

MARIAM - A Chatbot for Farmers

Software Engineering Mini Project

Submitted by:

Samagra Singh 9919103165

Aviral Agarwal 9919103144

Mansi Joshi 9919103146

Vedant Agarwal 9919103135



Department of CSE/IT

Jaypee Institute of Information Technology University, Noida

ACKNOWLEDGEMENT

We would like to place on record my deep sense of gratitude to **Dr Ashish Kumar**, Assistant Professor, Jaypee Institute of Information Technology, for his generous guidance, help and useful suggestions.

I also wish to extend my thanks to our classmates for their insightful comments and constructive suggestions to improve the quality of this project work.

ABSTRACT

Deep learning methods possess many processing layers to understand the stratified representation of data and have achieved state-of-art results in several domains. Recently, deep learning model designs and architectures have unfolded in the context of Natural Language Processing (NLP). This has had a lot of applications in the real world ranging from speech bots replying to the user's general questions like Siri to computer generating live image captions by looking at the actions and stuff that is going on in that image. We have come really far but there is still a lot more to go.

We hereby, are putting forward our approach to provide trusted solutions to problems faced by farmers. We have shortened the scope of the project as the domain is very vast by limiting ourselves to general farmer queries and information.

Introduction

Most farmers in India are generally not scientifically educated regarding crops they grow and the practices they follow. Nowadays, whenever a farmer faces any problems regarding their produce, they trust their local sources or hearsay. These solutions are generally not very trusted and farmers incur loss on their produce. When the rural farmers lack access to knowledge and information that would help them achieve maximum agricultural yield, they are not only grope in the dark but are driven to the urban centers in search of formal employment, as the only option for survival.

The least expensive input for improved rural agricultural development is adequate access to knowledge and information in areas of new agricultural technologies. There have been short-comings of traditional print and library-based methods of providing such agricultural information to rural farmers who are generally illiterate and relatively remote from formal sources of information (e.g., extension stations, libraries). Farmers would benefit from global information, if information centers are cited in rural areas complete with all information and communication gadgets.

The problem that this project will be trying to solve is the same. Our NLP model shall take data related to agriculture from wikipedia and process it to generate proper response for farmer queries. The model shall be scalable and can also be used for application in other domains.

Software Requirements Specification

for

MARIAM: THE CHATBOT

Version 2.0

Samagra Singh 9919103165

Aviral Agarwal 9919103144

Mansi Joshi 9919103146

Vedant Agarwal 9919103135

Jaypee Institute of Information Technology- Sector 128

Table of Contents

Table of Contents ii Revision History ii

1. Introduction	1	1.1 Purpose	1	1.2 Document Conventions	1	1.3 Intended Audience and Reading Suggestions	1	1.4 Project Scope	1	1.5 References	1				
2. Overall Description	2	2.1 Product Perspective	2	2.2 Product Features	2	2.3 User Classes and Characteristics	2	2.4 Operating Environment	2	2.5 Design and Implementation Constraints	2	2.6 User Documentation	2	2.7 Assumptions and Dependencies	3
3. System Features	3	3.1 System Feature 1	3	3.2 System Feature 2 (and so on)	4										
4. External Interface Requirements	4	4.1 User Interfaces	4	4.2 Hardware Interfaces	4	4.3 Software Interfaces	4	4.4 Communications Interfaces	4						
5. Other Nonfunctional Requirements	5	5.1 Performance Requirements	5	5.2 Safety Requirements	5	5.3 Security Requirements	5	5.4 Software Quality Attributes	5						
6. Other Requirements	5	Appendix A: Glossary	5	Appendix B: Analysis Models	6	Appendix C: Issues List	6								

Revision History

Date Reason For Changes
Version

May 1, 2022 Initial Release Version 1.0

May 8, 2022 Grammatical Changes throughout the document

● Introduction

●.1 Purpose

Version 2.0

This is the Software Requirement Specifications (SRS) document for the Mariam ; Chatbot being created for the Minor Project of Jaypee Institute Of Information Technology. This SRS provides an overview of forward our approach to provide trusted solutions to problems faced by farmers. This document discusses the project's goals, parameters, target audience, user interface, requirements, design issues and users' view.

●.2 Document Conventions

All the headings are font size 18 , font type 'Times New Roman' and bold. Subheadings are font size 14, font type 'Times New Roman' and Bold. Normal Text is of font size 11 , font type 'Times New Roman'. Words that are in italic are given more emphasis as they contain any specific term that requires definition.

