

Title

FYP

15%

SIMILARITY INDEX

13%

ACADEMIC

2%

INTERNET

| | |
|-------------------------|---------------------------------|
| Date: | 2022-01-31 06:56:48(+00:00 UTC) |
| Report ID: | 61f78838134e2a4a9 |
| Word count: | 2381 |
| Character count: | 11769 |

Similar sources

| | | |
|----------|--|-------------|
| 1 | <ul style="list-style-type: none"> ● Requirements Engineering Academic | 0.9% |
| 2 | <ul style="list-style-type: none"> ● What is the difference between stemming and lemmatization? ● https://blog.bitext.com/what-is-the-difference-between-stemming-and-lemmatization/ Internet | 0.7% |
| 3 | <ul style="list-style-type: none"> ● System Modelling for Requirements Engineering ● Requirements Engineering Others | 0.6% |
| 4 | <ul style="list-style-type: none"> ● Generic User Interface Development ● Mobile Computing Principles,2004 Others | 0.6% |
| 5 | <ul style="list-style-type: none"> ● Using the Text-to-Speech API with C# Qwiklabs ● https://www.qwiklabs.com/focuses/2178?locale=it&parent=catalog Internet | 0.6% |
| 6 | <ul style="list-style-type: none"> ● Using the Text-to-Speech API with C# - Google Codelabs ● https://codelabs.developers.google.com/codelabs/cloud-text-speech-csharp Internet | 0.6% |
| 7 | <ul style="list-style-type: none"> ● Using the Text-to-Speech API with C# Google Cloud Skills ... ● https://www.cloudskillsboost.google/catalog_lab/1180 Internet | 0.6% |
| 8 | <ul style="list-style-type: none"> ● TWEETS AND TRUTH ● Alfred Hermida ● Journalism Practice,2012 Academic | 0.5% |
| 9 | <ul style="list-style-type: none"> ● Requirement checklist for blog in web application ● Karan Gupta,Anita Goel ● International Journal of System Assurance Engineering and Management,2012 Academic | 0.5% |

1. Software design and modeling

Software designing is basically the behavioral and structural overview of project which is used for better understanding of a project. For software modelling we have used the UML language in order to describe the software design explicitly.

Software modeling is a vital part in software development process. For our project we used the Kanban development process for managing and improving the work flow.

The activity diagram for this project is shown in figure 5.1. The diagram represents the whole flow of data and information between different components.

