MBEYA UNIVERSITY OF SCIENCE AND TECHNOLOGY

COLLEGE OF INFORMATION & COMMUNICATION TECHNOLOGY

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING

COURSE: COMPUTER ENGINEERING

MODULE NAME: SENIOR PROJECT II

MODULE CODE: COB 4227

LEVEL: UQF 8

STUDENT NAME: VICTOR ANTHONY ANDREA

REG No: 19100534050059

UE NUMBER: UE/BCE/22/17253

SUPERVISOR: LIHENDIME MADEMBWE

COORDINATOR: SHEDRACK NJUGUNYA

PROJECT TITLE: DEVELOPMENT OF INTELLIGENT STUDENT HELP DESK SYSTE

DECLARATION

I hereby solemnly declare that the project report entitled "Development of Intelligent Student Help Desk System" is based on my own work carried out during the course of my study.

I further certify that the work contained in the report is original and has been done by me under the general supervision of my supervisor and has not been submitted for any other degree/diploma/certificate in this university or any other universities.

Student Name

……………………………

Signature

……………..

Date

……………..

CERTIFICATION

This is to certify that the project entitled “Development of Intelligent Help Desk System” is submitted by Victor Anthony Andrea in partial fulfillment of the requirement of Bachelor degree of Engineering in Computer offered by department of Computer Science and Engineering from Mbeya university of Science and Technology (MUST).

Supervisor Name

…………………………….

Signature

……………..

Date

………………

ABSTRACT

The main purpose of this project is to develop an intelligent student help desk system in colleges which is capable to assist students during registration process and gives useful information when they need.

An Intelligent Help Desks are software applications that simulate a conversation with a user. They are used to automate interactions with users, and can be used to provide information, answer questions, and perform tasks. An Intelligent Help Desk are often used in customer service, Schools, banks and so on.

An Intelligent Help Desk is a set of software program that is intended to help consumer or customer support workforce. It is the system that can receive, process and reply to a customer or client inquiries across a range of contact points. Intelligent Help Desk allows better control of all the process performed in the system, it has the following important attributes that it can be used to control the system (Entity, Entries, Flow and Control or Application logic).

This system will use Machine learning algorithms and techniques that will produce the desired output based on user input.

ACKNOWLEDGEMENT

First of all, I would like to take this opportunity to thank the Almighty GOD who gave me the strength till this moment. Also, I would like to thank my project supervisor LIHENDIME MADEMBWE for the positive inputs that he has been giving me and saving his time for me whenever I am in need of his assistance.

I would also like to thank our teaching staff and panel for their contribution to this project including the suggestion and recommendations they gave me which help this project to reach this point. Lastly, I would like to thank my classmates who are always there to correct me whenever I approached them.

TABLE OF CONTENTS

DECLARATION i

CERTIFICATION ii

ABSTRACT iii

ACKNOWLEDGEMENT iv

LIST OF FIGURES 1

CHAPTER ONE 4

INTRODUCTION 4

1.1 GENERAL INFORMATION 4

1.2 HISTORICAL BACKGROUND 4

1.3 PROBLEM STATEMENT 5

1.4 PROJECT OBJECTIVES 5

1.4.1 MAIN OBJECTIVE 5

1.4.2 SPECIFIC OBJECTIVES 5

1.5 SIGNIFICANCE OF THE PROJECT 6

1.6 SCOPE OF THE PROJECT 6

1.7 CASE STUDY 6

1.8 LIMITATIONS OF THE PROJECT 6

CHAPTER TWO 7

LITERATURE REVIEW 7

2.1 INTRODUCTION 7

2.2 BACKGROUND 7

2.3 EXISTING SYSTEM 8

2.4 RELATED WORKS 8

2.5 PROPOSED SYSTEM 9

2.6 ARCHITECTURE OF PROPOSED SYSTEM 10
2.7 SYSTEM MECHANISM (HOW IT WORKS) 10
2.8 FLOW CHART 11
CHAPTER THREE 12
METHODOLOGY 12
3.1 INTRODUCTION 12
3.2 DATA COLLECTION METHODS 12
3.2.1 PRIMARY DATA COLLECTION 12
3.2.2 SECONDARY DATA COLLECTION 13
3.2.3 QUESTIONNAIRE 13
3.3 DOCUMENTATION 14
3.4 DESIGN APPROACH 14
3.4.1 ADVANTAGES OF PROTOTYPING-BASED METHODOLOGY. 15
CHAPTER FOUR 17
DATA COLLECTION AND ANALYSIS 17
4.1 CHAPTER OVERVIEW 17
4.2 PRIMARY DATA 17
4.3 SECONDARY DATA 20
4.3.1 PROGRAMMING LANGUAGE 20
4.3.2 IDE OR ENVIRONMENT FOR CODING ML 21
4.3.3 PROCESSOR 22
4.3.4 OPERATING SYSTEM 22
4.3.5 MACHINE LEARNING MODEL/TECHNIQUES 23
4.4 PRIMARY DATA ANALYSIS 25
4.4.1 ANALYSIS OF THE PRIMARY QUESTIONS. 25

4.4.2 SECONDARY DATA ANALYSIS ... 29

CHAPTER FIVE ... 37

SYSTEM DESIGN AND IMPLEMENTATION ... 37

5.1 CHAPTER OVERVIEW ... 37

5.1.1 OBJECTIVE AND GOALS OF THE PROJECT ... 37

5.1.2 USE CASE DIAGRAM ... 37

5.1.3 INTERACTION BETWEEN COMPONENTS ... 40

5.2 SYSTEM DESIGN ... 40

5.3 SYSTEM IMPLEMENTATION ... 41

5.3.1 DEVELOPMENT PROCESS AND METHODOLOGIES ... 42

5.3.2 CHALLENGES FACED DURING IMPLEMENTATION AND SOLUTION OBTAINED ... 55

5.4 TESTING APPROACH AND METHODOLOGIES ... 56

5.4.1 TEST RESULTS AND METRICS ... 56

5.4.2 ACHIEVEMENTS AND OUTCOMES ... 57

5.5 FUTURE ENHANCEMENTS AND FURTHER DEVELOPMENT ... 57

5.6 DISCUSSION OF RESULTS ... 57

5.7 CONCLUSION AND FUTURE WORK ... 57

5.8 RECOMMENDATIONS ... 58

APPENDECIES ... 59

PROJECT COST SHEET ... 59

PROJECT MILESTONE ... 60

REFERENCES ... 61

LIST OF FIGURES

Figure 1. Architecture of proposed system 10
Figure 2. System mechanism (how it works)........ 10
Figure 3. System flow chart 11
Figure 4. Design approach 15
Figure 5. Analysis question one........ 25
Figure 6. Analysis question two........ 26
Figure 7. Analysis question three........ 26
Figure 8. Analysis of question four........ 27
Figure 9. Analysis question seven. 28
Figure 10. Analysis question nine........ 28
Figure 11. Developer use case. 38
Figure 12. User use case. 38
Figure 13. System design........ 40
Figure 14. Libraries to work with. 41
Figure 15. Example of intents........ 42
Figure 16. Image of intents. json file. 43
Figure 17. Image of training file. 45
Figure 18. Image of training deep learning model........ 48
Figure 19. Image of a code containing user interface and functionality........ 52
Figure 20. UI interface........ 53
Figure 21. example of a user conversation with bot. 55
Figure 22. Image of metric evaluation of the accuracy of the model. 56
Figure 23. Image of bot response time........ 56

LIST OF TABLES

Table 1. Data collected on different Programming Languages. 20

Table 2. Data collected on different IDE's 21

Table 3. Data collected on different Processor specification.. 22

Table 4. Data collected on different Operating system.. 23

Table 5. Data collected on different machine learning models/techniques 24

Table 6. Enrolment trends of students 29

Table 7. Fee structure.. 30

Table 8. Important documents needed. 30

Table 9. Project Cost Sheet... 59

Table 10. Project Milestone. 60

LIST OF ABREVIATIONS.

LSTM = Long Short-Term Memory.

ML = Machine Learning.

AI = Artificial Intelligence.

DL = Deep Learning.

CNN = Convolution Neural Network.

RNN = Recurrent Neural Network.

DRL = Deep Reinforcement Learning.

GAN = Generative Adversarial Networks.

IDE = Integrated Development Environment.

GPU = Graphical Processing Unit.

RAM = Random Access Memory.

CPU = Central Processing Unit.

NLP = Natural Processing Language.

API = Application Programming Interface.

CHAPTER ONE

INTRODUCTION

1.1 GENERAL INFORMATION

This project work on the development of an Intelligent Student Help Desk System built to save time of student registration process which is processed manually by students to straight the line for a long time, also the system will help the student which step to follow after the previous one has been done.

When I got this idea of developing this system, the general working of the system was to create a registration system that helps students to process their data and information quickly, second is to create the system for Identity Card to be processed automatically, as when student already gives the information in the earlier steps of registration like Names, Passport, School information and Course he or she studied the Administration office gets this information automatically and print the ID of that student regards he or she already pay the fees needed to be paid.

In the existing system here in Mbeya University of Science and Technology the student needed to arrange the forks in a line so as to be saved.

So, looking at the said factors, the system which I built it will help the students to do this process in easy way but due to time constraints in this project development of the system for student Identity Card was constrained.

1.2 HISTORICAL BACKGROUND

In Tanzania especially in large institutions or Universities now the use of Intelligent Student Help Desk System in student registration process, virtual assistance has not been grown well, is still having a problem of managing a time to complete the process in a short period of time, it took long time to complete the registration processes.

The present invention used at Mbeya University of Science and technology in registration process is registering students and assists manually by having some people wearing Ask Me reflector. Ask Me people they have a time they save students but there is a time they get tired; they cannot perform the task perfectly and sometimes they delay to save someone. Moreover, they have moderate speed during registering the students at morning, when it comes to afternoon and proceed the energy of saving people decreases.

A further disadvantage of using the current system is that it is necessary for a student to be there physically and it allow one person to be saved at a time in a line. As we have known the population of students grows daily, managing a large number of students it takes a while.

1.3 PROBLEM STATEMENT

The intelligent help desk system can take over a lot of tasks to perform I could be virtual assisting in e-learning, student registration process, answering questions, provide information and so on. Let’s look what are some of the problems that make me to propose to develop this Intelligent Student Help Desk System.

Firstly, the current existing system of student registration is done manually, students needed to be there physically to do step by step process in his or her registration.

Secondly, for the student to take a passport for the Identity Card (student ID), they took a long time to stand over the line for taking a picture, then after taking a picture they wait for two or three days or even a week plus to wait for their ID, sometimes after that two or three days you go there to take your ID and you found that your ID is not yet done. That is the wastage of time.

Due to the situation and factor mentioned above and due to time constraints, I decided to develop the system (Intelligent Student Help Desk System) that will have the ability to assist students and provides step by step process to follow during their registration process. And for the future come will be recommended in recommendation stage which things to be done for this system.

1.4 PROJECT OBJECTIVES

The objectives of this project are categorized into two main parts which are main objectives and specific objectives.

