TRIBHIVAN UNIVERSITY
INSTITUTE OF ENGINEERING
THAPATHALI CAMPUS

A Project Proposal
On
‘Samadhan’ – A Local Government Service QueryBot

Submitted By:

Aarogya Bhandari (THA077BCT001)
Aayush Pokharel (THA077BCT002)
Piyush Luitel (THA077BCT031)
Prashant Bhusal (THA077BCT034)

Submitted To:

Department of Electronics and Computer Engineering
Thapathali Campus
Kathmandu, Nepal

December, 2023

ACKNOWLEDGEMENT

We extend our gratitude to the Department of Electronics and Computer Engineering at the Institute of Engineering, Thapathali Campus, for providing us the opportunity to broaden our understanding of programming, as well as delve into the realms of Natural Language Processing (NLP) and Artificial Intelligence (AI) throughout the execution of this project. Special thanks are offered to our supervisor, Er. Bisikha Subedi, whose guidance not only enriched our programming skills but also facilitated our exploration into the fascinating domains of NLP and AI. Additionally, we acknowledge our supervisor's valuable suggestion for incorporating a user-friendly approach in creating this project, which significantly enhanced its usability and accessibility.

Aarogya Bhandari (THA077BCT001)

Aayush Pokharel (THA077BCT002)

Piyush Luitel (THA077BCT031)

Prashant Bhusal (THA077BCT034)

ABSTRACT

This project is poised to unfold under the Agile software development model, emphasizing a phased approach. In the initial phase, our focus will be on the development and deployment of a user-friendly system centered around citizenship services. Subsequent phases will extend this approach to taxation, birth certificates, and health insurances, ensuring a comprehensive and iterative development process.

Additionally, to enhance user experience, our project will feature an innovative element-a conversational bot. This bot will not only facilitate interactive communication but will also incorporate a dynamic component: the ability to showcase animated videos illustrating government service procedures. This integration aims to provide users with a visually engaging and informative experience, ensuring a more effective and user-friendly interaction with the system.

Keywords: *Rasa-NLU, Rasa-core, spaCy, Scikit-learn, Rasa pipeline, POS tagging, Tokenization, Semantics, Entity Extraction, Intent Recognition, Docker*.

TABLE OF CONTENTS

List of Figures

List of Tables

List of Abbreviations

API - Application Programming Interface

AWS - Amazon Web Services

CSS - Cascading Style Sheets

DAO - District Administration Office

FAQ - Frequently Asked Questions

HTML - Hypertext Markup Language

IDE - Integrated Development Environment

NLP - Natural Language Processing

NLG - Natural Language Generation

POS - Part-Of-Speech

Rasa NLU - Rasa Natural Language Understanding

UI - User Interface

URL - Uniform Resource Locator

UX - User Experience

1 INTRODUCTION

1.1 Background

In Nepal, citizens face a significant hurdle when navigating government services due to a notable lack of effective communication channels within government offices. Individuals trying to obtain important services are often left in the dark about the necessary steps, required documents, and how long the process might take.

1.2 Problem Definition

In Nepal, people often find themselves facing a considerable challenge when dealing with government services, mainly because effective communication channels within government offices are notably lacking. Those trying to secure essential services are frequently left in the dark about the required steps, necessary documents, and the expected duration of the process. The absence of a central hub or clear guidance system means there's no go-to resource for navigating the complexities of various government processes. Currently, information sharing is textual and tedious to read, making it difficult for individuals to receive the right guidance. This unreliable communication setup creates a noticeable divide between government offices and citizens, making it increasingly challenging for people to access the services they need and leaving them feeling frustrated and disconnected.

1.3 Motivation

As of now, there's no central hub or clear system that guides people through the ins and outs of various government processes. The current ways information is shared are scattered and informal, making it even more challenging for individuals to get the right guidance. This lack of a reliable communication system creates a gap between government offices and citizens, making it harder for people to access the services they need and leaving them feeling frustrated and disconnected. Imagine a solution that changes this narrative. We propose creating a friendly chatbot, ‘Samadhan’ designed to be like a virtual guide for anyone seeking local government (Ward, Municipality and DAO) services in Nepal.

1.4 Objectives

- To develop a chatbot 'Samadhan' which solves the governmental services query in Ward, Municipality and District Administration Office (DAO) in English language with a user-friendly approach.
- To deploy Samadhan in a website.

1.5 Scope and Application

Samadhan aims to act as a virtual facilitator, using technology to provide the public with accurate and accessible information about government processes with a user-friendly approach. Activities include a communication network that can provide guidance on steps required, documentation required and estimated timelines for tasks and animated video guidelines for user-friendly experience.

