Final Year Project

News Analytics as a Service

Iteration 2 Report

FYP Team

Mehmood Amjad 19I-0472

Muhammad 19I-0561

Danyal Faheem 19I-2014

Supervised by

Dr. Muhammad Arshad Islam

FAST School of Computing

National University of Computer and Emerging Sciences

Islamabad, Pakistan

2022

Students' Submission

This includes the title of the report and its occasion.

Anti-Plagiarism Declaration

This is to declare that the above FYP report produced under the:

Title: **News Analytics as a Service**

is the sole contribution of the author(s) and no part hereof has been reproduced on as it is basis (cut and paste) which can be considered as Plagiarism. All referenced parts have been used to argue the idea and have been cited properly. I/We will be responsible and liable for any consequence if violation of this declaration is determined.

Date: 25-12-2022

Student 1

Name: Mehmood Amjad

Signature:

Student 2

Name: Muhammad

Signature:

Student 3

Name: Danyal Faheem

Signature:

Supervisor (Faculty)

Name: Dr. Muhammad Arshad Islam

Signature:____________________

Authors' Declaration

This states Authors' declaration that the work presented in the report is their own, and has not been submitted/presented previously to any other institution or organization.

Abstract

News Analytics as a Service is an automated platform that displays NEWS plotted on a map of Pakistan that can be filtered based on time frame, location and key words. This is achieved by scrapping the NEWS and then extracting the focus time and focus location out of the NEWS documents and then plotting them according to the focus location.

Executive Summary

Whenever a person wants to know about a certain event that occurred over a period of time, he would have to consult every NEWS article that was published in that period of time. However, it has been noticed that the publishing time of the NEWS rarely ever corresponds to the focus time of the NEWS i.e, the time that the NEWS is actually referring to. Similarly, if someone wants that data of the NEWS over a span of time based on a specific location, the work becomes more tedious and it has also been noticed that the headline of the NEWS sometimes does not correspond to the actual focus location of the NEWS i.e, the location that the NEWS is actually referring to.

NAaaS is therefore a platform which automatically scraps the NEWS off of online sources and extracts the focus time and focus location out of the NEWS documents. Then, it provides a map interface which plots the NEWS based on a map of Pakistan based on their focus location. It even allows users to filter the NEWS based on the timeframe, location as well as key words so that their tedious work is now done by the computer.

The project is divided into 5 major components:

1. Data Gathering
2. Data Streaming
3. Temporal extraction
4. Spatial extraction
5. Geographical information system

In the first iteration of our project, we must set up an intermediate structure of our system which consists of a scrapper that fetches textual data from targeted sources and after gathering, we had to set up data pipe lining which led to the use of big data tool Kafka. And in the end we send the continuously fetched data to a cluster (having only one node for now). Which takes the stream of data and applies Temporal and Spatial extraction. Our implementation has to find out the focus time and focus location of the stream getting from the producer side of the kafka (scrapper). We must have to justify how our models are working and try to optimize the given resources and extract the correct information.

In the second iteration of our project, our main focus was setting up a cluster of Apache Spark. This will help us to process a huge amount of news in a short span of time. We used the PySpark module of python and docker containers to set up an active connection of Apache Spark with a single master node and multiple slave nodes. Furthermore, We configured the database for storage of our data. To improve the accuracy in the extraction of focus location, we used fuzzy matching. We also focused on improving the extraction of focus time to make the results more accurate.

Table of Contents

Authors' Declaration **3**

Abstract **4**

Executive Summary **5**

1. Introduction **10**

2. Project Vision **11**

2.1. Problem Statement 11

2.2. Business Opportunity 11

2.3. Objectives 11

2.4. Project Scope 13

2.5. Constraints 13

2.6. Stakeholders 14

2.7. Target audience 14

3. Software Requirement Specifications **14**

3.1. List of Features 14

3.2. Functional Requirements 15

3.3. Quality Attributes 15

3.4. Non-Functional Requirements 16

4. Iteration Plan **16**

5. Iteration 1 **17**

5.1 Use Cases 17

5.1.1 Use Case Diagram 17

5.1.2 High Level Use Cases 18

5.1.2.1 Filter map based on keywords 18

5.1.2.2 Filter map based on location 18

5.1.2.3 Filter map based on timeframe 18

5.1.3 Expanded Use Cases 19

5.1.3.1 Filter map based on keywords 19

5.1.3.2 Filter map based on location 19

5.1.3.3 Filter map based on timeframe 20

5.2 Operational Contracts 21

5.2.1 Filter map based on keywords 21

5.2.2 Filter map based on location 21

5.2.3 Filter map based on timeframe 22

5.3 System Sequence Diagrams 22
5.3.1 Filter map based on keywords 22
5.3.2 Filter map based on timeframe 23
5.3.3 Filter map based on location 23
5.3.4 Filter map based on timeframe, keywords, location 24
5.4 Domain Model 25
5.5 Sequence Diagrams 26
5.5.1 Filter map based on timeframe 26
5.5.2 Filter map based on keywords 26
5.5.3 Filter map based on location 27
5.5.4 Scrap, produce and consume NEWS 27
5.5.5 Filter map based on timeframe, keywords, location 28
5.6 Entity Relationship Diagram 29
5.7 Relational Schema Diagram 30
5.8 Class Diagram 31
5.9 Architecture diagram 32
6. Iteration 2 33
6.1 Use Cases, System Sequence Diagrams, Sequence Diagrams 33
6.2 Architecture Diagram 33
6.3 Data Flow Diagrams 34
6.3.1 Level 0 34
6.3.2 Level 1 34
6.3.3 Level 2 35
6.4 State Machine Diagram 36
6.5 Entity Relationship Diagram 37
6.6 Relational Schema Diagram 38
6.7 Class Diagram 39
6.8 Interface 40
6.8.1 Dummy interface with Geographical Information System 40
7. Implementation Details 40
7.1 Iteration 1 40
7.1.1 Scrapper Module 40
7.1.2 Geographical Specificity Module 42
7.1.3 Temporal Specificity Module 43
7.1.4 Kafka Producer-Consumer Module 46
7.2 Iteration 2 46
7.2.1 Temporal Specificity Module 46
7.2.2 Geographical Specificity Module 48

7.2.3 Database handling 49
7.2.4 Apache Spark distributed processing 50
7.2.5 Application Programming Interface 50
References **51**

1. Introduction

According to human psychology, anything that has a pictorial representation has far more impactful meaning compared to the bulk of data to be read. Relevant information extraction from the text is a hot domain these days, but the major problem is the amount of data. In the case of NEWS articles, there is tons of content published every second. The data is getting big. Extracting information from a text and storing it somewhere and then retrieving it when it's required becomes difficult every day due to the exponential growth in data.

Imagine someone wants to know about a specific event occurred on a specific time, He has to go on some NEWS website and manually check whether the event occurred is important for him/her or not and if he/she wants to get the information about one specific event/topic but over a time frame, things get more clumsy. E.g X wants to know about the number of robberies in city Y, area B reported in the NEWS from January 2021 to January 2022. If such information is not collected by any third-party organization. He himself has to go through the NEWS and check the robbery keyword and make sure the location is city Y, area B. X has to work quite hard to get this data for 1 year, But what about 2 years ? Or not just robbery, any other key-word/topic he might be interested in

NAaaS provides a one-stop solution to many of the problems related to the NEWS articles. It provides you quick access to all the NEWS regarding the selected event/topic of NEWS in the provided time frame and plots it on the Map, which provides a better understanding of NEWS relevance to the location and time. To deal with the big-data NAaaS internal structure is to be distributed over the cluster of computers, allowing distributed computing, help increase the response time, query execution of the user and extraction of data from different sources with stream processing techniques using big data tools. For the front-end, NAaaS has a GIS geographic information system, allowing people to get a more clear image of NEWS relevant to the location with granularity up to union councils (In Pakistan).

2. Project Vision

2.1. Problem Statement

News documents consist of a focus time (time/date the NEWS is referring to) and a focus location (location to which the NEWS is referring to) as well as a creation time (time/date the NEWS was created/uploaded at). Similarly, NEWS might have a focus location as well.

However, it has been observed that the focus time and the focus location usually do not relate to the creation time of the document nor the Area mentioned in the headline of the document. Therefore, it is usually left up to the reader to interpret the temporal specificity and the geographical or spatial specificity of the document by themselves.

Therefore, we aim to specify the focus location as well as focus time of a NEWS document. Furthermore, we want to plot the NEWS on a map of Pakistan by placing markers for each NEWS

2.2. Business Opportunity

Our System is fully automated and gets NEWS from different sources automatically and then plots on the map. All the NEWS channels can buy our service to give ease to their audience. Also help people for research purposes and gather information quickly and efficiently. It will also help government sector, identifying hotspot of criminal activities, better way to represent election results and much more

We can also extend this idea, specific to a domain e.g by changing the data-set of countries with companies and scrapping stock market data we can help users to get a clear image of companies profits and losses over the time and help them invest in profitable companies. Creating a ML module over the data-set will create a prediction model based on the data we collected, allowing the authorities to work on certain areas before the incident even happens.

2.3. Objectives

1.Web based GIS, Geographic Information system.
2. Scraping techniques.
3. Text information Extraction and APIs.
4. Real time data processing.

5. Big data tools for data processing
6. Finding NEWS relevant to the topic, focused temporal and special relevance of the NEWS
7. Distributive computation to save time and increase efficiency
8. Automating the tasks
9. Reducing the manual work

As mentioned earlier, finding patterns becomes much easier on a map. Filtering the NEWS through specific keywords can make the map useful to a number of different professions. Therefore, one possible use case could be that of a crime map for a certain area. A police analyst could use such a map to figure out hotspots for crime.

CrimeMapping is such an application that provides crime NEWS displayed on a map; however, no such application exists for Pakistan. A sample example of how CrimeMapping works can be seen in Figure 1.

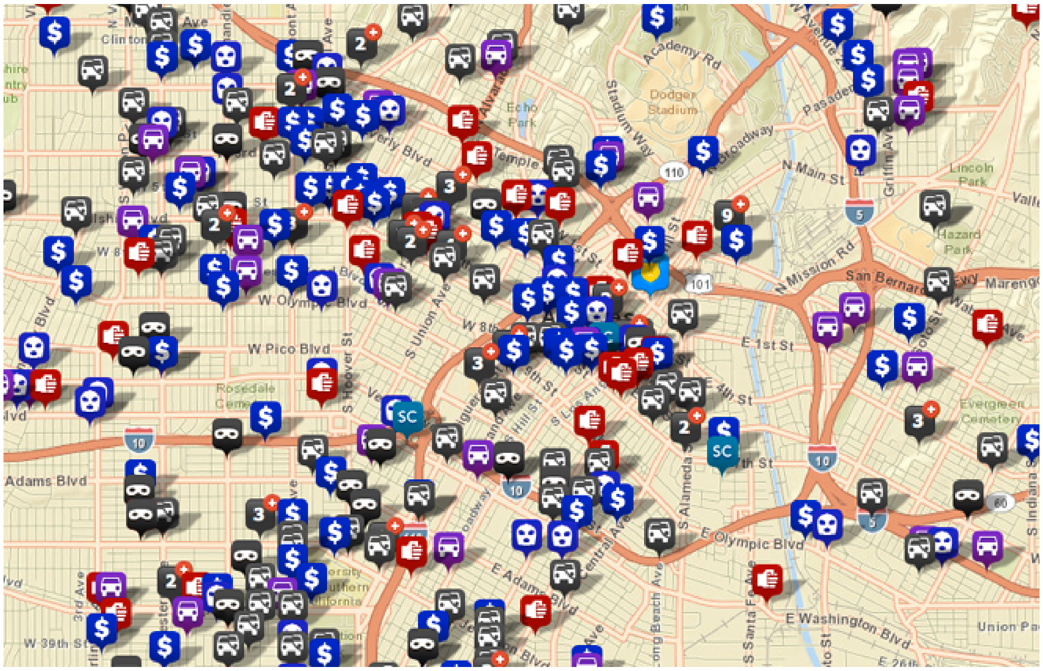


Figure 1: A map of Los Angeles as shown by the CrimeMapping application, displaying crime records as markers.

Another possible use case could be that of a property dealer making a prediction of the price of property in a specific location going up or down by locating patterns related to the construction of roads, airports, public transport nearby.

2.4. Project Scope

The project is aimed at creating a web application with an integrated GIS. The application will be able to show the NEWS on the map, however, the application is limited to only showing the results. It cannot find any geographically specific patterns that might be visible in the map or perform analysis of the kind. That is left up to the user.

Moreover, the NEWS to be displayed can only be displayed of NEWS that contain mention of a location. Displaying NEWS of a focus time of something happening in the future is not possible as well. Due to limited availability of digital NEWS documents, the application cannot display NEWS that were published very long ago.

We will be using geographical information available with the Election Commission of Pakistan to map the NEWS with a high level of granularity.

We will also try to enable analysis of live streams of NEWS, in which case the application will be deployed using Apache storm and the development model will be adapted according to the framework.

2.5. Constraints

1. The data set of Pakistan election commission has redundancy moreover there are certain places which have the same name as of its tehsil and district e.g Rawalpindi, We must have to extract the correct relevance from the article scrapped.
2. The data set also has some names that have separate spellings as mentioned in the NEWS articles. For example, in some places Muzaffargarh is written as one word whereas in our dataset it is written as two separate words i.e, Muzaffar garh.
3. Extracting focus time and focus location out of real time data is quite slow and inefficient and working with a huge amount of data will require distributed computing.
4. Some articles can mention multiple different locations at once. Deciding which one location to choose will be tricky.
5. Some articles are based on events happening in the future. These kinds of articles have to be filtered out as they are of no use to us.

2.6. Stakeholders

- NATIONAL RESEARCH PROGRAM FOR UNIVERSITIES

2.7. Target audience

1. **Civil servants and law enforcement agencies:** Someone working in a civil office could filter the NEWS based on the region he is assigned to and regularly update himself with reports in his region. NEWS could be filtered by crime types such as robbery and by region to identify crime hotspots.
2. **Relief workers:** Relief workers could easily keep track of provisions distributed in different regions by filtering out the NEWS based on keywords.
3. **Businessmen:** Businessmen could filter NEWS based on organizations or products to keep track of different NEWS regarding competition in their respective product lines.
4. **Business aficionados:** People who regularly ingest NEWS can use our program as well. Moreover, NEWS websites can integrate our program in their websites and applications so that their users can have another form of NEWS view as well.

