaster Status Tracking Activity Diagram

### **Sequence Diagrams**

Figure 4.6(a) shows the sequence diagram for the log in use case.

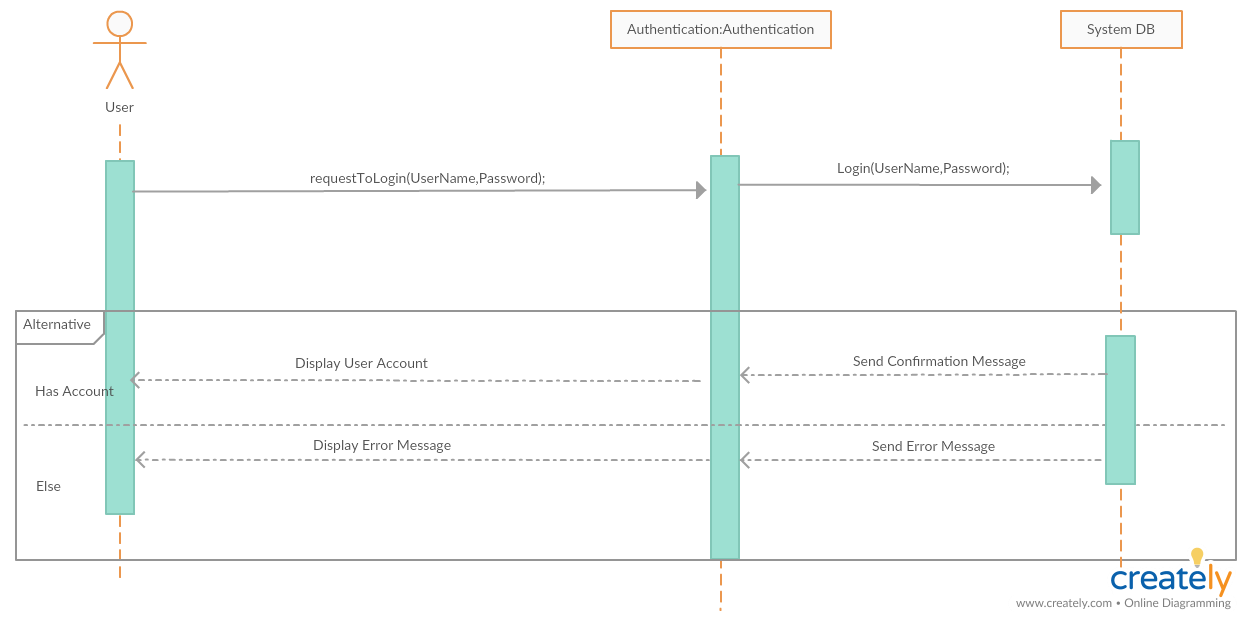


Figure 4.6(a) Log-In Sequence Diagram

Figure 4.6(b) shows the sequence diagram for users’ interactions with system.

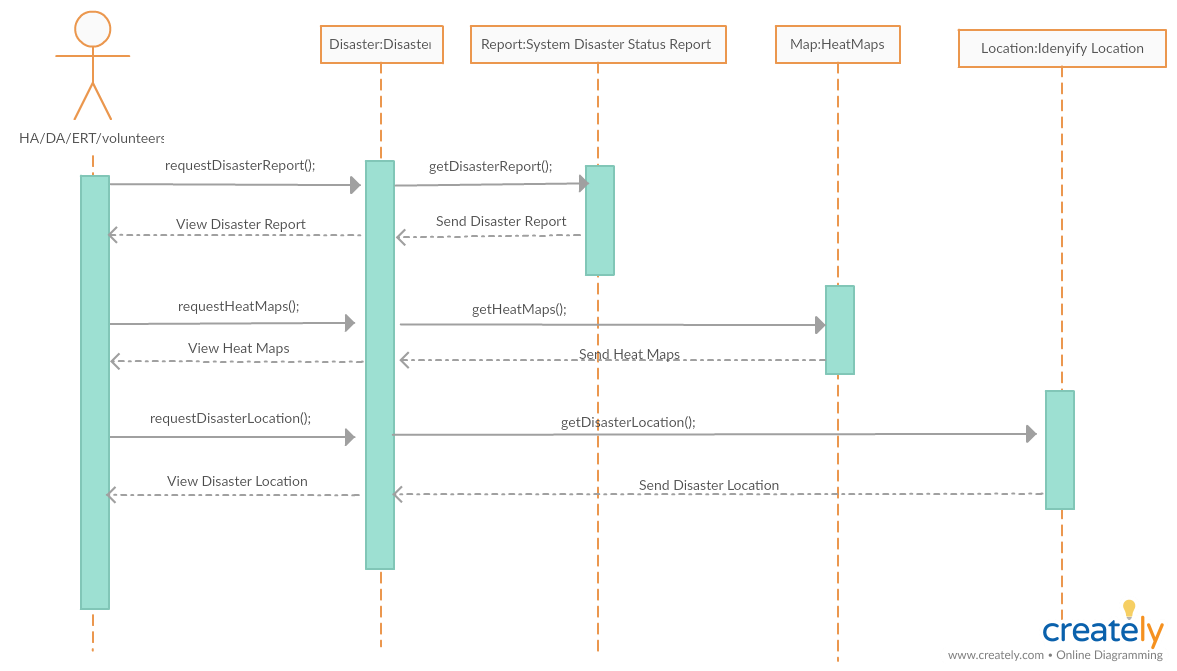


Figure 4.6(b) User Interactions Sequence Diagram

Figure 4.6(c) shows the sequence diagram for admin interactions with system.