The applications of ‘Samadhan’ are:

- Information accessibility: Provide citizens with instant access to accurate and up-to-date information on government services like queries on citizenship and tax.
- Guidance and assistance: Offer step-by-step guidance for individuals seeking government services (citizenship, tax, health insurance, etc.) from application procedures to required documentation. Assist users in understanding the expected timelines for various processes, reducing uncertainty and frustration.
- User-Friendly Interface: Develop an intuitive and friendly chatbot interface accessible via web and mobile platforms, ensuring ease of use for citizens.
- Transparency and Accountability: Enhance transparency in government processes by providing clear and consistent information to citizens. Foster accountability by offering a traceable record of interactions, ensuring a reliable source of information.

2 LITERATURE REVIEW

The increasing integration of Artificial Intelligence (AI) into daily life is evident through the production of intelligent agents, such as chatbots, which perform diverse tasks from labor to complex operations [1]. Chatbots, an elementary yet widespread form of intelligent Human-Computer Interaction (HCI) [2], simulate human-like conversation through text or voice, employing Natural Language Processing (NLP) to understand multiple human languages.

Their popularity stems from several advantages, including platform-independence, immediate user accessibility, and re-engagement of inactive users through notifications. The widespread utility and advantages of chatbots underscore their significance in modern AI-driven interactions. In the private sector, chatbots, also referred to as conversational agents, have seen extensive utilization in various domains like banking, media, tourism, and retail. These AI-driven solutions, exemplified by virtual assistants like Siri, Alexa, and Google Now, have transformed customer service interactions and transactional processes [3].

2.1 Related Works:

2.1.1 Muna by Diyo.ai:

Diyo.ai creation Muna is conceived as a user-friendly guide to facilitate hassle-free communication between citizens and their local government whether to handle application processes properly, document a clarifying analysis, or provide guidance on workshops, Muna emerges as a versatile partner. Acting as a reliable source of general information on government services, Muna addresses public inquiries on aspects of local governance.

Furthermore, Muna is now incorporated into Lalitpur Metropolitan Municipality and Butwal Sub-Metropolitan Municipality. In addition to its role in information dissemination, Muna not only facilitates public participation but also contributes to the development of responsible and accountable local government in areas of meaningful implementation difficult to address public complaints, empowering individuals to express concerns about issues such as traffic, public waste disposal, or road conditions. [4]

2.1.2 AskGov by Singapore government:

AskGov.sg, a Singapore government-led initiative acts as a comprehensive website and interactive chatbot, providing citizens with a versatile platform to inquire about government services In this digital space, users the user can refer their queries to various government agencies, including the finance ministry, health including ministry, public - Transport council, immigration, checkpoint authority, etc. The integrated chatbot feature ensures that citizens can easily interact with the platform and get fast and accurate answers to their queries.

The user-friendly AskGov.sg homepage not only displays a careful selection of frequently asked questions but also uses chatbot functionality to provide concise and informative answers This unique integration provides accessibility and visibility transparently, enabling citizens to access government services and speak up in real-time effortlessly to clarify doubts or gather information AskGov.sg stands as a prime example of the usefulness of technology to engage citizens encourage government communications and facilitate access to critical information. [5]

2.2 Related theory

2.2.1 Rasa-NLU

Rasa NLU (Natural Language Understanding) is an open-source natural language processing library primarily used for understanding and extracting intent and entities from user messages. Operating as part of the Rasa framework, it facilitates the creation of conversational AI applications. Rasa NLU employs machine learning-based models to decipher the intentions behind user messages (intents) and extract relevant information (entities) crucial for responding effectively in chatbot interactions. Its pipeline-based architecture allows for the configuration of various components, including tokenizers, featurizers, intent classifiers, and entity extractors, enabling developers to tailor the NLU pipeline to suit specific use cases and data. By training on annotated datasets, Rasa NLU learns to recognize patterns in user input, enhancing its ability to accurately comprehend natural language queries and generate appropriate responses in conversational applications.

2.2.2 Rasa-core

Rasa Core, an integral component of the Rasa framework, is an open-source dialogue management library used for building conversational AI applications. It orchestrates the flow of conversations in chatbots by employing machine learning-based models for managing dialogue policies. Operating on a probabilistic framework, Rasa Core utilizes reinforcement learning techniques, particularly the LSTM-based neural networks, to predict and determine the next best action or response based on the conversational context and history of interactions. Its dialogue management system allows for handling multi-turn conversations by learning from user inputs and feedback, enabling the chatbot to make contextually appropriate decisions and responses. Rasa Core's flexibility lies in its capability to create custom dialogue policies, such as rule-based policies or machine learning-based policies, empowering developers to design sophisticated conversational flows and optimize chatbot interactions for diverse use cases and domains.