1.4.1 MAIN OBJECTIVE

The main objective of this project is development of Intelligent Student Help Desk System.

1.4.2 SPECIFIC OBJECTIVES

1. To build a comprehensive knowledge base to provide students information.
2. To train the algorithm using machine learning algorithms.

3. Fine-tuning of the parameter to improve performance of a bot.
4. To develop chat interface.

1.5 SIGNIFICANCE OF THE PROJECT

- Quick and easy access to information.
- Time management.
- Help to redirect the asked venue by the student.
- Cost reduction**:** AI Help Desk can help to reduce the cost of college enquiries by automating many of the tasks involved in the process.
- Improved customer satisfaction**:** this Intelligent Student Help Desk System can help to increase student satisfaction by providing a more personalized and efficient service.

1.6 SCOPE OF THE PROJECT

The scope of this project will end in query handling, student registration, venue allocation and provide useful information.

1.7 CASE STUDY

The case study of this project is Mbeya University of Science and Technology (MUST).

1.8 LIMITATIONS OF THE PROJECT

- It is used only in Mbeya University.
- It is developed purposely for helping MUST students in registration process and not in other domains.
- It does not process the images (passport) for the Identity card creation.
- It does not do any payment integration process.
- It does not allow mixing of languages when asking something at the same time.

CHAPTER TWO

LITERATURE REVIEW

2.1 INTRODUCTION

Literature review is a piece of academic writing demonstrating the knowledge and understanding of the academic literature on a specific topic placed in context. OR Is a survey of scholarly sources) related to a specific topic or research questions, it is often written as part of a thesis, dissertation, or research paper, in order to situate your work in relation to existing knowledge, example of sources are books, journal and articles.

2.2 BACKGROUND

An Intelligent Help Desks are software applications that simulate a conversation with a user. They are used to automate interactions with users, and can be used to provide information, answer questions, and perform tasks. An Intelligent Help Desk are often used in customer service, Schools, banks and so on.

An Intelligent Help Desk is a set of software program that is intended to help consumer or customer support workforce. It is the system that can receive, process and reply to a customer or client inquiries across a range of contact points. Intelligent Help Desk allows better control of all the process performed in the system, it has the following important attributes that it can be used to control the system (Entity, Intents, Flow and Control or Application logic).

- Intents: An intent is a user's intention or goal when interacting with a Help Desk. For example, if a user asks a Help Desk a question, the intent of the user is to get an answer to the question. If a user asks a Help Desk to book a flight, the intent of the user is to book a flight.
- Entities: An entity is a piece of information that is relevant to the user's intent. For example, if a user asks a Help Desk to book a flight, the entity is the flight that the user wants to book.
- Dialog: A dialog is a conversation between a user and a Help Desk. A dialog is made up of one or more turns. A turn is a single interaction between a user and a Help Desk. For example, if a user asks a Help Desk a question, the dialog is made up of one turn. If a user asks a Help Desk a question, and the Help Desk asks the user a follow-up question, the dialog is made up of two turns.

Intelligent Help Desk allow better control over the intents, entities, dialog flow, by controlling these attributes a system can produce the correct answer. These attributes are very powerful (allowing for the system to recognize the patterns, the output will be produced if the pattern is matched otherwise it will give wrong output).

The system will be accurate and perfect producing the correct output if it has large volume of data for training (to have more features in the system).

2.3 EXISTING SYSTEM

At Mbeya University now there is no Intelligent Student Help Desk System that help to assist students in helping them during registration process, hence the people just wait in the queue or line for getting help. For that case it results in wastage of time and sometimes a person can stand in the line for a long time just going to ask something like what process do i go now after paying the school fees?

In the existing system the student stands in front of student registration table and waiting for one student to be saved and then followed by another one, the disadvantage of the system is that a person can stand in the line for a long time and comes to found that he or she left some steps before the current stage he or she is.

2.4 RELATED WORKS

1. (Sharma, 2020) developed a system for a college enquiry chatbot system which has been built by using Artificial Intelligence algorithms. This bot analyses user's query and understands user messages. The system has modules like Online chatbot.

Drawback: the drawback of the system is that it does not allow or process data in more than one language. It uses English language only.

2. (Roel P. Masongsong and Maria Amelia E. Damian , 2016) design and develop an automated Help Desk. This system was deployed to Metro Manila to test its functionality and usefulness.
3. (Binde Ponle Gwanmak1 & Godwin Thomas2, 2020) development of a Helpdesk Support System used for useful information distribution In Jos University.
4. (Nitesh Thakur, Akshay Hiwrale, Sourabh Selote, Abhijeet Shinde and Prof. Namrata Mahakalkar) developed an artificial chatbot using NLP (Natural Language Processing) which can

be done in two ways the first via written text and the second is via verbal or voice communication. Written communication is much easier than the verbal communication.

Drawback: The drawback of the system is that it does not allow or process data in more than one language. It uses English language only.

5. (Jain, 2019) developed a "College Enquiry Chatbot Using Iterative Model", International Journal of Scientific Engineering and Research (IJSER), ISSN (Online): 2347-3878, Volume 7 Issue 1, January 2019, pp 80-83.

Drawbacks: cannot control the student and staff information at a time and the second one is that it does not allow or process data in more than one language.

6. (Amey Tiwari, Rahul Talekar, Prof.S.M.Patil, 2017) , "College Information Chat Bot System", International Journal of Engineering Research and General Science, ISSN 2091-2730, Volume 5, Issue 2, March-April, 2017, pp 131-137.

7. (Harsh Pawar , Pranav Prabhu, Ajay Yadav, Vincent Mendonca , Joyce Lemos, 2018), "College Enquiry Chatbot Using Knowledge in Database", a chatbot is designed by them using knowledge in database. The proposed system has Online Enquiry and Online Chatbot System. The development is done using various programming languages by creating a user-friendly graphical interface to send and receive response.

2.5 PROPOSED SYSTEM

- Due to some of the drawbacks of some available systems, my proposed system it going to solve a problem of time management and use two languages (Swahili and English) so as to make it easier and give user a choice of his/her language.
- In proposed system we are going to use Machine learning techniques and algorithms to build a proposed system. One of the Algorithm that is going to be used is called LSTM (Long Short-Term Memory) which is the best algorithm that can be applied for both small and large amount of data.

2.6 ARCHITECTURE OF PROPOSED SYSTEM

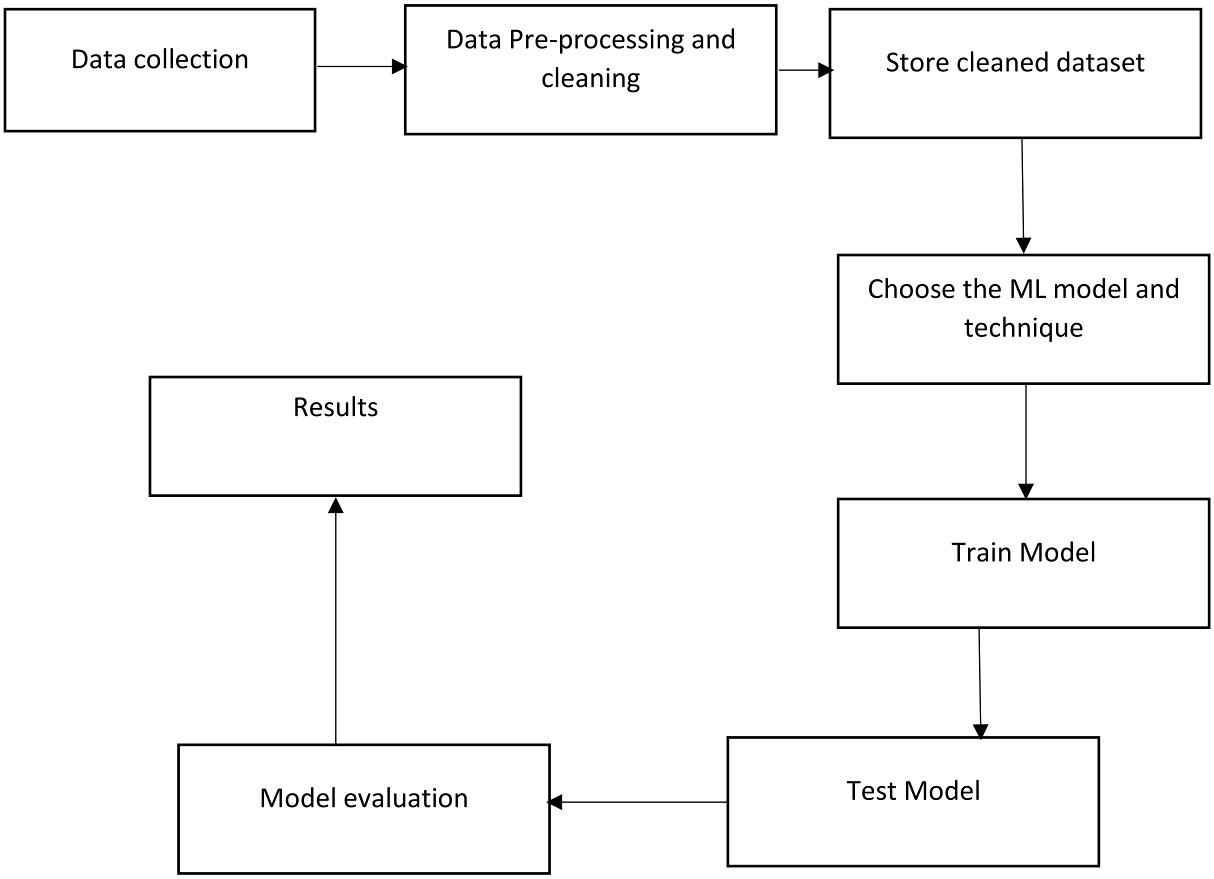


Figure 1. Architecture of proposed system

2.7 SYSTEM MECHANISM (HOW IT WORKS)

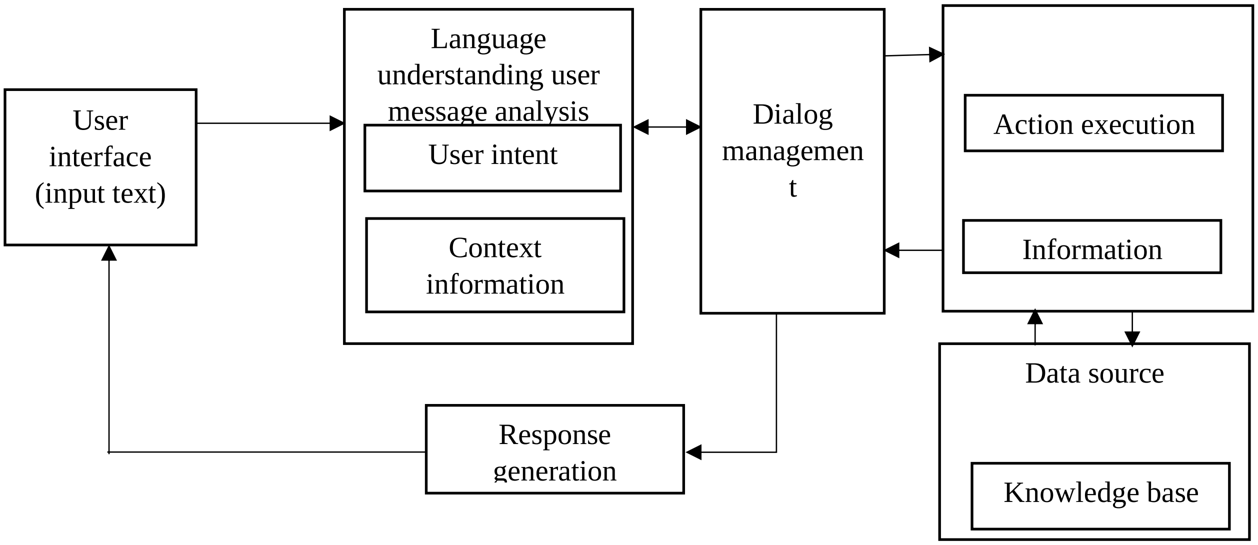


Figure 2. System mechanism (how it works)