●.3 Intended Audience and Reading Suggestions

This SRS is intended to be read and understood by developers, project managers, users, testers and documentation writers. The SRS contains all the information of the chatbot starting from the project scope

, functional and non functional requirements, system features and ending with the future scope and

revision history. Developers are suggested to go through the system features for the better understanding of the project.

●.4 Project Scope

This project is a new development to create an interactive chatbot . The model shall take data related to agriculture from Wikipedia and process it to generate proper response for farmer queries. The model shall be scalable and can also be used for application in other domains.

● Overall Description

●.1 Product Perspective

We hereby, are putting forward our approach to provide trusted solutions to problems faced by farmers. We have shortened the scope of the project as the domain is very vast by limiting ourselves to general farmer queries and information.

●.2 Product Features

Most farmers in India are generally not scientifically educated regarding crops they grow and the practices they follow.

Nowadays, whenever a farmer faces any problems regarding their produce, they trust their local sources or hearsay. These solutions are generally not very trusted and farmers incur loss on their produce.

The problem that this project will be trying to solve is the same. The model shall take data related to agriculture from Wikipedia and process it to generate proper response for farmer queries. The model shall be scalable and can also be used for application in other domains.

•.3 User Classes and Characteristics

•.4 Operating Environment

The system is also designed to be user-friendly. The software will operate in a python environment(Python IDE) including any operating system(Windows/Mac/Linux).

•.5 Design and Implementation Constraints

Below are the hardware requirements of the system and software requirements of the system respectively.

HARDWARE REQUIREMENTS:

- Ram : 1 GB (Minimum)
- Hard Drive : 32 GB (Minimum free space : 1 GB)
- Processor : 1 GHz

SOFTWARE REQUIREMENTS :

1. Python IDE(VS code/Pycharm)
2. Operating system : WINDOWS /MAC / LINUX

•.6 Assumptions and Dependencies

we assume to implement all the features discussed in the scope of work and make the chatbot as accurate as possible. We want to continue working on this chatbot, develop it to the best of our abilities and then release it as an open source package so that other interested developers can participate in the development of this model.

● System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

●.1 System Feature 1

<Don't really say "System Feature 1." State the feature name in just a few words.>

3.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

3.1.1 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

3.1.2 Functional Requirements

● Communicates in Hindi:

- All communication with the user is done in Hindi
- Queries are provided to the chatbot in Hindi.
- Responses to the said queries are also provided back to the user in Hindi.

● Query responses:

- Capable of responding to all queries regarding farming and plant diseases

● Tokenization and Lemmatization:

- All the extracted data is tokenized and lemmatized for further processing

● Cosine Similarity:

- This similarity is used for searching the keyword in user query in the corpus in order to build a response

● Web Scraping:

- A web page is scrapped in order to make the corpus

●.1 System Feature 2 (and so on)

● External Interface Requirements

●.1 User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

●.2 Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

●.3 Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

●.4 Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

● Other Nonfunctional Requirements

●.1 Performance Requirements

The performance of our software is at its best when the following are regularly done

- Password Management
- Regular Database Archiving
- Virus Protection

Safety Requirements

- Any users who make use of the system need to hold a Login ID and password.

•.2 Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

•.3 Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

● Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: Issues List

< This is a dynamic list of the open requirements issues that remain to be resolved, including TBDs, pending decisions, information that is needed, conflicts awaiting resolution, and the like.>
. Normal Text is of font size 11 , font type ‘Times New Roman’.

Words that are in italic are given more emphasis as they contain any specific term that requires definition.

FUNCTIONAL REQUIREMENTS:

- **Communicates in hindi:**
 - All communication with the user is done in hindi
 - Queries are provided to the chatbot in hindi.

- Responses to the said queries are also provided back to the user in hindi.
- **Query responses:**
 - Capable of responding to all queries regarding farming and plant diseases
- **Tokenization and Lemmatization:**
 - All the extracted data is tokenized and lemmatized for further processing
- **Cosine Similarity:**
 - This similarity is used for searching the keyword in user query in the corpus in order to build a response
- **Web Scraping:**
 - A web page is scrapped in order to make the corpus

NON - FUNCTIONAL REQUIREMENTS:

- The bot should reply in under 30 seconds to users' queries.
- Talking to the bot should feel like talking to a person (the bot should not reply instantly, 1-2 second delay between responses).
- The users have to authenticate themselves before being able to query information from the bot.
- The user credentials should not be shared with anyone.