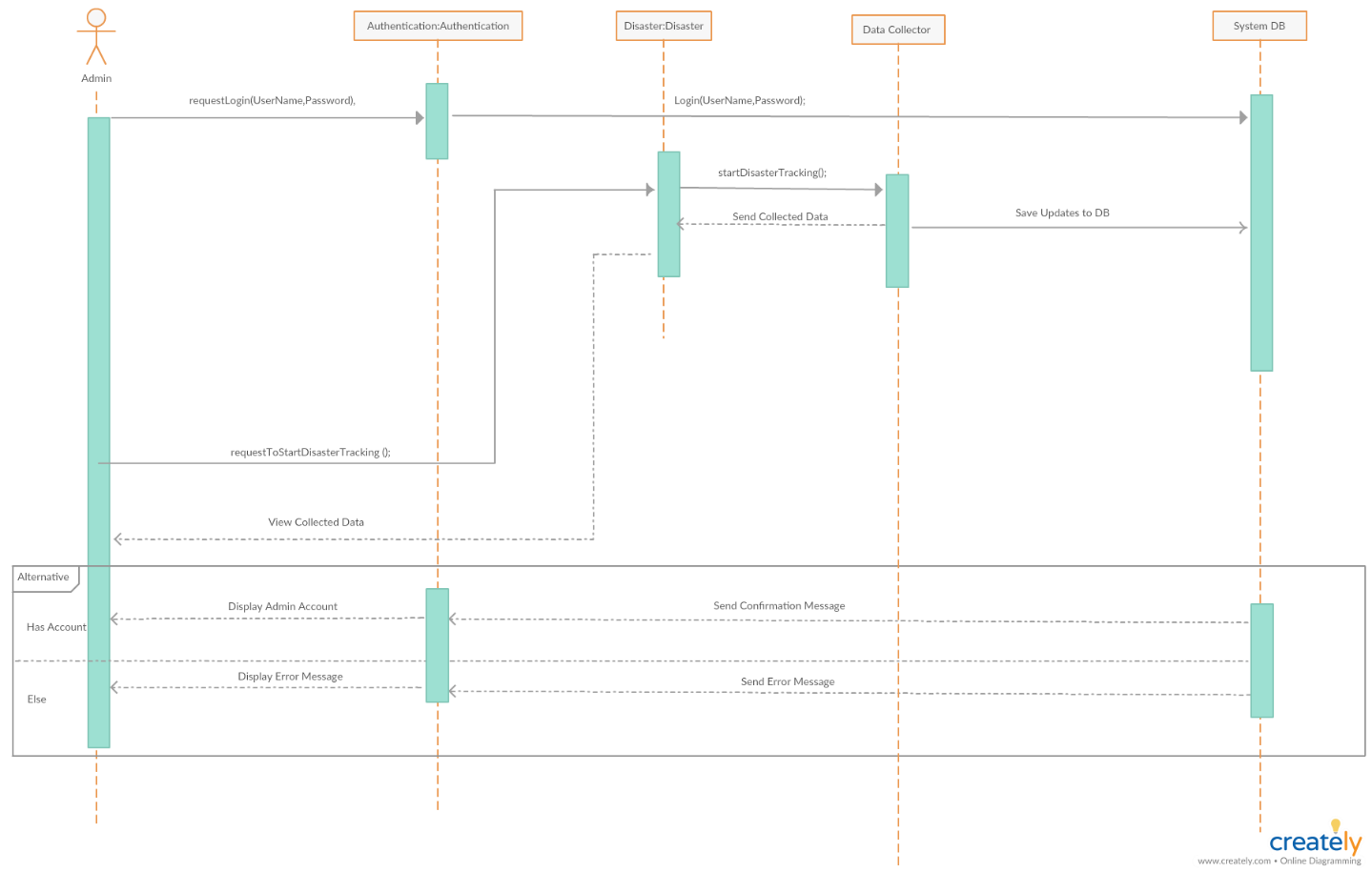
****

Figure 4.6(c) Admin’s Interactions with System

### **State Transition Diagram**

Figure 4.7 shows the sequence diagram for the log in use case.

