HIGH LEVEL DESIGN DOCUMENT

Performance Evaluation of Real Time twitter data for Business Analytics using cloud platform

UE18CS390A – Capstone Project Phase – 1

Submitted by:

Gagan Mahesh	**PES1201800793**
Akash K	**PES1201801760**
TS Yogesh	**PES1201801723**
Abhijit Mohanty	**PES1201801293**

Under the guidance of

Prof. Silviya Nancy J
Professor
PES University

January - May 2021

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)

TABLE OF CONTENTS

Note:

Section – 1 & Section 2	**Common for Product Based and Research Projects**
Section 3 to Section 11	**High-Level Design for Product Based Projects.**
Section 12	**High-Level Design for Research Projects.**
Appendix	**Provide details appropriately**

1. Introduction

The high level Design Document highlights the concept and necessary detail for implementing Real Time business analytics models using cloud platforms. A combination of use case, activity and sequence diagrams provides a detailed idea behind the product. These diagrams show the relationship between the users and the admins and the cloud platform(system) and the interaction between them. Using these diagrams it's easier to come up with implementation.

2. Current System

The current system presently does not provide real time analysis based on the twitter platform. The current analysis system first creates a dataset and then performs analytics whereas our proposed system will fetch the tweets real time and provide the respective output to the client real time itself. Hence removing the need to store the data in a dataset - optimizing storage.

3. Design Considerations

3.1. Design Goals

- Goal:To process summarise tweets with minimal user interaction and simple User Interface.
- Provide multiple tweets processing features via an API
- Guidelines:
- All UI components will interact with the MI model through rest APIs.
- Purpose of Usage:
- The proposed system suppresses the need of storing the tweets in datasets, uses cloud platforms for intensive computations and hosting, hence utilizing maximum amount of resources for the application.
- The use of rest APIs and the use of AWS cloud platform automatically enhances security, speed of the system. The privacy of the users is protected through the use of the twitter APIs.

3.2. Architecture Choices

Cloud Architecture:

Serverless Machine Learning Architecture on Cloud Platform

We leverage the use of various services provided by the cloud provider to satisfy the requirements of our project.

General Cloud Architecture

- Create a cloud function -API
- The client sends request (with a query string) to the hosted API endpoint
- The cloud function trigger performs tasks such as:
  ->Runs predictions using the trained deployed Machine Learning Algorithms
  ->Send Response to the user in a JSON format with the result of predictions

Serverless Technology and Event Based Triggering

->We use the concept of serverless Technology -that is the server is a distant concept and is invisible to customers

->Actions are performed by triggering the function by events

->Functions run short-lived tasks

Business Analytics(ML) Architecture:

Summarisation

- The alternate choice we had for our model was the traditional Bi-Directional Encoder-Decoder architecture.This architecture was outdated and the accuracy was quite saturated. But with additional techniques such as a pointer generator and converge mechanism did give accuracy close to SOA, but this involves complex code and a lengthy tuning process.

  Pros:

  ->Ability to train an end-end model on source and target output and generate variable length output for variable length input.

  Cons:

  -->Not accurate for longer input sequences. Involves lengthy code and difficulty in fine tuning.
- The current model chosen is Huggingface Transformers.Transformers is based on the same encoder-decoder architecture and attention mechanism.There are different transformers available for different NLP tasks .The T5 model is chosen for summarisation.The model has pre-trained weights.So fine-tuning a transformer model is an easy process where we chose the respective tokenizer for the transformer and give the processed data as input.

  Pros:

  ->The model architecture is already built and has trained weights.

  -->Transformers unlike other models are not sequential and unlike other recurrent neural network architecture,so they can be parallelized and training bigger models takes comparatively shorter time.

  -->Transformers moreover don't use RNN and use majorly attention mechanism.

-->Transformers are the latest SOA architecture currently in NLP and are used by various tech giants like google,Apple etc.

Cons:

-->Further research is going on for improving the accuracy of Abstractive Summarisation.

3.3. Constraints, Assumptions and Dependencies

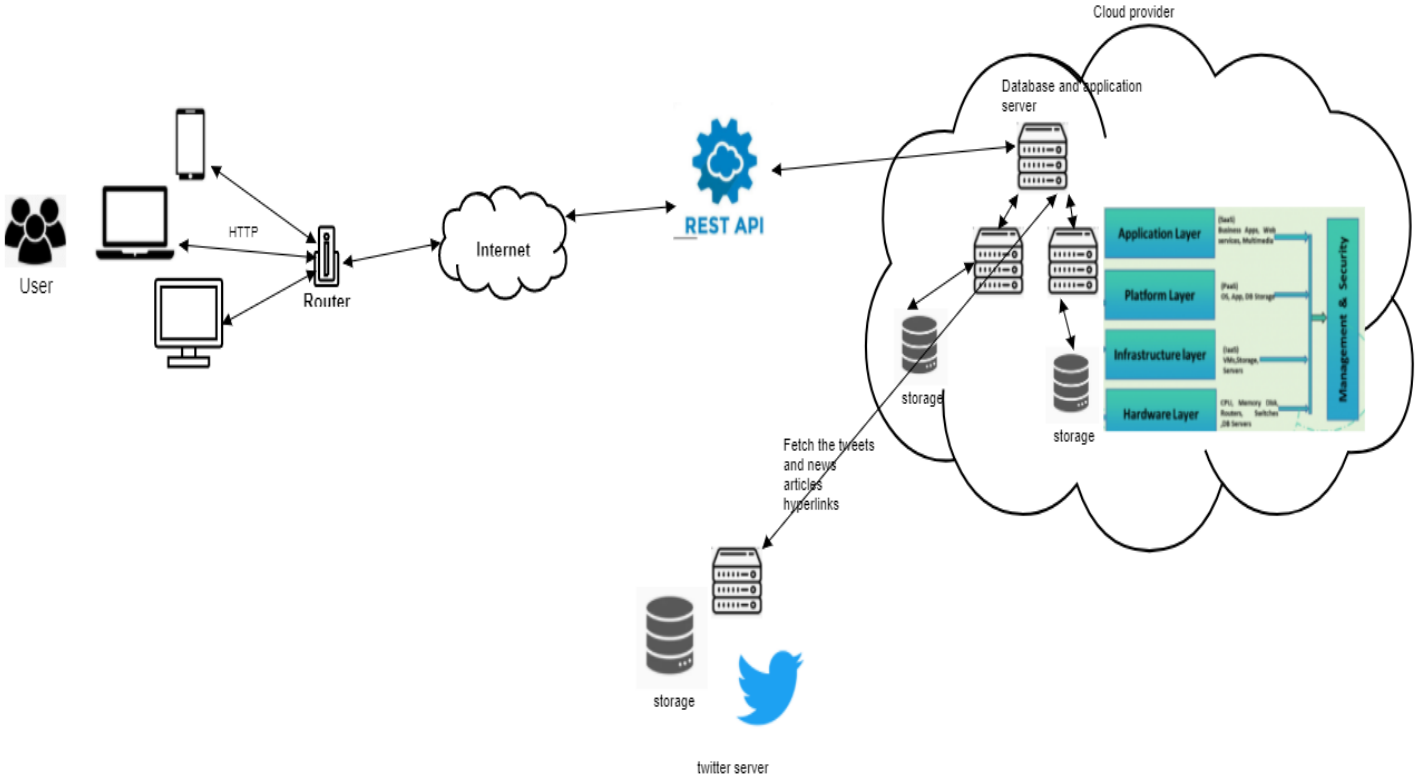
- Interoperability requirements
  - Consistent internet connection is required.
  - Latest web browser should be installed.
  - Since the system is cloud hosted, there are minimal issues regarding interoperability of the product.
- Interface/protocol requirements
  - The system requires http protocol since it's a client-server cloud based system.
- Data repository and distribution requirements
  - The data requirement is that the allowed language of the tweets and articles are restricted to English only.
  - News related tweets should have a hyperlink to the original article.
  - Training dataset is extracted from the CNN repository.
- Discuss the performance related issues as relevant.
  - As size of data increases, and more operations are performed parallelly the number of GPU's and CPU's will need to be increased
- End-user environment.
  - Consistent internet connection.
  - Latest web browser installed.
- Availability of Resources.
  - AWS platform takes care of the availability of resources through its fault detection and prevention features.
- Hardware or software environment
  - Dedicated GPUs to accelerate the training time and performance of the analytics models.
- Discuss issues related to deployment in target environment, maintainability, scalability, availability, etc.
  - Since the system will be cloud assisted there is no issue of maintainability, scalability and availability as these are provided by