User story

- As a farmer I should be able to access an application that helps me answer all my questions regarding agriculture.
- As a farmer I want to be able to type in my query, in my native tongue , which happens to be Hindi, and get the response in Hindi as well.
- As a farmer I don't have access to laptops or personal computers, so I want the application to work on all mobile devices.¹

Link to Project: <https://github.com/atishaya01/mariamFarmerChatbot>

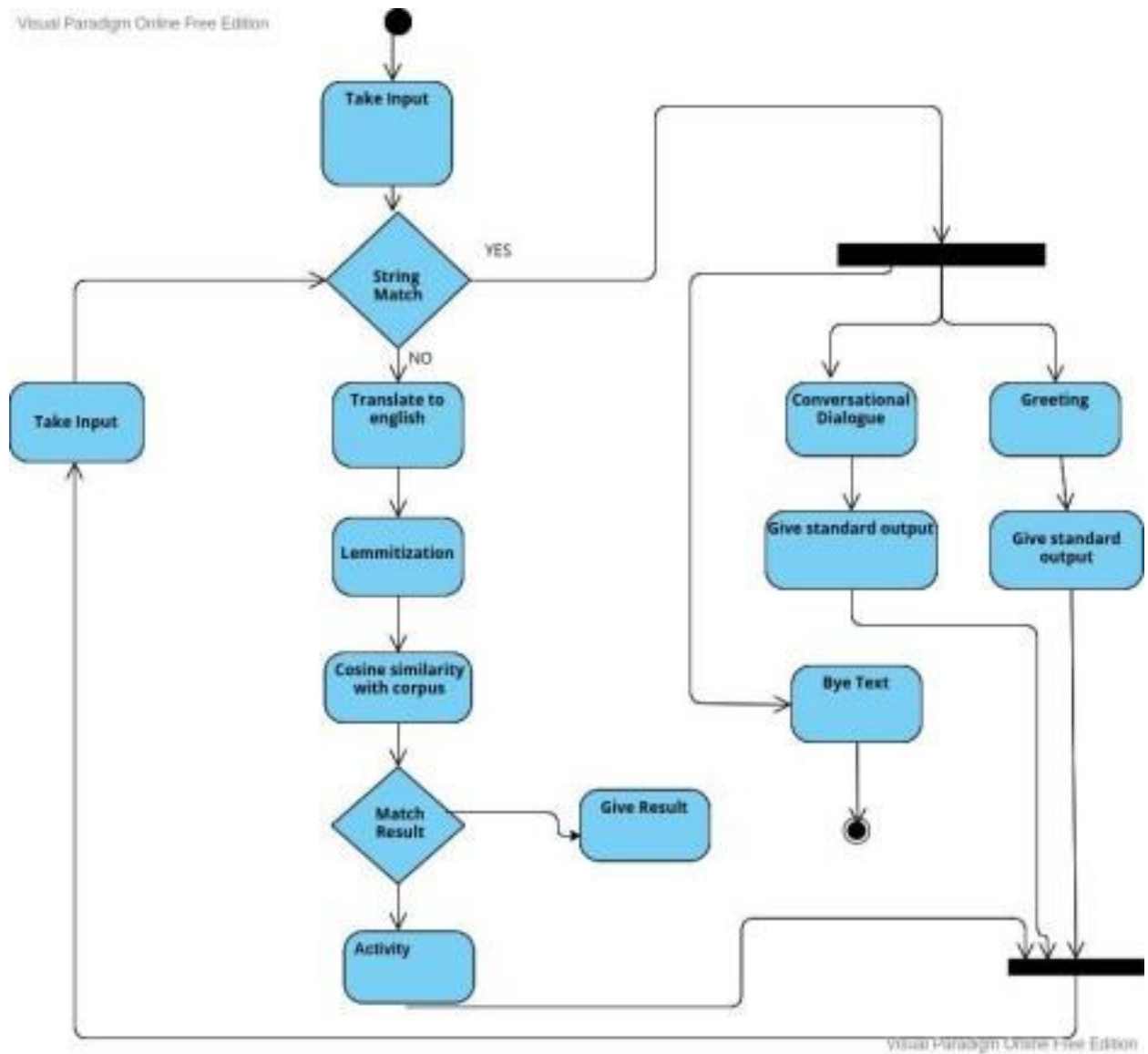