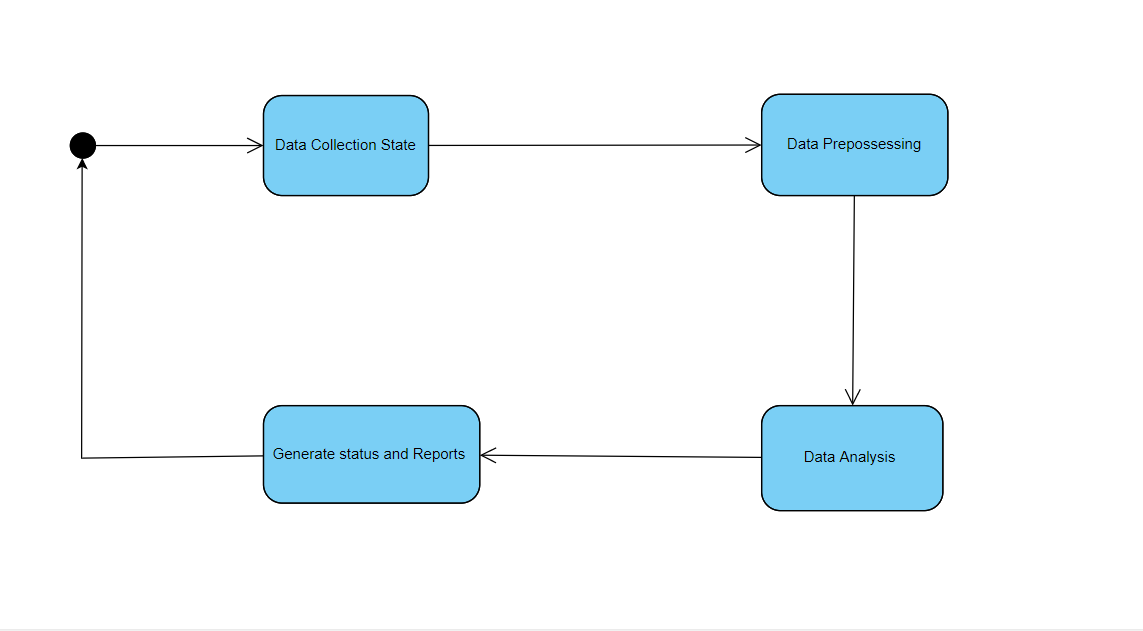
****

Figure 4.7 Sequence Diagram

### **Database Design**

Figures 4.8 and 4.9 describe the design of the ERD and the Database Schema for the system.