2.2.3 Rasa pipeline

Rasa NLU employs the "spacy_sklearn" pipeline, integrating various NLP libraries like spaCy, scikit-learn, and sklearn-crfsuite for robust language comprehension, particularly focused on intent classification and entity extraction in chatbots and AI assistants. This pipeline comprises specific components: "tokenizer_spacy," utilizing spaCy for tokenization and part-of-speech annotation; it leverages spaCy's specialized neural models optimized with convolutional layers, residual connections, batch normalization, and maxout non-linearity. GloVe vectors [6] from tokens concatenate to form sentence feature vectors, used for multiclass SVM-based intent classification via "intent_classifier_sklearn." Additionally, the "ner_crf" component, employing a CRF classifier from sklearn-crfsuite, utilizes words and POS tags as time steps and states, respectively, to extract entities by computing the most probable tag set based on word features such as capitalization and POS tagging. [7]

3 METHODOLOGY AND SYSTEM ARCHITECTURE

3.1 Software Development Life Cycle (SDLC):

For the development of this noble digital channel for interaction between government and citizens we will be adapting the Agile Software Development Model. This model uses iterative and incremental software development approach for better understanding of requirements and provide effective product.

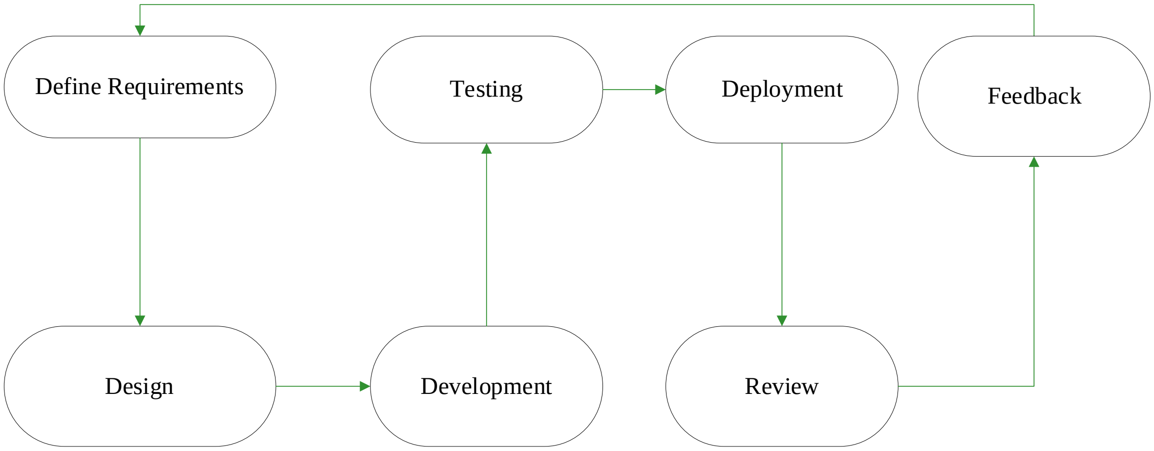


Figure 3-1 Agile Software Development

Iteration 1: The most common government services are issuing citizenship. So, all the information regarding the citizenship issuing process will be gathered at first. The related queries are documented and implemented in design providing the best outcome and effective response. In addition, an animated video will be prepared for this query so that, the information can be easily expressed and understood by general public.

Iteration 2: After successfully implementing the citizenship issuing procedure, the next phase of our project will be focused on tax-related queries. The same procedure will be followed to provide the effective response. To increase the expressiveness of the response, short animated video will also be provided.

Further services like queries regarding insurance, national ids, birth certificate, etc. will be added in next iterations.

3.2 Deployment

First, the trained Rasa model needs to be integrated into a web environment which is achieved by utilizing Rasa's REST API, allowing the chatbot to communicate with the website. Next, a frontend user interface for the chatbot is created using web development technologies such as HTML, CSS, and JavaScript. This interface serves as the bridge between the user and the Rasa backend. The communication between the frontend and the Rasa backend is facilitated through HTTP requests. To deploy a Rasa chatbot in the cloud, we must package the Rasa model and server using a containerization tool like Docker .The deployment process also involves hosting the Rasa server on a web server. This can be done using platforms like Docker or deploying the Rasa server on cloud services such as AWS, Azure, or Google Cloud. Configure the necessary network settings and ensure the server is accessible over the internet. Once deployed, users can interact with the Rasa chatbot on the website, receiving responses generated by the NLU (Natural Language Understanding) and dialogue management capabilities of the Rasa framework.