UML Diagrams

USE CASE DIAGRAM



Activity Diagram

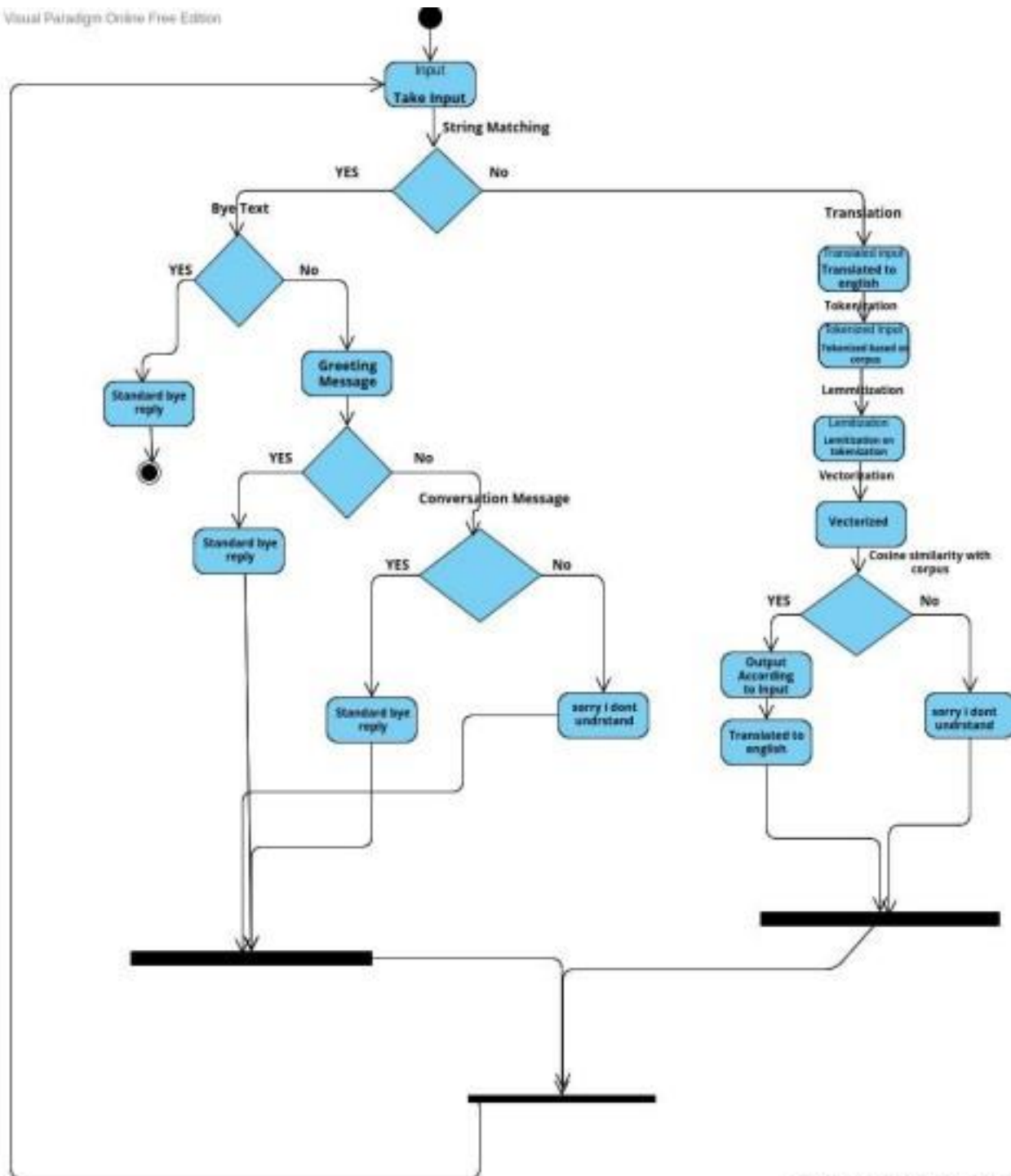
Visual Paradigm Online Free Edition



Visual Paradigm Online Free Edition

State Diagram

Visual Paradigm Online Free Edition



Visual Paradigm Online Free Edition

Code Refactoring :

(i) Rename Variable:

Renamed an ambiguous variable to give it a more meaningful name:

```
tth = translator_hindi(human_text)
```

to

```
translated_human_text=translator_hindi(human_text)
```

Code Screenshots :