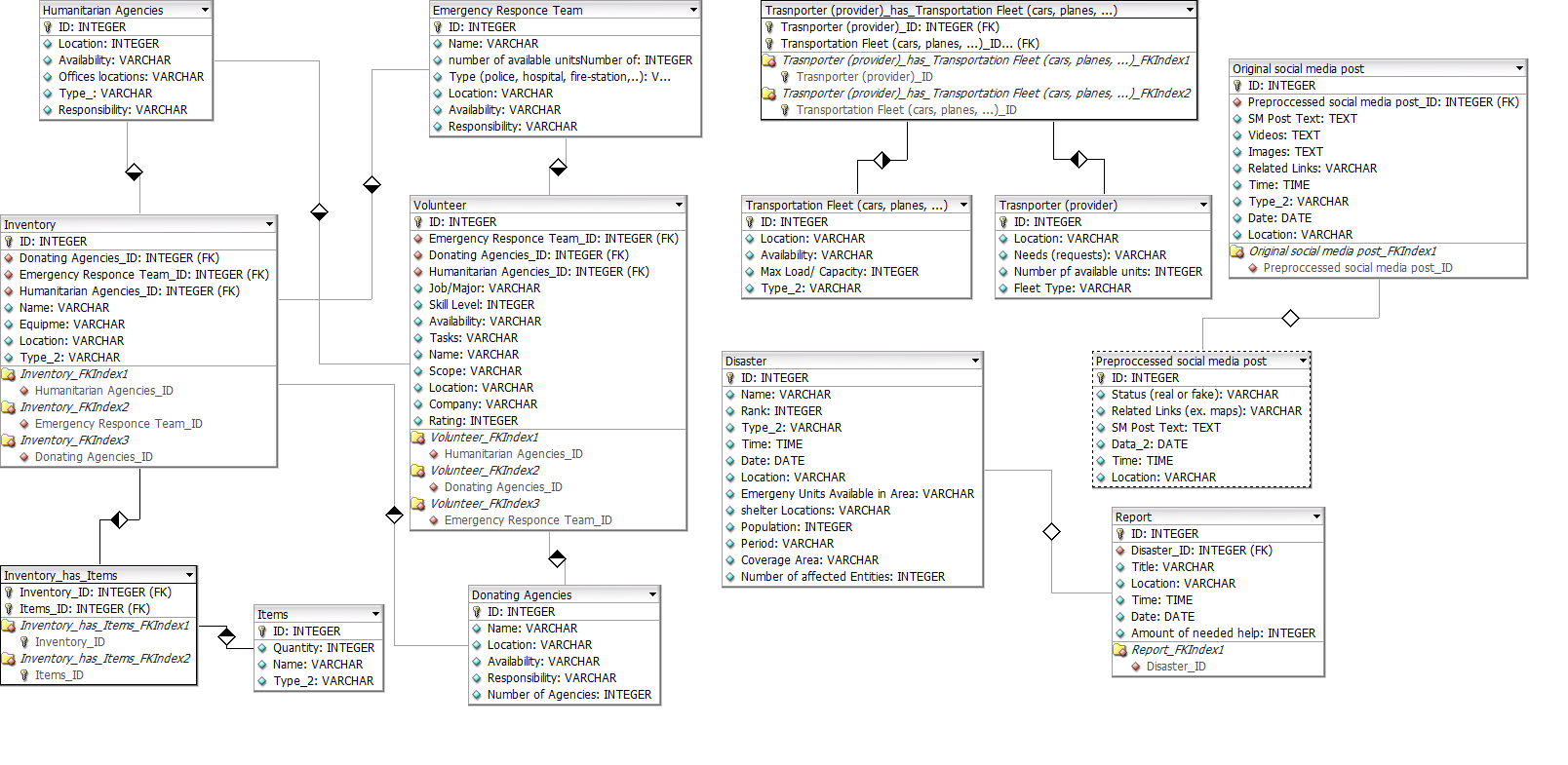


Figure 4.8 Entity Relationship Diagram

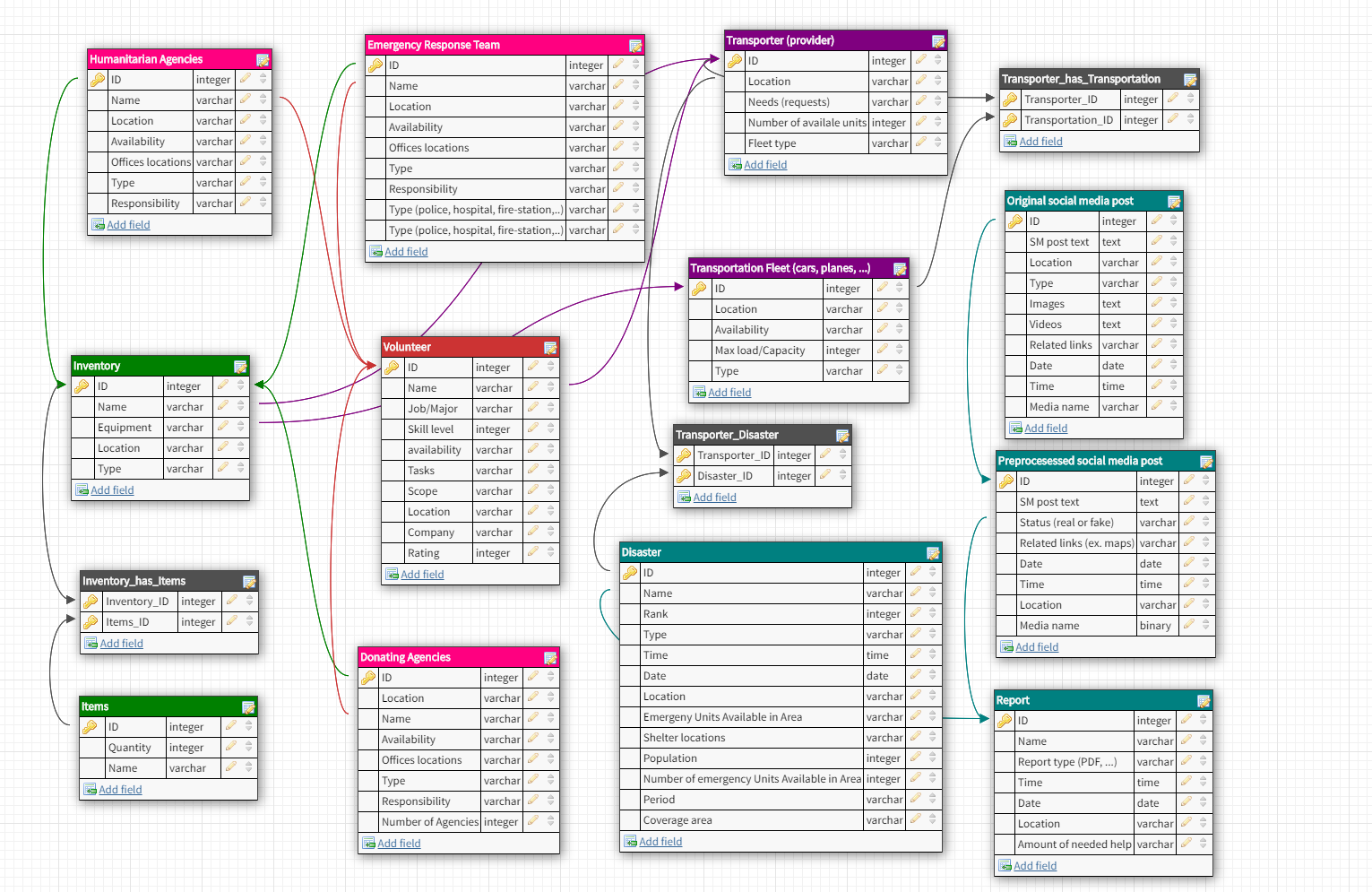


Figure 4.9 Database Schema

# **Chapter 5 Implementation**

### **5.1 General Overview & Technologies Used**

The platform deploys a machine learning model (logistic regression), a total of 15 packages and libraries. Spyder IDE was used to run the Python code, Power BI Desktop to generate dashboard and Power BI Service to publish dashboard online. Nine data descriptive figures are generated through the Python code, seven data descriptive figures appear on the online Power BI report. Dataset was provided through figure8 opensource service.