the cloud vendor. The availability issue is handled by different data centers, the scalability issue is handled by distributed computing.

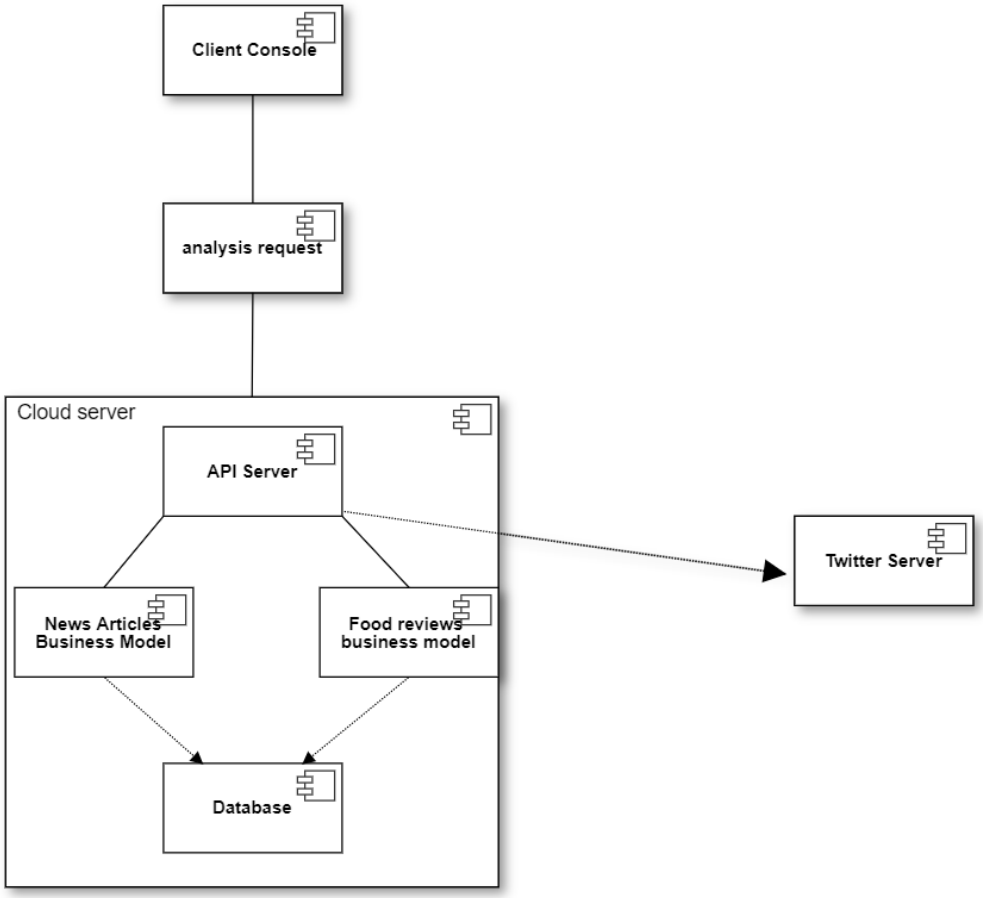
- Any other requirements described in the Requirements Document.]
    - Software Requirements
        - Tweepy,
        - Python
        - Rest-API
        - Flask
        - Keras,
        - Postman
        - HTML
        - ReactJS
- Assumptions:
    - The user should use english language only.
    - The user should provide meaningful hashtags.

