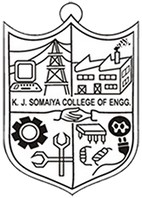
**K. J. SOMAIYA COLLEGE OF ENGINEERING**

**SOMAIYA VIDYAVIHAR**



**Source Adaptive Disinformation Detection**

Software Design Description

**PROJECT ID: 19**

**Project By:**

|  |  |
| --- | --- |
| Heet sakaria (Team leader) | 1714110 |
| Vedang parasnis | **1714100** |
| Hritik Jaiswal | **1714086** |

*Under the guidance of –* **Dr. Irfan Siddavatam**

**DEPARTMENT OF INFORMATION TECHNOLOGY A. Y. 2020-2021**

**Contents**

1. [Introduction 3](#_TOC_250023)
   1. [Design Overview 3](#_TOC_250022)
   2. [Requirement Traceability Matrix 3](#_TOC_250021)
2. System architecture design 4
   1. [Chosen System Architecture 3](#_TOC_250019)
   2. [System Interface Description 5](#_TOC_250017)
      1. [User Interface 5](#_TOC_250016)
      2. [Software Interface 5](#_TOC_250015)
3. [Detailed Description of Components 5](#_TOC_250014)
   1. [Component](#_TOC_250013) 1: User Interface 6
   2. [Component](#_TOC_250012) 2: Web Crawler 6
   3. [Component](#_TOC_250011) 3: Database 6
   4. Component 4: Machine learning model 6
4. [System Architecture 7](#_TOC_250007)
   1. [Use Case Specification Using Template 0](#_TOC_250006)7
      1. Use case diagram - 1 7
      2. Use case diagram - 2 8
      3. Component diagram - 3 9
5. [Data](#_TOC_250002) flow specification 10
   1. Level 0 DFD 10
   2. Level 1 DFD 11

# Introduction

## Design Overview

Disinformation detection system is a system for detecting disinformation in fake news. It is s a web based application that enables the users from field of information security, information law and also normal users to determine the disinformation in the news articles and provides awareness so as to prevent getting fallacious information about the given topic Disinformation identification System mainly works to identify the fabricated content in the news and provide adaptive based approach to adapt itself and thereby enabling itself to a state as to determine with confidence that the given information generated from this source is of high chance being fake using machine learning .We are following an hybrid based approach which is and ensemble and feedback based approach by analyzing transitive relations and two level ensemble approach so as the domain in determining fake news can work on variety of news as an input .We focus on providing accurate results as of proving scrapping data from various sources so as to provide wider category for classification. For Working in real time, we are scraping articles from various renowned sources and thereby passing this raw data to our machine Learning models based on source generated by data and prediction provided as a analytical report to End User.

## Requirement Traceability Matrix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | User | Machine Learning Models | Database | Web Scrapper |
| Article Scrapping | Y | Y | N | Y |
| Twitter Tweets Scrapping |  |  |  | Y |
| Fast and Secure Scrapping |  |  |  | Y |
| Inteligent Model Selection |  | Y | N |  |
| Hyperparameter Tuning |  | Y |  |  |
| Previous Response results | Y |  | Y |  |
| Predictive analysis Report | Y | Y | Y |  |

# 

# System Architectural Design

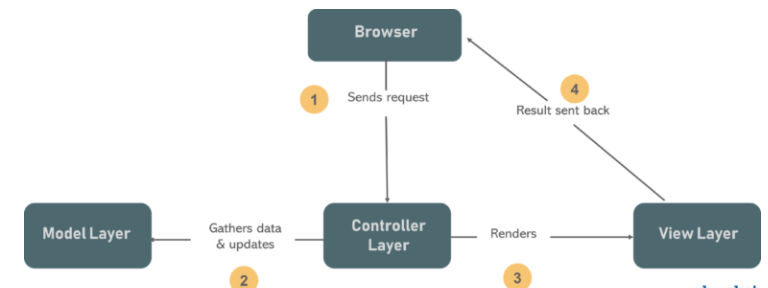
## Chosen System Architecture

## The chosen system architecture for this project is MVC. MVC is a software design pattern for developing a web application. MVT Structure consists of three parts namely Model, View and Controller.

## Model acts as the interface of the data. It is responsible for maintaining data. It is the logical data structure behind the entire application and is represented by a database (generally NoSQL database such as MongoDB, Redis).

## The View is the user interface what we see in our browser when we render a website. 3 It is represented by HTML/CSS/JavaScript/TypeScript). A template consists of static parts of the desired HTML output as well as some special syntax describing how dynamic content will be inserted.