3.3 System Architecture

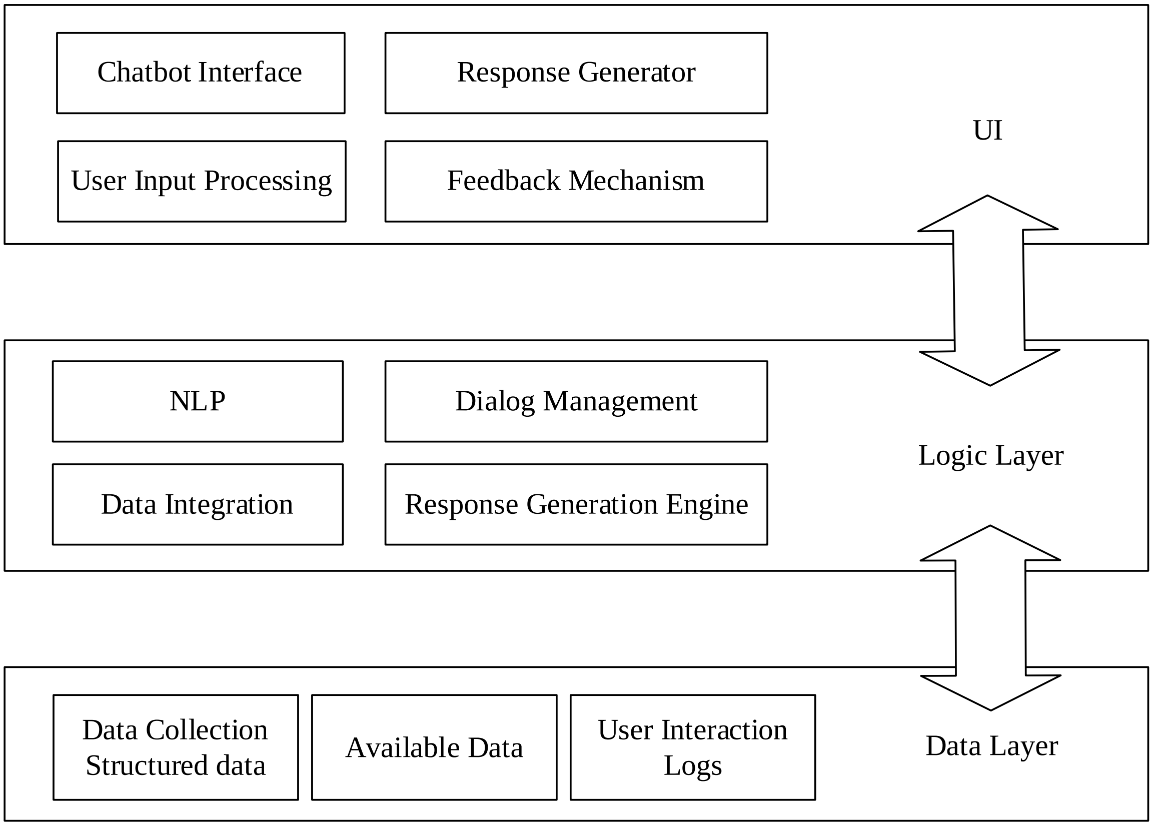


Figure 3-2 Three-tier architecture of proposed approach

Figure 3-2 provides a visual representation of the chatbot system, highlighting the key components and their interactions. The effectiveness of the chatbot relies on the seamless coordination of these blocks to provide accurate and timely information to users.

3.4 Parts of Program

The parts of the programs as shown in Figure 3-2 are:

3.4.1 UI Layer

3.4.1.1 Chatbot Interface

The Chatbot Interface is the graphical or conversational platform through which users interact with the chatbot. The design considerations involve creating an intuitive and user-friendly web-interface for seamless interaction. Elements include input/output sections where users enter queries and receive responses from the chatbot.

3.4.1.2 Response Generator

This component will generate responses based on the users' queries and the information available within the system. The response will be provided to the user in the form of text and animated video.

3.4.1.3 User Input Processing and Feedback Mechanism

Input Processing involves handling user queries received through the UI. It will perform preprocessing tasks, such as text cleaning, tokenization, and entity recognition, to understand user intent and extract relevant information.

The Feedback Mechanism will capture user feedback on the provided responses, helping to improve the chatbot's accuracy and performance over time.