2.8 FLOW CHART

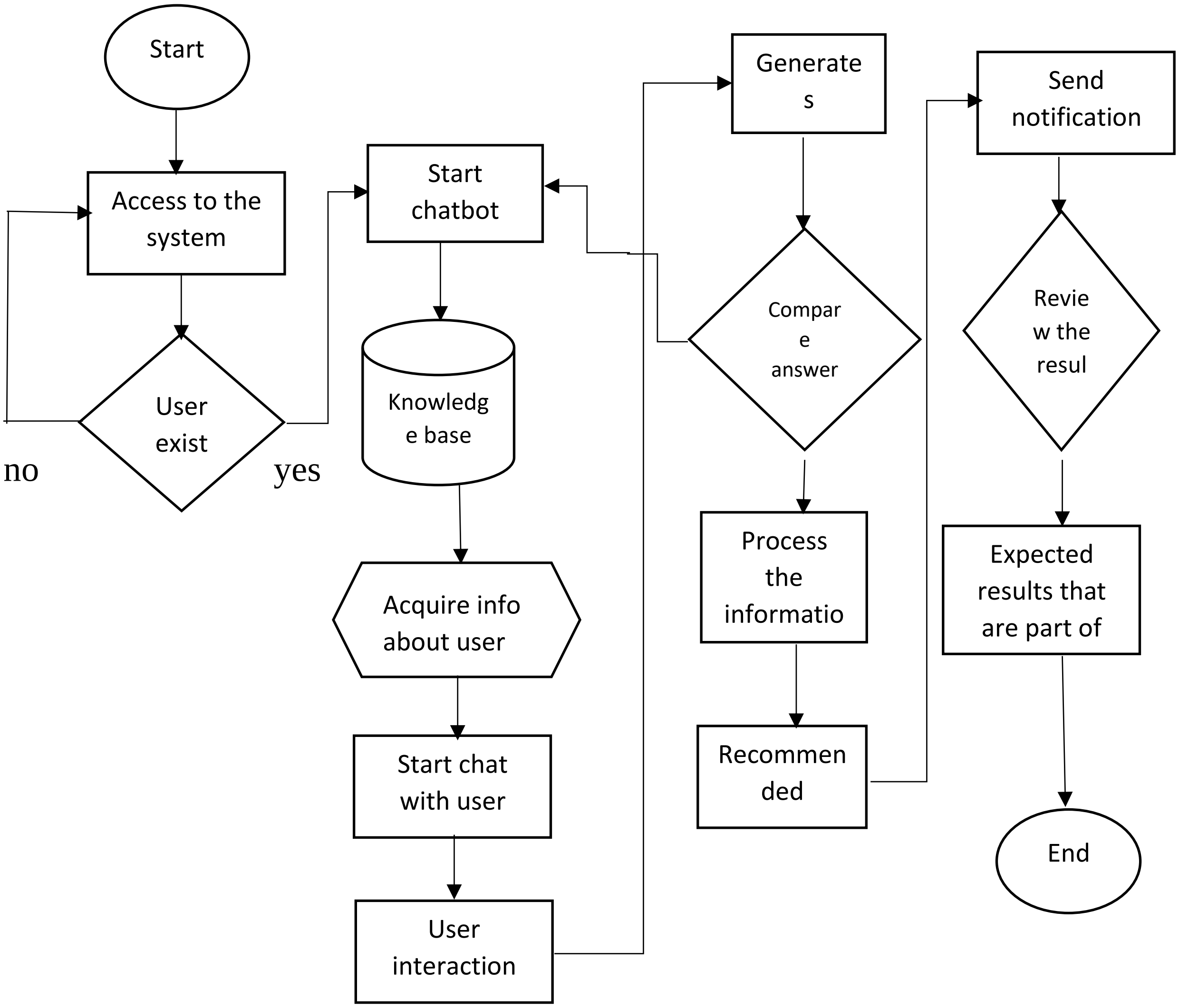


Figure 3. System flow chart

CHAPTER THREE

METHODOLOGY

3.1 INTRODUCTION

This chapter explains different steps that I will use in order to accomplish the objective of my project it shows critically analyzed step by step to be taken during undergoing the project from the beginning to the end. Also, express which kind of method that I can use in data collection and design approach.

3.2 DATA COLLECTION METHODS

Data collection is systematic process which used to describe a process of gathering data and preparing data. The purpose of data collection is to obtain information to keep on record, to make decision about important issues, to pass information on to others.

In order to design the system which will work and be able to produce the required output. Different methods are used in data collection but the mostly used methods are questionnaire, observations, existing records, interviews and experiment.

Information of existing system, the need for new system, parameters and components required for the design of the proposed system has been considered in this part.

There are two types of data that is being used in data collection which are primary data and secondary data.

3.2.1 PRIMARY DATA COLLECTION

Type of data that collected by or that has been generated by the researchers directly through interviews, surveys, experiments, Observations, questionnaires especially designed for understanding and solving the research problem at hand. A primary data collected directly from the original source, so it’s the first-hand data collected from a field.

The proposed system will use this kind of method to collect information through questionnaires. This method takes much time and money because it needs to travel to interview them, and in case of observation it needs more much time to observe until information needed are acquired.

3.2.2 SECONDARY DATA COLLECTION

Data gathered from studies, surveys, or experiments conducted by individuals other than the current researchers for alternative research purpose.

The data has been previously gathered from primary sources and is easily accessible to researchers for their own research purposes.

Source of secondary data includes journal articles, reviews, internet searches, libraries and blogs. In this proposed system also prefer to use documentation to collect secondary data.

3.2.3 QUESTIONNAIRE

Is a set of printed or written questions along with a choice of answers, devised for the purposes of a survey or statistical study and is printed or typed in a sequence form OR Is a list of questions or items used to gather data from respondents about their attitudes, experiences or opinions?

Typically, individuals receive questionnaires through postal mail, where they are requested to respond to the questions and return the completed questionnaire. The recipients are anticipated to read and comprehend the questions and provide their answers in the designated spaces provided within the questionnaire. The questionnaire is carefully designed to present the necessary information in the form of a series of inquiries that can be easily understood and answered by the recipients.

3.2.3.1 SAMPLE QUESTIONS

The following are the sample of the questions that will be used in questionnaire.

1. Do you ever try to use any Intelligent Help Desk system to do registration?
2. Do you think to develop an Intelligent Help Desk System for student’s registration will reduce the manual work?
3. Do you think to develop Intelligent Help Desk will reduce the time for registration process?
4. What is the accurate of the existing system?
5. What is the minimum time which is used by the existing system to complete the registration process?
6. What new system will help you?
    - ☐ Request information instantly.
    - ☐ Ask a question to the system.

☐ To know which step you need to follow from where you end up.

☐ Others, mention here

……………………………………………………………………..

7. How do you recommend this system to be installed in the University?

☐ Highly recommended.

☐ Recommended.

☐ Not recommended.

3.2.3.2 ADVANTAGES OF QUESTIONNAIRE

i. A significant advantage of questionnaires is their consistency: all participants are presented with identical questions.

ii. This method is cost-effective, regardless of the size of the target population.

iii. It eliminates interviewer bias since respondents answer the questions in their own words.

iv. Respondents are afforded ample time to contemplate and provide thoughtful answers.

v. Questionnaires enable broad reach, making it convenient to engage respondents residing in remote locations.

3.3 DOCUMENTATION

Documentation is the material that provides official information or evidence or that serves as a record or evidence provided for information and ideas borrowed from others including primary and secondary sources. I use this method as it enables to gather information about various Algorithms and techniques required to develop the proposed system.

3.4 DESIGN APPROACH

In this project, the method which I will use to establish my project is Prototyping based model. Prototyping model performs Analysis, designing and implementation phase. All these three phases are performed at the same time and repeatedly in a cycle unit up to when the system is complete. I choose to use iterative prototyping model because this model allows to go back and make changes.