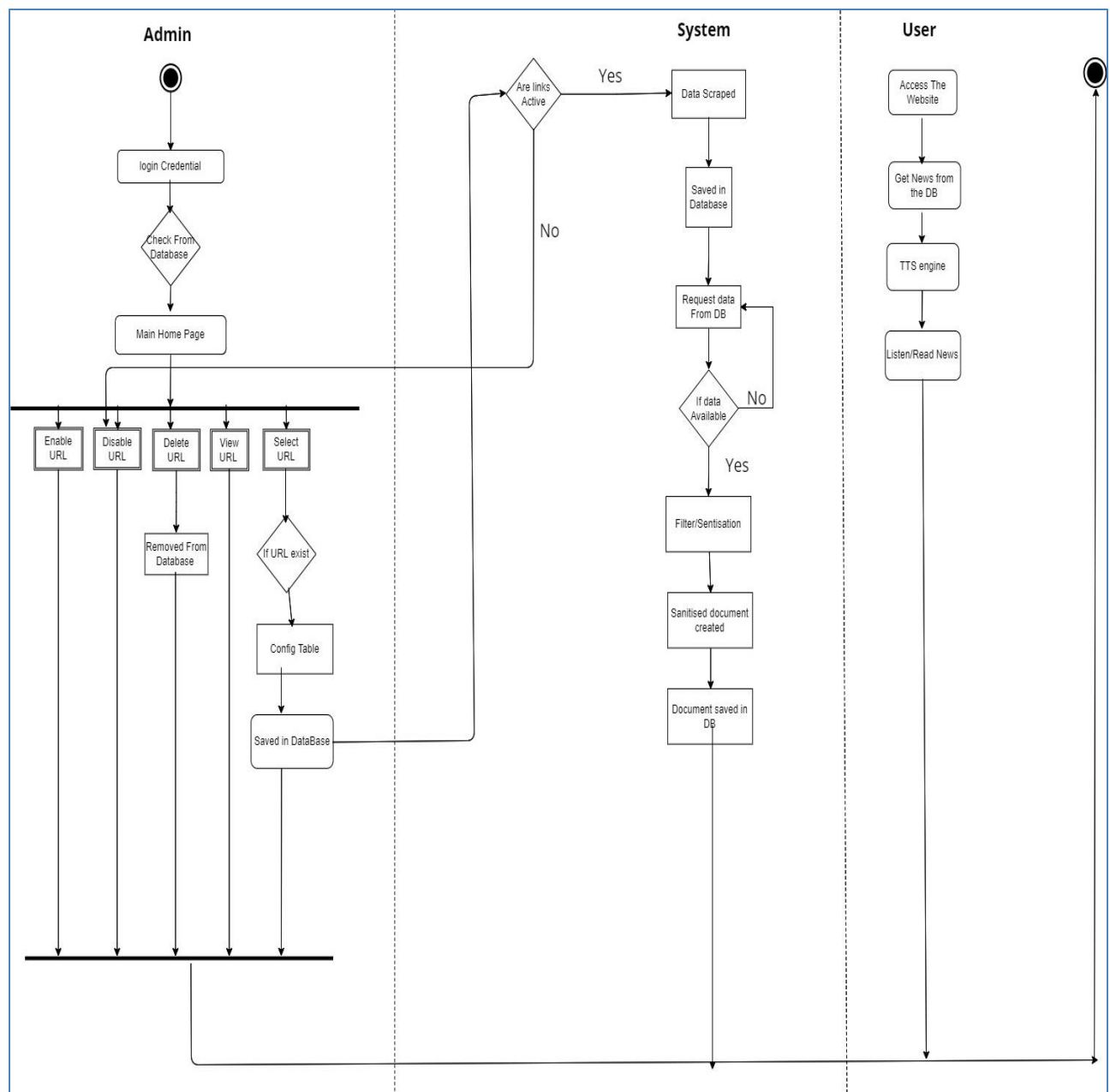


Figure 5.1: Activity Diagram

1.1 UML Diagrams

UML diagrams basically are used to visualize the project, its different processes and states at the time of development of the project. Below we have used different UML diagrams to envision the different fragments of our project.

1.1.1 Class Diagram

Class diagrams are a useful way to model the whole system and the objects present in that system, along with the attributes and methods of each object. **Figure 5.2 here represents the class diagram of our system.** It consists of 11 different classes and the relationship (such as generalization, association, specialization etc.) between each of these classes

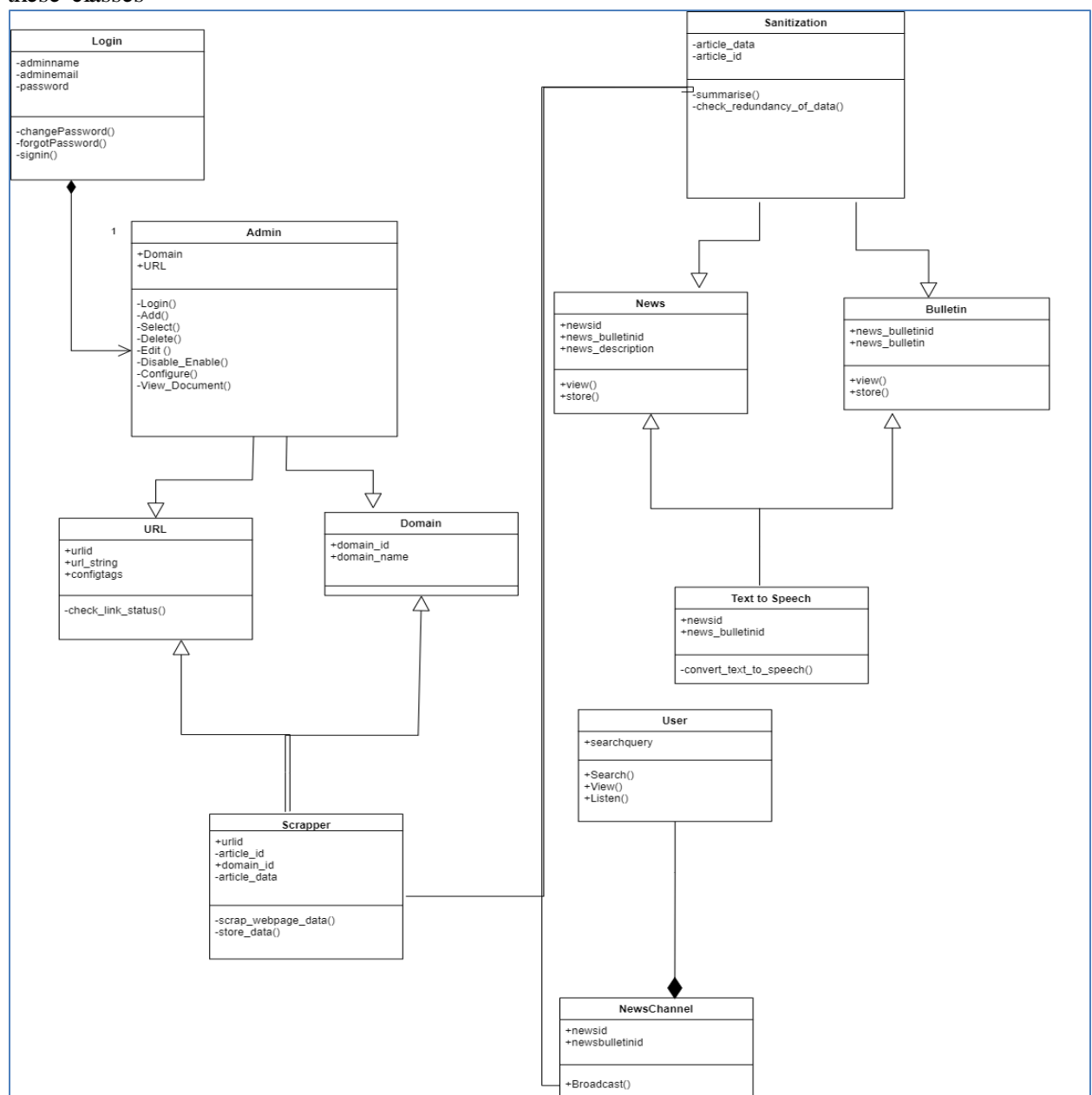


Figure 5.2: Class Diagram

1.1.2 Object Diagram:

Object diagrams are used to display the whole working of the project at a particular instant, with the help of an example. Figure 5.3 here represents the object diagram of our system.