3.4.2 Logical Layer:

3.4.2.1 Natural Language Processing (NLP):

NLP involves understanding and interpreting human language. It encompasses tasks like intent classification, entity recognition, sentiment analysis, etc. NLP helps the chatbot comprehend user queries and extract important information to provide accurate and contextually relevant responses.

3.4.2.2 Dialog Management:

Dialog Management will maintain the conversation flow, manages context, and determines the appropriate responses based on the current dialogue state. It uses state-tracking mechanisms to ensure coherent and natural interactions and may integrate with NLP for improved understanding.

3.4.2.3 Data Integration:

Data Integration involves combining data from knowledge base to provide comprehensive and accurate information to users.

3.4.2.4 Response Generation Engine:

Response Generation Engine will categorize the type of intent by extracting the entities and then according to the predefined response for each entity, it responds to the user. It is in the form of textual or user-friendly way in the form of animated video.

The flow of intent and responses will be defined in the form of story in '.yml' files

3.4.3 Data Layer:

3.4.3.1 Data Collection:

Involves gathering and compiling datasets required for training and improving the chatbot's performance. This data can include information on government services, FAQs, user queries, and other relevant data points.

3.4.3.2 User Interaction Logs:

Logs and stores user interactions with the chatbot. Helps in analyzing user behavior, identifying patterns, and improving the chatbot's performance based on user feedback.

This architecture emphasizes the interaction between the UI, Logical Layer (NLP, Dialog Management, Data Integration, Response Generation), and the Data Layer (Data Collection, User Interaction Logs) to create a robust and efficient governmental chatbot system. Integrating these layers effectively ensures accurate, accessible, and user-centric interactions, thereby enhancing the overall user experience and service delivery.

3.5 Design

3.5.1 UML Use Case Diagram

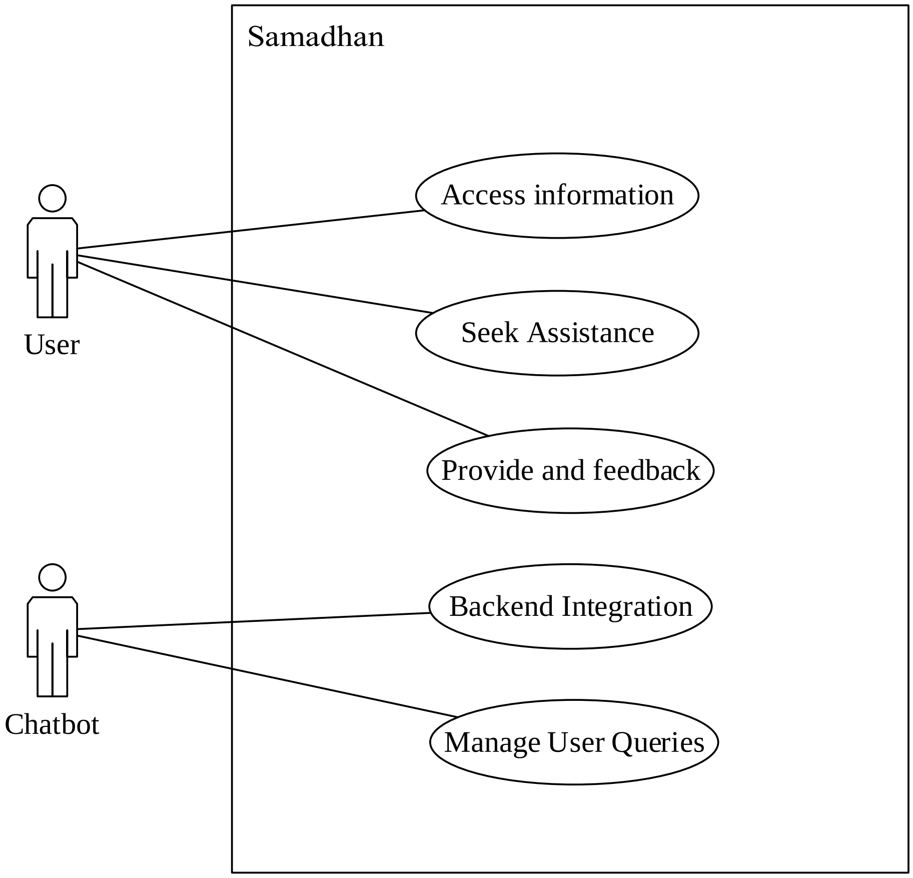


Figure 3-3 UML Use Case Diagram of main program

Figure 3-3 outlines the interactions between the user (actors) and the 'Samadhan' chatbot system to accomplish specific tasks. It showcases how users can access information, seek assistance, provide feedback, and how the chatbot manages user queries while interacting with backend systems to provide accurate information.