## 

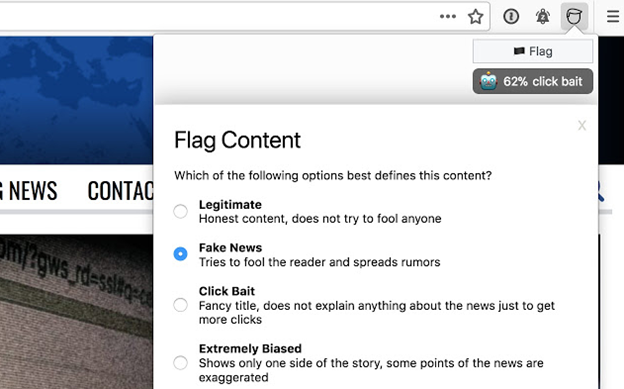


**Figure 1: Model View Controller (MVC)**

## System Interface Description

### User Interface

### The application will be web based, so the first screen of the application shows the input field where user will pass the raw input data into it and then it will click on the submit button and after that the raw data will be fed to model which will give the output and then resultant output is shown with help of radio button and there is feedback button for user to cross verify whether the predicted output and actual output (actual output in a sense that, we are assuming the user is rational which mostly giving true feedback which is based on some analysis) are matching or not, we will collect this data to generate some new insight, which help us to find performance measure of model.



**Figure 2: user interface**

### Software Interface

The software interface follows the Model-View-Controller (MVC) model for making and modeling data objects. The chosen operating system is Windows for its best support and user-friendliness. The interface must be managed via an external cloud instance, Application programming interface like AWS EC2 which controls AWS Sagemaker for our robust machine Learning models training and prediction. The application server has an ability to connect to an in-memory database or NoSQL database to store analytical results generated about the associate fake news sources from different sources. The interface is to be created using ReactJs as front-end library, ExpressJs/Flask for application server which is intern managed by our AWS EC2 instances in production environment we are trying to make a progressive web app, to provide faster production delivery leveraging the ability of cloud.

# 3. Detailed Description of Components

## Component 1: User Interface

|  |  |
| --- | --- |
| Responsibility | To access their respective application and its functionalities |
| Constraints | None |
| Composition | Internet access with updated browser version |
| Interactions | User can access application functionalities |

## Component 2: Web Crawler

|  |  |
| --- | --- |
| Responsibility | Generate raw data relevant to user interest and provide input to machine learning model. |
| Constraints | User must enter valid input |
| Composition | Internet, access to application with valid input |
| Interactions | Crawler and news sources/social media tweets and machine Learning Model |

## Component 3: Database

|  |  |
| --- | --- |
| Responsibility | To store the results of the model |
| Constraints | Enforce proper data type values |
| Composition | server-side scripting language such as node.js |
| Interactions | It interacts with the system for user data |

## Component 4: Machine Learning Model

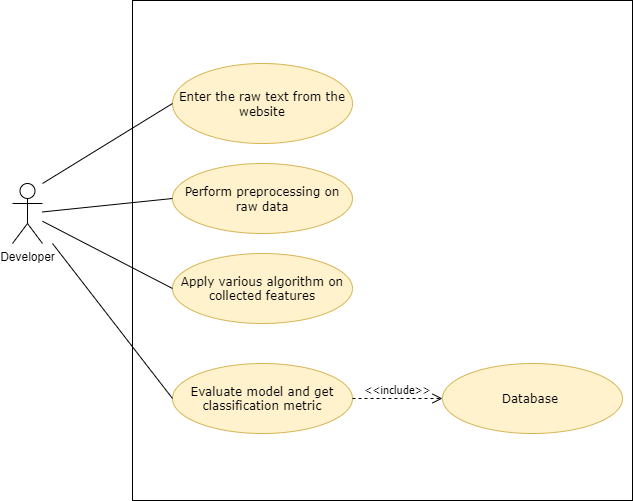
|  |  |
| --- | --- |
| Responsibility | To compute type of fake news and percentage being fake using NLP |
| Constraints | Input being preprocessed based on considered model input parameter |
| Composition | Machine Learning Languages like python, tensorflow |
| Interactions | It interacts with user for end result and crawler for raw input data |

# System Architecture

## Use Case Specification Using Template

## Following are the use case diagrams for our system that describe a set of actions (use cases) that the system should or can perform in collaboration with one or more external users of the system (actors).