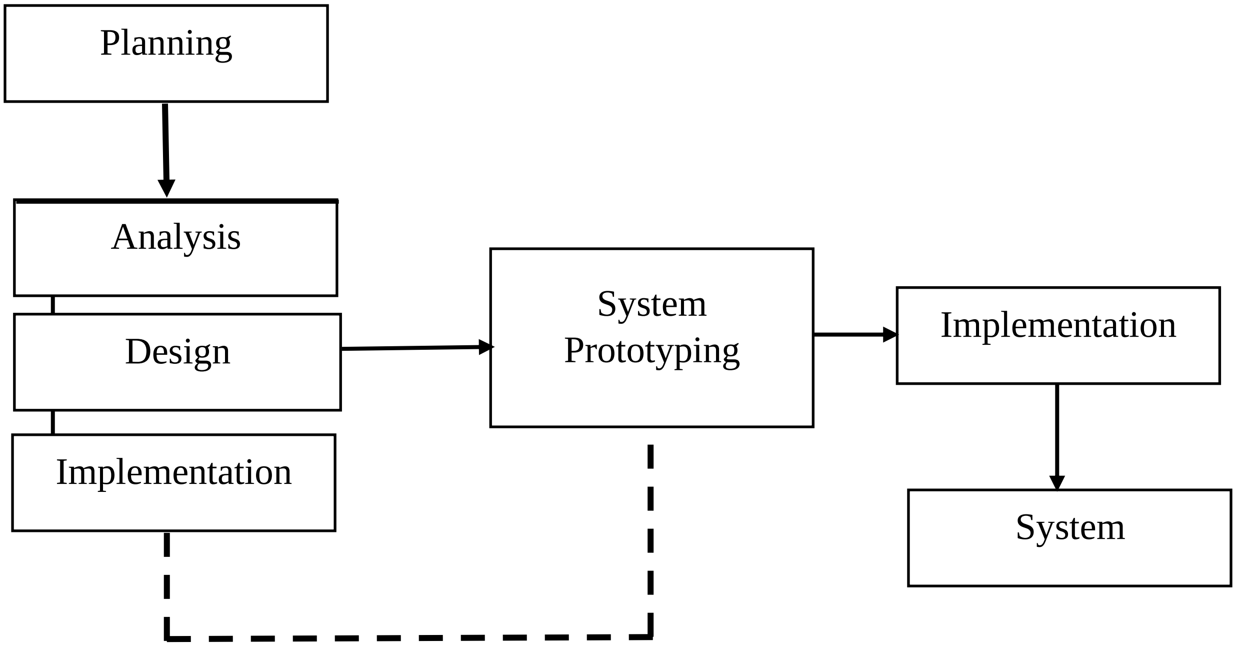


Figure 4. Design approach

3.4.1 ADVANTAGES OF PROTOTYPING-BASED METHODOLOGY.

1) Active involvement: In this model, the users are involved actively in the development phase and, thus, it is easier to develop the model according to their preferences. Since the users are actively involved, errors are detected in the preliminary stages, making the process easier.
2) Easy detection of missing functionality: The missing functionality in the prototype model is easily detectable. As a result of this, the failure risks are reduced.
3) Quick feedback: Since the customer has direct interaction with the prototype model, their feedback is much faster. These feedbacks are essential since they are taken into consideration while creating the final system. The customers can quickly provide their thoughts, report the changes required in the project, and then the developers can modify the project accordingly.
4) Customer satisfaction: The prototype model provides more significant customer satisfaction. The customer can feel the product and make actual contact with it from a very early stage and can also understand the needs of the product much better. They also enjoy the experience of being a part of developing the operating version of their project.
5) Flexibility: The prototype model is flexible. It can be easily changed and modified according to the preferences and needs of the customer or the developer. Furthermore, the developer can reuse the prototype for future use in more complicated projects.

6) Saves money: A prototype model can improve the ability to detect errors in the initial stages of the project. As a result, the inclusive cost and time of the project are reduced. Prototype models enable the developer to foresee the areas of expenditure that were not taken into consideration beforehand. It addresses the changes needed in the project before the adjustment process becomes expensive.

CHAPTER FOUR

DATA COLLECTION AND ANALYSIS

4.1 CHAPTER OVERVIEW

This chapter presents the methods used in collecting and analyzing data for the development of the Intelligent Student Help Desk System. Data collection and analysis are important steps in the development of any system as they help to ensure that the system meets the requirements of the users. This chapter describes the data collection methods, data analysis techniques, and the results of the analysis.

Data collection refers to the systematic gathering and measurement of information related to specific variables of interest. This process enables researchers to address project questions, test hypotheses, and evaluate outcomes in a structured manner. In projects, various tools are employed for data collection. The collected data can be categorized as either primary or secondary data.

Data analysis is an integral and critical component of any research endeavor. It involves the comprehensive examination and summary of the collected data. Data analysis employs analytical and logical reasoning to interpret the gathered information, aiming to identify patterns, relationships, or trends within the dataset.

4.2 PRIMARY DATA

These are data which researcher collects in from the field. Includes questionnaire, interview, survey, observation and experimentation. In this project, questionnaire was used. The following are the questionnaire for Intelligent Student Help Desk Registration System.

QN 1. Have you experienced difficulties with the school registration process?

☐ Yes
☐ No

QN 2. How often have you faced these difficulties?

☐ Day
☐ Week
☐ Monthly

- ☐ Rarely
- ☐ Never

QN 3. How easy or difficult do you find it to obtain information about school registration process?

- ☐ Very easy
- ☐ Easy
- ☐ Moderate
- ☐ Difficult
- ☐ Very difficult

QN 4. Have you ever used Intelligent Student Helpdesk before?

- ☐ Yes
- ☐ No

QN 5. If yes, how satisfied were you with your experience using Intelligent Student Helpdesk?

- ☐ Very satisfied
- ☐ Satisfied
- ☐ Neutral
- ☐ Dissatisfied
- ☐ Very dissatisfied

Intelligent Student Helpdesk Features and Preferences:

QN 6. What features would you like to see in an Intelligent Student Helpdesk Registration System that assist in registration process? (open-ended response)

……………………………………………………………………………………………………

…………………………………………………………………………………………………....

QN 7. How important is it to have an Intelligent Student Helpdesk Registration System that can provide step-by-step guidance through the school registration process?

- ☐ Very important
- ☐ Important
- ☐ Moderate
- ☐ Not important
- ☐ Not at all important

QN 8. How often would you like to receive updates or reminders about important school registration and payment deadlines?

- ☐ Daily
- ☐ Weekly
- ☐ Monthly

QN 9. Would you be willing to use an Intelligent Student Helpdesk Registration System to assist with school registration?

- ☐ Yes
- ☐ No

Additional Feedback:

QN 10. What do you think is the biggest benefit of having Intelligent Student Helpdesk Registration to assist with school registration? (Open-ended response)

……

……

QN 11. Is there any other suggestions or feedback you have for improving the Intelligent Student Helpdesk Registration? (Open-ended response).

……

……

4.3 SECONDARY DATA

These are data that has been already collected and documented by other researchers. They include documentation of scientific procedures, formulas from books, journals, articles and publication.

4.3.1 PROGRAMMING LANGUAGE

Data collected on different Programming Languages

Language	Popularity	Ease of use	Performance	Libraries/Frameworks
Python	Very popular	Easy to learn and use	Good performance	Tensorflow, Pytorch, Scikit-learn, Kera's, Pandas
R	Popular in academia	Easy to learn and use	Good performance	Caret, mlr, Tensorflow
Java	Popular in enterprise	Difficult to learn more than Python and R	Good performance	Deeplearning4j, Weka, MOA
C++	Good performance	Difficult to learn and use	Best performance	Tensorflow, Caffe, Torch
Julia	Growing in popularity	Easy to learn and use	Good performance	Flux.jl, MLJ.jl, Tensorflow.jl

Table 1. Data collected on different Programming Languages.

4.3.2 IDE OR ENVIRONMENT FOR CODING ML

Data collected on different IDE’s

Environment/IDE	Popularity	Ease of use	Functionality	Collaborative features
Jupyter Notebook	Very popular	Easy to learn and use	Interactive notebooks, data visualization	Easy to share and collaborate notebooks
PyCharm	Popular among developers	difficult to learn than Jupyter Notebook	Code completion, debugging, unit testing	Integrates with version control systems
RStudio	Popular among R users	More difficult to learn than Jupyter Notebook	Code completion, debugging, data visualization	Integrates with version control systems
Google colab	Growing in popularity	Easy to learn and use	Interactive notebooks, access to GPUs and TPUs	Easy to share and collaborate notebooks
Visual Studio Code	Growing in popularity	Easy to learn and use	Code completion, debugging, integrated terminals	Integrates with version control systems

Table 2. Data collected on different IDE’s

4.3.3 PROCESSOR

Data collected on different Processor specification

Specification	CPU Cores	Clock Speed	Cache Size	RAM	GPU
Intel Core i5	4-6	2.4-4.2 GHz	6-12 MB	8-16 GB	Integrated GPU
Intel Core i7	4-8	2.6-5.0 GHz	8-16 MB	16-32 GB	Integrated or dedicated GPU
Intel Core i9	6-10	2.9-5.3 GHz	16-20 MB	32-64 GB	Dedicated GPU
AMD Ryzen5	6-8	3.6-4.4 GHz	16 MB	8-16 GB	Integrated GPU
AMD Ryzen7	8-16	3.6-4.9 GHz	16-32 MB	16-32 GB	Integrated or dedicated GPU

Table 3. Data collected on different Processor specification

4.3.4 OPERATING SYSTEM

Data collected on different Operating system

Operating system	Compatibility	Performance	Ease of use	Cost
Windows	Some libraries may have limited support	Good with powerful hardware	User friendly	More affordable than macOS

Linux	Well-supported by many libraries	Excellent performance and customization options	Steeper learning curve	Generally, more affordable than macOS
macOS	Well-supported by some libraries	Good performance with powerful hardware	User friendly	Generally, more expensive than Windows and macOS

Table 4. Data collected on different Operating system

4.3.5 MACHINE LEARNING MODEL/TECHNIQUES

Data collected on different machine learning models/techniques

Model/Technique	**Strengths**	**Weaknesses**	**Why Choose**
Natural Language Processing (NLP)	Excellent for understanding and interpreting natural language queries of students.	May struggle with understanding queries that use unusual or informal language.	Choose NLP if the project involves a lot of textual data and the chatbot needs to understand and respond to student queries.
Convolutional Neural Networks (CNNs)	Ideal for analyzing images and extracting relevant information from them.	May not be useful for projects that do not involve image analysis.	Choose CNNs if the project involves analyzing images, such as documents or forms, related to the registration process.

Recurrent Neural Networks (RNNs)	Great for analyzing and understanding sequences of data, such as text or speech.	May struggle with longer sequences of data.	Choose RNNs if the project involves processing student queries and generating appropriate responses based on context.
Long Short-Term Memory (LSTM)	Excellent for handling long sequences of data and remembering previous interactions.	May be more complex to implement and train compared to other models.	Choose LSTMs if the chatbot needs to remember previous interactions with students and use that information to provide better assistance.
Transformer Models	State-of-the-art models for natural language processing tasks with a large pre-trained language model.	May require a lot of computational power and data for training.	Choose transformer models if you want to achieve state-of-the art performance on natural language processing tasks.
Generative Adversarial Networks (GANs)	Useful for generating synthetic data to train the chatbot.	May not be as effective in generating realistic or representative data compared to other methods.	Choose GANs if you want to generate synthetic data to train the chatbot, such as student queries and responses.

Table 5. Data collected on different machine learning models/techniques

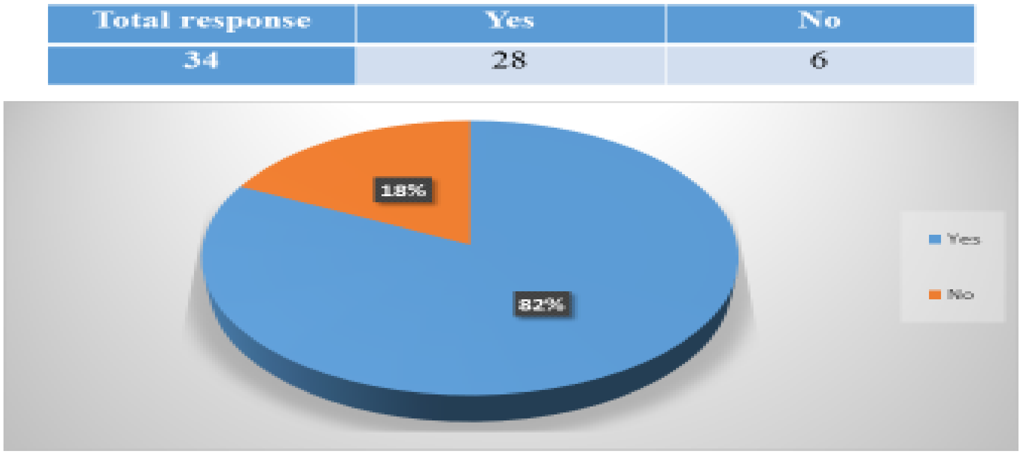
4.4 PRIMARY DATA ANALYSIS

The following are the analysis of the primary data based on the questionnaires.