3.5.2 DFD Diagram

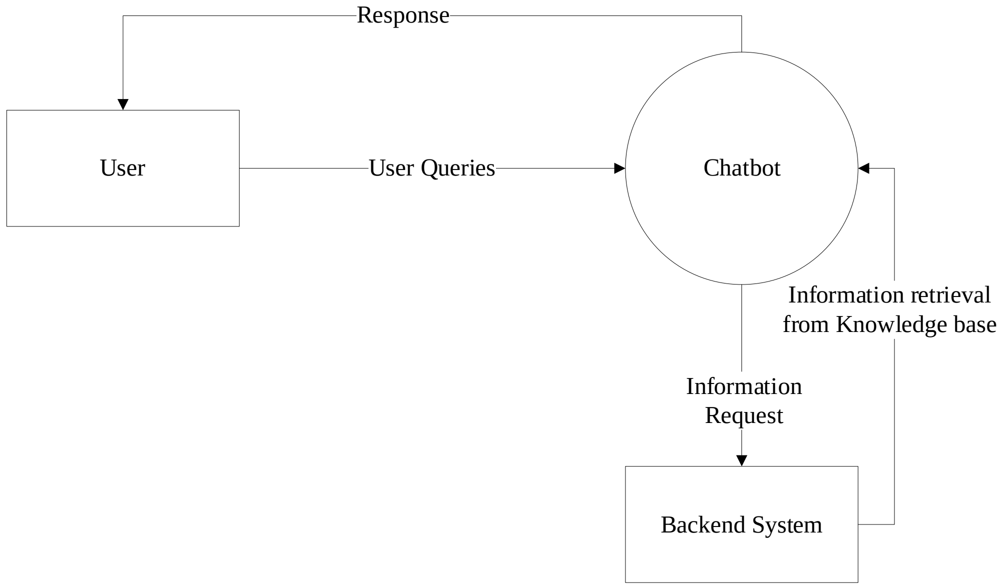


Figure 3-4 DFD Diagram of main program

Figure 3-4 shows Level-0 DFD which illustrates the main processes and interactions between the user, chatbot application, and backend systems. It showcases how users submit queries, how the chatbot processes and responds to these queries, and how it interacts with backend systems to fetch the information about government services enqired by user.

3.5.3 Interaction Flow:

- User inputs a query through the UI.
- The NLP Processor analyzes the query and extracts intent and entities.
- The Information Retrieval block fetches relevant data based on the user's intent and entities.
- The Response Generator formats the retrieved information into a user-friendly response.
- The UI displays the response to the user.
- In case of FAQs or general queries, the relevant block provides predefined responses.