▸ Responding to User Queries

the response will be generated based upon the cosine similarity of the vectorized form of the input sentence and the sentences in the corpora. The following script imports the TfidfVectorizer and the cosine_similarity functions Based on the cosine similarity of the user input to the corpus, a response is generated

```
[ ] 1 from sklearn.feature_extraction.text import TfidfVectorizer
    2 from sklearn.metrics.pairwise import cosine_similarity

[ ] 1 def generate_response(user_input):
    2     marian_response = ''
    3     article_sentences.append(user_input)
    4
    5     word_vectorizer = TfidfVectorizer(tokenizer=get_processed_text, stop_words='english')
    6     all_word_vectors = word_vectorizer.fit_transform(article_sentences)
    7     similar_vector_values = cosine_similarity(all_word_vectors[-1], all_word_vectors)
    8     similar_sentence_number = similar_vector_values.argsort()[0][-2]
    9
    10    matched_vector = similar_vector_values.flatten()
    11    matched_vector.sort()
    12    vector_matched = matched_vector[-2]
    13
    14    if vector_matched == 0:
    15        marian_response = marian_response + "I am sorry, I could not understand you"
    16        return marian_response
    17    else:
    18        marian_response = marian_response + article_sentences[similar_sentence_number]
    19        return marian_response

[ ] 1 word_vectorizer = TfidfVectorizer(tokenizer=get_processed_text, stop_words='english')
    2 all_word_vectors = word_vectorizer.fit_transform(article_sentences)
```

✦ Chatting with the chatbot

a function that allows us to chat with the chatbot that we just designed. To do so, we will write another helper function that will keep executing until the user types "Bye" or "see you later"

```
[ ] 1 continue_dialogue = True
2 print("मनसो, मैं अपनी दोस्त बनिय हू। अब तुमसे खेलें और तुम के संदेश में कोई भी प्रश्न पूछ सकते हैं।")
3 while(continue_dialogue == True):
4     print('तुम: ', end='')
5     human_text = input()
6
7     if human_text != 'अवधि' and human_text != 'बिदा':
8         if human_text == 'उपलब्ध' or human_text == 'जो बहुत बहुत धुनिया':
9             continue_dialogue = False
10            print("मनसो: अपनी सेवा में किसी भी समय")
11        else:
12            translated_human_text=translator_hindi(human_text)
13
14            translated_human_text=translated_human_text.text
15            if generate_greeting_response(translated_human_text) != None:
16                greeting_response_translated=translator_english(generate_greeting_response(human_text))
17                print("मनसो: " + greeting_response_translated.text)
18            else:
19                translated_human_text = translated_human_text.lower()
20                print("मनसो: ", end='')
21                response=generate_response(translated_human_text)
22                print(translator_english(response).text)
23                #article_sentences.remove(human_text)
24        else:
25            continue_dialogue = False
26            print("मनसो: अवधि और अपना स्वागत करता...")
```


Screenshots of application



Test Suite:

Testing done through Robustness testing

Our application only uses hindi for processing so we are testing 'hindi' , 'hinglish' & english for testing inputs And analyzing corresponding outputs

- For hindi greeting:

नमस्ते, मैं आपकी दोस्त मरियम हूँ। आप मुझसे खेती और कृषि के संबंध में कोई भी प्रश्न पूछ सकते हैं:
हम : नमस्ते
मरियम: आपका स्वागत है, मैं अच्छा हूँ और आप

- For 'Hinglish' greeting

हम : namaste
मरियम: अरे तुम कैसे हो?

- For english greeting

हम : Hi
मरियम: क्षमा करें, मैं आपको समझ नहीं पाया

- For hindi Query

हम : फसल क बार में बतए
मरियम: फसल के पोषक तत्वों के उपयोग को फसल रेटेंशन या परती अवधि जैसी सांस्कृतिक तकनीकों का उपयोग करके भी प्रबंधित किया जा सकता है। खाद का उपयोग या तो पशुधन को फसलदार किया जाता है, जहां चारा फसल बढ़ रही है, जैसे कि प्रबंधित गहन घुर्नी चराई में, या सूखे या तरल पौधों को फैलाकर। फसल भूमि या बरसात पर खाद।

- For 'Hinglish' query

हम : phasal kaisi hoti hai
मरियम: खदियों से पौधों को फलतु बनाने से उपज में कृद्धि हुई है, रोग प्रतिरोधक क्षमता और सुखा सहनशीलता में सुधार हुआ है, फसल में आलसी हुई है और फसल के पौधों के खाद और पोषण मूल्य में सुधार हुआ है। सामाजिकीकरण नयन और फसलन का फसल पौधों की विशेषताओं पर भी प्रभाव पड़ा है।

- For english query

हम : how should be the yield
मरियम: कृषि उपज बढ़ाने और लागत कम करने का प्रयास करती है। उर्वरकों और रोगजल्कों, शिकारियों और प्रतिस्पर्धियों (जैसे खरपतवार) को हटाने जैसे इनपुट के साथ उपज बढ़ती है।

- Terminating message

हम : अलविदा
मरियम: अलविदा और अपना ख्याल रखना...