3. Software Requirement Specifications

3.1. List of Features

1. Scrapping of NEWS from online NEWS sources.
2. Extraction of focus time and focus location from NEWS articles.
3. Filtering of NEWS plotted on map based on specific keywords.
4. Filtering of NEWS plotted on a map based on specific timeframes.
5. Filtering of NEWS plotted on a map based on specific locations.
6. Integrated Geographical Information System to plot NEWS on a map of Pakistan.
7. Real time processing of NEWS through Apache Storm.

3.2. Functional Requirements

1. The system shall scrap NEWS from online sources once every 1 hour at the start of the hour i.e, 12:00, 1:00, 2:00….
2. The system shall extract focus time and focus location information out of the NEWS documents.
3. The system shall refresh the webpage once every hour with an updated map.
4. The user should be able to filter NEWS based on multiple keywords.
5. The user should be able to filter NEWS based on locations which will be selected from the dropdown menu consisting of all possible choices.
    a. The user should be able to search inside the dropdown menu.
6. The user should be able to filter NEWS by providing a start date and end date and only view results that fall within that specific timeframe.
7. The system shall plot the NEWS on a map of Pakistan and markers should be placed at appropriate locations as per extracted in the focus location extraction module.
8. The user should be able to click on a NEWS marker and get access to details of the NEWS and a link to the original webpage where the article was acquired from.

3.3. Quality Attributes

1. **Performance:** As we are using high speed databases along with cached results as well as distributed computing, we will be able to perform high performance.
2. **Interoperability:** Our web application will be supported by all latest browsers available and will work smoothly.
3. **Maintainability:** We have divided our project into multiple modules and therefore, we would only need to make changes to each module separately making maintainability very easy.
4. **Scalability:** We are using Apache Spark and Apache Storm as our main computing platforms which themselves have the property of scalability, therefore we will be able to scale our app easily. Moreover, our database will be scalable over multiple nodes as well.
5. **Usability:** Our system will not be very complex as we are making it as abstract as possible and providing limited options to the user which are actually useful.

6. **Modifiability:** We have divided our project into multiple modules and therefore, we would only need to make changes to each module separately making maintainability very easy.
7. **Reusability:** We have divided our project into multiple modules and therefore, each module can be used in some other project easily.

3.4. Non-Functional Requirements

1. Large amounts of data will be processed in a distributed environment for faster processing.
2. The system will be easy to use and can be taught to someone with basic knowledge of computers in under 30 minutes.
3. System will cache queried results of up to a timeframe of 6 months for faster performance..
4. System will handle an extensive amount of data through Kafka streams.
5. System will process NEWS in real time through Apache Storm.

4. Iteration Plan

Iteration	**Time Frame**	**Modules to Cover**	
		Module	**Status**
1	September 2022 - October 2022	Scrapping online NEWS	Completed ▾
		Identification of NEWS focus time using Temporal Specificity	Completed ▾
		Identification of NEWS focus location using Spatial Specificity	Completed ▾
		Apache Kafka pipeline development	Completed ▾
2	November 2022 - December 2022	Development of Apache Spark processing model	Completed ▾
		Development of Apache Spark SQL	Completed ▾

		Database configuration with ECP dataset and Apache Spark model	Completed
3	February 2022 - March 2022	Development of Apache Storm module to process live stream of NEWS articles	Not started
		Development of web application	Not started
		Development of Geographical Information System module for the web application	Not started
4	April 2022 - May 2022	Docker images integration	Not started
		Website Deployment	Not started

5. Iteration 1

5.1 Use Cases

5.1.1 Use Case Diagram

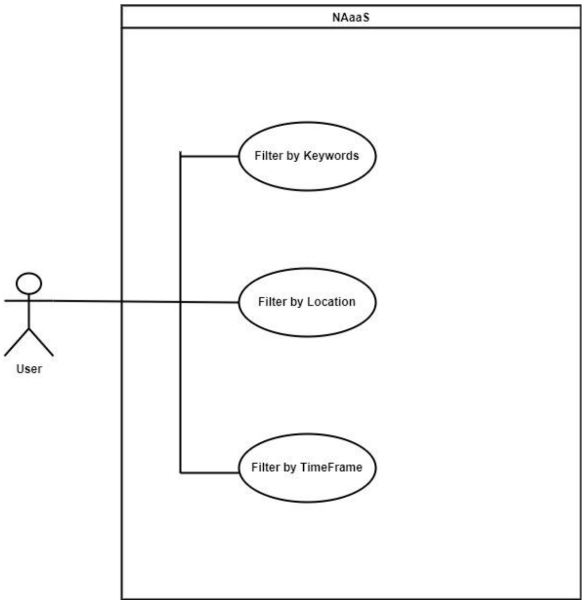


Figure 2: Use Case diagram

5.1.2 High Level Use Cases

5.1.2.1 Filter map based on keywords

Use Case	Filter map based on keywords
Actors	User
Type	Primary
Description	Users should be able to give keywords to find NEWS and the system should display all the NEWS related to those keywords.

5.1.2.2 Filter map based on location

Use Case	Filter map based on location
Actors	User
Type	Primary
Description	Users should be able to give the location of a NEWS and the system should display the NEWS in that specific location.

5.1.2.3 Filter map based on timeframe

Use Case	Filter map based on timeframe
Actors	User
Type	Primary
Description	Users should be able to give the start and end date of a NEWS and the system should display the NEWS in that

	timeframe.

5.1.3 Expanded Use Cases

5.1.3.1 Filter map based on keywords

<table>
<tr><td><b>Use case</b></td><td>Filter map based on keywords</td></tr>
<tr><td><b>Scope</b></td><td>The system under design</td></tr>
<tr><td><b>Primary Actor</b></td><td>User</td></tr>
<tr><td><b>Stakeholders</b></td><td>User</td></tr>
<tr><td><b>Postconditions</b></td><td>Map should be updated</td></tr>
<tr><td><b>Main Success Scenario</b></td><td><table><tr><th>User</th><th>System</th></tr><tr><td>1. User enters keywords in the input textbox</td><td>2. System updates the map with new markers displaying only NEWS that contain the entered keywords</td></tr></table></td></tr>
<tr><td><b>Extensions</b></td><td>2. If no NEWS exists that match the entered keywords, the system will ping the user with a no results found error.</td></tr>
<tr><td><b>Frequency of Occurrence</b></td><td>Possibly Every Minute</td></tr>
</table>

5.1.3.2 Filter map based on location

Use case	Filter map based on location
Scope	The system under design
Primary Actor	User

<table>
<tr><td><b>Stakeholders</b></td><td>User</td></tr>
<tr><td><b>Postconditions</b></td><td>Map should be updated</td></tr>
<tr><td><b>Main Success Scenario</b></td><td><table><tr><th>User</th><th>System</th></tr><tr><td>1. User chooses the location from the dropdown menu</td><td>2. System updates the map with new markers displaying only NEWS that are of the entered location.
3. System zooms the map bringing the entered location into view.</td></tr></table></td></tr>
<tr><td><b>Extensions</b></td><td></td></tr>
<tr><td><b>Frequency of Occurrence</b></td><td>Possibly Every Minute</td></tr>
</table>

5.1.3.3 Filter map based on timeframe

<table>
<tr><td><b>Use case</b></td><td>Filter map based on timeframe</td></tr>
<tr><td><b>Scope</b></td><td>The system under design</td></tr>
<tr><td><b>Primary Actor</b></td><td>User</td></tr>
<tr><td><b>Stakeholders</b></td><td>User</td></tr>
<tr><td><b>Postconditions</b></td><td>Map should be updated</td></tr>
<tr><td><b>Main Success Scenario</b></td><td><table><tr><th>User</th><th>System</th></tr><tr><td>1. User chooses the starting date for the timeframe.
2. User chooses the ending date for the timeframe.</td><td>3. System updates the map with new markers displaying only NEWS that fall in the timeframe.</td></tr></table></td></tr>
</table>

Extensions	1. If the user chooses an invalid date, the system will prompt the user with an invalid date error. 2. If the user chooses an invalid date, the system will prompt the user with an invalid date error.
Frequency of Occurrence	Possibly Every Minute

5.2 Operational Contracts

5.2.1 Filter map based on keywords

Contract OC#01:	
Operation	fetchNews(Keywords)
Cross Reference	Filter map with NEWS based on keywords
Precondition	1. News with that keyword exists.
Post Condition	1. An instant to fetch NEWS NewsFetcher is created. 2. NewsFetcher gets the NEWS from the database based on keywords. 3. NewsFetcher loads NEWS from the database and plots it on the map. 4. NewsFetcher caches the query.

5.2.2 Filter map based on location

Contract OC#02:	
Operation	fetchNews(Location)
Cross Reference	Filter map with NEWS based on selected location
Precondition	1. News is scraped, processed and stored in a database. 1. The selected location is among the available locations.
Post Condition	1. News is loaded from a database and plotted on a map. 2. The query is cached.

5.2.3 Filter map based on timeframe

Contract OC#03:	
Operation	fetchNews(timeFrame)
Cross Reference	Filter map with NEWS based on a specific timeframe
Precondition	1. News is scraped, processed and stored in a database. 2. The timeframe is among the possible timeframes.
Post Condition	1. News is loaded from a database and plotted on a map. 2. The query is cached.

5.3 System Sequence Diagrams

5.3.1 Filter map based on keywords

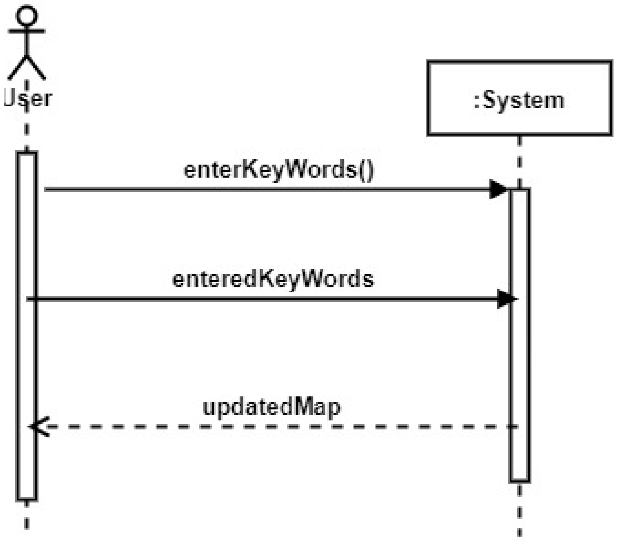


Figure 3: Filter map based on keywords system sequence diagram

5.3.2 Filter map based on timeframe

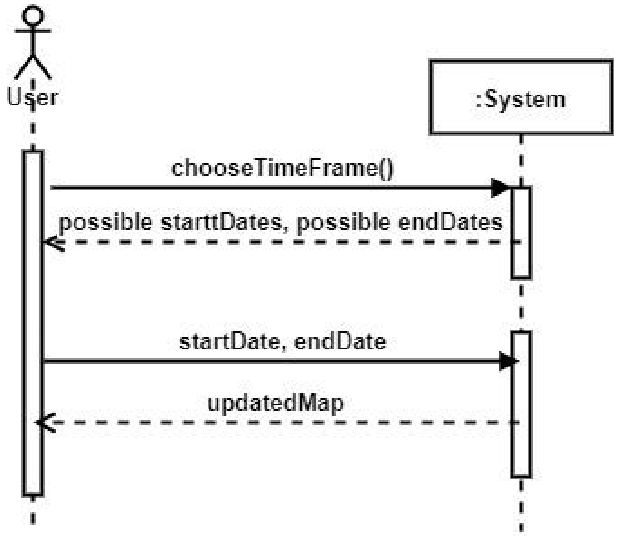


Figure 4: Filter map based on timeframe system sequence diagram

5.3.3 Filter map based on location

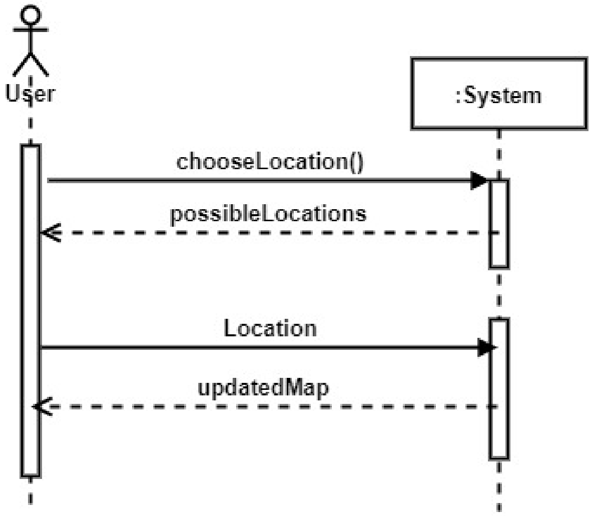


Figure 5: Filter map based on location system sequence diagram

5.3.4 Filter map based on timeframe, keywords, location

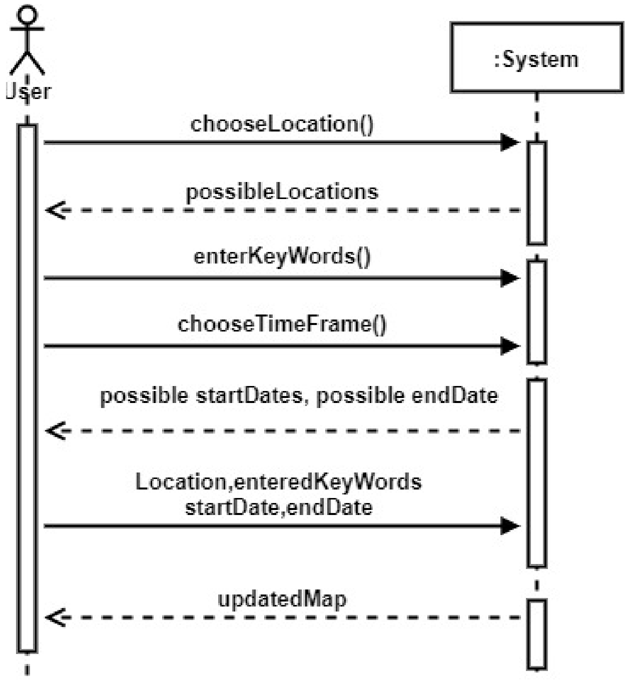


Figure 6: Filter map based on timeframe, keywords, location system sequence diagram