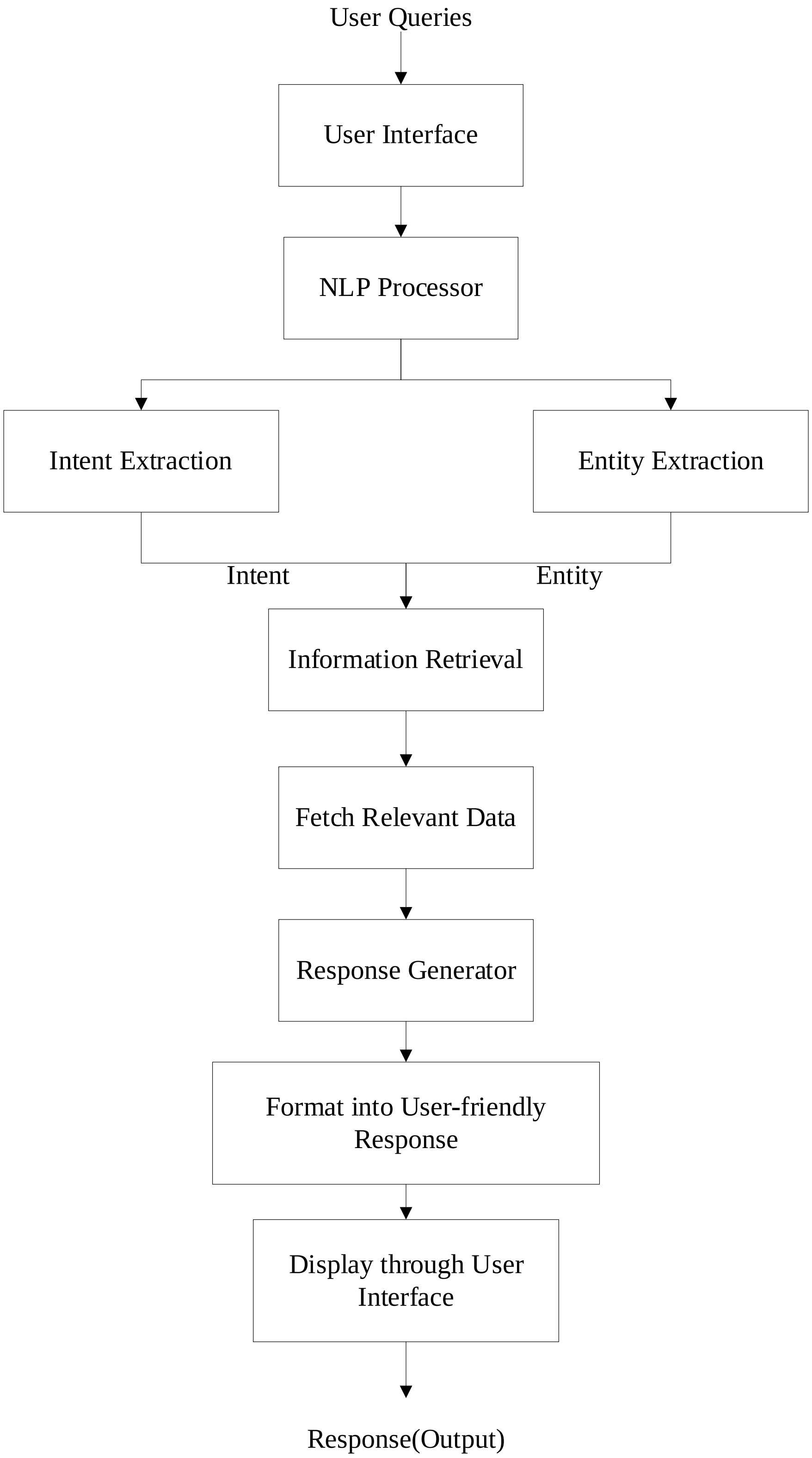


Figure 3-5 Work-Flow of the proposed system

4 REQUIREMENT ANALYSIS AND FEASIBILITY STUDY:

4.1 Software Requirements:

The softwares required for this project is listed below:

Table 4-1 Software Requirements

Software	**Reason**
Python	Python for coding and utilizing various modules and libraries.
HTML, CSS and Javascript	For front-end development
Rasa	Rasa for natural language understanding and natural language generation.
spaCy	spaCy for enhanced entity recognition and part-of-speech tagging.
VS code IDE	Visual Studio Code (VS Code) for development and project management.
Canva	Canva for the development of animated videos.

4.2 Technical Feasibility:

The technical feasibility assessment for our government services chatbot project revolves around evaluating the practicability of its implementation. Leveraging open-source tools such as Rasa NLU and spaCy, we aim to enhance the chatbot's ability to comprehend user queries by intent and entity extraction. Our development strategy prioritizes open-source libraries and frameworks, particularly relying on Python and its associated libraries. An integral aspect of our approach involves establishing a comprehensive knowledge base ensuring the chatbot's access to essential information. This application will be hosted on a website, providing users with an efficient and accessible solution.'

4.3 Operational feasibility:

The operational feasibility of our government services chatbot project is rooted in its practicality and effectiveness for day-to-day use. The seamless integration of the chatbot into existing government service processes ensures smooth operations, and the user-friendly design guarantees accessibility for citizens. The availability of data from the Kathmandu Metropolitan City website and the District Administration Office Kathmandu supports the development of the knowledge base of the chatbot. Considering the utilization of free software frameworks like Rasa and spaCy, the project stands as an operationally sustainable solution. The project's success depend upon the collaboration with relevant government offices and the responsiveness of the chatbot to user needs, making it a operationally viable tool for improved citizen engagement with government services.

4.4 Economic feasibility:

The economic feasibility of our government services chatbot project is favorable due to the utilization of open-source tools, eliminating any development costs associated with proprietary software. The adoption of frameworks like Rasa and spaCy, which are freely available, aligns with the project's commitment to cost-effectiveness. Furthermore, the decision to locally host the chatbot or leverage free hosting platforms ensures that there are no additional costs for hosting services. By strategically leveraging open-source resources and opting for no-cost hosting solutions, the project ensures economic viability and sustainability.