4.4.1 ANALYSIS OF THE PRIMARY QUESTIONS.

Analysis Question 1: Have you ever experienced difficulties with the school registration process?

- ☐ Yes
- ☐ No



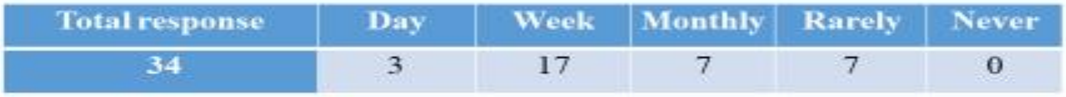
Total response	Yes	No
34	28	6

Figure 5. Analysis question one

The analysis of this question said that 28 said Yes, they experienced difficultness when doing registration which is equal to 82 percent, it seems to be a problem.

Analysis Question 2: How often have you faced these difficulties?

- ☐ Day
- ☐ Week
- ☐ Monthly
- ☐ Rarely
- ☐ Never



Total response	Day	Week	Monthly	Rarely	Never
34	3	17	7	7	0

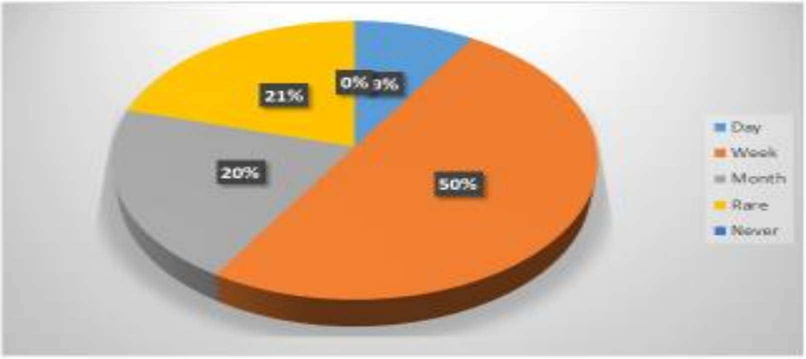
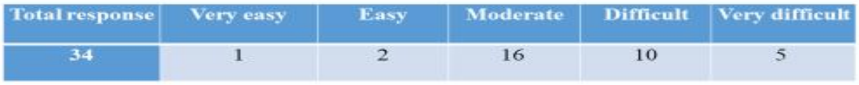


Figure 6. Analysis question two

The analysis of this question said that 24 people said they faced the challenge a week up to month they experienced this difficulty when doing registration which is equal to 70 percent, it seems to be a problem. So, it needs to be reduced to a minimal day.

Analysis Question 3: How easy or difficult do you find it to obtain information about school registration process?

- ☐ Very easy
- ☐ Easy
- ☐ Moderate
- ☐ Difficult
- ☐ Very difficult



Total response	Very easy	Easy	Moderate	Difficult	Very difficult
34	1	2	16	10	5

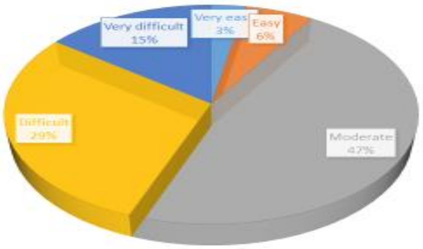


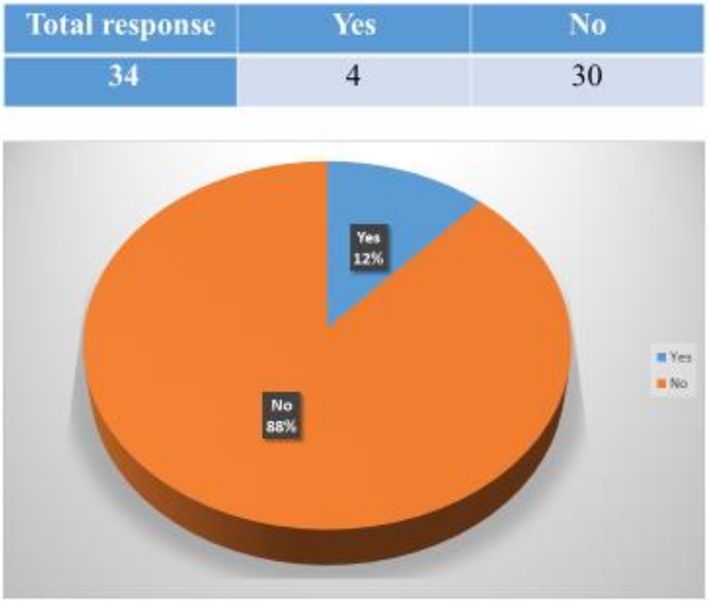
Figure 7. Analysis question three

The analysis of this question said that 16 people said they find this information moderate and 10 people they find it difficult and 5 people find it very difficult when doing registration which is equal to sum of 91 percent, it seems to be a problem. So, it seems that, this information is needed to be accessible easily.

Analysis Question 4: Have you ever used Intelligent Student Helpdesk before?

☐ Yes

☐ No



Total response	Yes	No
34	4	30

Figure 8. Analysis of question four.

The analysis of this question shows that 88 percent they never used help desk, the training or lesson about it is needed to be prepared.

Analysis Question 7: How important is it to have an Intelligent Student Helpdesk Registration System that can provide step-by-step guidance through the school registration process?

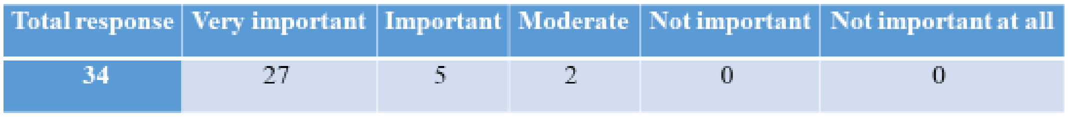
☐ Very important

☐ Important

☐ Moderate

☐ Not important

☐ Not at all important



Total response	Very important	Important	Moderate	Not important	Not important at all
34	27	5	2	0	0

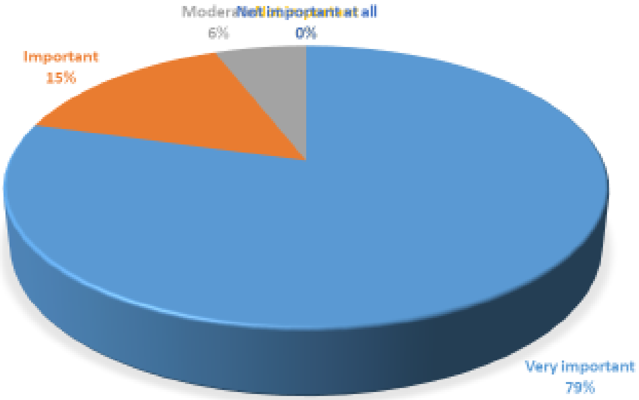


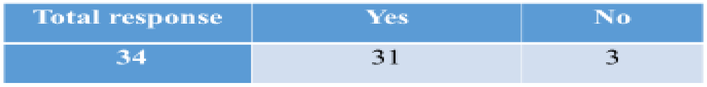
Figure 9. Analysis question seven.

The analysis of this question said that 27 people said it is important to have help desk which is equal to 79 percent.

Analysis Question 9: Will you be willing to use an Intelligent Student Helpdesk Registration System to assist with school registration?

☐ Yes

☐ No



Total response	Yes	No
34	31	3

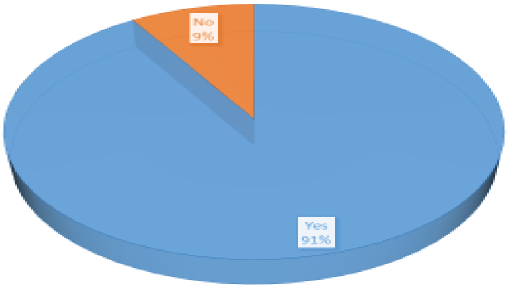


Figure 10. Analysis question nine.

The analysis of this question said that 31 people equal to 91 percent they will be willing to use the system.

Conclusion from Primary Data.

According to the views of the respondents, shows that there is a need to design a system that should be able to provide step by step guide to the students about registration processes and provide useful information when needed and respond to user prompts and queries. Also allocating the venues where it founds.

4.4.2 SECONDARY DATA ANALYSIS

The following are the analysis of the Secondary Data Analysis based on Programming languages, IDEs, Operating systems, Processor and Machine learning techniques/models.

4.4.2.1 GENERAL INFORMATION

1. Demographic data on the student population, such as age, gender, and geographic location. This information can be useful in developing a registration system that caters to the needs of different student groups.

2. Historical data on enrollment trends, such as the number of students who enroll each year, the programs they enroll in, and the time of year when enrollment peaks. This data can help anticipate demand and plan accordingly.(source of data from 2012 to 2018 is from https://www.tcu.go.tz/sites/default/files/Admission%20and%20Graduation%20Statistics.pdf) source of data from 2019/2020 to 2022/2023 is from https://mustnet.ac.tz/home

Year enrolment	2012 to 2013	2013 to 2014	2014 to 2015	2015 to 2016	2016 to 2017	2017 to 2018	2019 to 2023
No. of enrolled students	309	513	696	1053	712	796	8415

Table 6. Enrolment trends of students

3. Financial data on tuition and fees, including historical tuition rates, the costs of specific programs, and any financial aid or scholarship programs available to students. This data can help inform the payment and financial aid options offered in your registration system.

S/N	PROGRAM OFFERED	FEE TYPE	AMOUNT	PAYMENT MODE
1	Engineering courses	Tuition fee	1,100,000	Partial
2	Engineering courses	Direct cost	426,400	Exact
3	Agriculture courses	Tuition fee	900,000	Partial
4	Agriculture courses	Direct cost	426,400	Exact
5	Business Administration courses	Tuition fee	900,000	Partial
6	Business Administration courses	Direct cost	426,400	Exact
7	Education courses	Tuition fee	1,100,000	Partial
8	Education courses	Direct cost	426,400	Exact

Table 7. Fee structure

4. Data on student satisfaction with the registration process, gathered through surveys or feedback forms. This data can help identify pain points in the registration process and inform improvements to your system.

5. Historical data on the registration process, such as the number of students who complete registration on time, the average time it takes to complete registration, and common issues or errors encountered during registration. This data can help you optimize the registration process and improve efficiency for students.

6. Information on the documentation required for registration, such as academic transcripts, identification documents, and medical records. This data can help you design a registration system that clearly communicates what documentation is required and how to provide it.

S/N	TYPE OF DOCUMENT
1	Student Basic information (Name, Registration number, birth date, gender etc.)
2	Contact information (phone, email, address)
3	Next of kin information
4	Education background (certificates)
5	Attachments (passport, disability certificates)
6	Payment reports

Table 8. Important documents needed.