## Use case diagram – 1



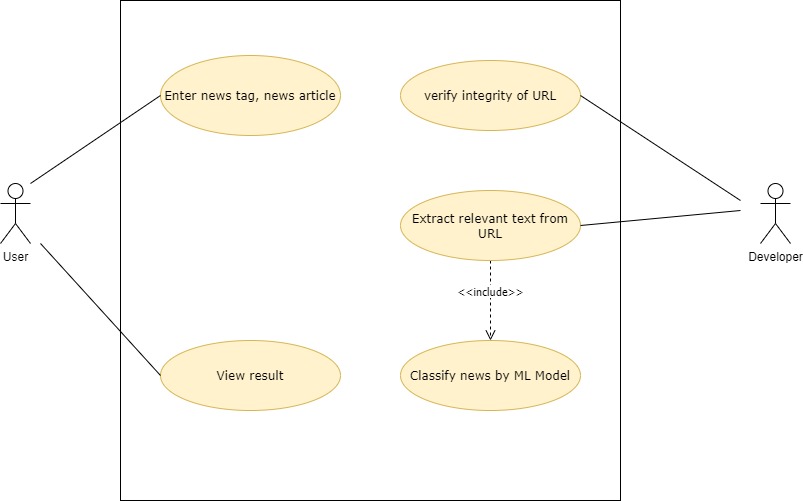
**Figure 5:** Developer prospective (Use Case Diagram)

## The classification System is the backbone of entire software. Figure 3.7-1 shows the use case related to classification system. The classification system extracts text from News URL and uses NLP to extract the required features. Then different machine learning algorithms are applied using the features and results are displayed to the user and stored in the database.

### Use Case Diagram 2

### User enters a News URL. System verifies the URL and extracts relevant text from the URL using a web crawler and then classified the news article as fake or credible using machine learning algorithms. After the result is computed the user can view the result.

### 

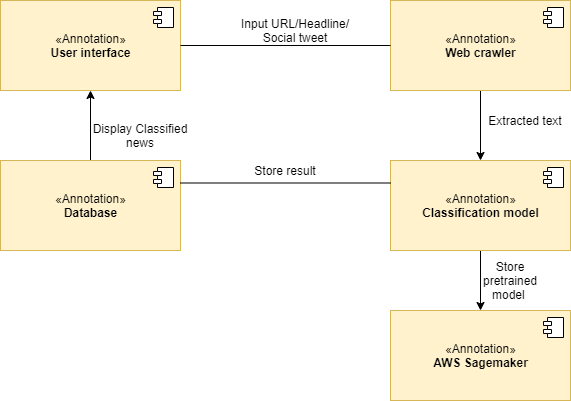


**Figure 6:** User-Developer prospective (Use Case Diagram)

**4.1.3 Component diagram**

UML Diagrams Following are the Unified Modelling Language (UML) diagrams that are intended to provide a standard way to visualize the design of our system.

Following is the component diagram, and describes the components to make the functionalities of the system



**Figure 7 : Component diagram**

The Figure 7 shows the overall view of the system showing all the different components and information that flows between these different components. User Interface is the view available to the user through which user interacts with the system. Our user interface will be web application. The User will input a News URL which will then be passed to the web crawler. Web crawler will crawl the URL and extract relevant text and pass it to the Classification System. Classification System will then extract the required features from the text and apply machine learning algorithms on the feature vector and will store the results in the Database.

# Data Flow Specification

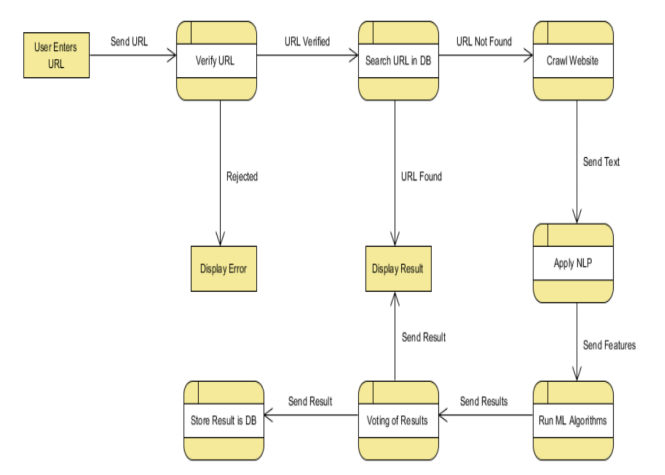
## Level 0 DFD

## 

**Figure 8:** DFD Level 0

Level-0 DFD is also known as a context diagram. It’s designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as a single bubble with input and output data indicated by incoming/outgoing arrows.

## Level 1 DFD



**Figure 9:** DFD Level 1

**Figure 9:** shows the flow the flow of data. First User sends the URL, error is displayed if entered text isn’t in URL format and else the URL is searched in Database in ‘Already Classified List’. If URL is found, it just displays the previous result, else the crawler crawls the website and scraps the relevant text. NLP is applied on text and features from NLP are processed by ML algorithms. Each Algorithm gives result, all the results are sent to Voting algorithm, and the result is displayed, and stored in Database.