4. High Level System Design

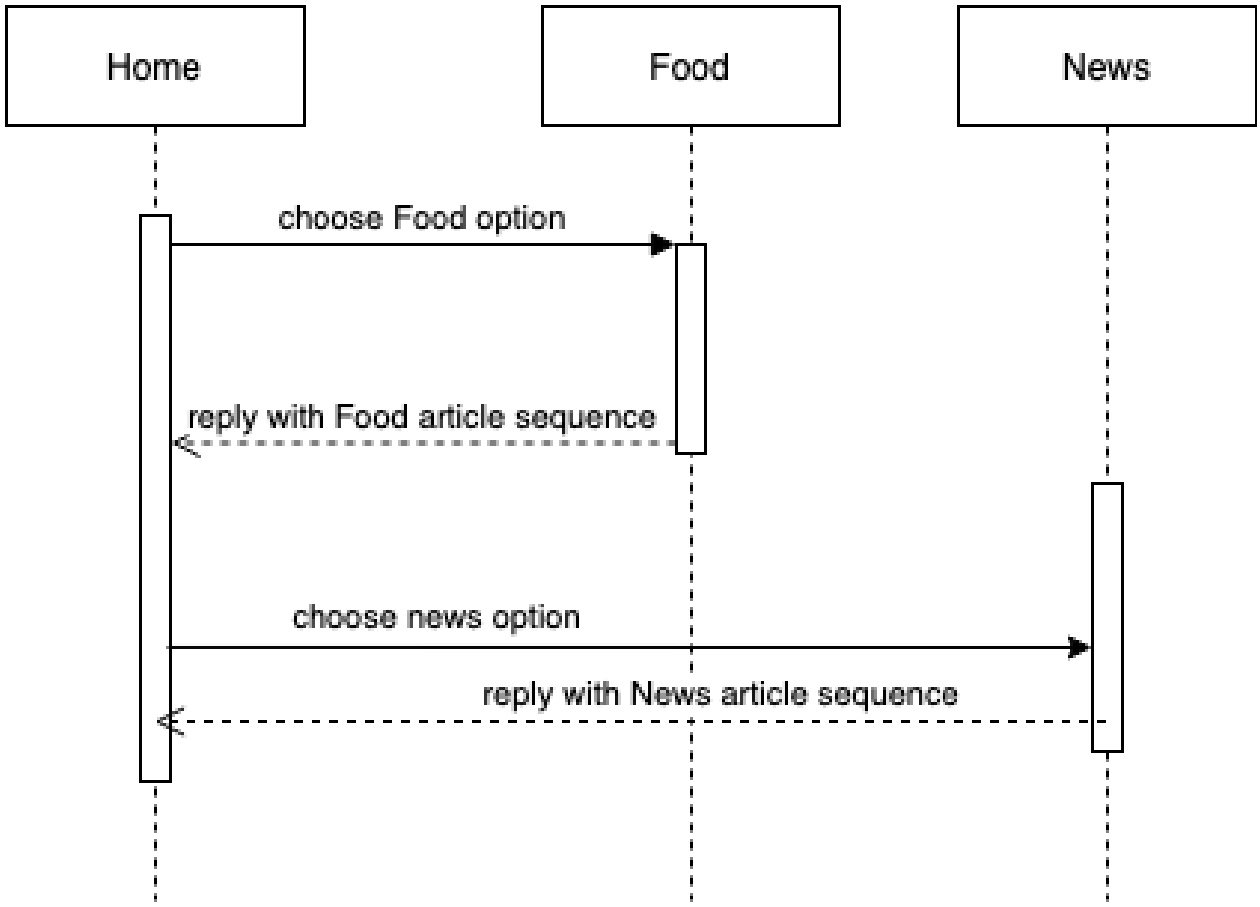
SYSTEM DESIGN:-



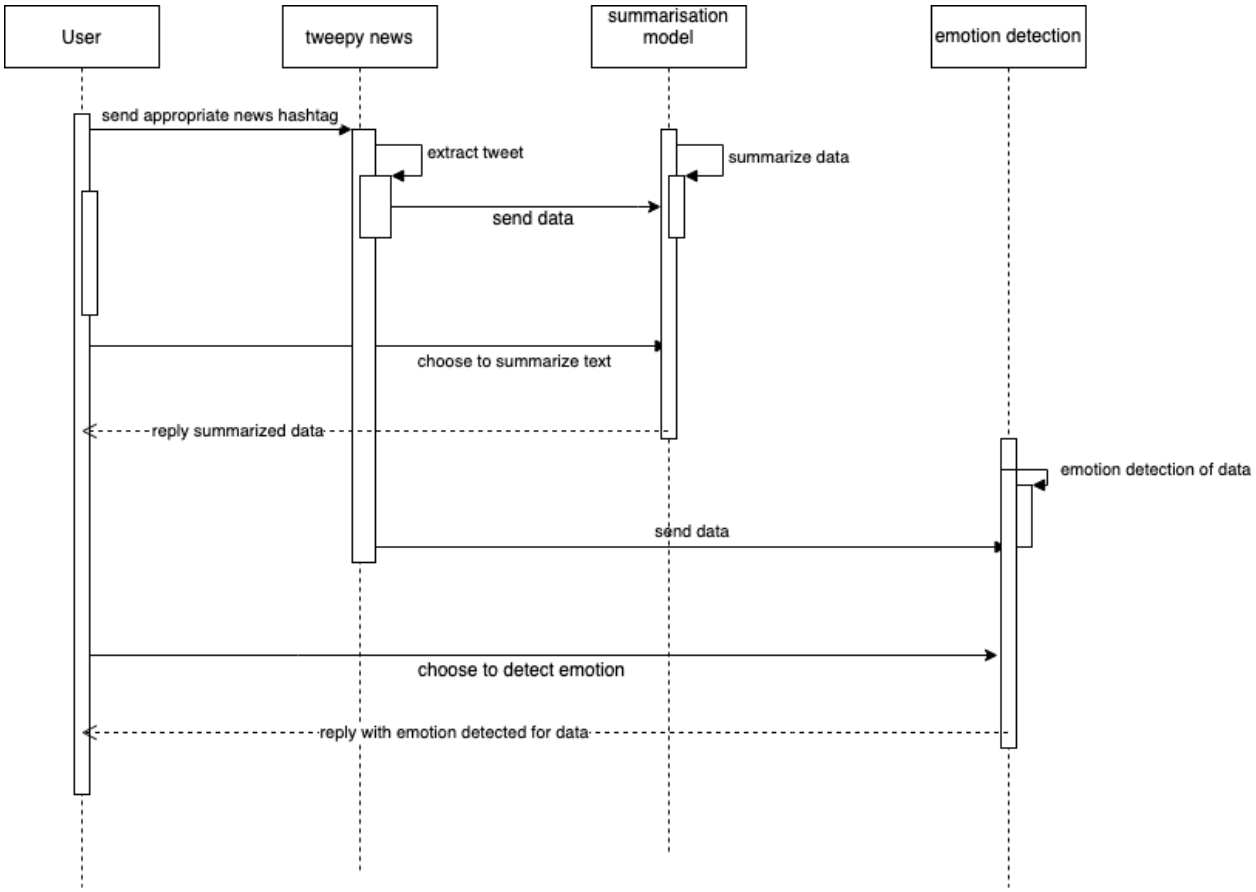
COMPONENT DIAGRAM:-



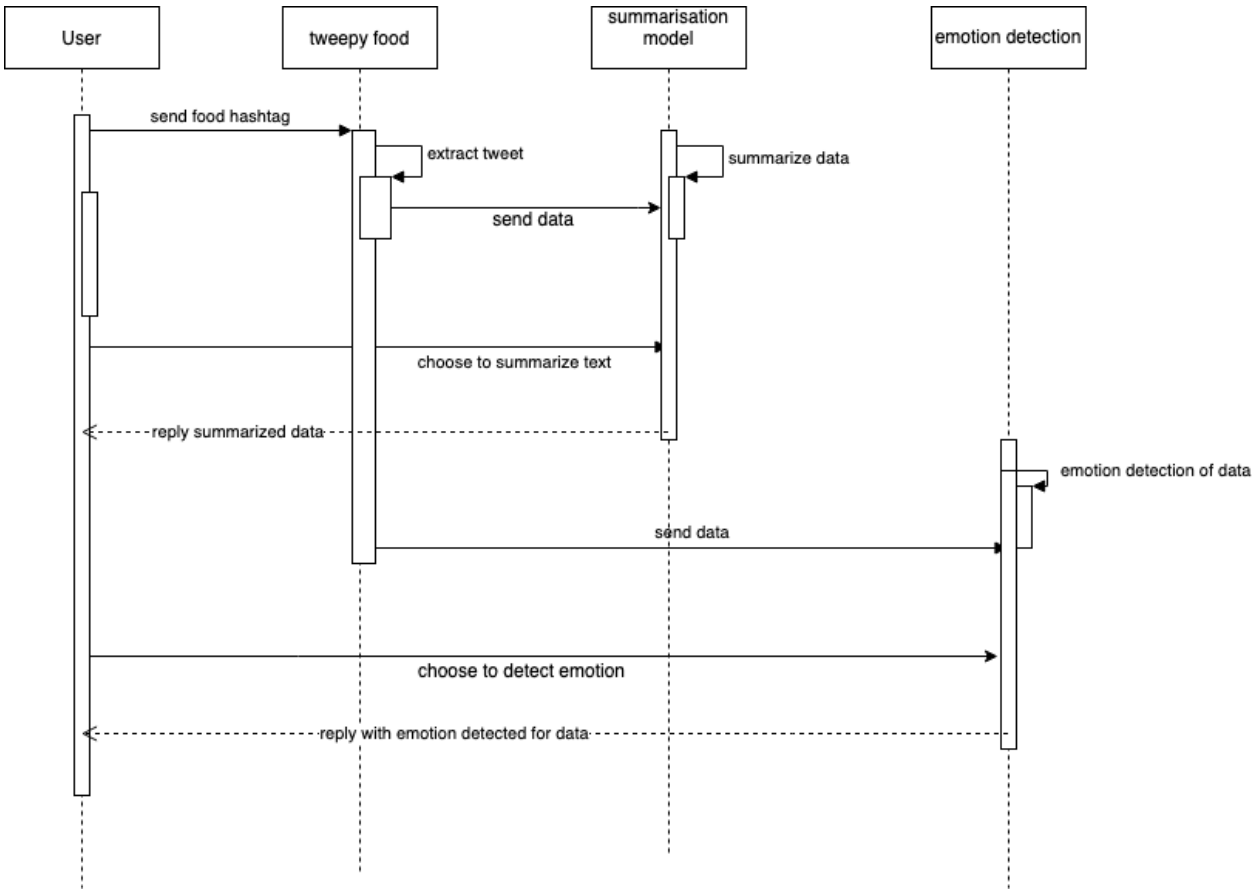
HOME PAGE SEQUENCE DIAGRAM:-



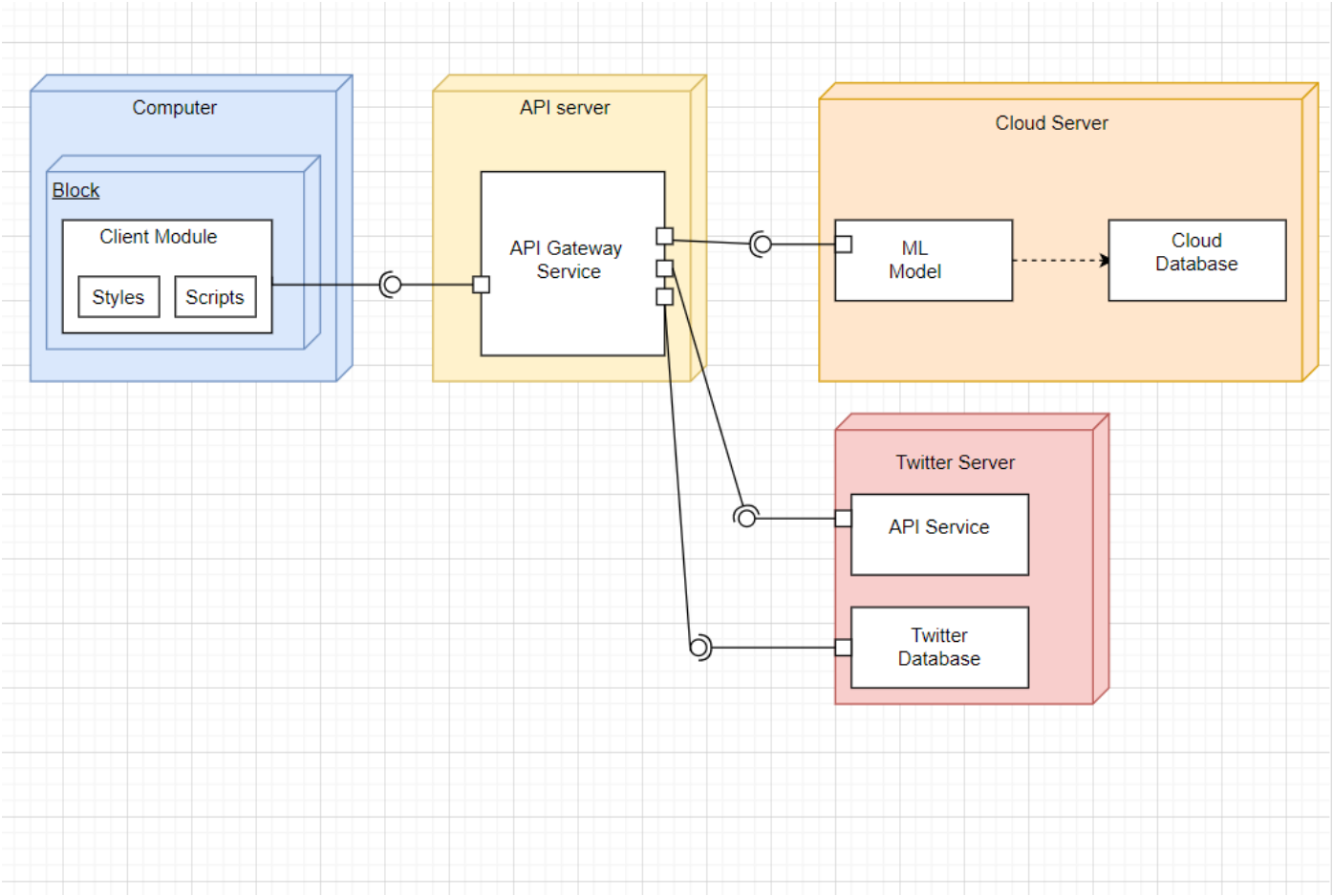
NEWS ARTICLE SEQUENCE DIAGRAM:-



FOOD ARTICLE SEQUENCE DIAGRAM:-



Deployment Diagram:-



1. Module –