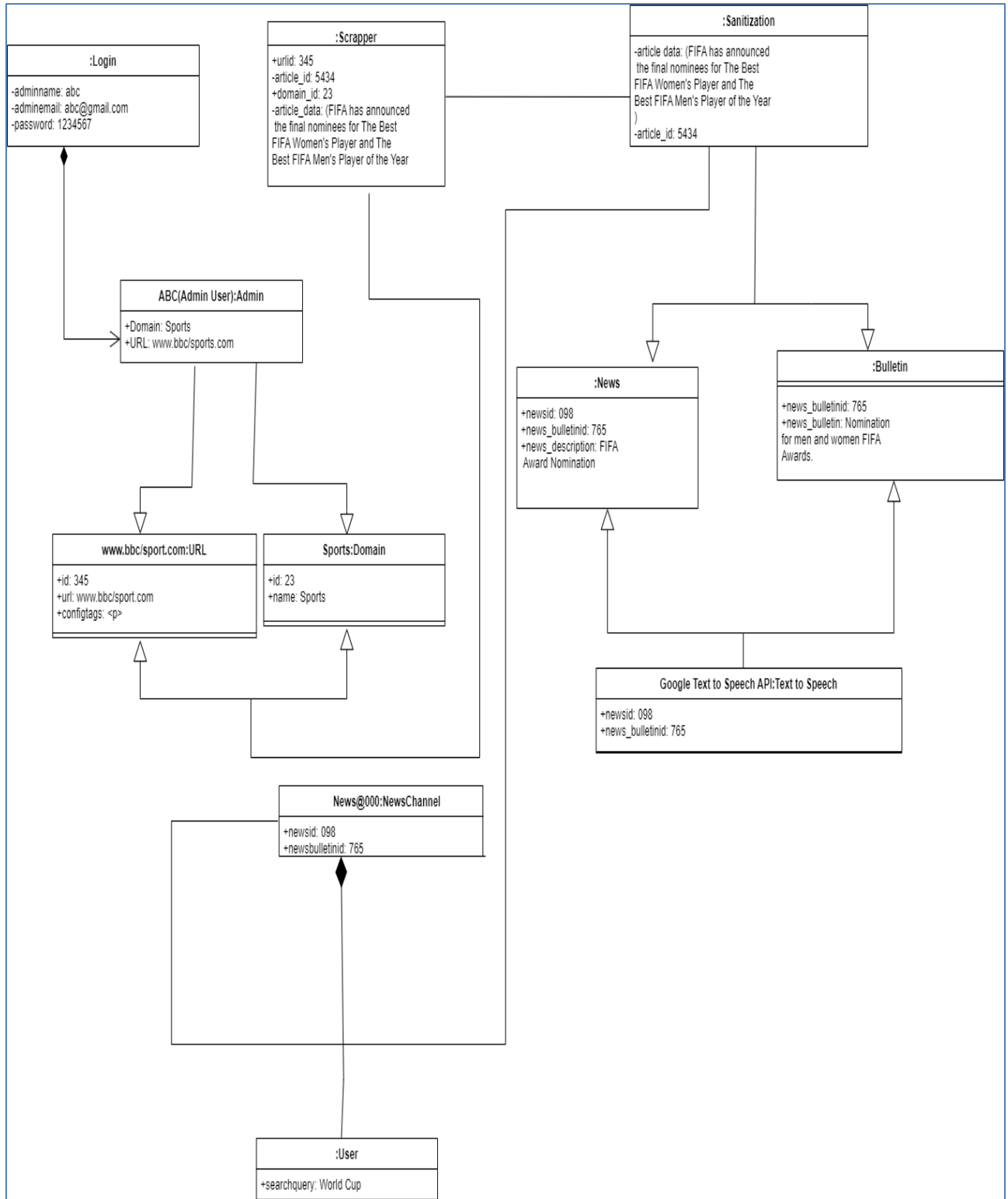


Figure 5.3: Object diagram

1.1.3 Sequence Diagram:

Sequence diagram represents the interaction of the user with the system keeping in view the time constraint. Figure 5.4 here represents the sequence diagram of our system. It consists of 2 actors (the user and the admin) and 3 processes in between (the system, server and database) and shows how each of the users interacts with each of the system process to complete their task.