5 EXPECTED OUTCOME

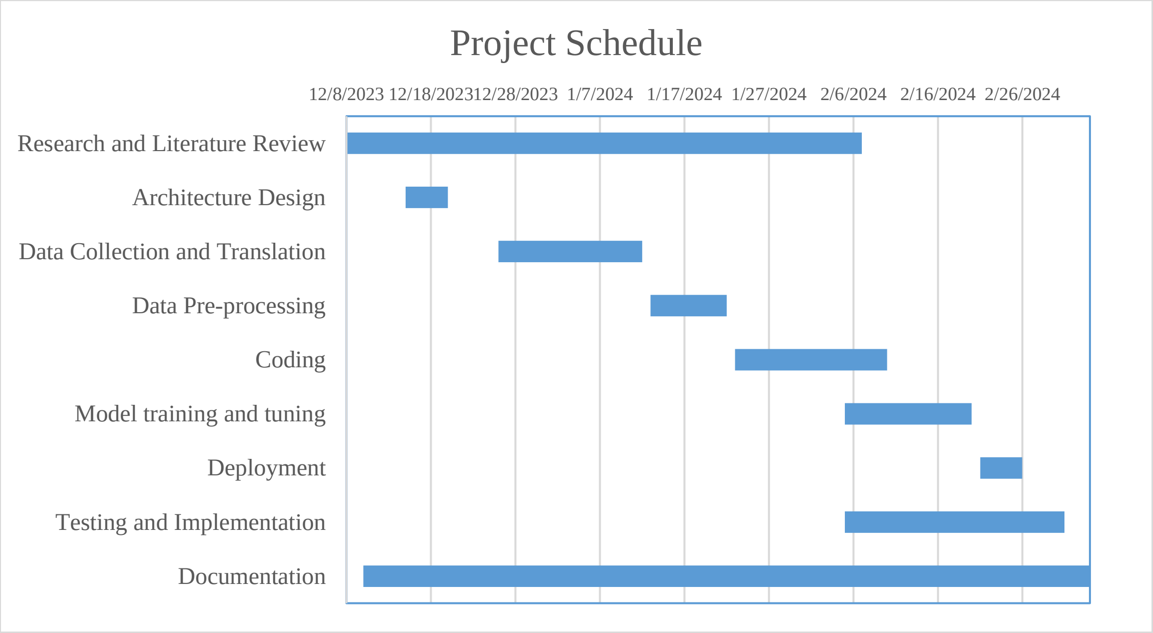
The expected results are:

- Intuitive Interface: 'Samadhan' will come with a simple design, making it easy to navigate government services. It will help to go through the process effortlessly.
- Comprehensive Information hub: ‘Samadhan' will be a go-to for information on different government services in Ward Office, Municipality Office and DAO. It will give all the details in the form of text and animated video required to make smart decisions at every step.
- Feedback Integration: 'Samadhan' will incorporate a feedback mechanism, underscoring the importance of user input. This feature will empower users to share experiences, contributing to the ongoing enhancement of the chatbot's functionality. The feedback mechanism will serve users to influence refinements, ensuring continuous improvement aligned with user expectations and needs.

6 PROJECT SCHEDULE

The timing of the project is shown in the table below:

Table 6-1 Gantt Chart Showing Project Schedule



REFERENCES

[1] R. K. Sharma, "An Analytical Study and Review of open source Chatbot framework, Rasa," *International Journal of Engineering Research & Technology (IJERT),* vol. 9, no. 6, 2020.

[2] H. Bansal and R. Khan, "A Review Paper on Human Computer Interaction," *nternational Journals of Advanced Research in Computer Science and Software Engineeing,* vol. 8, no. 4, 2018.

[3] L. Klopfenstein, S. Delpriori, S. Malatini and A. Bogliolo, "The Rise of Bots: A Survey of Conversational Interfaces, Patterns, and Paradigms," in *The 2017 Conference*, Urbino, 2017.

[4] Diyo.ai, "Localizing AI | DIYO.AI," [Online]. Available: https://diyo.ai/muna. [Accessed 15 12 2023].

[5] S. Government, "AskGov | Q&A: Your questions answered," Open Government Products, [Online]. Available: https://ask.gov.sg/. [Accessed 15 12 2023].

[6] J. Pennington, R. Socher and C. D. Manning, "GloVe: Global Vectors for Word Representation," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, 2014.

[7] A. Jiao, "An Intelligent Chatbot System Based on Entity Extraction Using RASA NLU and Neural Network," *Journal of Physics: Conference Series*, vol. 1487, no. 012014, 2020.

[8] E. Adamopoulou and L. Moussiades, "An Overview of Chatbot Technology," in *IFIP International Conference on Artificial Intelligence Applications and Innovations*, 2020.