   Code management is done using github

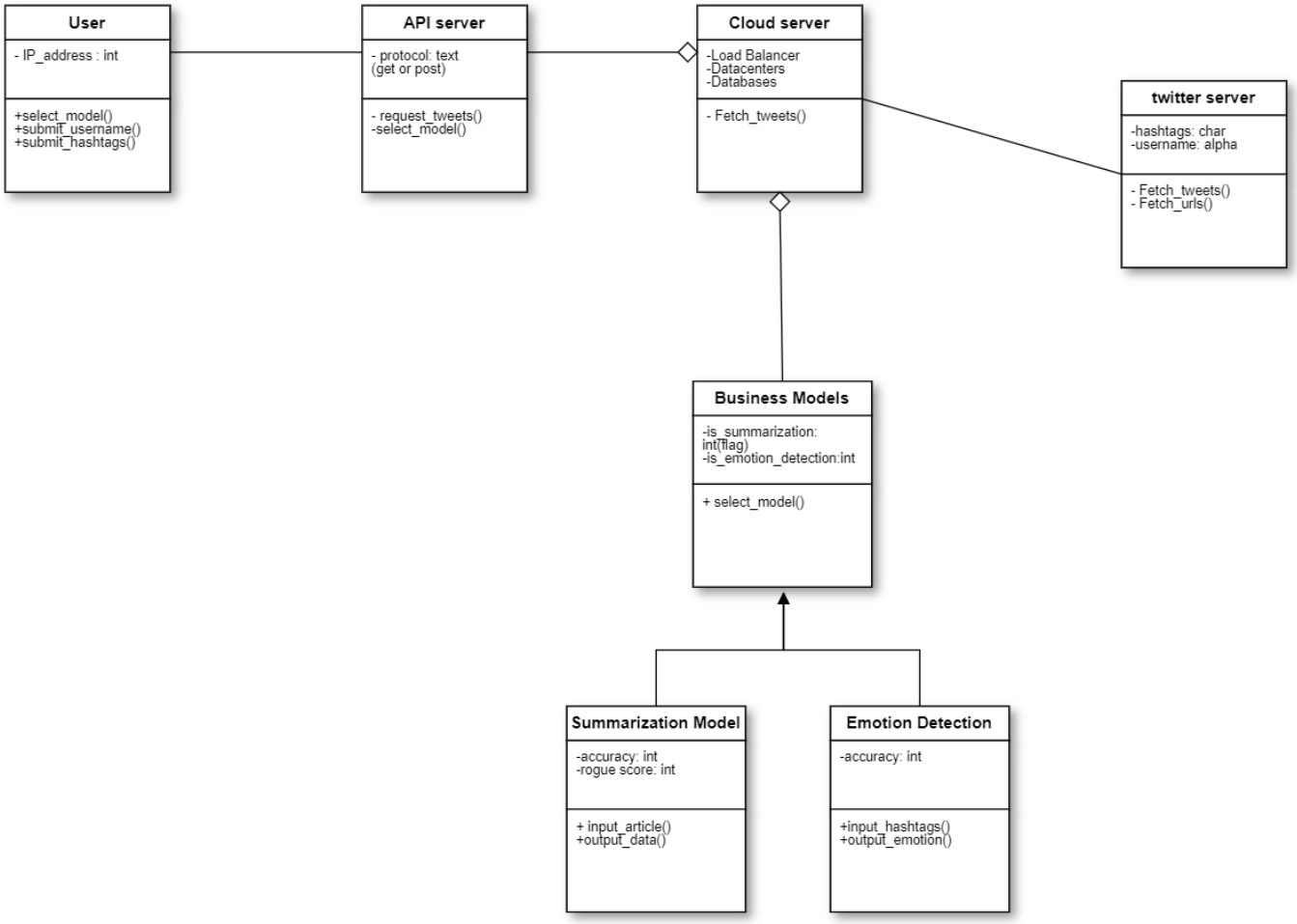
   - Multiple branches
     - Master
     - ML
       - .ipynb files
       - README file
     - tweets
       - tweets.csv(for test dataset)
       - .py files(all the codes for fetching the tweets based on username and hashtags)

   All other files will be stored in database provided by cloud provider(s3 in case of Amazon and google drive in case of Google)

2. Security – Describe the security features of the system.
    a. privacy of the user is protected by the use of twitter APIs, and also by the prevention of storage of user data in any database.
    b. the resources being used for the product are also fault tolerant and secured due to the usage of AWS cloud provider.