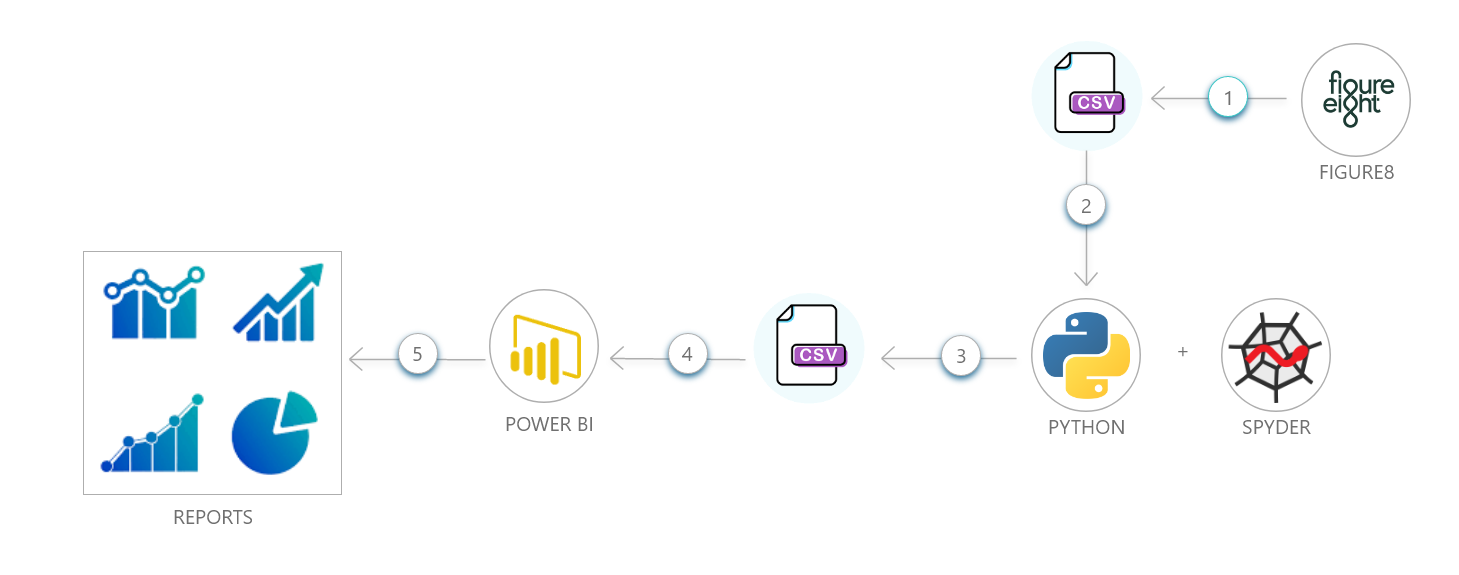


Figure 5.1 General Overview & Technologies Used

### **5.2 Algorithms in Depth**

This section describes 8 critical features in the platform; Collecting & Reading Dataset, Tokenization and Data Inspection, Data Preprocessing, Splitting Dataset, Classification of Tweets, Evaluation and Results and Reports Generation. The implementation details of each feature are described in the following subsections:

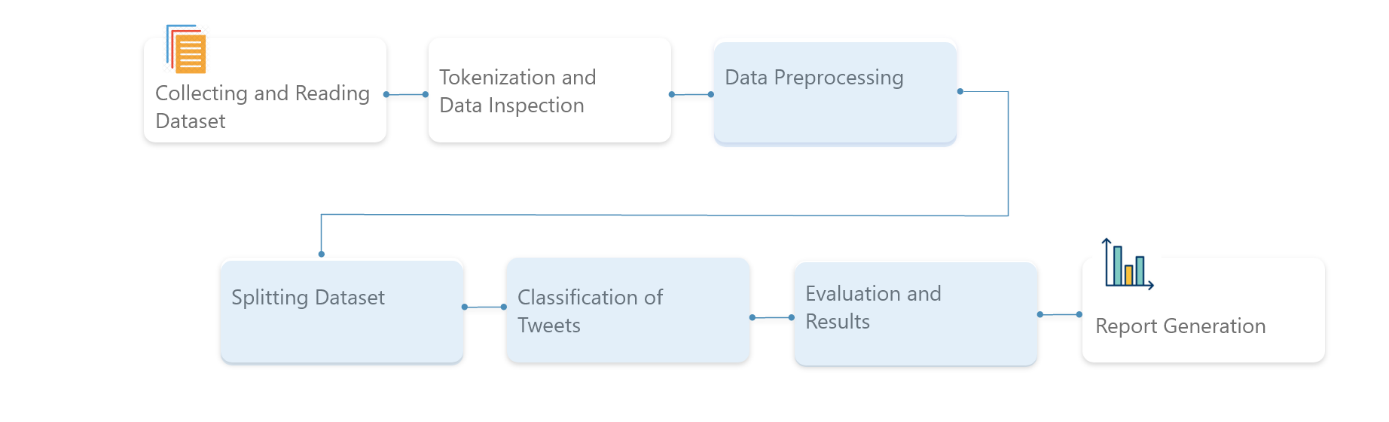


Figure 5.2 Machine Learning Workflow

### **5.2.1 Data Collection & Reading**

We collected over 10,000 crisis-related tweets from Twitter posted from August to September in 2015. CSV file snippet below shows the list of crisis keywords (e.g. ablaze, quarantine), crisis location and countries where they took place, and the relevancy of each tweet along with the tweet text, id and user id. We collected this dataset using Figure8. Figure8 (formerly known as Dolores Lab, CrowdFlower) is a human-in-the-loop machine learning and artificial intelligence company based in San Francisco. In 2015, Figure8 announced the Data For Everyone initiative, which included a collection of data sets available to researchers and entrepreneurs. In 2009, the company provided work for refugees in Kenya who completed microtasks; iPhone users donated their time by checking for accuracy through the app Give Work. After the 2010 Haiti earthquake, Figure8 again worked with Samasource to help Haitians find work through the application GiveWork. Figure8 provides different convenient ways to collect datasets from Twitter using the Twitter’s streaming API. One can use different data collection strategies. For example, collecting tweets that contain some keywords and are specifically from a particular geographical area/region/city (e.g. New York). [6]