5.4 Domain Model

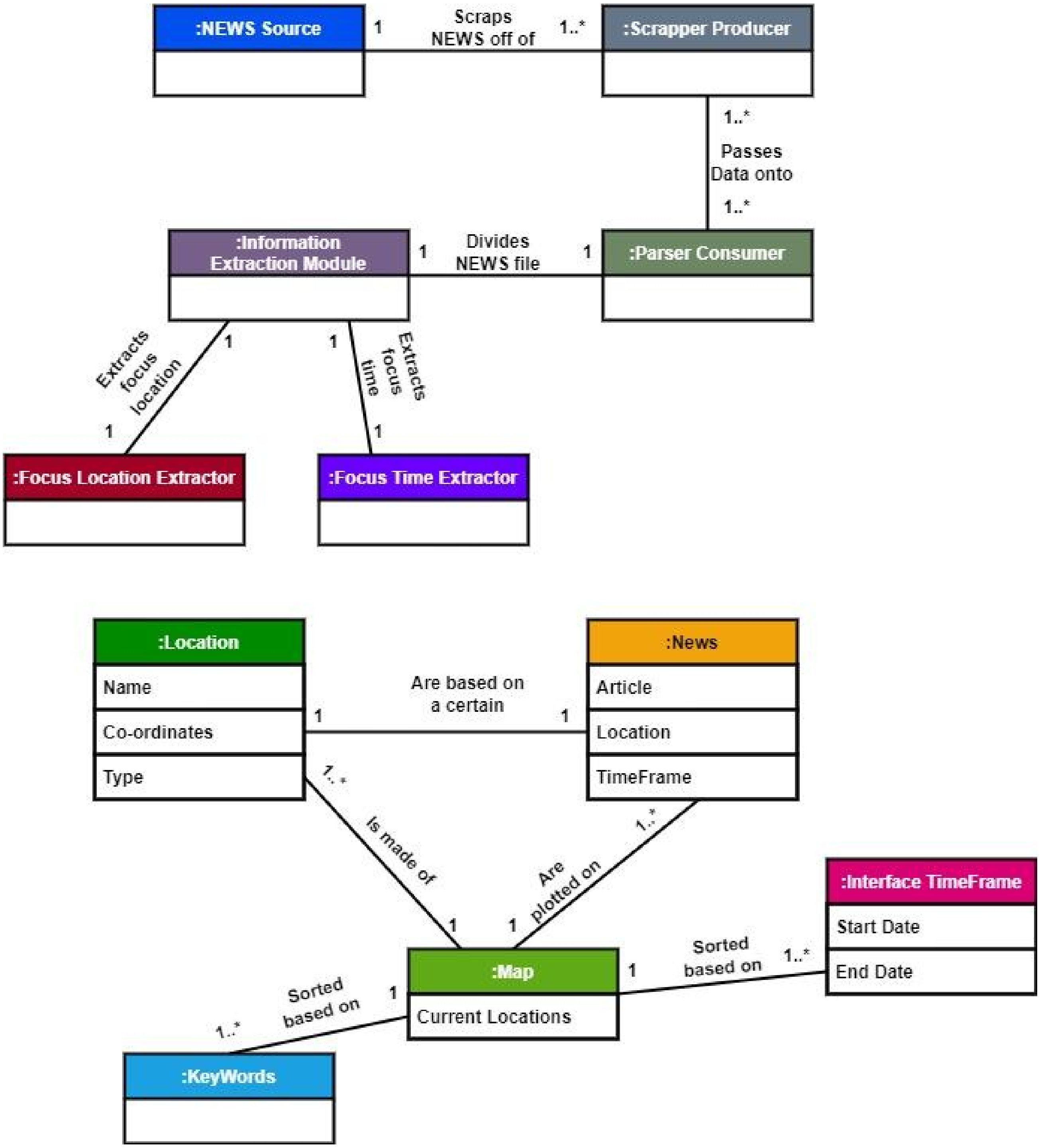


Figure 7: Domain Model

5.5 Sequence Diagrams

5.5.1 Filter map based on timeframe

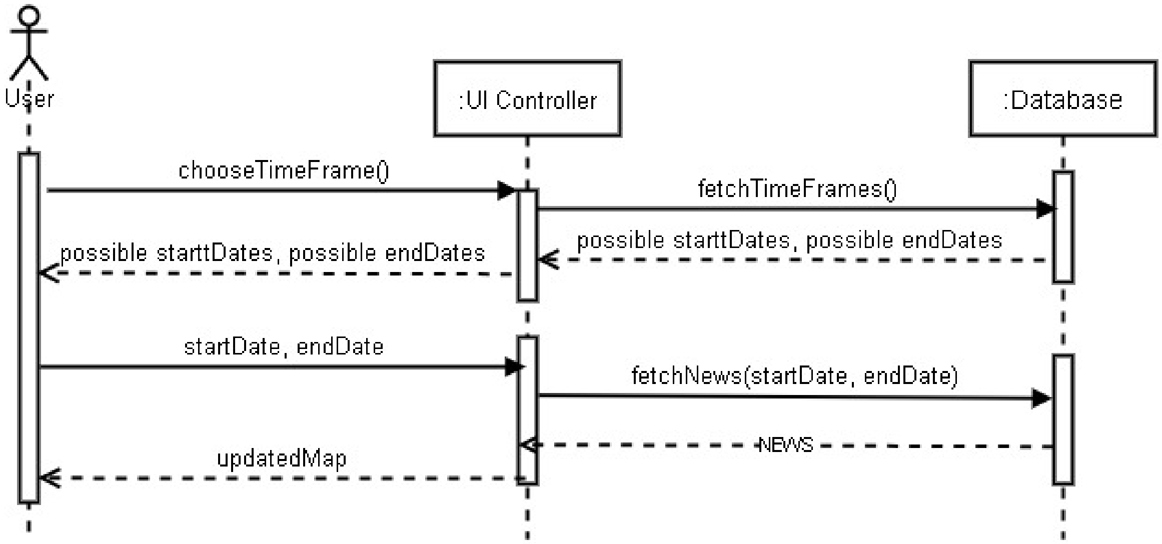

User
:UI Controller
:Database
chooseTimeFrame()
fetchTimeFrames()
possible starttDates, possible endDates
possible starttDates, possible endDates
startDate, endDate
fetchNews(startDate, endDate)
NEWS
updatedMap