5. Design Description

5.1. Master Class Diagram

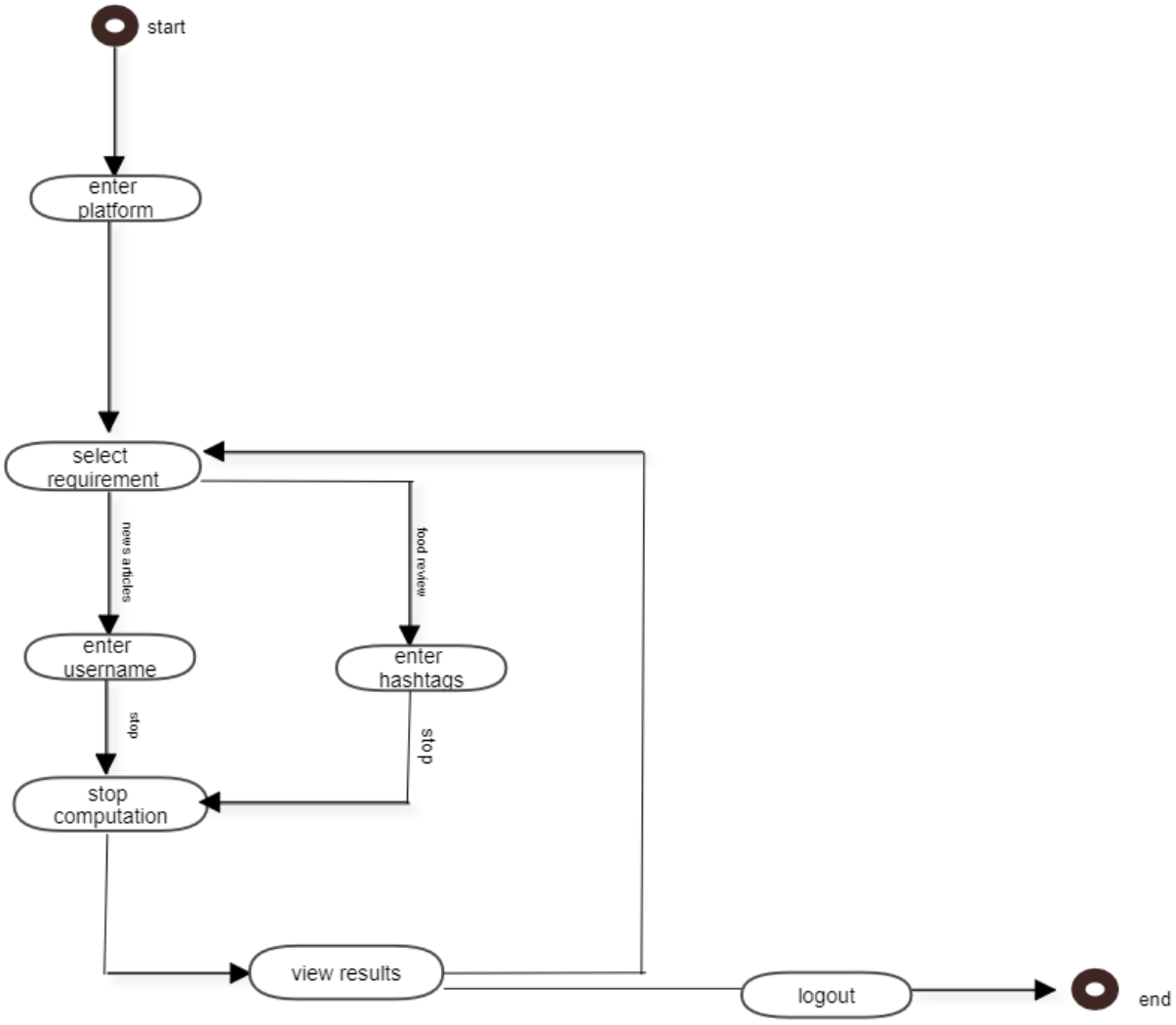


5.2. Reusability Considerations

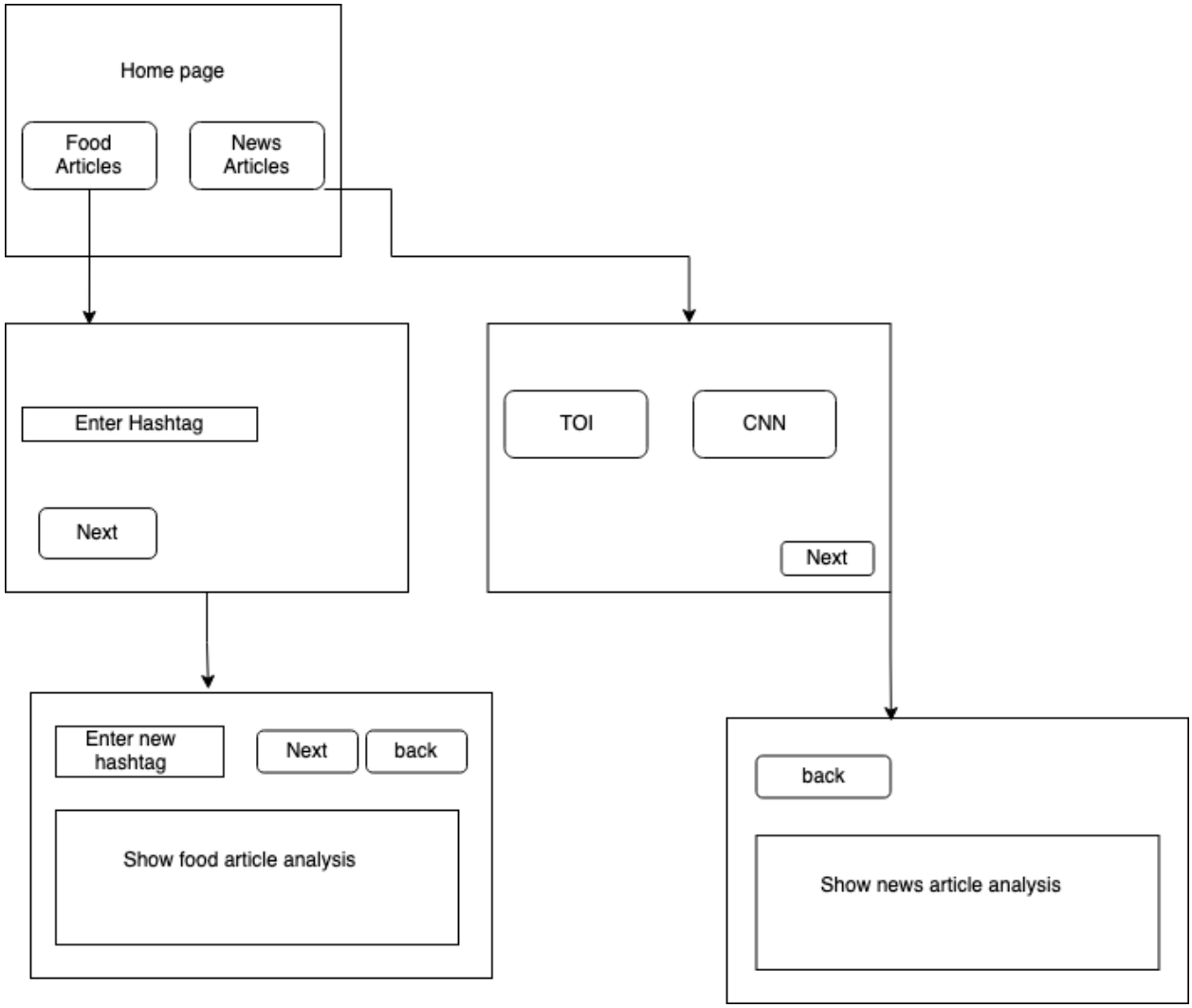
The ML models used for summarization and emotion detection are reused and some extra layers(mechanism) have been added on top of that.