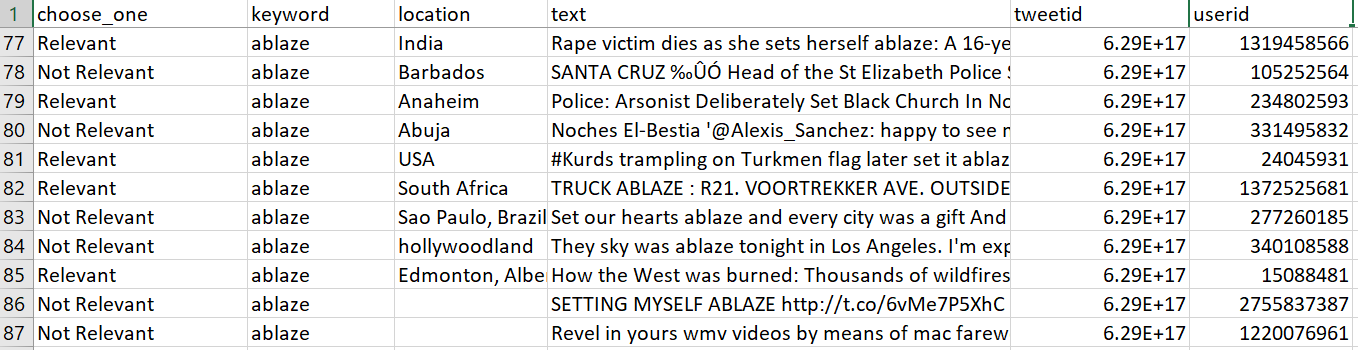


Figure 5.3 Dataset Snapshot



### **5.2.2 Tokenization & Data Inspection**

Tokenization is a step which splits longer strings of text into smaller pieces, or tokens. Larger chunks of text can be tokenized into sentences, sentences can be tokenized into words, etc.

For our task, we will tokenize our sample text into a list of words. This is done using NTLK's RegexpTokenizer() function.

from nltk.tokenize import RegexpTokenizer

token = RegexpTokenizer(r'\w+')#To separ

data["tokens"] = data["text"].apply(token.tokenize)

print(data.head())

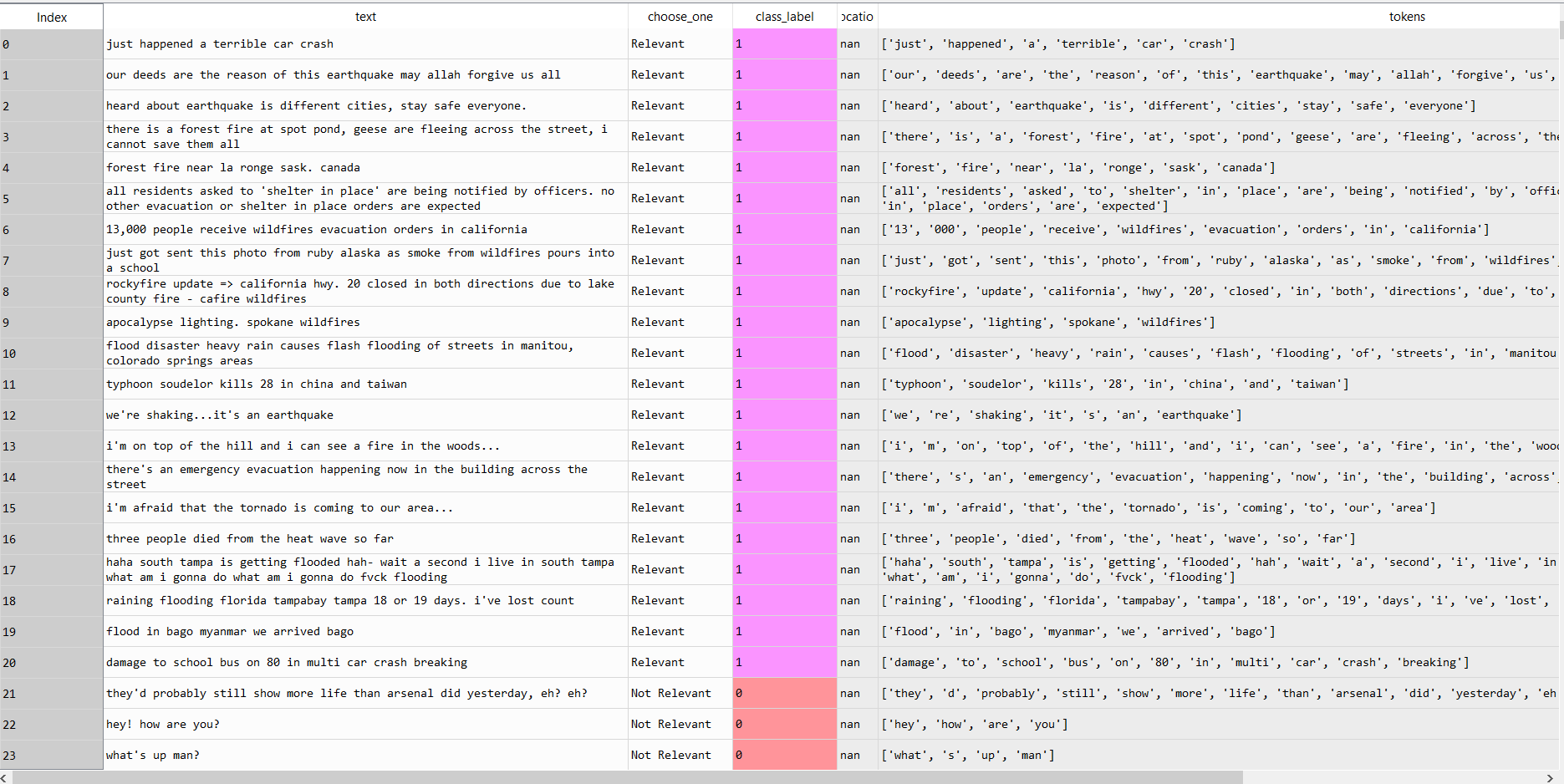


Figure 5.4 Dataframe

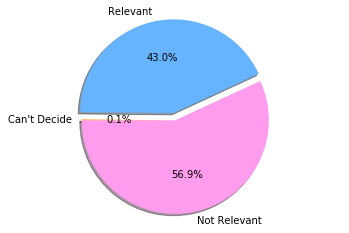


Figure 5.5 Data Inspection

### **5.2.3 Data Preprocessing**

Feature engineering is the method of converting raw data into features that improve representation of the underlying problem to the predictive models, ensuing enhanced model accuracy on test data.