Figure 8: Filter map based on timeframe sequence diagram

5.5.2 Filter map based on keywords

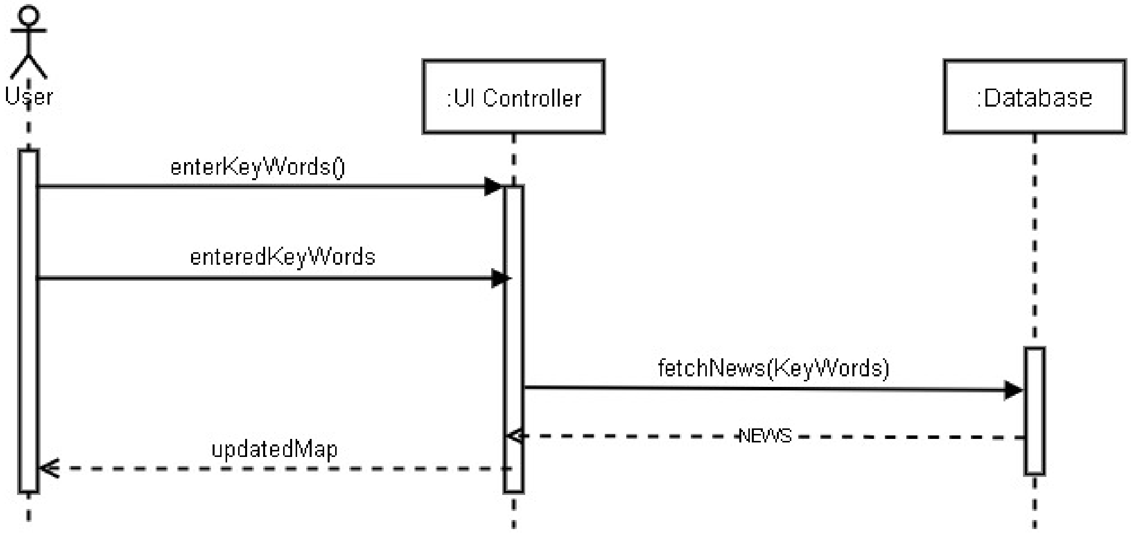


Figure 9: Filter map based on keywords sequence diagram

5.5.3 Filter map based on location

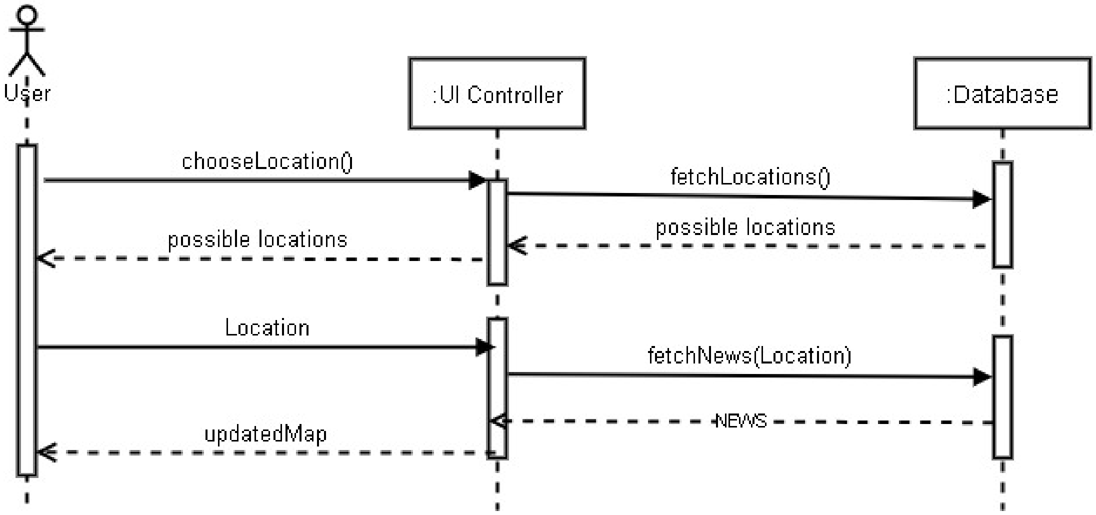


Figure 10: Filter map based on location sequence diagram

5.5.4 Scrap, produce and consume NEWS

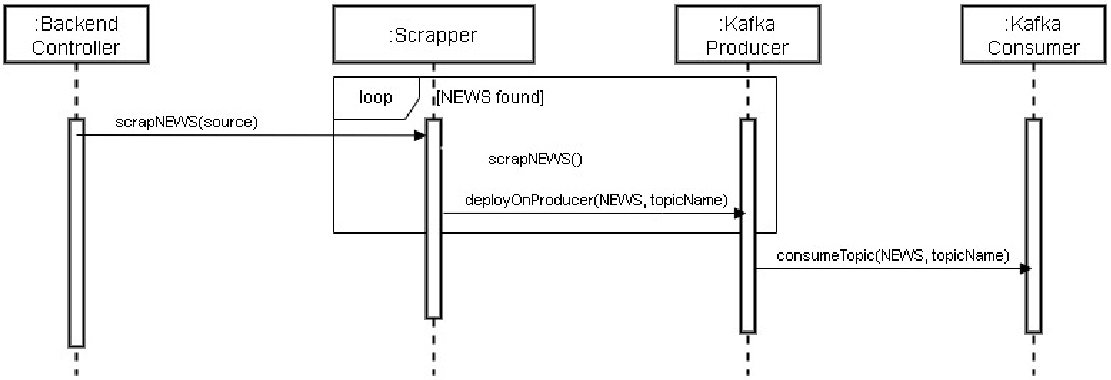


Figure 11: Scrap, produce and consume NEWS sequence diagram

5.5.5 Filter map based on timeframe, keywords, location

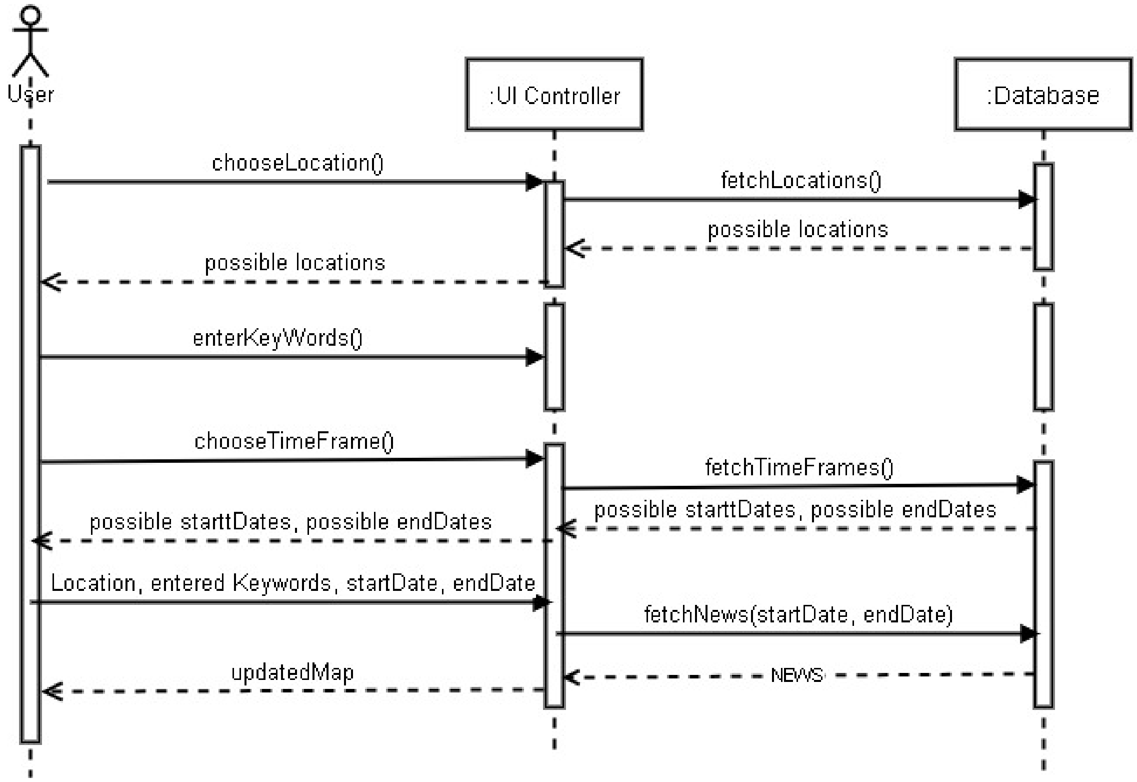


Figure 12: Filter map based on timeframe, keywords, location sequence diagram

5.6 Entity Relationship Diagram

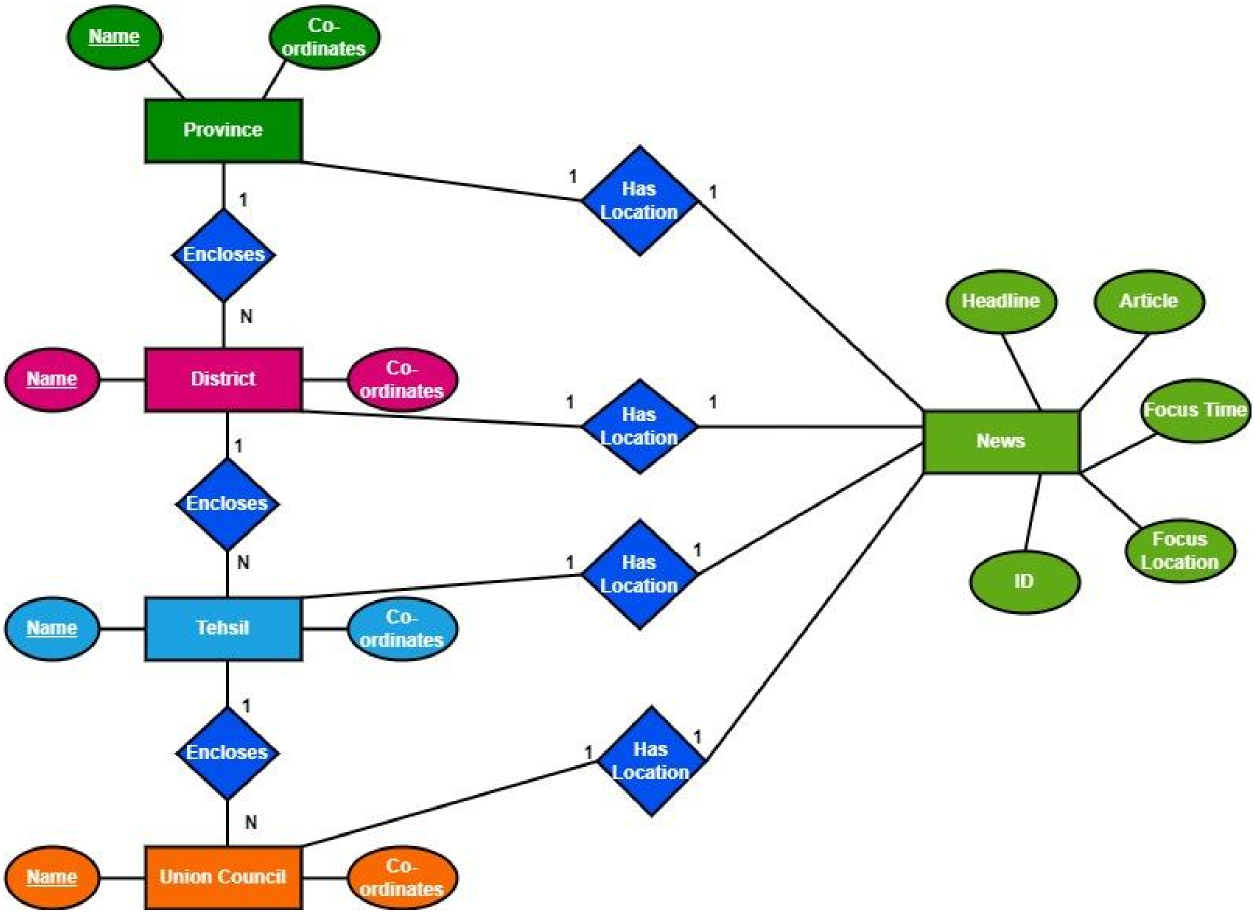


Figure 13:Entity relationship diagram

5.7 Relational Schema Diagram

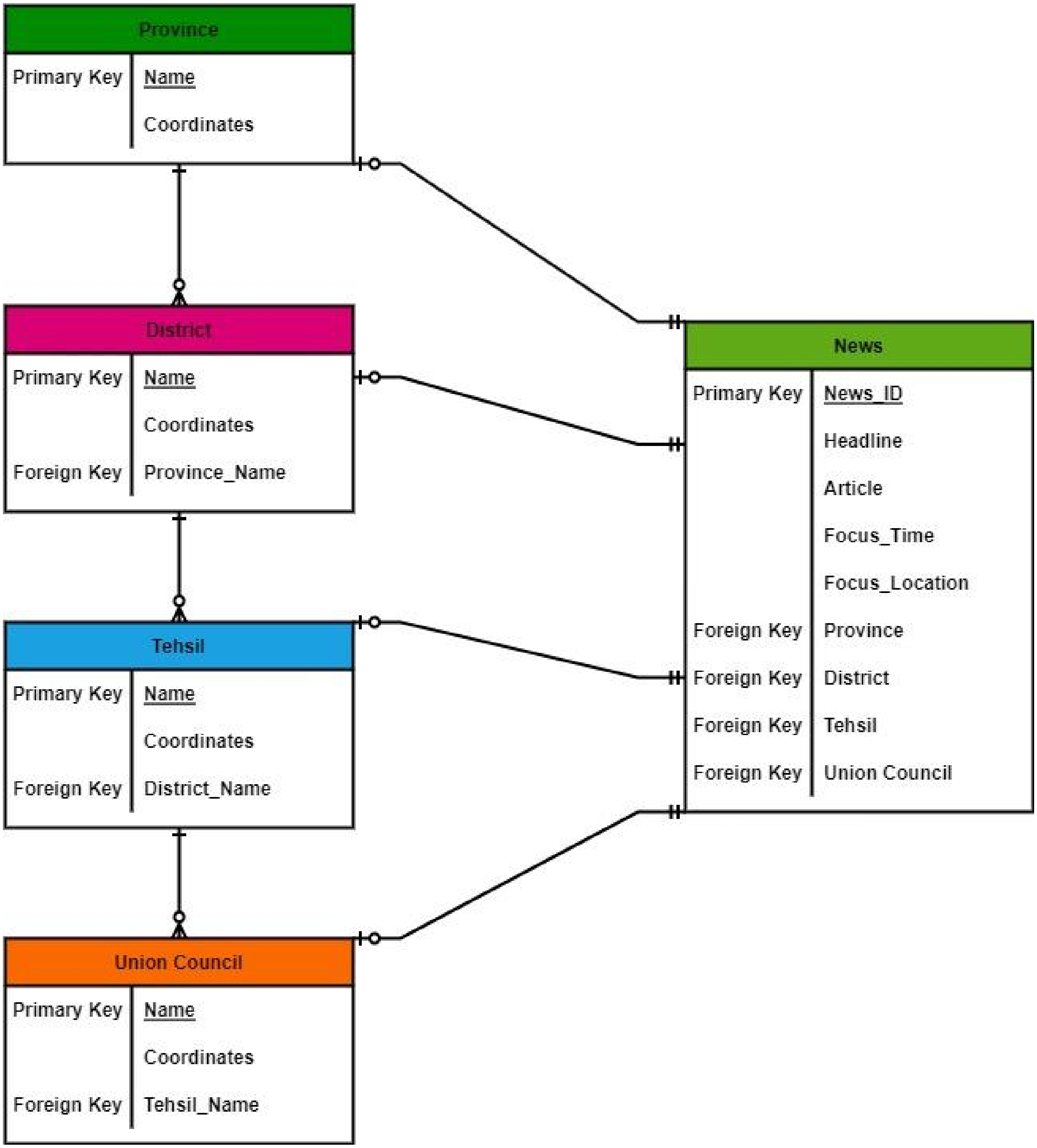


Figure 14: Relational Schema diagram

5.8 Class Diagram

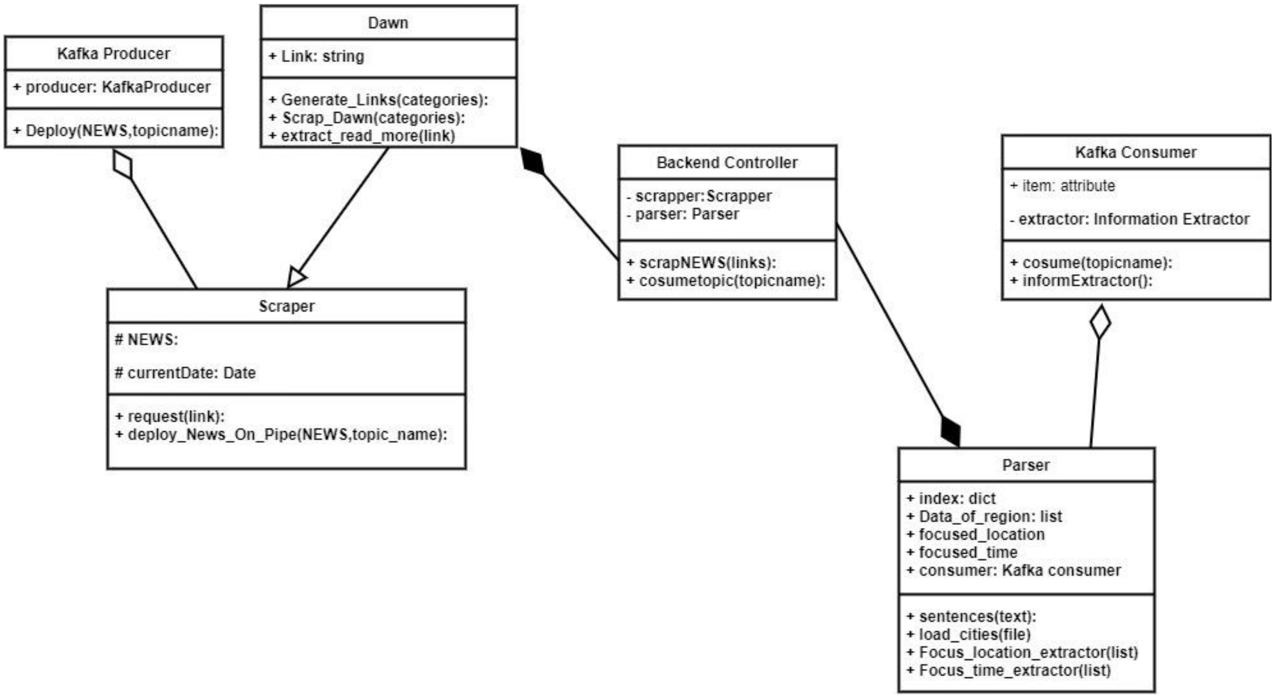


Figure 15: UML Class diagram

5.9 Architecture diagram

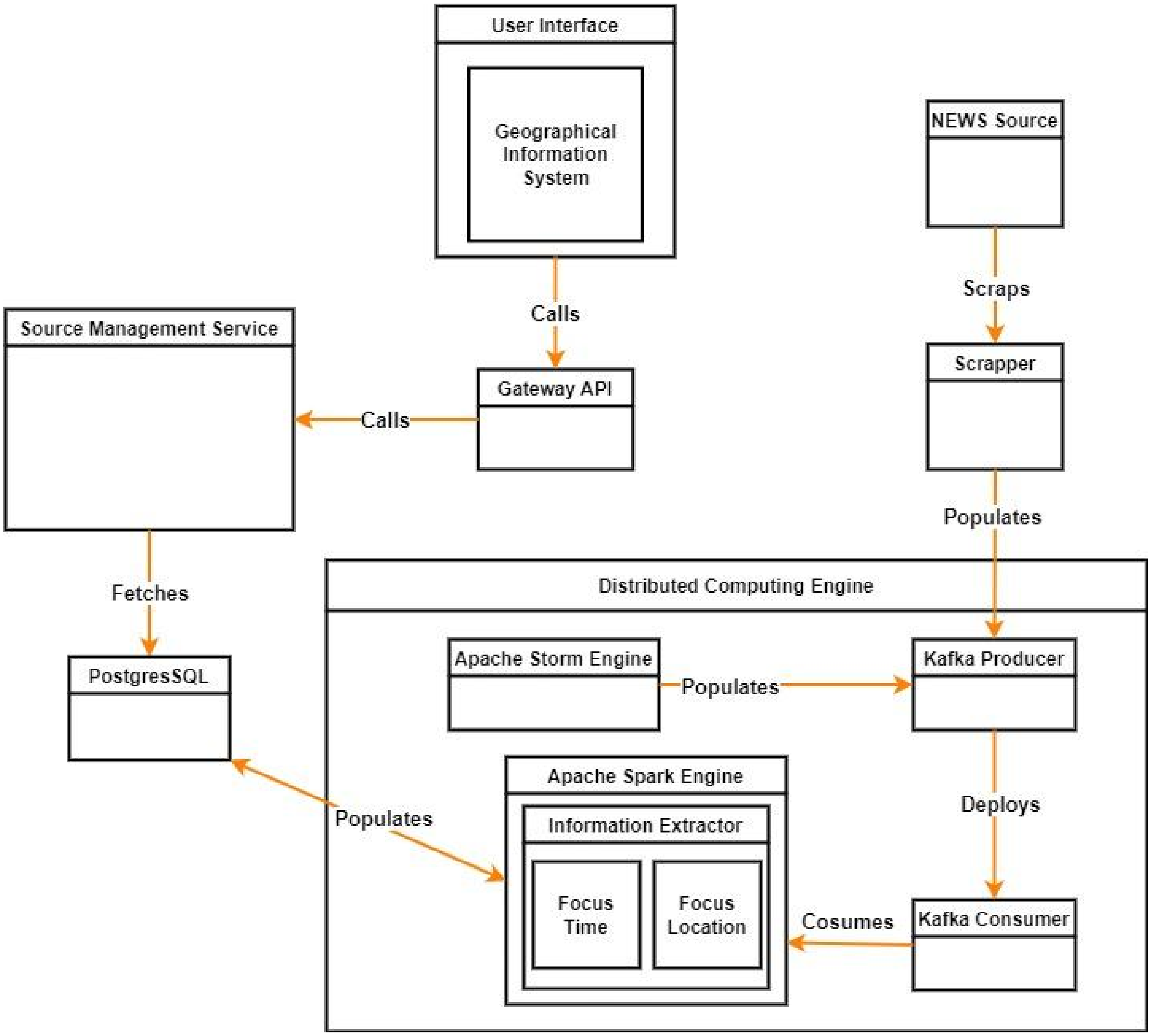


Figure 16: High Level Architecture diagram

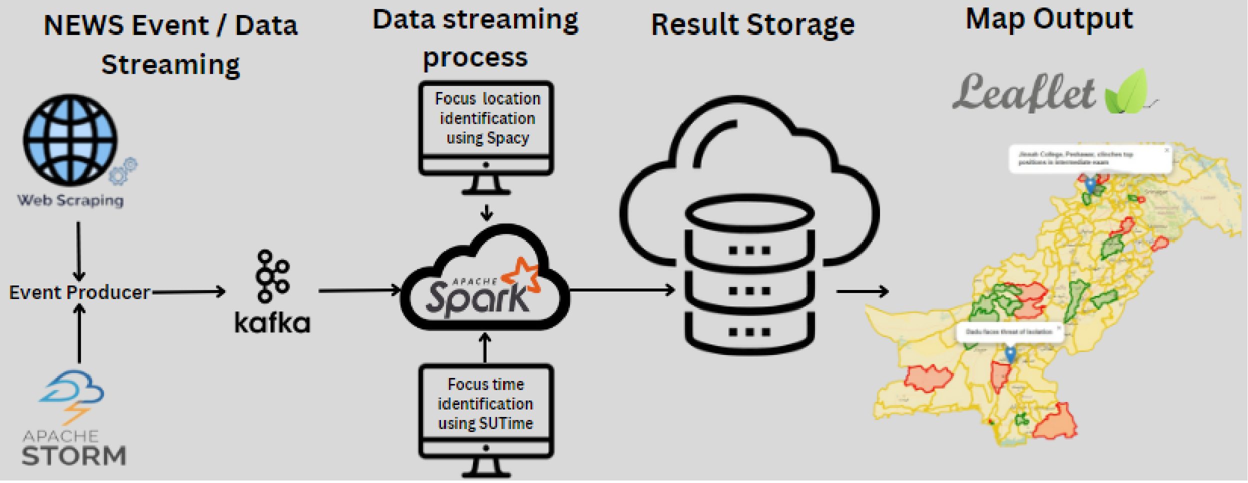

NEWS Event / Data Streaming
Web Scraping
Event Producer
APACHE STORM
kafka
Data streaming process
Focus location identification using Spacy
APACHE Spark
Focus time identification using SUTime
Result Storage
Map Output
Leaflet
Jinnah College, Peshawar, clinches top positions in intermediate exam
Dadu faces threat of isolation

Figure 17: Architecture diagram with tools

6. Iteration 2

6.1 Use Cases, System Sequence Diagrams, Sequence Diagrams

As per the recommendation of the FYP panel, we were asked not to make:

- Use case diagrams
- System sequence diagrams
- Sequence diagrams

As they were not applicable to our FYP. We were instead instructed to use Data Flow Diagrams and State Machines for the design of our project. If you wish to view our previous work, please refer to section 5. Iteration 1.

6.2 Architecture Diagram