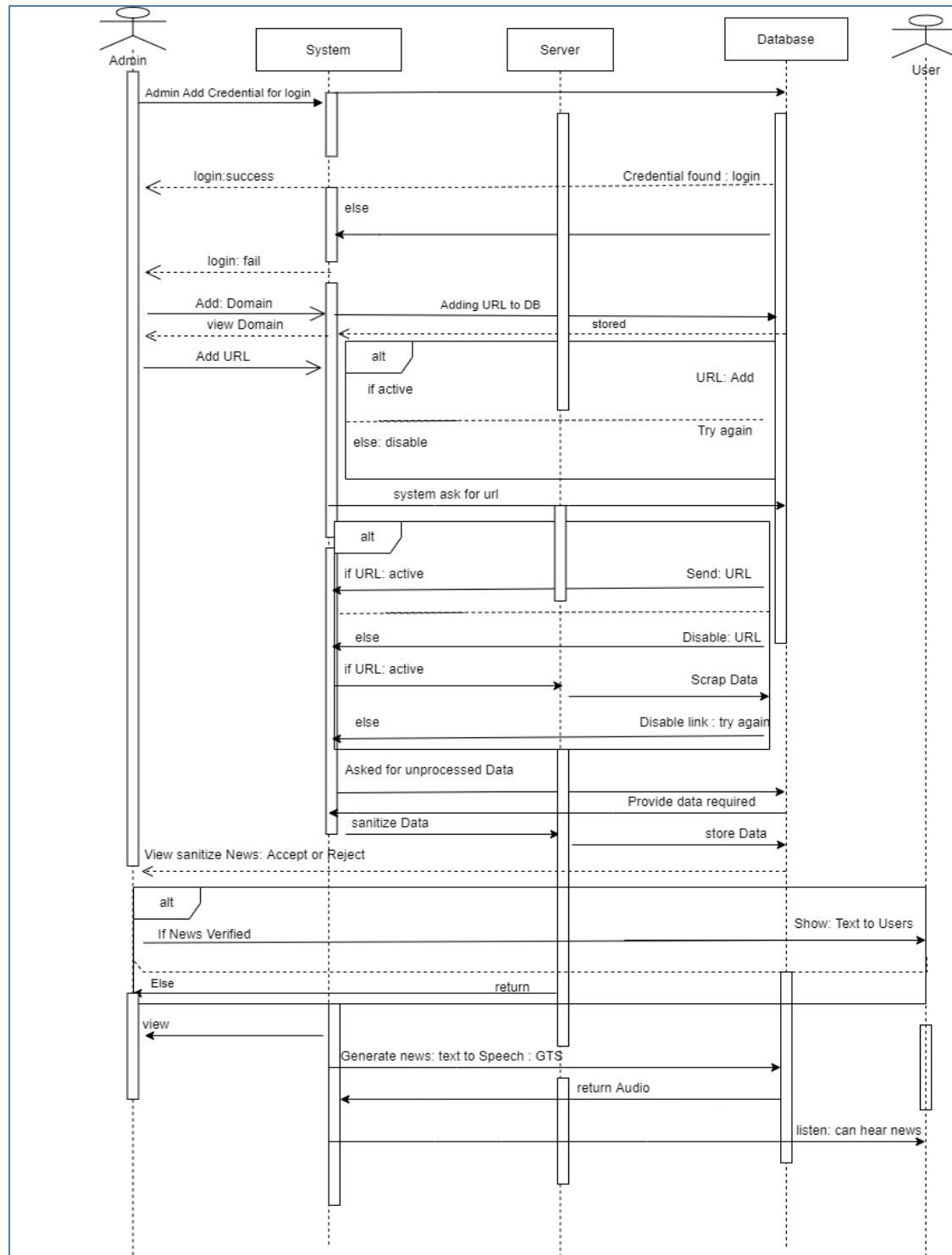


Figure 5.4: Sequence Diagram

1.1.4 Use case diagram:

Use case diagrams are used to define the interactions between the actors (i.e. the users) and the system. Figure 5.5 represents the use case diagram for our system. It consists of 3 actors (the admin, system and the user) and the different interaction processes with which each of these actors interact with.



Figure 5.5: Use Case Diagram

1.1.5 Entity Relationship diagram (ERD):

54%
An ERD diagram depicts the different entities present in the system and the relationship between each of those entities. Figure 5.6 below represents the entity diagram of our system.