To realize this, we first represented our csv file in a dataframe format, we also introduced a new feature by adding a column named “class label”. In this column we sorted our data into 3 groups based the “choose one” (relevancy) column where “1” represents “Relevant”, 0 represents “Not relevant” and “2” is for “Can’t decide”.

Next, we extracted columns “text”, “choose one”, “class label”, “location” and placed them into our dataframe for later use.



This step preprocesses the tweet content before creating the numeric vector. This task removes the tweets which contain any links or special characters from collected tweets. Finally, all text is converted to lower case.

****

### **5.2.4 Splitting Dataset**

Dataset is split at random 70% to 30% between training and testing.

#Splitting the Data

tweets=dataset[text].tolist()

labels=dataset[class\_label].tolist()

x\_train,x\_test,y\_train,y\_test=train\_test\_split(tweets,labels,test\_size=.30,random\_state=5)

### **5.2.5 Feature Extraction**

To make sense of huge amounts of collected Twitter text, we consider a basic operation, that is to transform the text into some numeric or vector representation. This numeric representation should illustrate important characteristics of the text. There are many such techniques, in this project we use TF-IDF (Term Frequency, Inverse Document Frequency) which means weighing words by how frequent they are in our dataset, disregarding words that are too frequent (the, of ..etc.), as they just add extra noise.

from nltk.corpus import stopwords  
list=stopwords.words('english')  
tfidf\_vec=TfidfVectorizer(stop\_words=list)  
x\_train\_tfidf=tfidf\_vector.fit\_transform(x\_train)  
x\_test\_tfidf=tfidf\_vector.transform(x\_test)

### **5.2.6 Classification of Tweets**

This classifies as multiclass categorization problem in which cases are categorized into one of several classes. Specifically, we aim at learning a predictor h : X → Y, where X is the set of tweet text and Y is a finite set of categories. For this purpose, we use Logistic Regression for its simplicity, and is ease of interpretation.

#Fitting a classifier

from sklearn.linear\_model import LogisticRegression

Logistic\_model\_tfidf=LogisticRegression( C=30.0,class\_weight='balanced', solver='newton-cg', multi\_class='multinomial', n\_jobs=-1, random\_state=5)

#train the model

#test the model

Logistic\_model\_tfidf.fit(x\_train\_tfidf, y\_train)

predicted\_tfidf=Logistic\_model\_tfidf.predict(x\_test\_tfidf)

### **5.2.7 Evaluation and Results**

The classification report visualizer displays the precision, recall, F1, and support scores for the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1-score** |
| 0 | 0.80 | 0.83 | 0.81 |
| 1 | 0.76 | 0.73 | 0.74 |
| 2 | 0.0 | 0.0 | 0.0 |
| avg | 0.78 | 0.78 | 0.79 |
| Accuracy: 77.44% | | | |

Table 5.1 Model Performance

Dataset was successfully annotated with predefined three labels at 77.44% accuracy on average which is acceptable.

### **5.2.8 Report Generation**

****

Figure 5.3 Power BI Workflow

In order to turn our data into meaningful visualizations, Power BI is a business analytics service by Microsoft. Its purpose is to provide interactive visualizations and business intelligence abilities with an interface where its easy enough for end users to create their own reports and dashboards.

Power BI provides cloud-based BI services, known as "Power BI Services", along with a desktop based interface, called "Power BI Desktop". [7]

Firstly, we import different result datasets in CSV file format onto Power BI Desktop. Second, we apply a set of modifications using python scripts on the datasets when needed in order to illustrate data as desired.

Lastly, we use Power BI Services to produce interactive dashboards that are accessible online.