There has been no change in the architecture diagrams. Please refer to 5.9 Architecture Diagram.

6.3 Data Flow Diagrams

6.3.1 Level 0

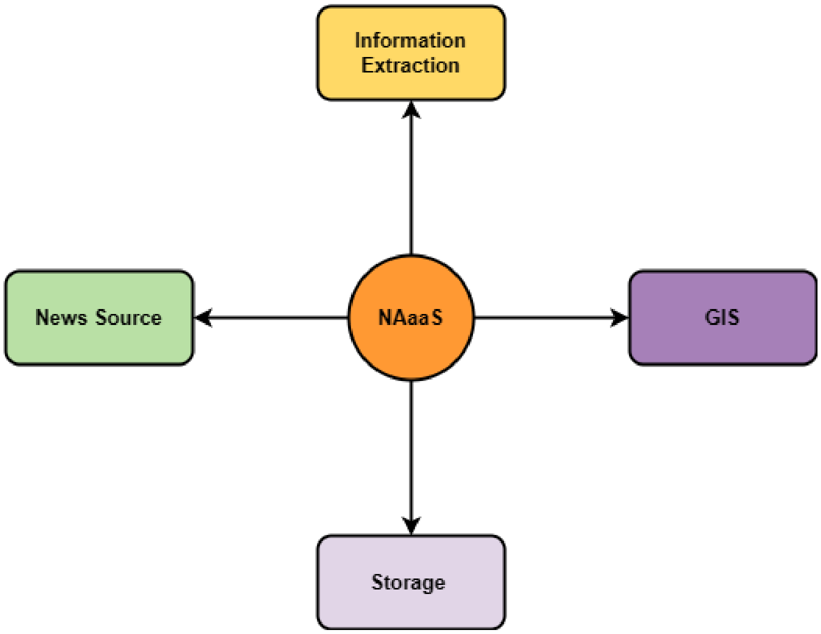


Figure 18: DFD Level 0

6.3.2 Level 1

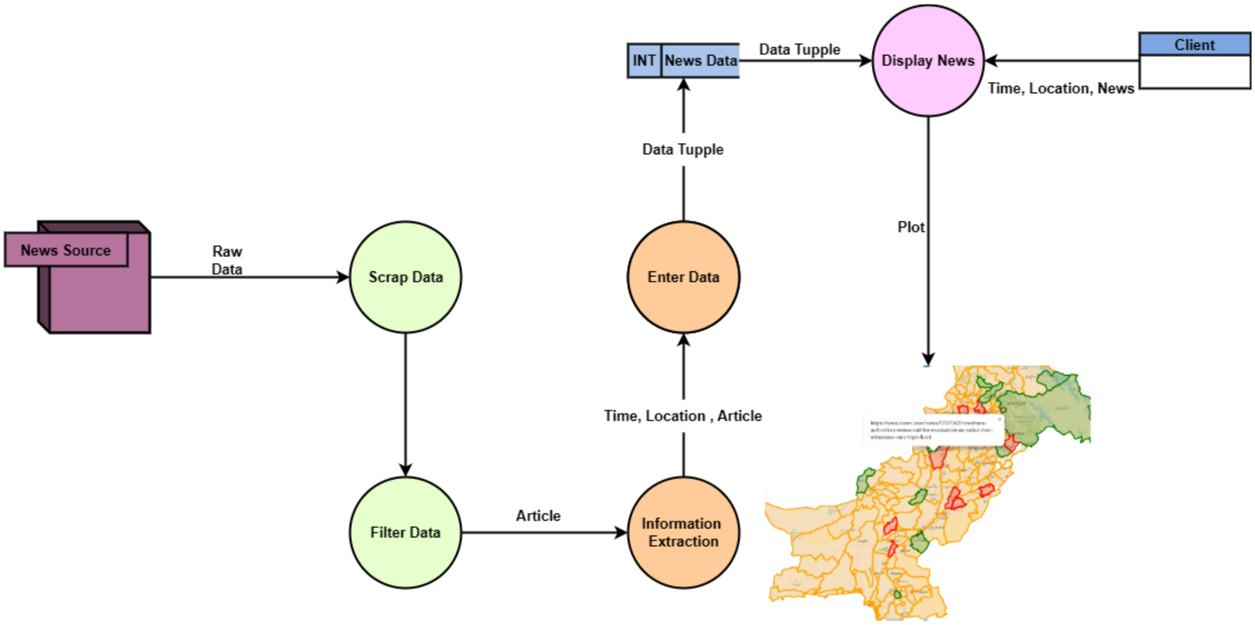
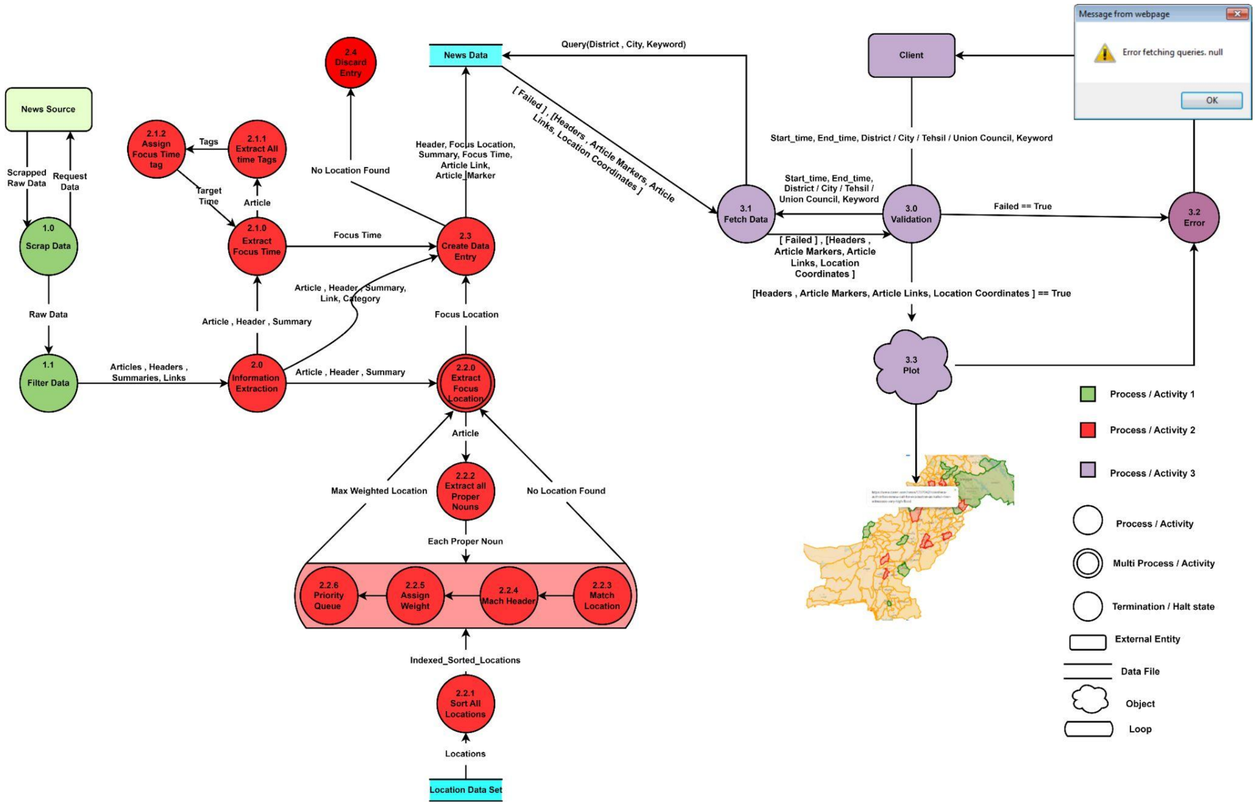


Figure 19: DFD Level 1

6.3.3 Level 2

As this diagram was too large, it might not be visible. Please visit this Link for a better view.



Message from webpage

Error fetching queries. null

OK
News Source
Scrapped Raw Data
Request Data
1.0 Scrap Data
Raw Data
1.1 Filter Data
Articles , Headers , Summaries, Links
2.0 Information Extraction
Article , Header , Summary
2.1.0 Extract Focus Time
Article
2.1.1 Extract All time Tags
Tags
2.1.2 Assign Focus Time tag
Target Time
Focus Time
Article , Header , Summary, Link, Category
2.3 Create Data Entry
Header, Focus Location, Summary, Focus Time, Article Link, Article_Marker
No Location Found
2.4 Discard Entry
News Data
Focus Location
Article , Header , Summary
2.2.0 Extract Focus Location
Article
2.2.2 Extract all Proper Nouns
Max Weighted Location
No Location Found
Each Proper Noun
2.2.3 Match Location
2.2.4 Mach Header
2.2.5 Assign Weight
2.2.6 Priority Queue
Indexed_Sorted_Locations
2.2.1 Sort All Locations
Locations
Location Data Set
Query(District , City, Keyword)
[Failed] , [Headers , Article Markers, Article Links, Location Coordinates]
3.1 Fetch Data
Client
Start_time, End_time, District / City / Tehsil / Union Council, Keyword
Start_time, End_time, District / City / Tehsil / Union Council, Keyword
3.0 Validation
[Failed] , [Headers , Article Markers, Article Links, Location Coordinates]
Failed == True
3.2 Error
[Headers , Article Markers, Article Links, Location Coordinates] == True
3.3 Plot
🟩 Process / Activity 1
🟥 Process / Activity 2
🟪 Process / Activity 3
Process / Activity
Multi Process / Activity
Termination / Halt state
External Entity
Data File
Object
Loop