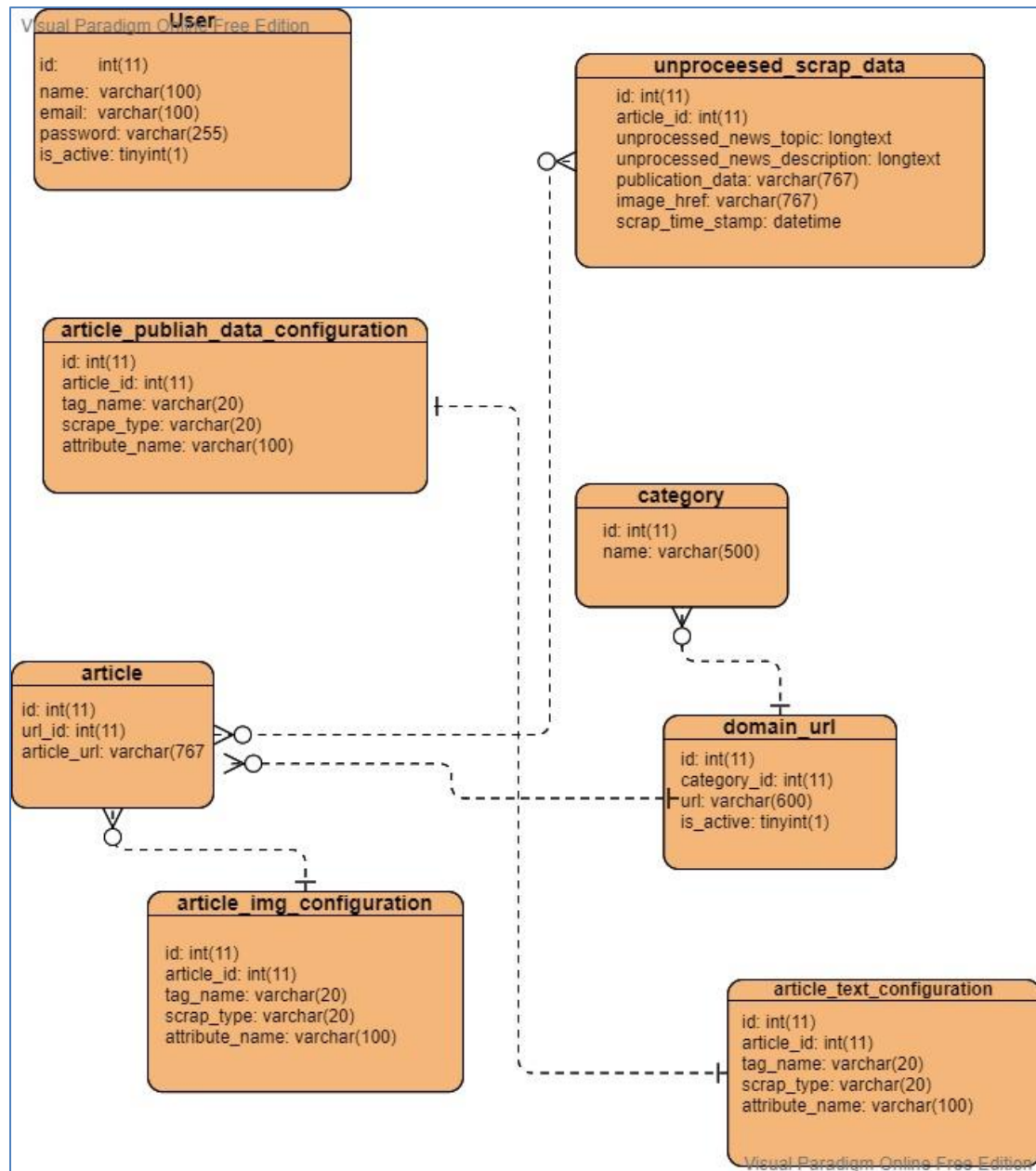


Figure 5.6: ERD Diagram

1.2 FrontEnd

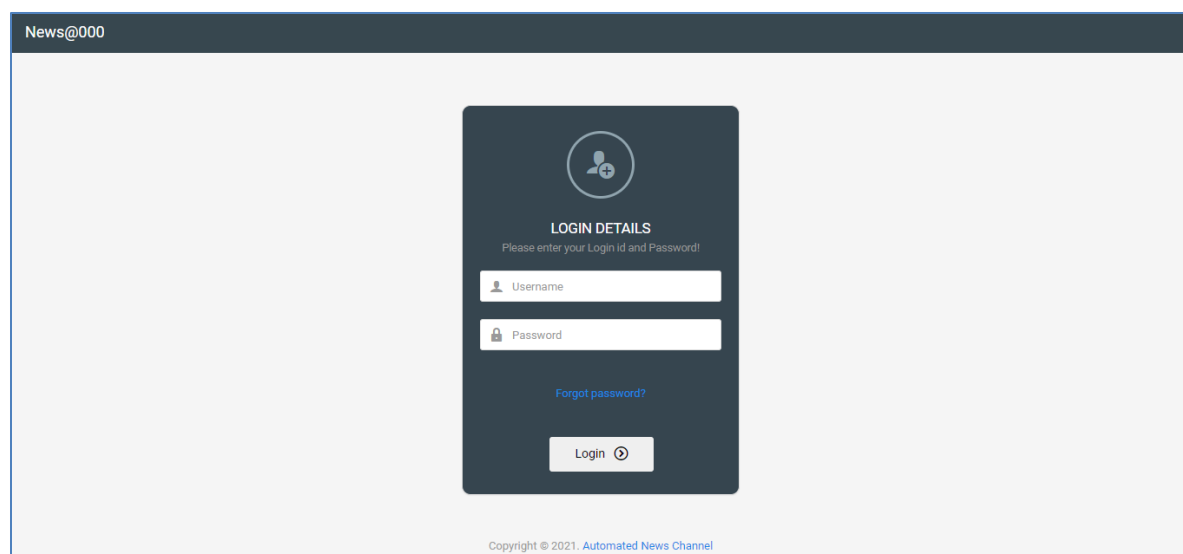


Figure 5.7: Admin login page

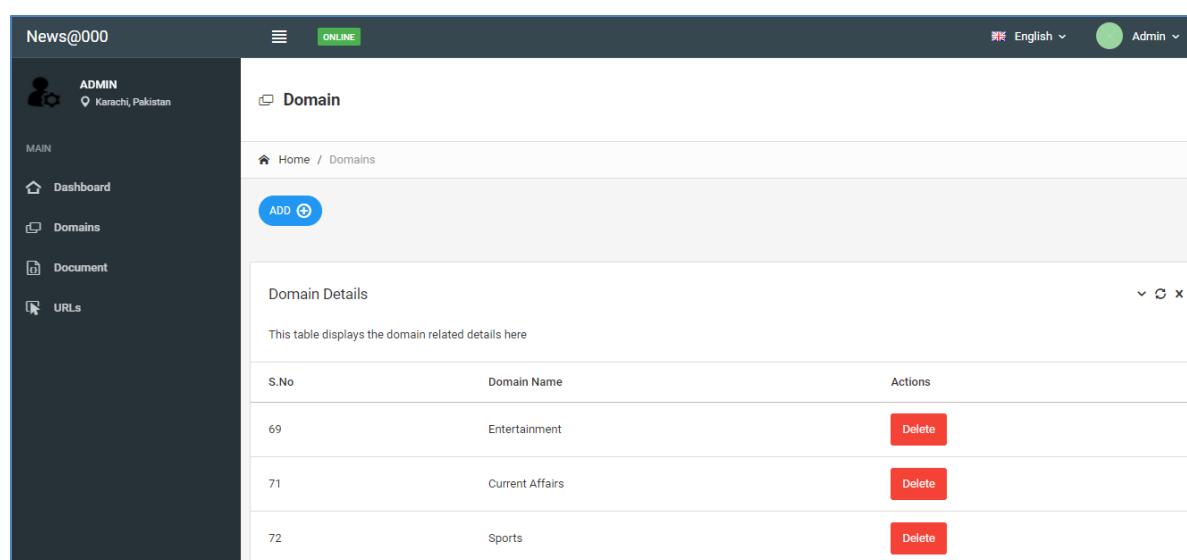


Figure 5.8: Domain information page

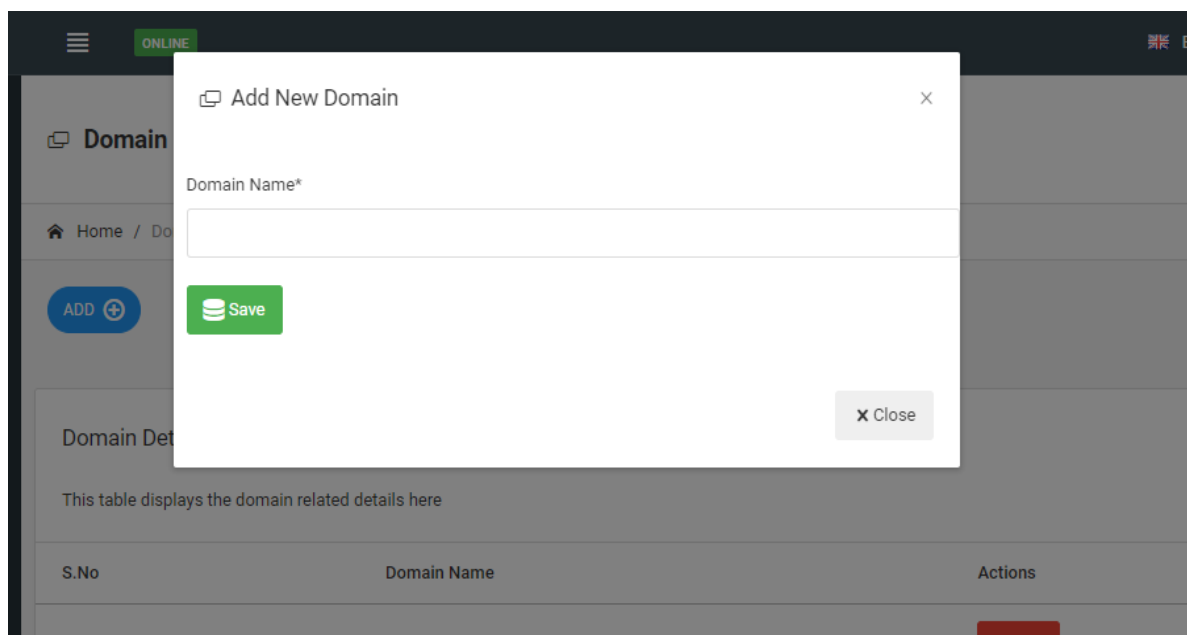


Figure 5.9: Add new domain interface

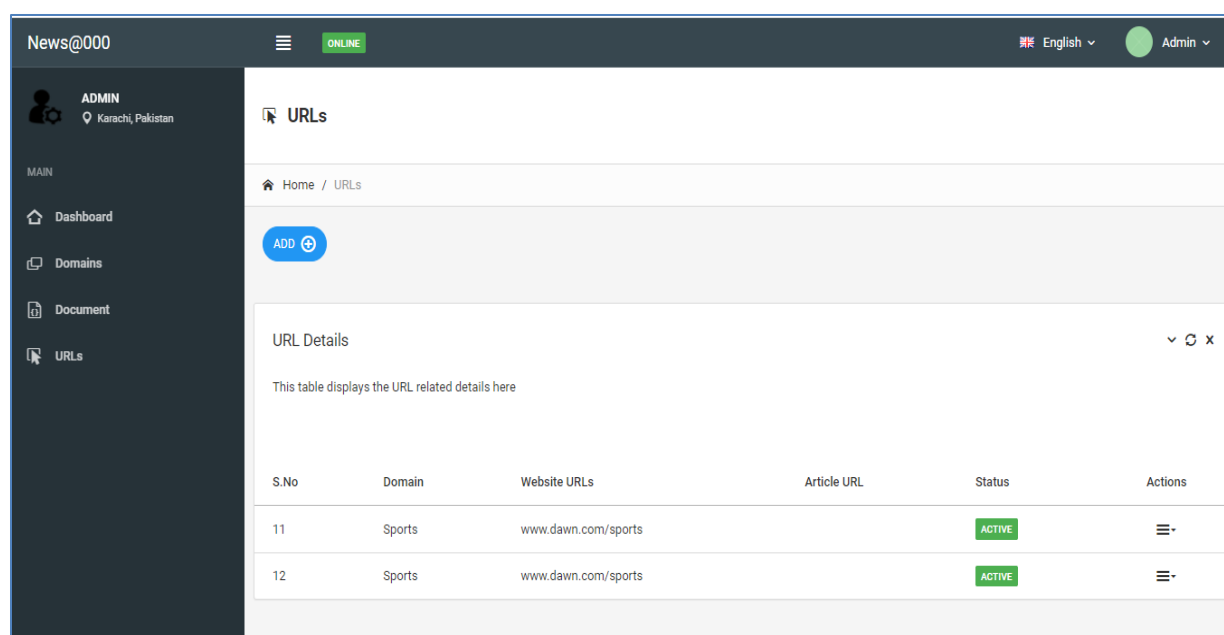


Figure 5.10: URLs info page

News@000 ONLINE English Admin

ADMIN
Karachi, Pakistan

MAIN

- Dashboard
- Domains
- Document
- URLs

Document

Home / Documents

Sanitised News will be displayed here

Edit Disable

| S.No | Domain Name | News | Status |
|------|-------------|--------------------------|----------|
| 1 | Sports | news displayed here..... | ACTIVE |
| 2 | Weather | news displayed here..... | DISABLED |
| 3 | sports | news displayed here..... | DISABLED |

Figure 5.11: Distilled news page

2. Algorithm analysis and complexity

^{59%} This section discusses the algorithms we used during the development of this project. It discusses each algorithm separately and the time and space complexity of the system in general.

2.1 Scrapper Algorithms:

There are many algorithms for web scrapping but we are using following algorithms which are best suited. Here we are using combination of BeautifulSoup 4 and Selenium to reduce the data scrapping time.