7. Process observation data: You can observe the registration process in real-time to identify bottlenecks, errors, or inefficiencies. This can be done by shadowing students as they complete registration, observing staff members as they assist with registration, or reviewing video recordings of the registration process.

By collecting and analyzing these types of secondary data, you can gain insights into the needs and experiences of your student population, and design a registration system that is user-friendly, efficient, and effective.

3.4.1.1 ANALYSIS ON PROGRAMMING LANGUAGES

Language	Popularity	Ease of use	Performance	Libraries/Frameworks
Python	Very popular	Easy to learn and use	Good performance	Tensorflow, Pytorch, Scikit-learn, Keras, Pandas
R	Popular in academia	Easy to learn and use	Good performance	Caret, mlr, Tensorflow
Java	Popular in enterprise	Difficult to learn more than Python and R	Good performance	Deeplearning4j, Weka, MOA
C++	Good performance	Difficult to learn and use	Best performance	Tensorflow, Caffe, Torch
Julia	Growing in popularity	Easy to learn and use	Good performance	Flux.jl, MLJ.jl, Tensorflow.jl

Conclusion on Programming Language used is Python. As you can see from the table above, Python is the most popular language for machine learning due to its ease of use and vast collection of libraries and frameworks.

3.4.1.2 ANALYSIS ON IDE

Environment/IDE	Popularity	Ease of use	Functionality	Collaborative features
Jupyter Notebook	Very popular	Easy to learn and use	Interactive notebooks, data visualization	Easy to share and collaborate notebooks
PyCharm	Popular among developers	More difficult to learn than Jupyter Notebook	Code completion, debugging, unit testing	Integrates with version control systems
RStudio	Popular among R users	More difficult to learn than Jupyter Notebook	Code completion, debugging, data visualization	Integrates with version control systems
Google colab	Growing in popularity	Easy to learn and use	Interactive notebooks, access to GPUs and TPUs	Easy to share and collaborate notebooks
Visual Studio Code	Growing in popularity	Easy to learn and use	Code completion, debugging, integrated terminals	Integrates with version control systems

Conclusion on IDE to use is Jupyter notebook. As you can see from the table above, Jupyter Notebook is the most popular environment for coding machine learning due to its easy to use and interactive notebooks. It’s also easy to share and collaborate notebooks with others. Also it is accessible for offline mode to access libraries.

4.4.2.4 ANALYSIS ON PROCESSOR

Specification	CPU Cores	Clock Speed	Cache Size	RAM	GPU
Intel Core i5	4-6	2.4-4.2 GHz	6-12 MB	8-16 GB	Integrated GPU
Intel Core i7	4-8	2.6-5.0 GHz	8-16 MB	16-32 GB	Integrated or dedicated GPU
Intel Core i9	6-10	2.9-5.3 GHz	16-20 MB	32-64 GB	Dedicated GPU
AMD Ryzen5	6-8	3.6-4.4 GHz	16 MB	8-16 GB	Integrated GPU
AMD Ryzen7	8-16	3.6-4.9 GHz	16-32 MB	16-32 GB	Integrated or dedicated GPU

Conclusion to Processor used is Intel Core i7. In general, a Processor with more CPU cores, a high clock speed and a larger cache size is better for machine learning tasks. In addition, having sufficient RAM and a dedicated GPU can greatly improve performance especially for deep learning tasks. However, the choice of machine with good processor it depends on the size of data and complexity of your dataset and model as well as your budget. In our case we use Intel Core i7, Processor 2.3 and RAM 4 GB is enough.

4.4.2.5 ANALYSIS ON OPERATING SYSTEM TO USE

Operating system	Compatibility	Performance	Ease of use	Cost
Windows	Some libraries may have limited support	Good with powerful hardware	User friendly	More affordable than macOS

Linux	Well-supported by many libraries	Excellent performance and customization options	Steeper learning curve	Generally more affordable than macOS
macOS	Well-supported by some libraries	Good performance with powerful hardware	User friendly	Generally more expensive than Windows and macOS

Conclusion on Operating system used is Window and Linux. It's important to note that the specific factors to consider when choosing an operating system for machine learning may vary depending on your specific needs and preferences. For example, if you already have experience with Linux and prefer it over other operating systems, then the learning curve may not be as much of a factor for you. Similarly, if you are working with certain libraries or frameworks that are better-supported on a specific operating system, then that may be the deciding factor for your choice of operating system. In this project we use both Windows and Linux.

4.4.2.6 ANALYSIS ON MACHINE LEARNING MODEL/TECHNIQUES

Model/Technique	Strengths	Weaknesses	Why Choose
Natural Language Processing (NLP)	Excellent for understanding and interpreting natural language queries of students.	May struggle with understanding queries that use unusual or informal language.	Choose NLP if the project involves a lot of textual data and the chatbot needs to understand and respond to student queries.
Convolutional Neural Networks (CNNs)	Ideal for analyzing images and extracting relevant information from them.	May not be useful for projects that do not involve image analysis.	Choose CNNs if the project involves analyzing images, such as documents or

			forms, related to the registration process.
Recurrent Neural Networks (RNNs)	Great for analyzing and understanding sequences of data, such as text or speech.	May struggle with longer sequences of data.	Choose RNNs if the project involves processing student queries and generating appropriate responses based on context.
Long Short-Term Memory (LSTM)	Excellent for handling long sequences of data and remembering previous interactions.	May be more complex to implement and train compared to other models.	Choose LSTMs if the chatbot needs to remember previous interactions with students and use that information to provide better assistance.
Transformer Models	State-of-the-art models for natural language processing tasks with a large pre-trained language model.	May require a lot of computational power and data for training.	Choose transformer models if you want to achieve state-of-the art performance on natural language processing tasks.
Generative Adversarial Networks (GANs)	Useful for generating synthetic data to train the chatbot.	May not be as effective in generating realistic or representative data compared to other methods.	Choose GANs if you want to generate synthetic data to train the chatbot, such as student queries and responses.

Conclusion on Machine learning model we are going to use is Sequential model and NLP. Choosing the model or technique, it will depend on the specific requirements and goals of the project. For example, if the project involves a lot of textual data and the chatbot needs to understand and respond to student queries, then NLP or RNNs may be more suitable. If the goal is to achieve state-of-the-art performance on natural language processing tasks, then transformer models may be the best choice. It's important to carefully consider the strengths and weaknesses of each model/technique and choose the one that best fits the project's requirements and goals.

CHAPTER FIVE

SYSTEM DESIGN AND IMPLEMENTATION

5.1 CHAPTER OVERVIEW

System design involves the establishment of various system elements, such as modules, architecture, components, interfaces, and data, in accordance with the specified requirements.

5.1.1 OBJECTIVE AND GOALS OF THE PROJECT

The objective of the Intelligent Student Help Desk System for Registration is to provide an automated solution for students to easily navigate the registration process. The system aims to enhance the user experience, provide accurate information, and assist students in resolving registration-related queries and help them in giving procedures to follow for registration.

In the context of this project, system design encompasses the process of determining the elements of a system, including modules, architecture, components, interfaces, and data, in alignment with the specified project requirements. Specifically, the system design phase will involve creating and documenting the Use Case diagram, developing a Dataflow Chart, and designing the architecture that will be utilized. These activities are essential for establishing a clear understanding of the system's functionality, data flow, and structural composition, ensuring effective implementation and successful project outcomes.

5.1.2 USE CASE DIAGRAM

A use case diagram is a visual representation illustrating the potential interactions between a user and a system. It presents different scenarios of user actions and the corresponding system responses. Use case diagrams typically encompass a range of use cases and diverse user roles, often accompanied by additional diagram types to enhance understanding. Use cases are commonly depicted as circles or ellipses, while actors are typically represented as stick figures. The Use case of this project is as follows;

- ➢ Developer use case.
- ➢ User use case.

Developer use case.

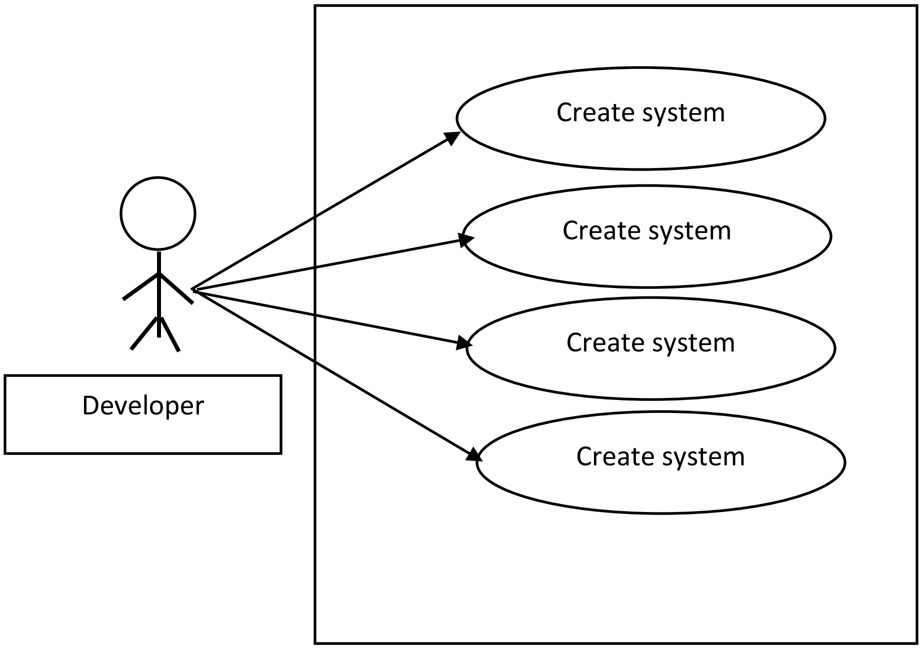


Figure 11. Developer use case.

User use case.