The .py file which is fetching the tweets will be reused as many times the user requests.

6. State Diagram (include as appropriate)



7. User Interface Diagrams



8. Report Layouts

1)Our product uses REST APIs for all interactions between the client and the analysis models being hosted on the cloud platform. Various protocols in REST architecture such as GET, PUT, DELETE, POST will be used for different kinds of interactions between the two.

2)An API documentation (generated by POSTMAN after testing the entire system) will be provided to the client. An interactive website will also be built to demonstrate the use of these APIs.

3)For news article analysis, since the entirety of the article has to be considered, it would be inefficient to continuously run the real time tweets extraction system (for extracting the hyperlinks present in the tweet, linking it to the actual news articles). Hence the system will limit the tweet (or hyperlink extraction in this case) to at most 10 items (or hyperlink extractions).
4)For food review analysis, the tweets generally posted by users will be brief (in terms of length of the tweet), hence extracting the tweets in real time doesn't generally need a set threshold. Hence, the limit for the data being extracted will be relaxed as compared to news article analysis.
5)All the analysis models will be hosted on the cloud, with the dataset being hosted in the storage services being provided by the cloud provider. Preferably, AWS services could be used for this purpose.

9. External Interfaces

User Interface

- Home page:- This page consists of two buttons which leads to two difference sections - news analysis and food article analysis.
- News article:- This section corresponds to news article analysis. The user is given an option to choose any major news cooperation among the given options. Based on the option chosen, the user will be given options to choose a trending topic. The trending topic chosen will be sent to another webpage which in turn will show all the analysis done on that webpage.
- Food article:- This section corresponds to food article analysis. The user will type in a hashtag in a given textbox. This data will be used to get all the trending tweets based on the hashtag. The articles and user reviews got from this data will then be used to perform analysis which will then be displayed in another webpage.

Software interface

- REST APIs will be developed using Flask and python. The API will then be used in the frontend components.
- JSON format will be used to transfer messages between frontend and backend.
- POSTMAN will be used for API testing.
- An API document will be generated by using the above software.

The communication between backend and frontend will take place with the help of REST APIs. The following REST APIs are designed.

GET /food/<food_hashtag>

This is used to access specific pages

The response is the entire page content serialised in a JSON if it exists, or a 404 error if it does not exist

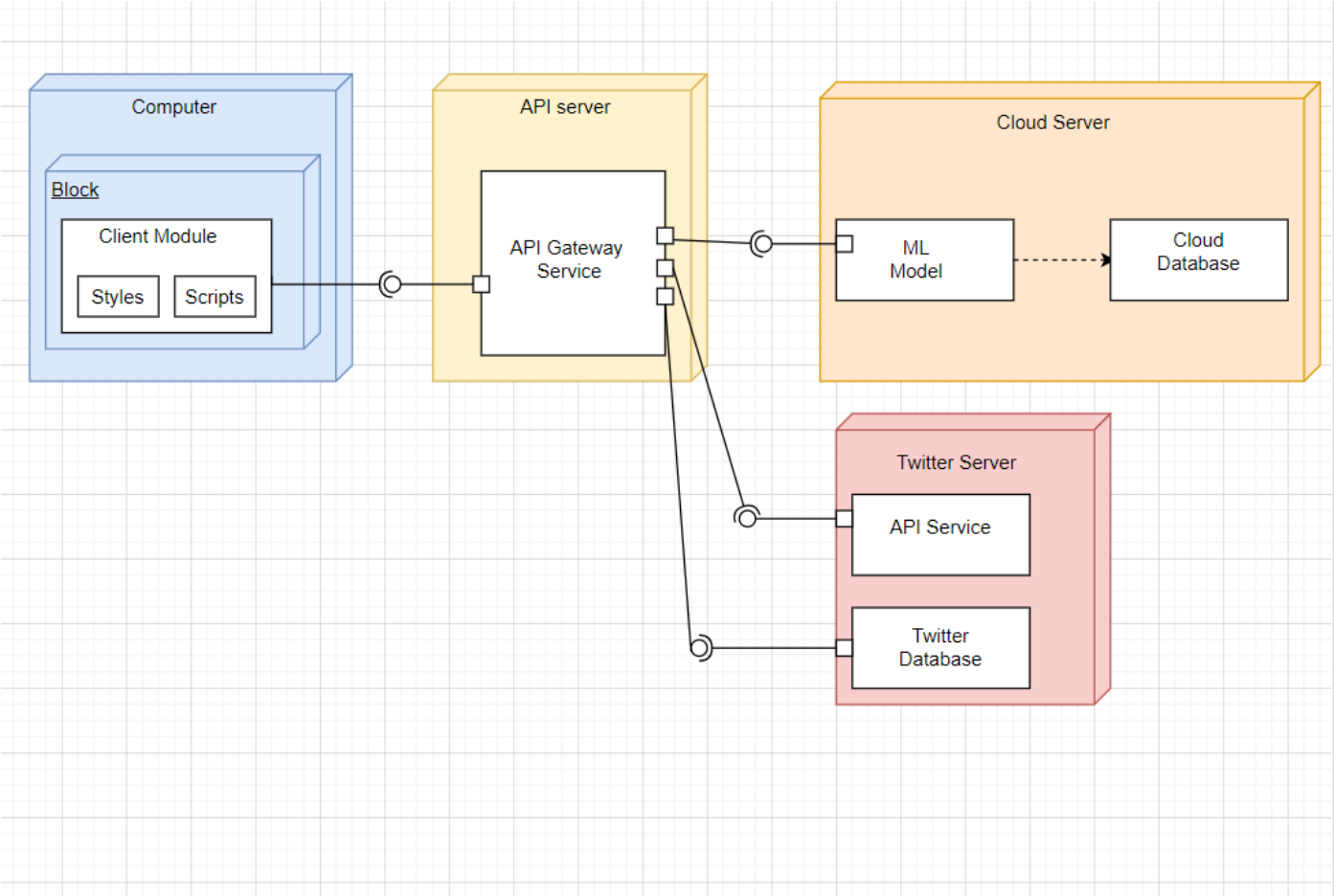
The structure of the JSON is as follows:

```
{
        "hashtags": "given hashtag",
        "contents": [ response ], ...
}
```