2.1.1 BeautifulSoup4:

It is a Python library that is use for creating parse tree of parsed pages which is used to pull out the data from HTML and XML files, which is beneficial for web scrapping.

2.1.2 Selenium:

Selenium is also a ^{64%}python library which is used for web scrapping. It is a tool for testing web applications. Selenium is an ideal library for performing scrapping with the browser automation function

For most of the scrapper function we used BS4, however BeautifulSoup4 was unable to scrap the data from those web pages where JavaScript is disable so in contrast with this we used Selenium which scrapped the data in the browser itself by calling its module i.e. web driver which opens the web page in the browser and enables the JavaScript of the web page (though this process is time consuming compared to web scraping by BeautifulSoup4). Therefore, we are using combination of Beautiful Soup 4 and Selenium to reduce the data scrapping time, where Selenium is used to open the web page in the browser for articles where JavaScript was disabled and then we use BeautifulSoup4 to scrap the data from it.

2.1.2.1.1

2.1.3 Working of algorithm:

^{70%} The steps of the parser algorithm are as follows:

- ➔ Our system right now caters for three domains (i.e. sports, technology and entertainment) so the admin can select any of that domain
- ➔ The admin can give different url links, to scrape data, w.r.t above mentioned domains (the format of the link should be "channel_name/domain_name" for example dawn.com/sports).
- ➔ ^{51%} These URLs would be saved in the URLs table in database
- ➔ The parser would then parse all the articles from the given link.
- ➔ The parser ^{58%}would work in such a way that it would save all the article urls in the database. Then parse each of these articles one by one.
- ➔ Each article/article url would be assigned a unique id so when the data is once again scrapped from the links after a set time period (such as 4 hours) we can

compare the new articles with the previously scrapped articles present and see if any of the new articles scrapped are the repeated ones.

- The above mentioned points would be repeated for all of the url links. Data would be gathered in this way and saved in database.
- Next we now need to start building config table
- For this the admin the particular url that he wants to configure
- And then select the desired tag and the related class or id method in that.
- These values and the related data would then be saved in database.

2.2 Summarization and Filtration Algorithms:

After getting the new data by through parser, the system now needs to summarize and filter the data. For this purpose, we are using transformers which are tuned for the training and testing of data. Transformers basically are used for changing one sequence into another with the aid of encoder and decoder. Here we are using BERT as an encoder and GPT-3 as a decoder.

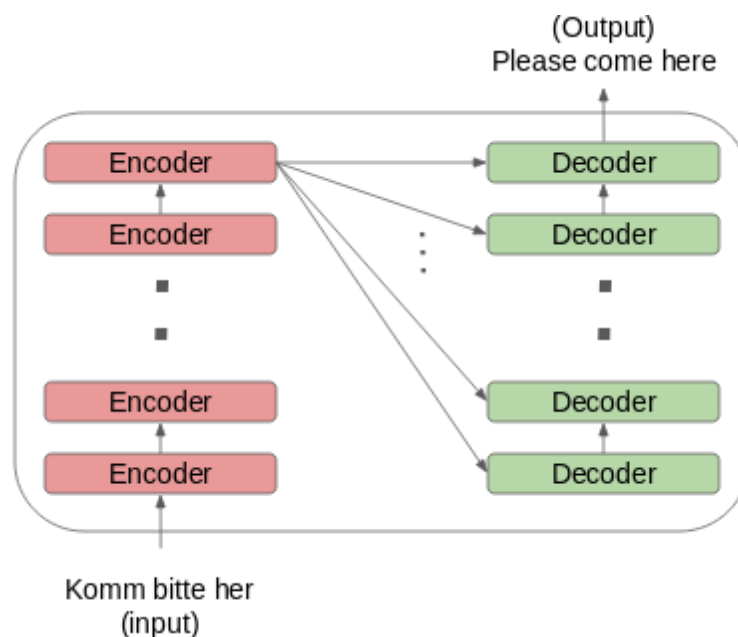


Figure 6.1: A representation of transformer.

2.2.1 BERT:

BERT an open source library which stands for Bidirectional Encoder Representations from Transformers works as an encoder in the transformer. It is a machine learning framework for NLP processing. BERT is a program that uses surrounding text to help computers grasp the meaning of ambiguous words in text. The BERT framework was trained by means of Wikipedia text and can be fine-tuned using question and answer datasets.

The working of BERT is such that every output element is connected to the other input element and the weightings between them are dynamically calculated based on the connection between them.

2.2.2 GPT-3:

GPT-3 stands for Generative Pre-Trained Transformer, it is a unidirectional transformer which takes few demonstrations to comprehend tasks and carry them out. It is built with 175 billion parameters which is 100 times more than GPT-2 transformer. It has abilities like writing articles which are difficult to recognize that whether the article is written by a human or by a computer.

2.2.3 Working of algorithm:

The news summarization domain consists of 2 parts. The first is the filtration part, the second is the news summarization/aggregation part. Below we have described the steps for both of them:

Filtration of data consists of the following steps:

- Convert Text to lower case
- Remove Special Characters/Unwanted Characters
- Correction of any typos using libraries such as TextBlob, Pyspell Checker, autocorrect Library
- Normalization i.e. finding similar words with different forms of tense (such as running, ran into run). It includes 2 steps
 - Stemming
 - Lemmatization

Stemming uses the stem of the word while lemmatization uses the context in which the word is being used.
- Tagging which includes finding parts of speech such as noun, pronoun, verb etc.
- Applying Chunking & Chinking

The process for summarizing the data includes following steps:

- Perform Cleansing of Data using the steps of filtration of data.
- The cleanse data would be sent to machine learning/deep learning model → (i.e. Sending data to Transformer)
- Tune transformer according to the requirement
- Consist of 2 phases training and testing. In the training phase we will train the model to generate a summary from a given news article using a dataset of 7000-8000 articles. The testing phase would then consist of our own system producing summarized weather news reports.