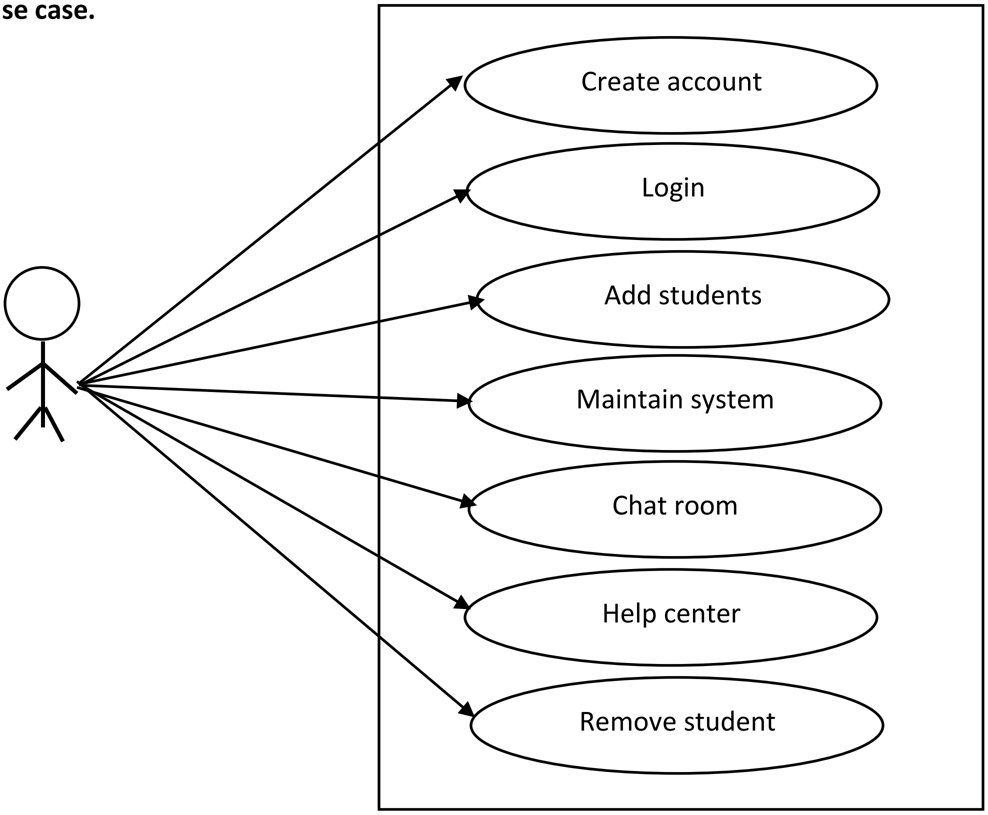
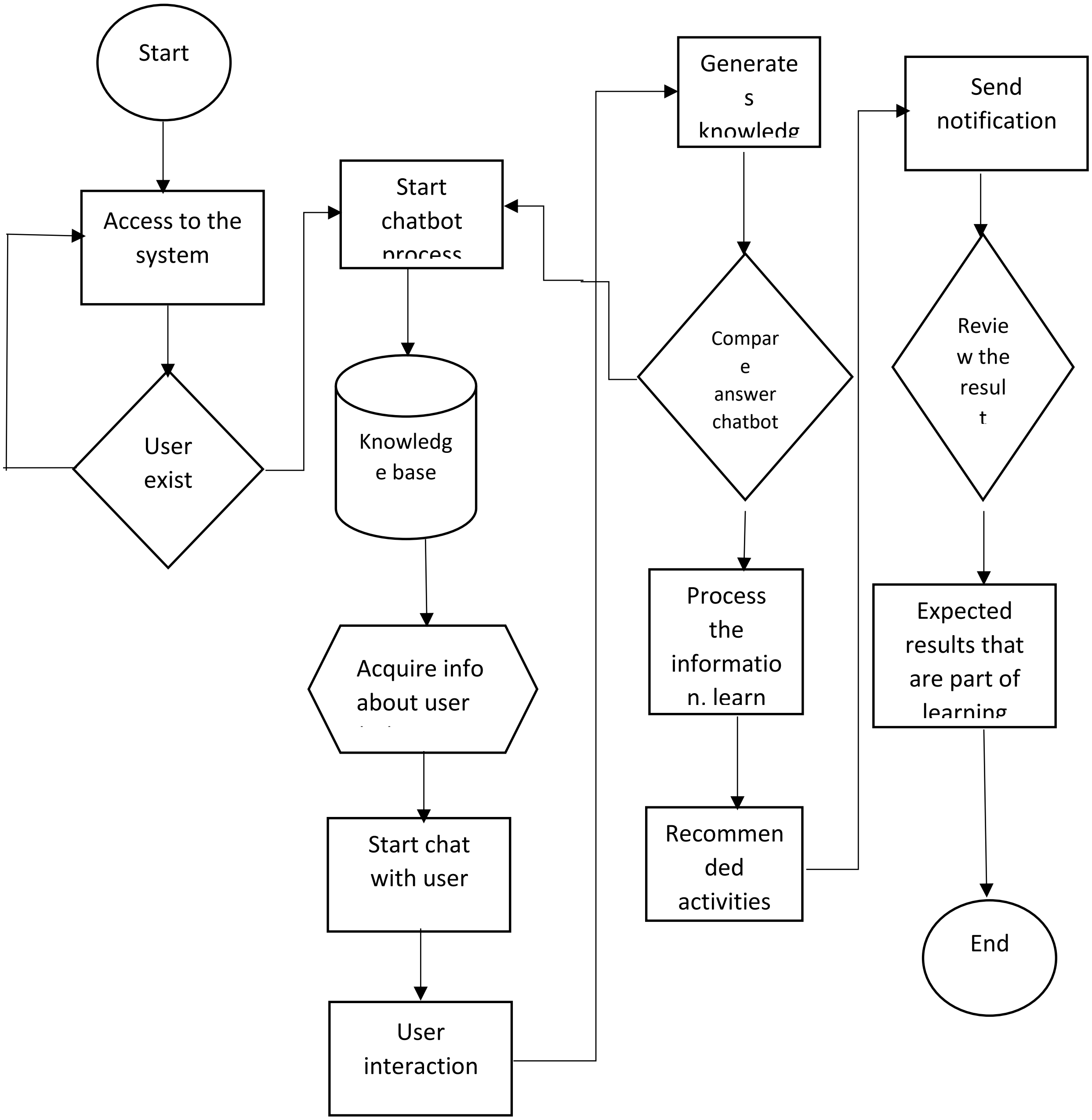


Figure 12. User use case.

FLOW CHART


Start
Access to the system
User exist
Start chatbot process
Knowledge base
Acquire info about user
Start chat with user
User interaction
Generates knowledg
Compare answer chatbot
Process the information. learn
Recommended activities
Send notification
Review the result
Expected results that are part of learning
End

5.1.3 INTERACTION BETWEEN COMPONENTS

User Interface: The user interface allows students to interact with the system. It provides a chat-like interface where students can enter their queries or requests related to registration.

Natural Language Processing Module: The input from the user interface is processed by the natural language processing module. This module is responsible for understanding the user's intent and extracting relevant information from their queries.

Knowledge base: The database/knowledge base stores relevant information, such as course schedules, registration rules, and frequently asked questions. It provides the necessary data to the backend server for generating responses.

5.2 SYSTEM DESIGN

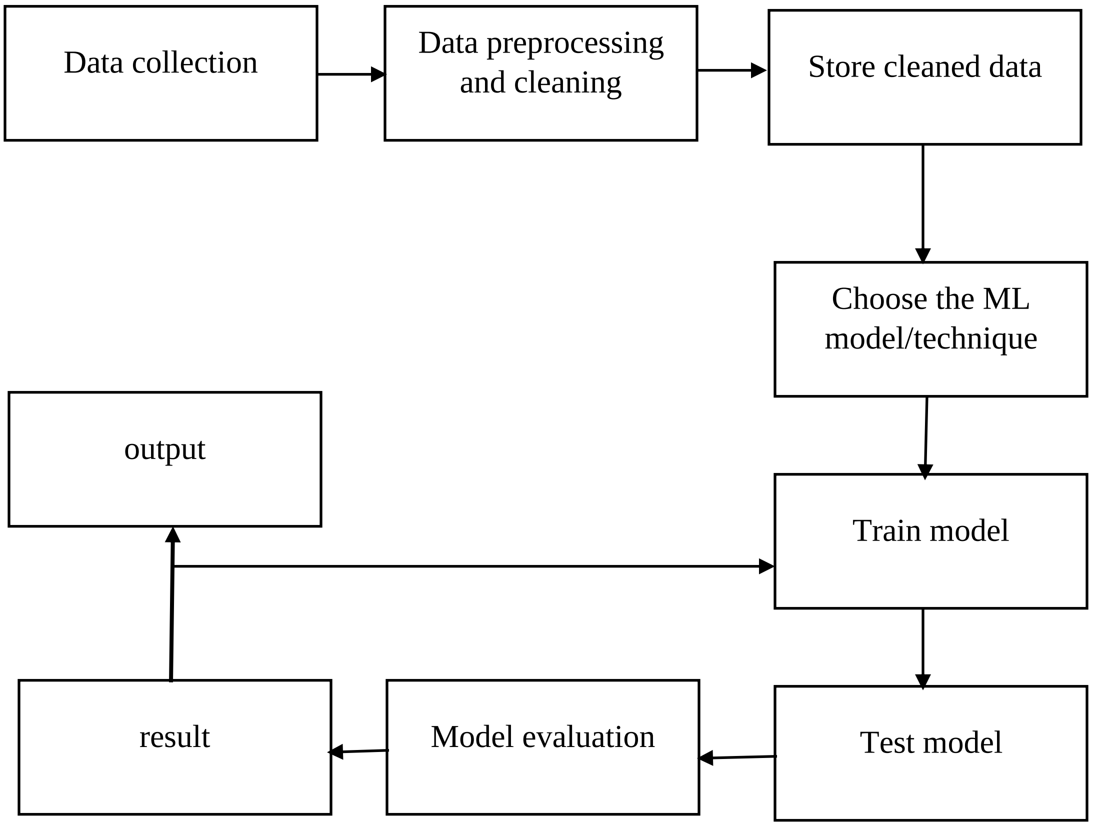


Figure 13. System design.

5.3 SYSTEM IMPLEMENTATION

Frameworks, Libraries, and Technologies used:

Python: The implementation of the system utilizes Python as the primary programming language.

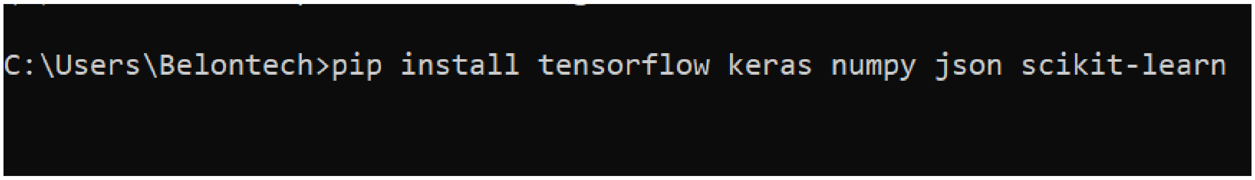
Natural Language Processing (NLP) Libraries: The system uses NLP libraries, such as NLTK (Natural Language Toolkit), for tasks like tokenization, lemmatization, and intent classification.

Keras and TensorFlow: These frameworks are employed for training and deploying the chatbot model used in the natural language processing module. Tensorflow, Keras and Scikit-learn frameworks is chosen because it is important to consider factors such as the ease of use, availability of resources and documentation, community support, scalability, and the availability of pre-trained models or APIs.

JSON: The system uses JSON as the data format for storing intents, responses, and other relevant information.