Figure 20: DFD Level 2

6.4 State Machine Diagram

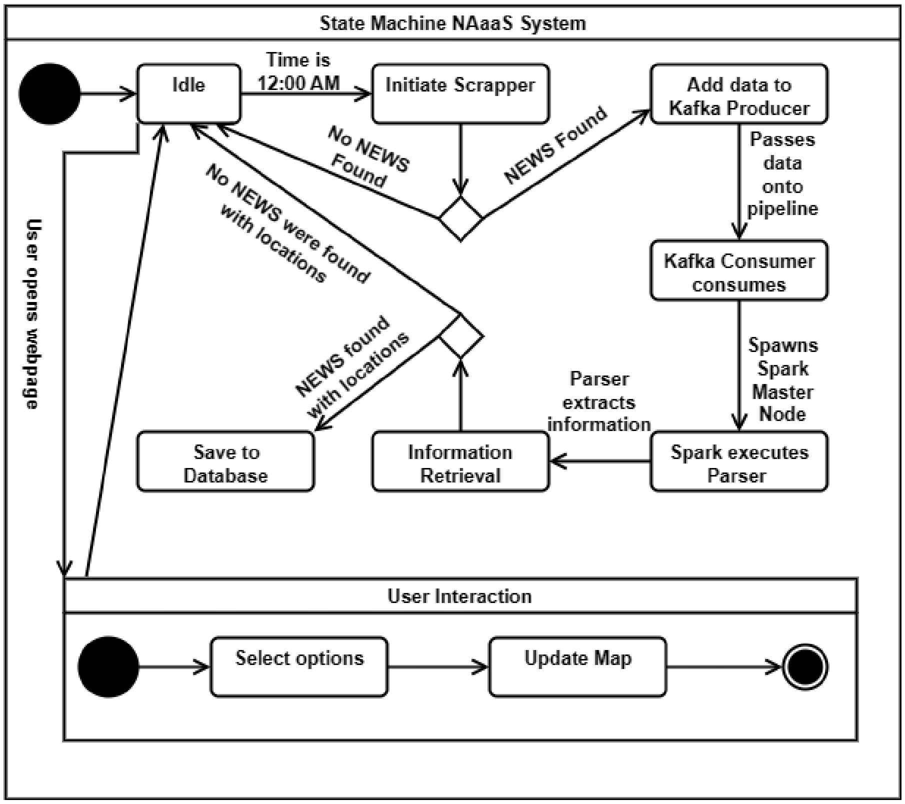


Figure 21: State Machine Diagram for the entire system

6.5 Entity Relationship Diagram

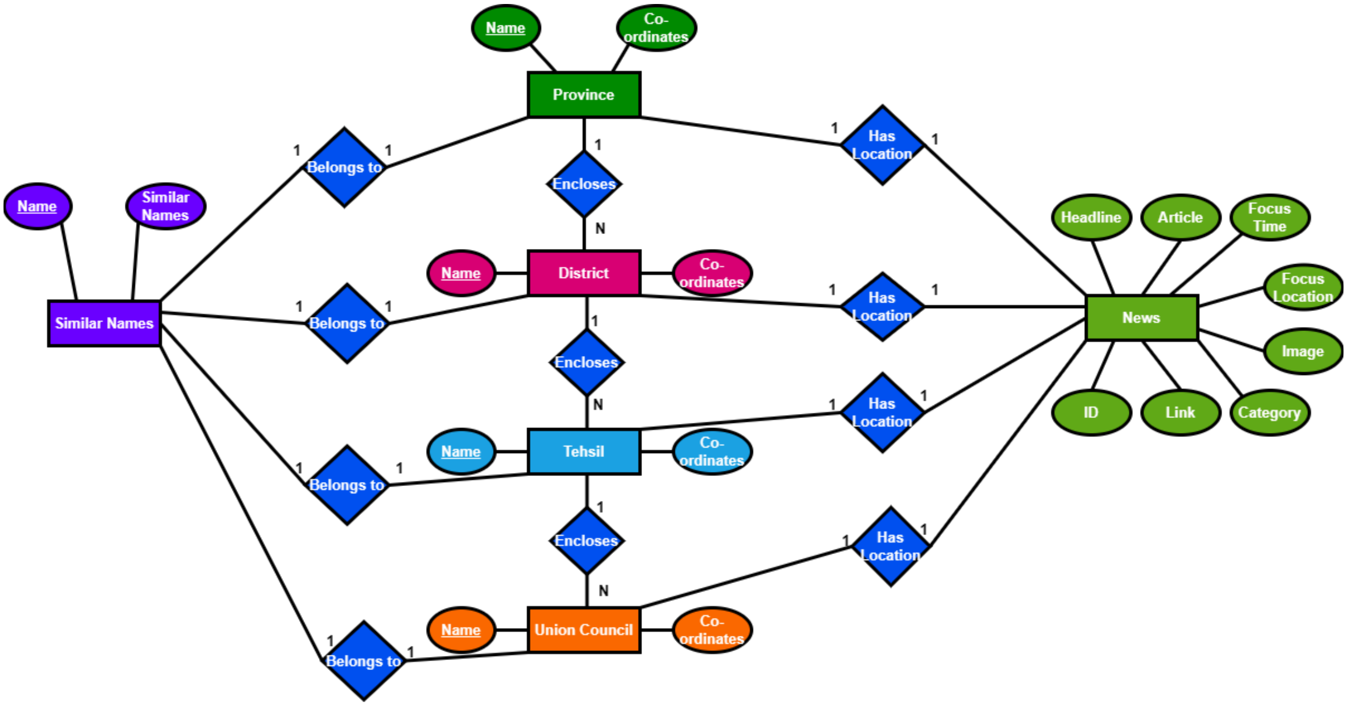


Figure 22: Entity Relationship Diagram

6.6 Relational Schema Diagram

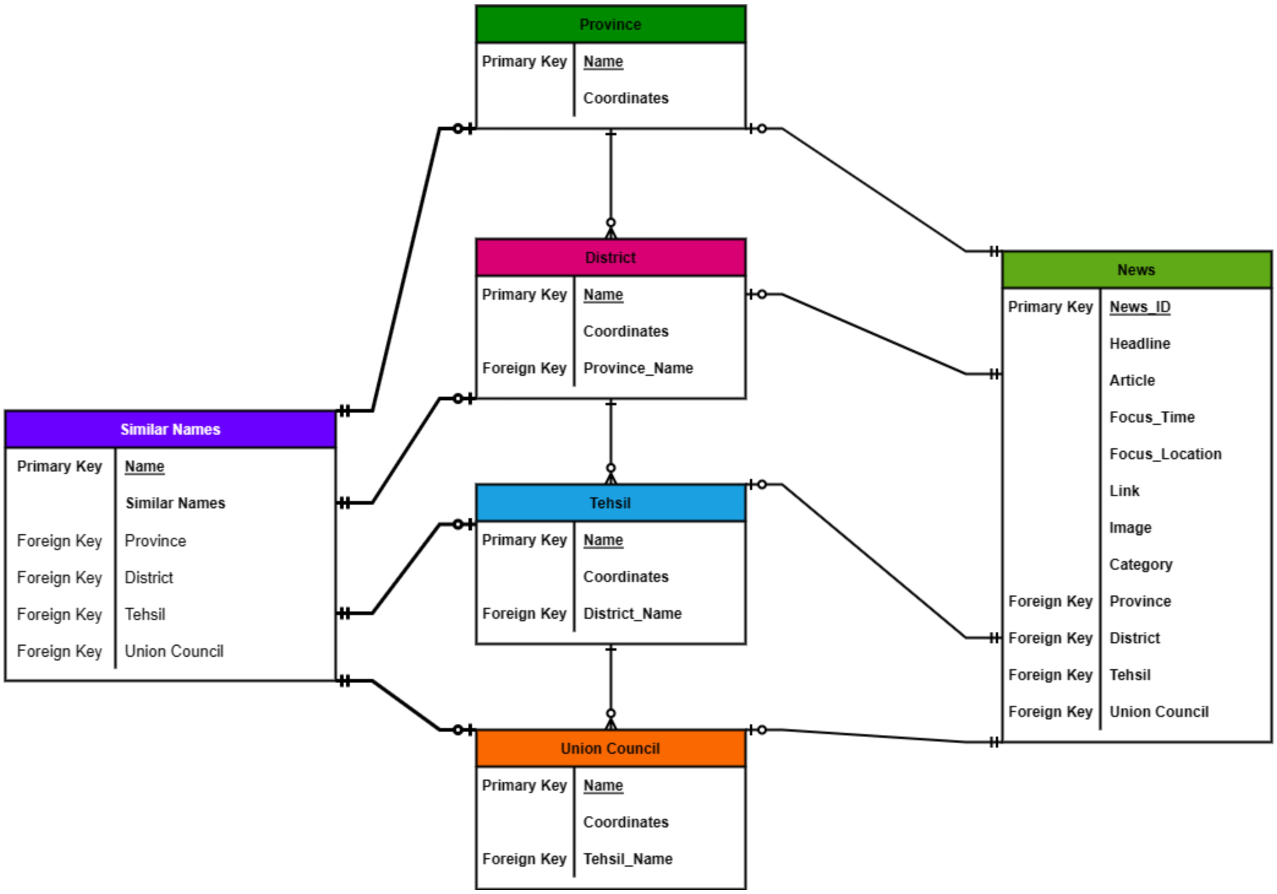


Figure 23: Relational Schema Diagram

6.7 Class Diagram

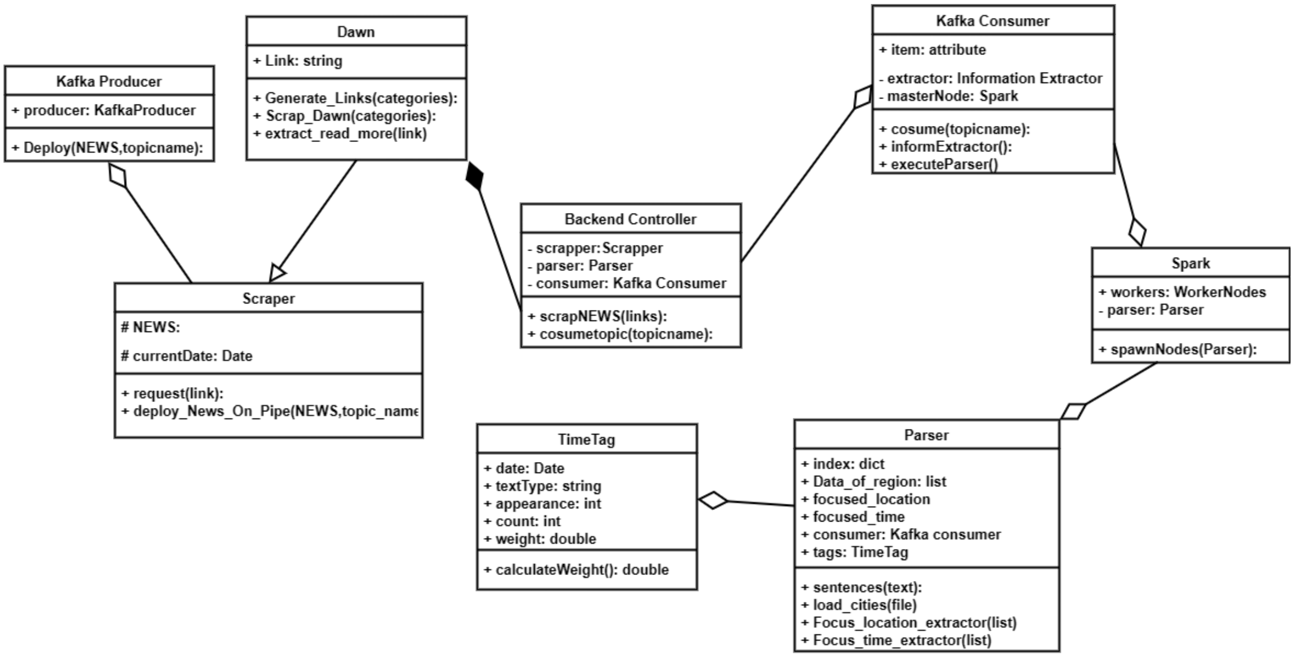
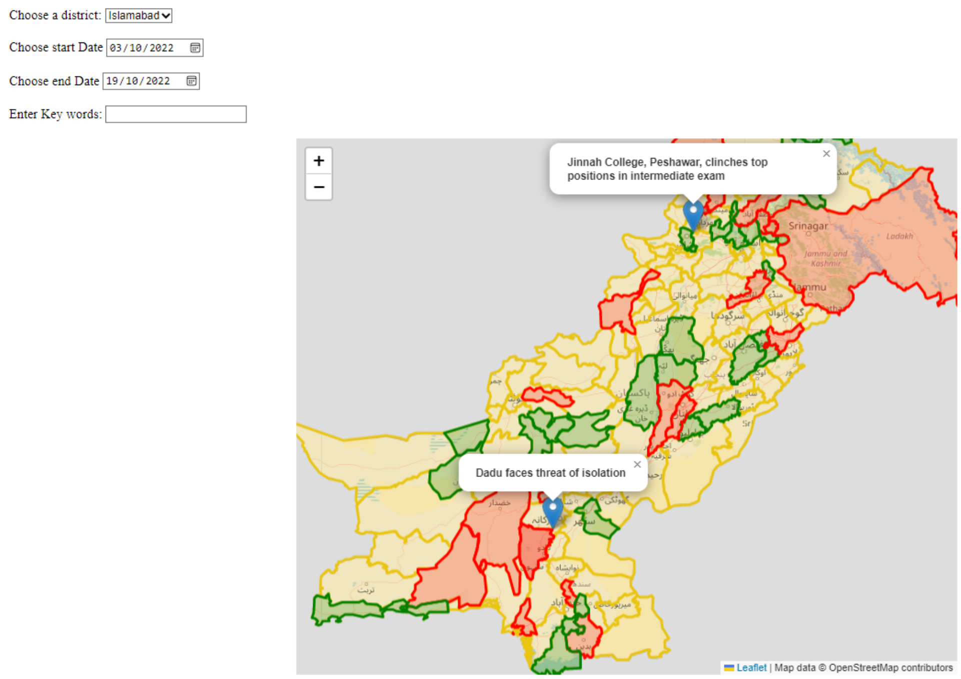


Figure 24: Class Diagram

6.8 Interface

6.8.1 Dummy interface with Geographical Information System



Choose a district: Islamabad

Choose start Date 03/10/2022

Choose end Date 19/10/2022

Enter Key words:

Jinnah College, Peshawar, clinches top positions in intermediate exam

Dadu faces threat of isolation

Figure 25: Dummy Interface with GIS

7. Implementation Details

7.1 Iteration 1

7.1.1 Scrapper Module

For this module, we require scraper to scrape the target webpage. We are using the urllib, datetime, bs4 and kafka packages for this reason. A flow of the program is specific for DAWN scraper and steps are as follows:

1. Determine a generic link
2. Define categories of NEWS available on the website
3. Get previous date
4. Create links by appending categories and previous dates.
5. Request to get the web page
    a. If webpage is successfully retrieve, load it in beautiful soap

i. Extract Header, summary and read more (Full article) link

ii. Request to get the full article

   1. If successfully received

      a. Load in beautiful soap

      b. Extract full article in correct grammar

      c. Create Header, Summary and article list

      d. Convert list to JSON object

      e. Throw on Kafka pipeline

   2. Else, try it again after 10 sec

b. Else, try it again after 10 sec

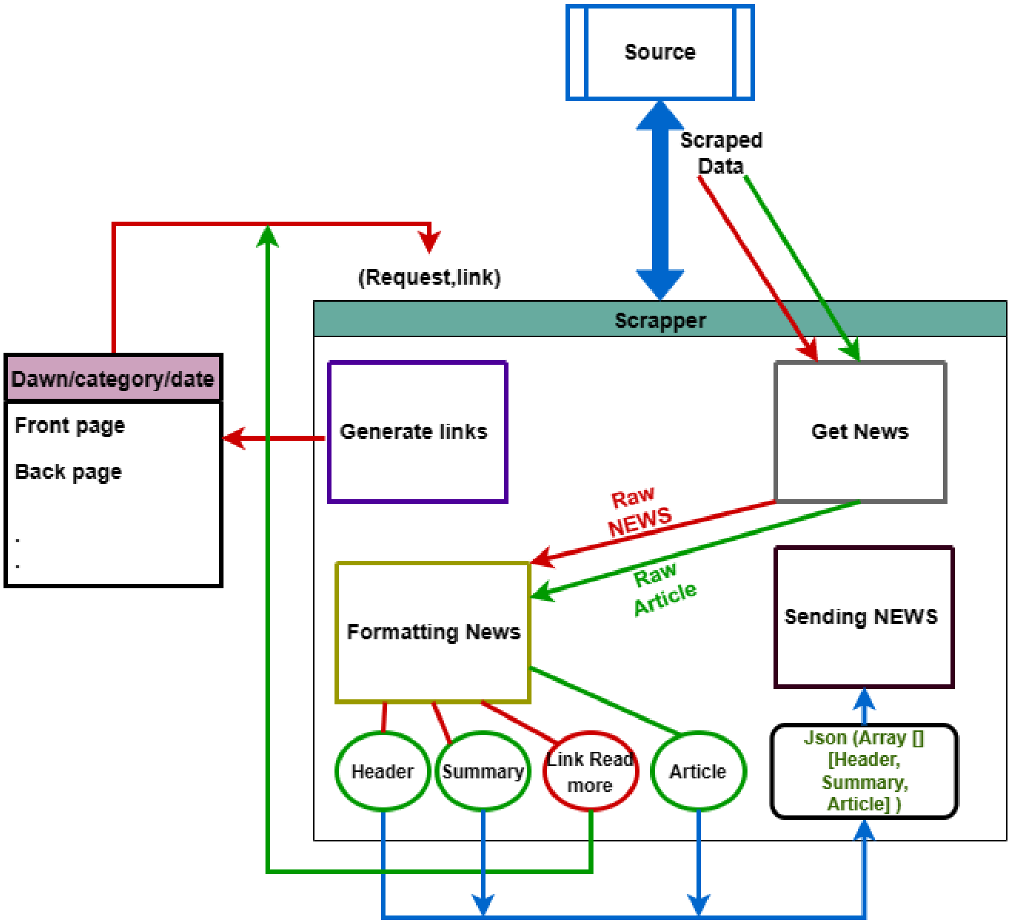


Figure 26: Scrapper Module Architecture

7.1.2 Geographical Specificity Module

For this module, we require pronoun tags that can tag location mentions inside documents. We are using the spacy, nltk, pandas, re and kafka packages for this reason. A flow of the program is as follows:

1. Get JSON object from kafka pipeline
2. Convert JSON into 3d list Header, summary and News Article
3. Load data set of Election Commission of Pakistan (Areas) in a list
4. Sort list
5. Create index based on starting alphabet
6. Extract pronouns from the received NEWS.
    a. Extract the start and end indexes of list (Areas) based on the first alphabet of pronoun
    b. If first word of area matches with the pronoun, get words from NEWS such as len(Pronoun) = = len(Area)
    c. If Pronoun matches to area, get words before and after from the News according to window size defined
        i. Check extracted words from (c) with header
            1. If matches, assign weight as normal_weight+scaling_factor to pronoun
            2. Else assign normal weight to pronoun
        ii. Maintain the list of pronouns as cities
    d. Get city as focused location whose weight is highest

A detailed flowchart of the above program can be seen in Figure 27.

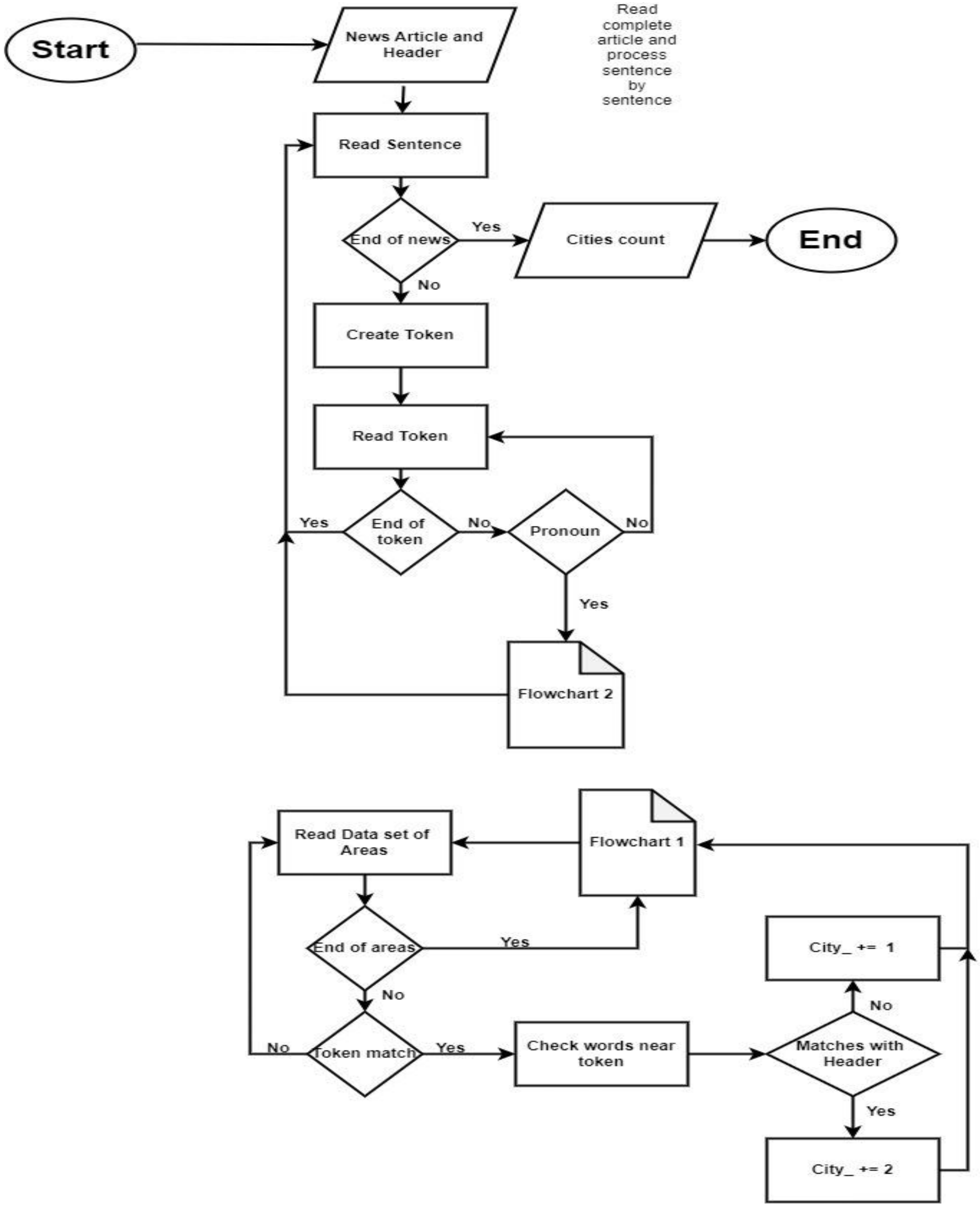


Figure 27: Focus location extraction flowchart

7.1.3 Temporal Specificity Module

For this module, we require temporal taggers that can tag time mentions inside documents. We are using the SUTime package for this reason. A flow of the program is as follows:

1. Extract Temporal taggers of the Header of the NEWS.
    a. If date tag is present and one that does not refer to the future, assign that as focus time. Else, move on.
2. Extract Temporal taggers of the Summary of the NEWS.
    a. If date tag is present and one that does not refer to the future, assign that as focus time. Else, move on.
3. Remove publication date from the Details of the NEWS so that it does not interfere with the temporal taggers.
4. Extract Temporal taggers of the Details of the NEWS.
    a. If date tag is present and one that does not refer to the future, assign that as focus time. Else, assign publication time as focus time.

A flowchart of the above program can be seen in figure 28.

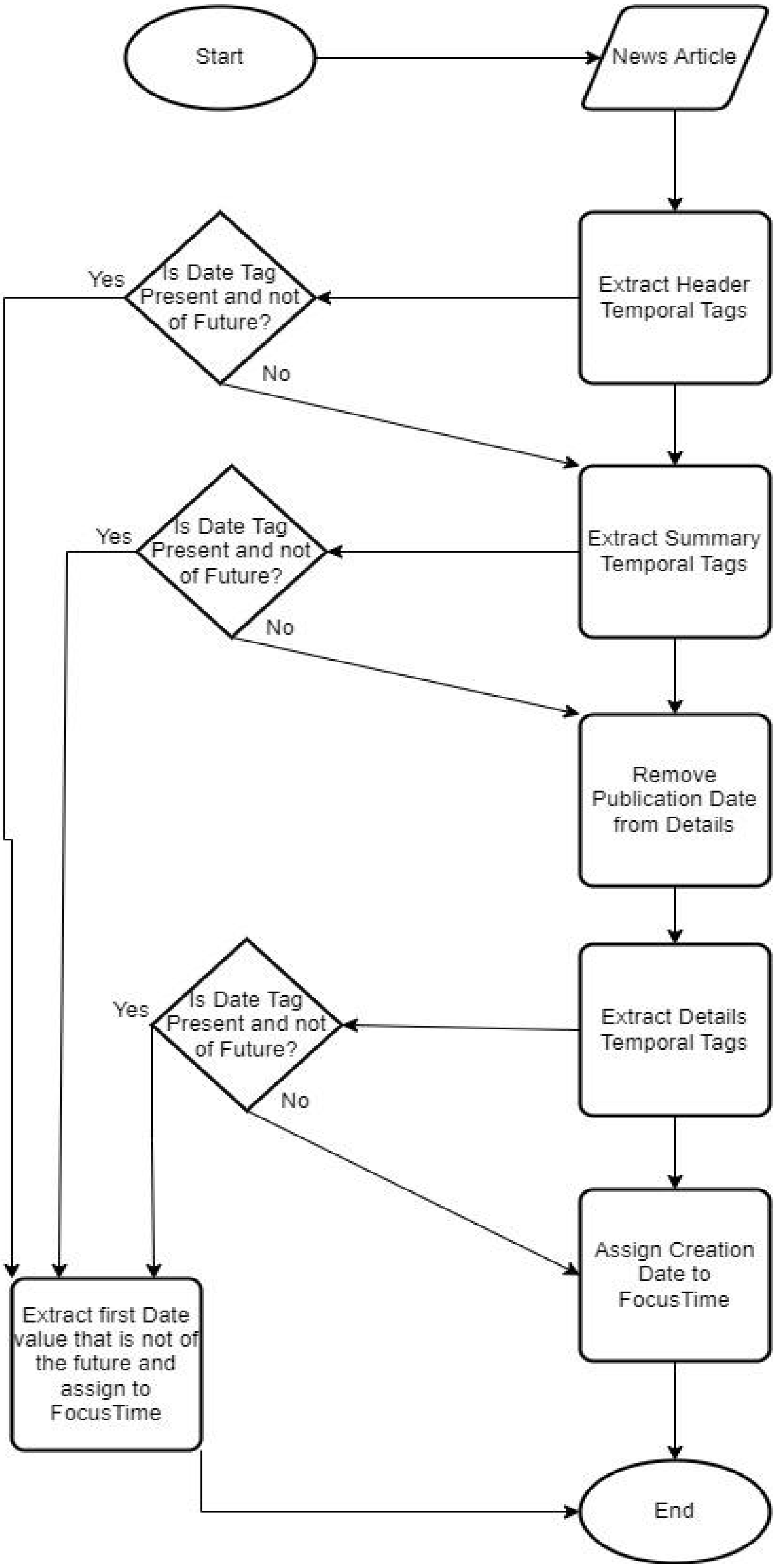

Start
News Article
Extract Header Temporal Tags
Is Date Tag Present and not of Future?
Yes
No
Extract Summary Temporal Tags
Is Date Tag Present and not of Future?
Yes
No
Remove Publication Date from Details
Extract Details Temporal Tags
Is Date Tag Present and not of Future?
Yes
No
Assign Creation Date to FocusTime
Extract first Date value that is not of the future and assign to FocusTime
End

Figure 28: Focus time extraction flowchart

We have noticed that the earliest possible mention of the NEWS in the document relates to the focus time of the NEWS in almost 80% of the cases, therefore we have decided to use that as a base for our focus time extraction.

7.1.4 Kafka Producer-Consumer Module

1. A cluster of Kafka brokers was set up using docker.
2. Using Kafka-Python library, Kafka Producer class was set up connected to a kafka broker.
3. Similarly, Kafka Consumer class was set up connected to the same kafka broker as the producer.
4. Kafka-Producer gets the NEWS and stores them on topics replicated and distributed on multiple nodes.
5. Kafka-Consumer reads from those topics and sends the data to the information extraction module.

7.2 Iteration 2

7.2.1 Temporal Specificity Module

We have updated our temporal specificity module as well as its implementation. For this module, we require temporal taggers that can tag time mentions inside documents. We are using the SUTime package for this reason. A flow of the program is as follows:

1. Extract tags from the header, summary, details.
2. Filter out everything else from tags except date tags.
    a. If no tags are left after this, assign the current date as focus time.
3. Remove duplicates from date tags.
4. Assign weights to the tags on the following basis:
    a. If tag belongs to header: $10 \ * Position$
    b. If tag belongs to summary: $5 \ * Position$
    c. If tag belongs to details: $2 \ * Position$

    Where position is the number of characters before the date tag appears in the text.
5. Sort the weighted date tags according to the weights.
6. Assign the highest weight date tag as the focus time.

A flowchart of the program can be seen in Figure 29.