2.3 **Text-to-Speech Algorithm:**

62%

Text to Speech (TTS in short) is used to convert a desired text into a speech form. With more advancement in technology TTS engines became a necessary component to take inputs and give outputs in speech form as opposed to the previous computers which used only the written input from keyboards or typewriters.

2.3.1 **GTTS (Google Text-To-Speech):**



65%

The Google Cloud Text-to-Speech API (Beta) allows developers to embed naturally-sounding synthetic human voice into their applications as playable audio. The Text-to-Speech API converts text input to audio data such as MP3 and LINEAR16.

***Note: The study of algorithm is in progress and is subject to further changes.

3. Achievements

According to given POCS we have achieved following points:

-  System is flexible to cater multiple types of news
- Fetching of news are configurable
- System working with three domains
-  Any user can fetch data by using the APIs
- Fetch history would be maintained for viewing

4. Appendices

66% In this chapter a detailed overview of the project is provided.

4.1 **Appendix A: Project Executive Summary**

The project “Automated New Reporting Channel”, this project is to automate the process of news reporting channel by creating a web app which would aggregate the news from different news reporting websites (by scrapping data from it), filter & summarize that data and then present it. For scraping of data, we use combination of libraries such as Beautiful-Soup and Selenium and for speech recognition; we will be using Google API.

4.2 **Appendix B: Project Overview**

This project is an Automated News Channel which has following functionalities:

- **50%** There would be an admin at the backend who can Log-in to the system.
- The admin would then be able to select the list of URLs/News Channel Websites, feed them into the database.
- On those websites web scrapping technique would be applied to gather the required data related to news content. This technique would be applied after every set time interval period to refresh the new content.
- Once the data is scrapped it would be filtered, summarized, checked for redundancy and a bulletin is made out of it.
- **51%** The bulletin would be converted from written to oral form using text to speech
- The news will be available for viewing and hearing in bulletin forms
- It would then be presented by any animated broadcaster from our website

4.3 **Appendix C: Project Objectives**

60% News presented in a creative way to the user in the form text as well as audio.

-Search option available to the user to search desired news

-Animated character will present the news

4.4 **Appendix D: Project Scope**

^{53%} The scope of this project includes and excludes the following items.

4.4.1 **In Scope:**

- ^{62%} The development of an automated news channel to produce news.
- ^{62%} This System would be based on different domains such as technology, sport etc.
- The development of an automated news channel to produce news.
- This System would be based on different domains such as technology, sport etc.
- NLP based training of the system so that the system could produce accurate results.
- ^{52%} Testing of the system before making it accessible for the public.
- User can search any news by giving related keyword.

4.4.2 **Out Scope:**

- Web security.
- Power Failure.
- Internet Failure.
- Further enhancement of this system in other news related fields (such as current affairs)
- Enhancement of system such that present the news with relevant pictures.
- NLP based training of the system so that the system could produce accurate results.
- ^{52%} Testing of the system before making it accessible for the public.
- User can search any news by giving related keyword.

4.5 **Appendix E: Deliverables Produced**

- Project Deliverable 1: Our Project Deliverable will be a Web Application which will be access by users by searching the link
- Project Deliverable 2: It will present the news on the front home page which will be deliver by animated character.
- ^{51%} Project Deliverable 3: Search option available for the user to find desired news by given keyword.
- ^{54%} Project Deliverable 4: News available in three different domains (technology, sports, etc.)

4.6 **Appendix F: Project Estimated Effort/Cost/Duration**

4.6.1 Estimated Effort Hours:

We will put in a total of 4 hours per day for each member

4.6.2 Estimated duration:

| <i>Milestone</i> | <i>Date completed</i> | <i>Deliverable(s) completed</i> |
|-------------------------|-----------------------|--|
| <i>Project planning</i> | <i>15/11/21</i> | <ul style="list-style-type: none"> • <i>Project definition</i> • <i>Work plan.</i> |
| <i>Milestone 1</i> | <i>3/12/21</i> | <ul style="list-style-type: none"> • <i>Actor use cases</i> • <i>Block diagram</i> • <i>Component diagram</i> |
| <i>Milestone 2</i> | <i>28/1/22</i> | <ul style="list-style-type: none"> • <i>Development of Front-end</i> • <i>Admin Backend</i> • <i>Database Connection</i> • <i>Sequence diagram</i> • <i>Actor diagram</i> • <i>Use case Diagram</i> • <i>Object diagram</i> |

4.7 Appendix G: Project Assumptions

In order to identify and estimate the required tasks and timing for the project, certain assumptions and premises need to be made. Based on the current knowledge today, the project assumptions are listed below. If an assumption is invalidated later, then the activities and estimates in the project plan should be adjusted accordingly.

- **Assumption #1:** User can search desired news using keyword in search bar
- **Assumption #2:** User can search in three domains.
- **Assumption #3:** User can hear audio.
- **Assumption #4:** Admin will add link
- **Assumption #5:** Admin will configure the news according to the domain

4.8 Appendix I: Project Approach

- 1) Use NLP (Natural Language Processing) for speech recognition
- 2) Use HTML, CSS BOOTSTRAP, JAVASCRIPT for front-end.
And for backend we will use Python.
- 3) Database for storage of data

4.9 *Appendix: J: Tools and Technologies*

We used Visual Studio (2019) as IDE along with git and Github for code integration.