Dashboards are as follows:

~~~~

Figure 5.4 Dashboard (1/3)

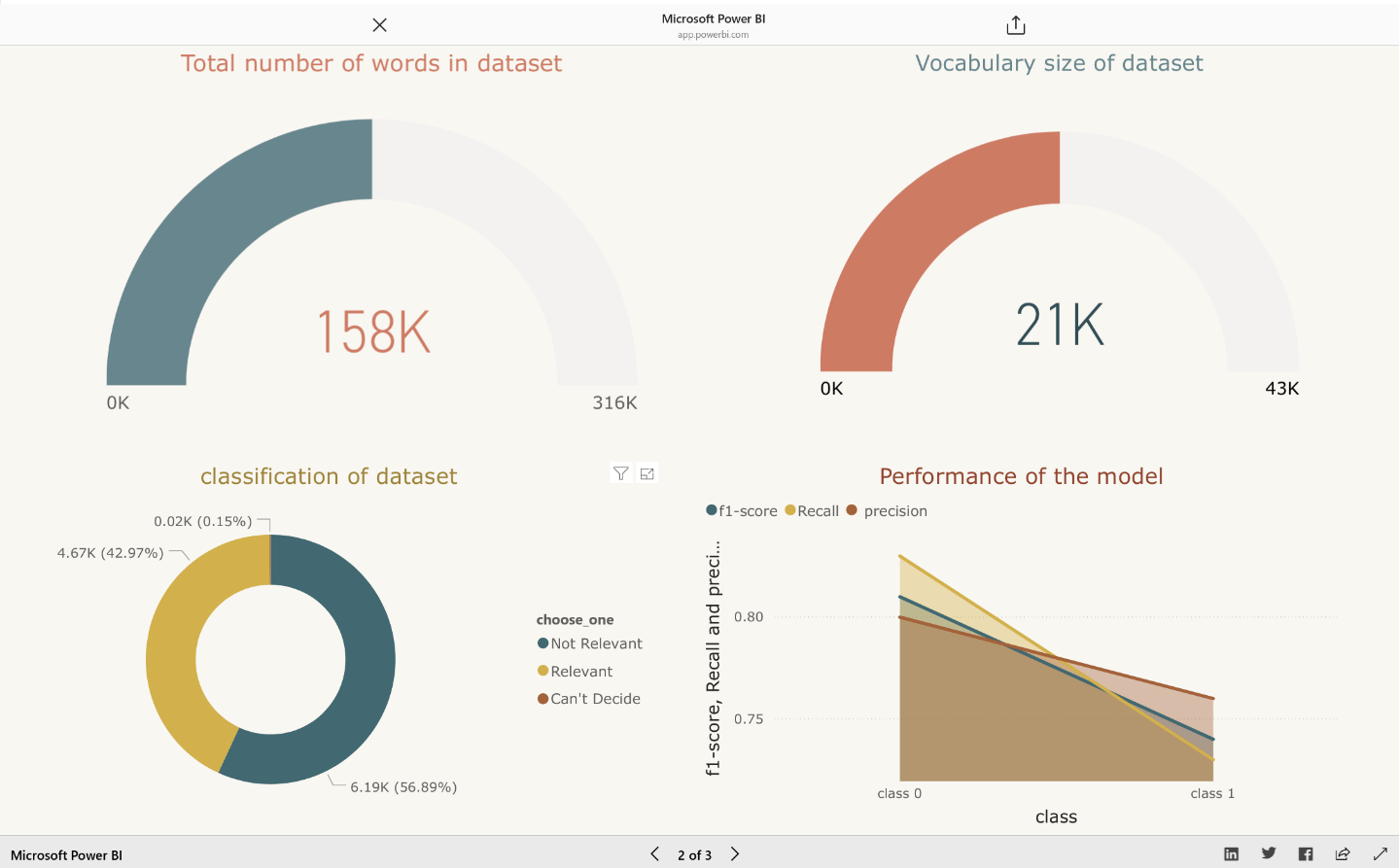
****

Figure 5.5 Dashboard (2/3)

****

Figure 5.6 Dashboard (3/3)

# **Chapter 6 Testing**

### **6.1 Comparison Testing**

### **6.1.1 Model Performance**

Execution 1:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Predicted** | |
|  |  | **Positive** | **Negative** |
| **Actual** | **Positive** | 986 (TP) | 409 (FN) |
| **Negative** | 318 (FP) | 1541 (TN) |

For above model Accuracy = 77.44%. Counter Vectorizer was used for word vectorization inside the model. For Execution 2 we tried a slightly more subtle approach. On top of our bag of words model, we use a TF-IDF (Term Frequency, Inverse Document Frequency) as previously mentioned in section 5.2.4.

Execution 2:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Predicted** | |
|  |  | **Positive** | **Negative** |
| **Actual** | **Positive** | 1024 (TP) | 374 (FN) |
| **Negative** | 325 (FP) | 1534 (TN) |

For above model Accuracy = 78.39%. Here we get better TP and FN result and slightly better accuracy.

### **6.1.2 End to End Test**

**Test ID:** TC1\_Reports

**Assumption:** Load the test data set and verify it has an acceptable output

|  |  |  |  |
| --- | --- | --- | --- |
| **Input** | **Expected Output** | **Pass/Fail** | **Comments** |
| List of locations all locations | Heat map | Fail | 7,000 locations were displayed out of 9,000 |
| List of total words | Size Figure | Pass | - |
| List of Relevancy | Donut Chart | Pass | - |
| Performance Metrics | Area Chart | Pass | - |
| List of top disasters | Tree Map | Pass | - |
| List of irrelevant words | Line & Clustered Column Chart | Pass | - |

# **Chapter 7 Conclusions & Future Work**

We present Twitter text consisting of over 10,000 tweets. We implement a machine-learning classifier to confirm the efficiency of the datasets. We believe that these resources and the tools built using them will ultimately prove useful for humanitarian institutions.

### **Image & Video Classification**

We believe employing deep learning through image classification model will significantly enhance performance and output of our model, as a larger scope of social media users share imagery and video at the occurrence of any disaster.

### **Using Bigger Dataset**

Larger dataset leads to higher accuracy of the model.

## **References**

|  |  |
| --- | --- |
| [1] | D. Rabadi, "Social Media Data Analytics for Emergency Response". |
| [2] | "[2]," [Online]. Available: http://ihelplogistics.com/. |
| [3] | "[3]," [Online]. Available: https://www.facebook.com/about/crisisresponse/. |
| [4] | "[4]," [Online]. Available: https://www.redcross.org/. |
| [5] | "[5]," [Online]. Available: https://www.fema.gov/. |
| [6] | [Online]. Available: https://en.wikipedia.org/wiki/Figure\_Eight\_Inc.. |
| [7] | [Online]. Available: https://en.wikipedia.org/wiki/Power\_BI. |