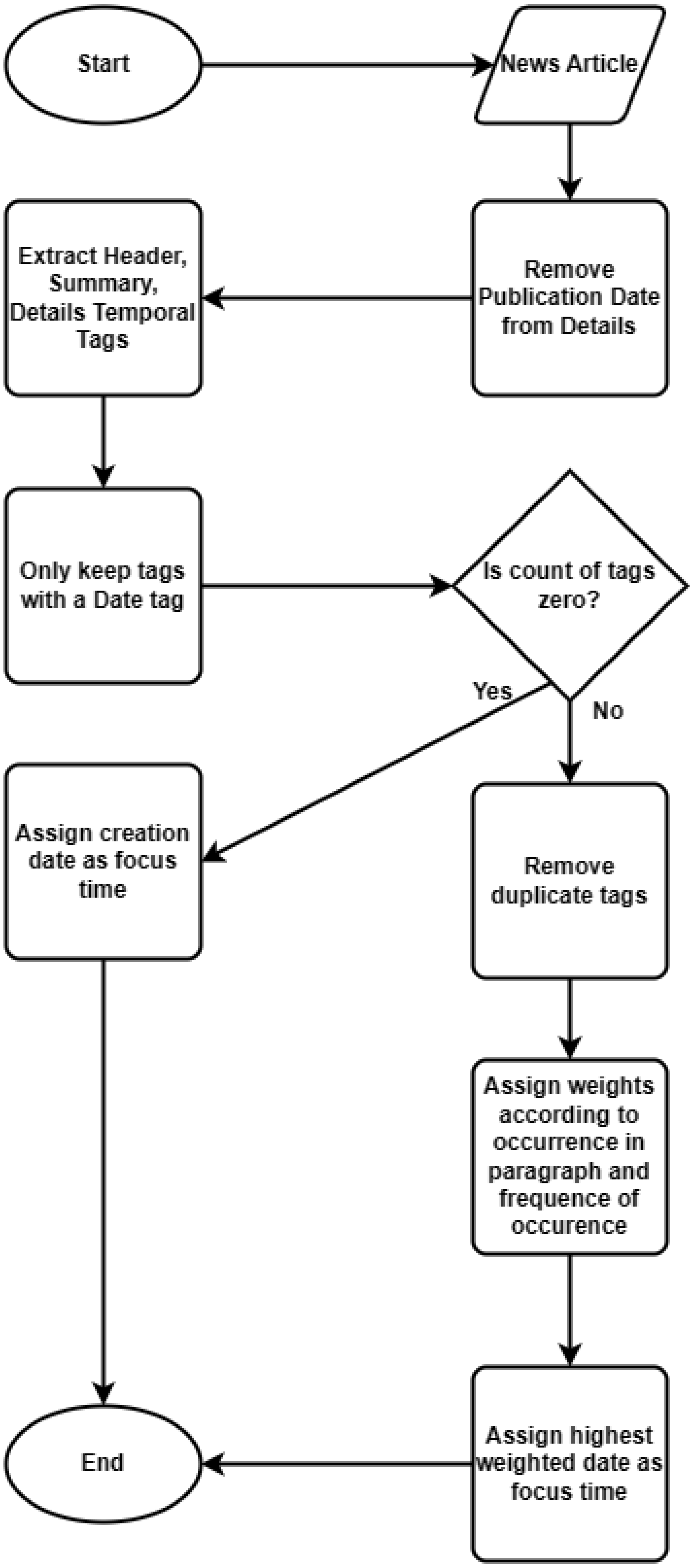

Start
News Article
Remove Publication Date from Details
Extract Header, Summary, Details Temporal Tags
Only keep tags with a Date tag
Is count of tags zero?
Yes
No
Assign creation date as focus time
Remove duplicate tags
Assign weights according to occurrence in paragraph and frequence of occurence
Assign highest weighted date as focus time
End

Figure 29: Temporal Specificity Module Flowchart

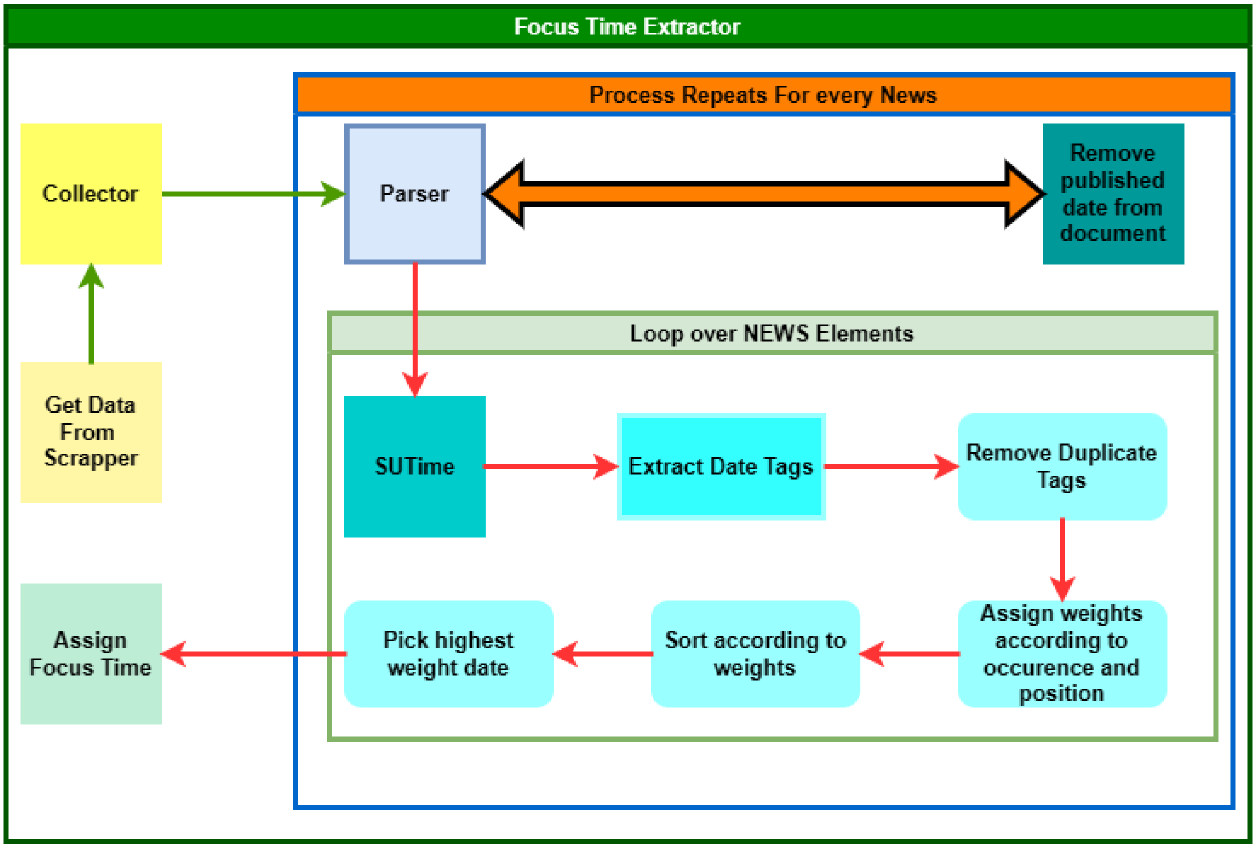


Figure 30: Focus Time Extractor Architecture

7.2.2 Geographical Specificity Module

We started facing some accuracy problems in the extraction of focus locations in our geographical specificity module. Hence, we used fuzzy matching which will help us in extracting the correct focus location by improving the matching of locations found on the news with our stored locations. This greatly increased the accuracy of our extracted focus locations.

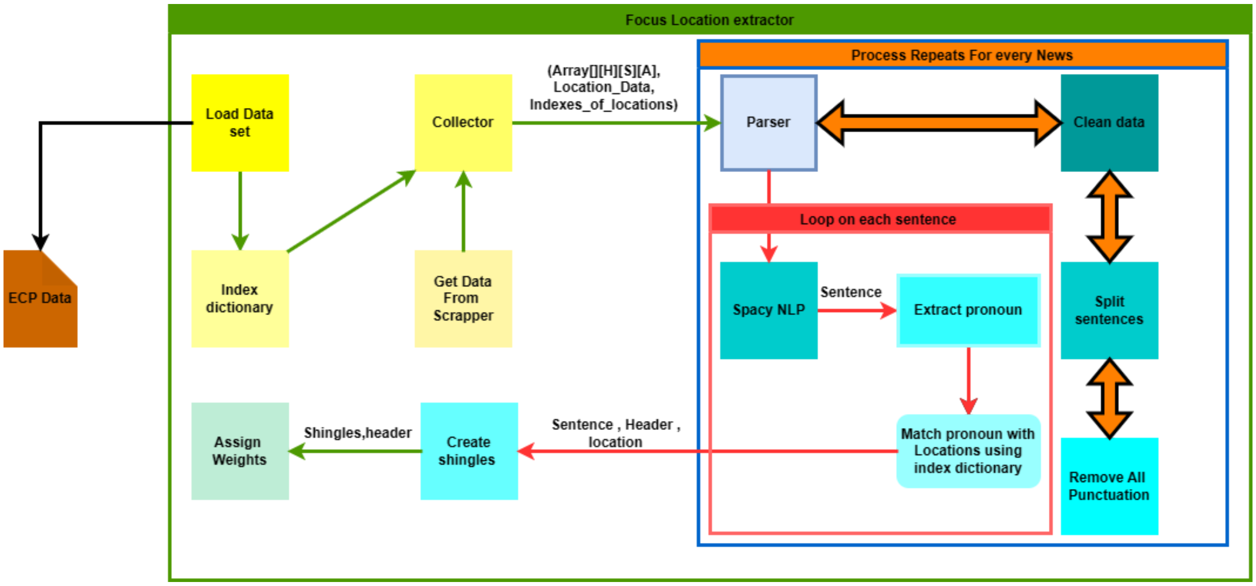


Figure 31: Focus Location Extractor Architecture

7.2.3 Database handling

As we had a special scenario with our database where we have to choose the location and find out the level of that city, i.e, if it belongs to Province, District, Tehsil, Union council, refer to 6.4 Entity Relationship Diagram for the ERD diagram, we had to traverse multiple tables and multiple levels of foreign keys were involved. This was catered using the following algorithm.

1. Search in the province table.
   a. If found, update the NEWS table with the other 3 entries as NULL.
2. Search in the district table.
   a. If found, look for the province the district belongs to and update the NEWS table with the other 2 entries as NULL.
3. Search in the tehsil table.
   a. If found, look for the district the tehsil belongs to. Then, search for the province the district belongs to.
   b. Update the NEWS table with the Union Council entry as NULL.
4. Search in the Union council table.
   a. If found, look for the tehsil the union council belongs to. Then, search for the district the tehsil belongs to and consequently also find the province as well.
   b. Update the NEWS table.

7.2.4 Apache Spark distributed processing

1. A cluster of Apache Spark was deployed using docker for large scale data processing.
2. PySpark module of python was used and it was connected to the spark container deployed.
3. A spark session was started with the help of PySpark and a master(driver) node was initialized.
4. The master node then distributed the processing to slave nodes with the help of cluster manager.
5. The slave nodes performed processing individually which helped in reduction of processing time.

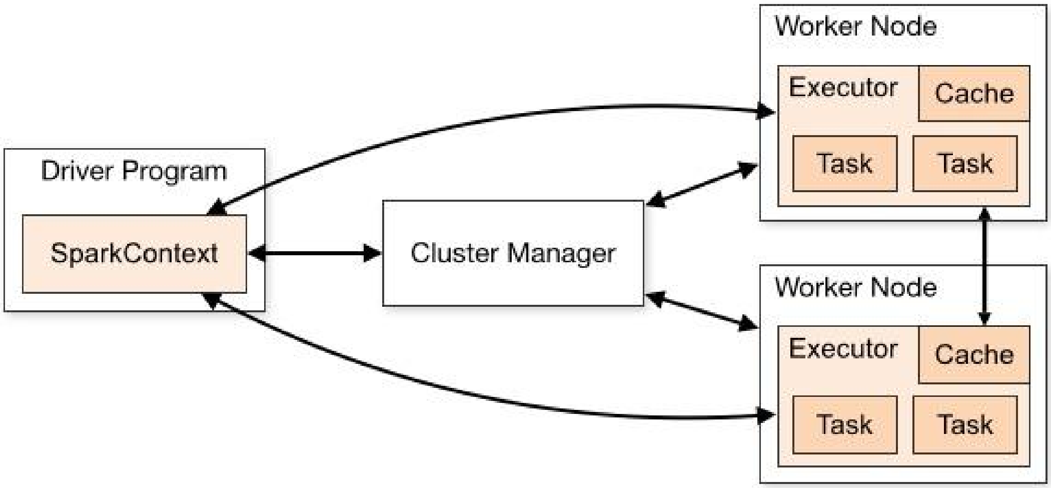


Figure 32: Spark Cluster Architecture

7.2.5 Application Programming Interface

We used GOLANG to make our api. Instead of using the simple router package, we used the gorilla mux package which will help us in setting up a mutual transport level security in the future if we ever want to make our application secure in the future. The main function of our api is to get the json responses from the web interface and then use these responses to generate SQL Queries for our database. These queries are then returned back to plot on the GIS.

References

CrimeMapping.com - Helping You Build a Safer Community, https://www.crimemapping.com/. Accessed 22 October 2022.

“Identification of Temporal Specificity and Focus Time Estimation in News Documents.” *Capital University of Science & Technology*, https://cust.edu.pk/static/uploads/2020/02/150_Shafiq-ur-Rehman-Khan-CS-2019-cust-isb-prr.pdf . Accessed 22 October 2022.

“Pakistan Union Council Boundaries along with other admin boundaries dataset.” *Humanitarian Data Exchange*, 28 August 2021, https://data.humdata.org/dataset/pakistan-union-council-boundaries-along-with-other-admin-boundaries-dataset. Accessed 23 October 2022.

“The.” *The Stanford Natural Language Processing Group*, https://nlp.stanford.edu/software/sutime.shtml. Accessed 22 October 2022.

“TimeTrails: A System for Exploring Spatio-Temporal Information in Documents.” *Database Systems Research Group*, https://dbs.ifi.uni-heidelberg.de/files/Team/jannik/vldb_poster.pdf. Accessed 22 October 2022.