Communication Interfaces

- The client software will run best on Google Chrome 10 and above, Mozilla Firefox 4 and above and Internet Explorer 8 and above.
- The software product can be accessed from any device such as PC, laptop and phone.
- Communication Protocols:
- HTTP
- HTTPS (for security measures)
- The images which will be stored in the database will be base-64 encoding.
- The necessary encryption will be provided by HTTPS.
- The software does not require high data transfer rate and can be used even on slower internet speeds.
- Synchronisation is not required since there is no real time communication.
- Messages between the frontend and the backend are communicated through REST APIs and are described in the previous section.

10. Packaging and Deployment Diagram



11. Help

Proper documented API documentation will be provided to the user. This will be generated using the POSTMAN software.

e,g www/websitename?username=TOI

All the readme files will be provided to the user as to how to enter the username and hashtags

Sample web page will be hosted to demonstrate the usage of API which can be used by different clients.

12. Design Details

1.1. Novelty

- Using the SOA different transformer architectures model for Summarisation and LSTM architecture for emotion detection

1.2. Innovativeness

- Numerous different transformers will be used for training on our Dataset ,which are built on different architectures and choose the one with highest ROUGE score and not just hyperparameter tuning

1.3. Interoperability

- The use of transfer Learning concept makes it easy to fine-tune and reuse the model on different datasets.

1.4. Performance

- Making the use of CUDA cores and tensor core equipped GPU to fasten the training process of the Neural Network.

1.5. Security

- The use of rest APIs and the use of AWS cloud platform automatically enhances security, speed of the system. The privacy of the users is protected through the use of the twitter APIs.

1.6. Reliability

- Highly accurate result as we aim to have the ROUGE score for summarisation above 40 and good F1 score for emotion detection.Moreover the use of AWS cloud platform ensures reliability to the users which is provided by the vendor

1.7. Maintainability

- The product doesn't require any maintenance from the client side,constant updation of the model on new dataset and hyperparameter tuning is done by the backend team.

1.8. Portability

- Since the trained model is hosted as an API on cloud ,it can be accessed anywhere easily via different thin and flat clients

1.9. Legacy to modernization

1.10. Reusability

- Certain features of the product such as emotion detection can be used for different custom input ,via the user interface.

1.11. Application compatibility

- Since the model is hosted on cloud the application does not require any additional software installation ,it requires only a compatible browser.

1.12. Resource utilization, Etc.,

- The model training uses extensive GPU resources for faster training which is a one time process.Once the training is done ,hosting the product uses minimal computational resources.

Appendix A: Definitions, Acronyms and Abbreviations

[Provide definition of all terms, acronyms and abbreviations required for interpreting this High Level Design document.]

JSON =JavaScript Object notation is a format for storing and transporting data in the form of key-value pairs

HTTP =Hypertext Transfer Protocol is an application-layer protocol use for transmitting hypermedia documents .

REST API =Representational state transfer is a software architectural style which uses a subset of HTTP which is used in interactive applications that use web service.

Neural Network=Neural Network is a machine learning algorithm which learns to recognize the underlying pattern in a data through a process that mimics the way the brain works.

LSTM=Long Short-Term Memory is an artificial recurrent neural network architecture used in deep learning which uses feedback connections.

GPU=Graphics Processing Unit is specialized electronic circuit used for gaming and faster matrix multiplication for Machine Learning Models

F1 score=A measure of model's accuracy on a dataset which combines precision and recall model.

ROUGE score=(Recall-Oriented Understudy for Gisting Evaluation) is an metric which calculates the similarity between a candidate document and a collection of reference documents.

Transformers=The Transformer in NLP is a novel architecture that aims to solve sequence-to-sequence tasks while handling long-range dependencies.

Appendix B: References

- A. S. Halibas, A. S. Shaffi and M. A. K. V. Mohamed, "Application of text classification and clustering of Twitter data for business analytics," 2018 Majan International Conference (MIC), Muscat, Oman, 2018, pp. 1-7, doi: 10.1109/MINTC.2018.8363162.
- [1] Avudaiappan.T, Jenifer, Sisay tumsa, Subashrree T.Jayansankar "Twitter sentiment analysis using neural network"
- [2] Carlo Aliprandi, Federico Neri, Federico Capeci ,et.al, "Sentiment Analysis on Social Media." IEEE Computer Society Washington, 2012
- [3]Chandu Parmar,Ranjan Chaubey ,Kirtan Bhatt,Reena Lokare "Abstractive Text Summarization using Artificial Intelligence" 2019
- [4]Minh-Thang Luong, Hieu Pham, Christopher D. Manning "Effective Approaches to Attention-based Neural Machine Translation" 2015
- [5]Ankush Chatterjee, Kedhar Nath Narahari, Meghana Joshi and Puneet Agrawal "EmoSense at SemEval-2019 Task 3: Bidirectional LSTM Network for Contextual Emotion Detection in Textual Conversations" 2019
- [6] Ashish Vaswani,Noam Shazeer,Niki Parmar,Jakob Uszkoreit,Llion Jones,Aidan N. Gomez,Łukasz Kaiser,Illia Polosukhin "Attention is All you Need" 2017
- https://ai.googleblog.com/2020/02/exploring-transfer-learning-with-t5.html

https://towardsdatascience.com/serverless-machine-learning-architecture-on-leading-cloud-platforms-c630dee8df15

https://creately.com/blog/diagrams/component-diagram-tutorial/

https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/

Appendix C: Record of Change History

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

#	Date	Document Version No.	Change Description	Reason for Change
1.				
2.				
3.				

Appendix D: Traceability Matrix

[Demonstrate the forward and backward traceability of the system to the functional and non-functional requirements documented in the Requirements Document.]

Project Requirement Specification Reference Section No. and Name.	DESIGN / HLD Reference Section No. and Name.
2.3 General Constraints, Assumptions and Dependencies	3.3Constraints, Assumptions and Dependencies
4.1 User Interfaces	7. User Interface Diagrams
4.3	