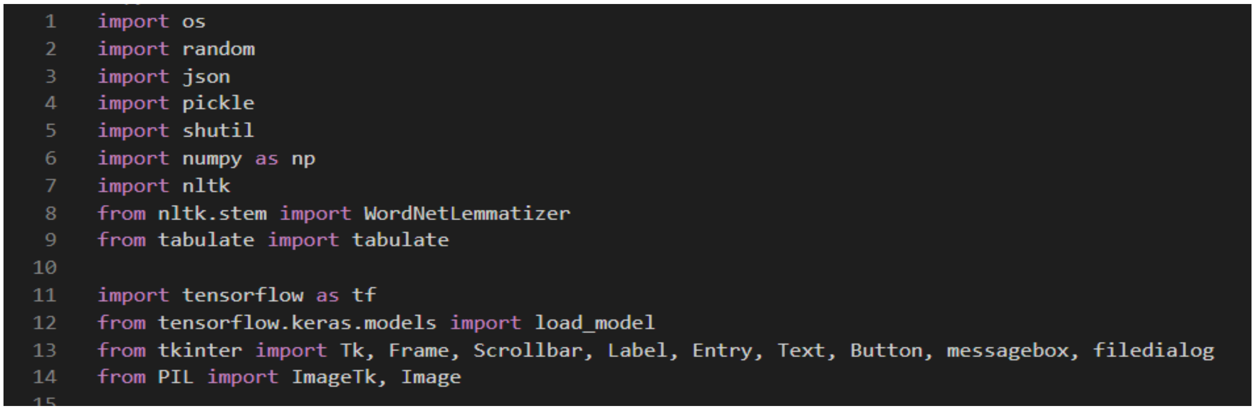
So, to work with these frameworks and libraries you need to install and import in the file.

This is how to install the libraries and framework.



```
C:\Users\Belontech>pip install tensorflow keras numpy json scikit-learn
```

After being installing, you need to import in the you want to use it.



```python
import os
import random
import json
import pickle
import shutil
import numpy as np
import nltk
from nltk.stem import WordNetLemmatizer
from tabulate import tabulate

import tensorflow as tf
from tensorflow.keras.models import load_model
from tkinter import Tk, Frame, Scrollbar, Label, Entry, Text, Button, messagebox, filedialog
from PIL import ImageTk, Image
```

Figure 14. Libraries to work with.

5.3.1 DEVELOPMENT PROCESS AND METHODOLOGIES

Iterative prototyping: An iterative development methodology is adopted to facilitate iterative development, continuous feedback, and frequent updates based on user requirements and feedback and test performed.

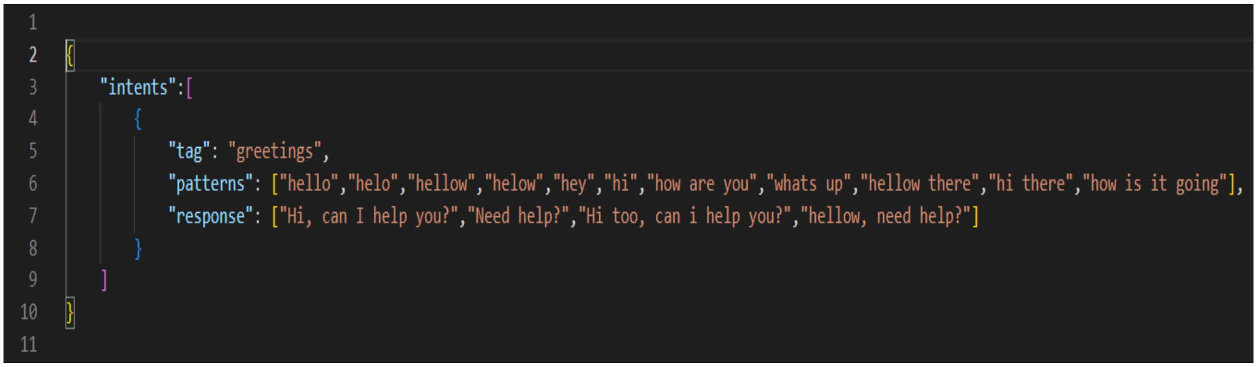
Iterative Testing and Refinement: The system undergoes iterative testing to identify and fix issues, improve accuracy in intent classification, and enhance the chatbot's responses.

In the development process, first data is needed to be collected first in json file so as to be easy to be readable in machine learning models. As json collects data in form of having a Key and Value for that key. And in this json file all of the intent is in intents file having a tag, patterns and responses and you can add more intents as you want as per your needs of project and having this more intent classes means you have a lot of data for the bot to learn and understand the functionalities.

Tag: refers to a unique identifier or label assigned to an intent. It helps categorize and identify different user queries or statements. Each tag represents a specific intention or topic that the chatbot is trained to recognize and respond to.

Patterns: These are the examples or messages or text input of user queries or statements that correspond to a particular intent. These patterns serve as training data for the chatbot, helping it learn and recognize user inputs associated with specific intents.

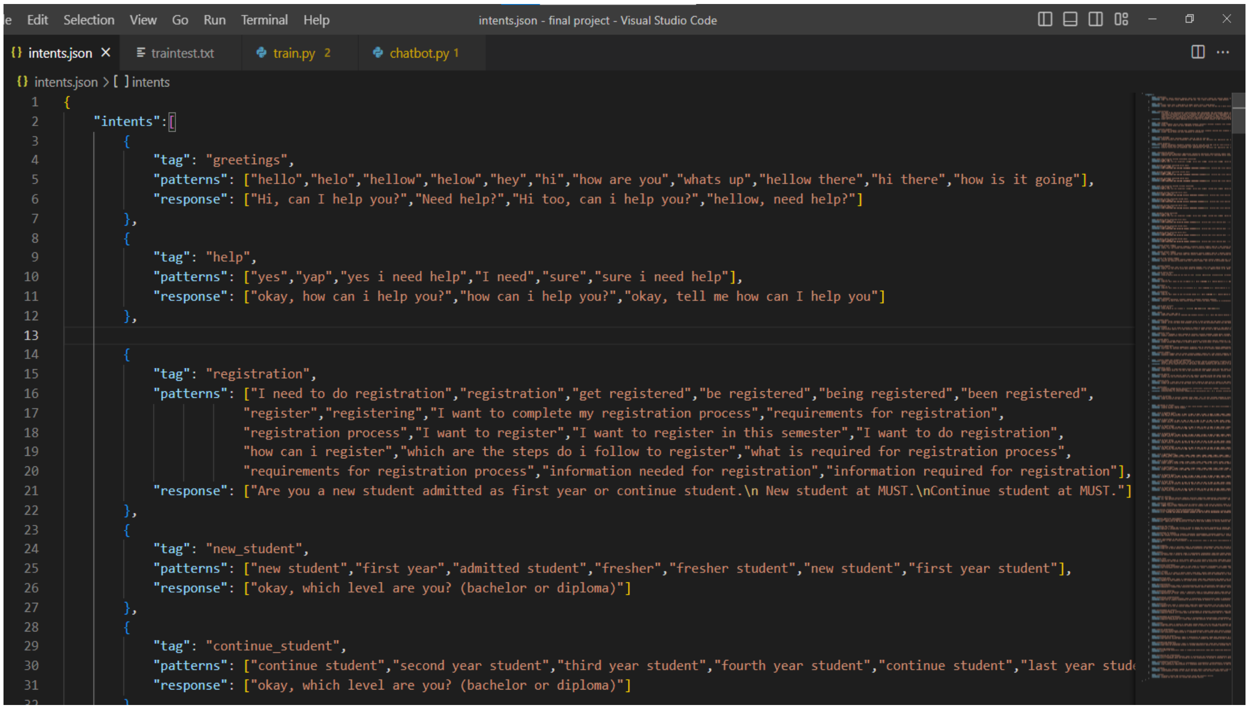
Responses: These are the predefined messages or actions that the chatbot provides when it recognizes a specific intent in the user input.



```

{
    "intents":[
        {
            "tag": "greetings",
            "patterns": ["hello","helo","hellow","helow","hey","hi","how are you","whats up","hellow there","hi there","how is it going"],
            "response": ["Hi, can I help you?","Need help?","Hi too, can i help you?","hellow, need help?"]
        }
    ]
}

```

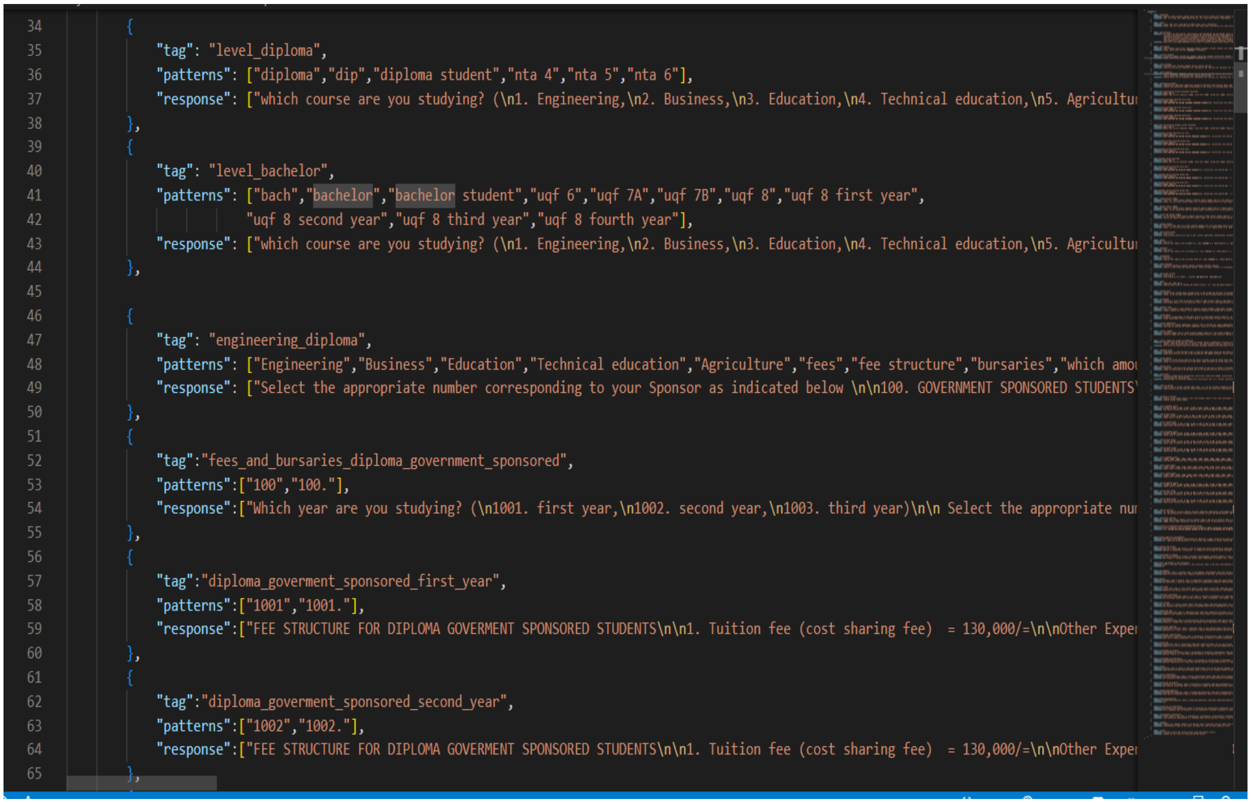
Figure 15. Example of intents.

This is the example of the intents in the json file that contains information to for training.

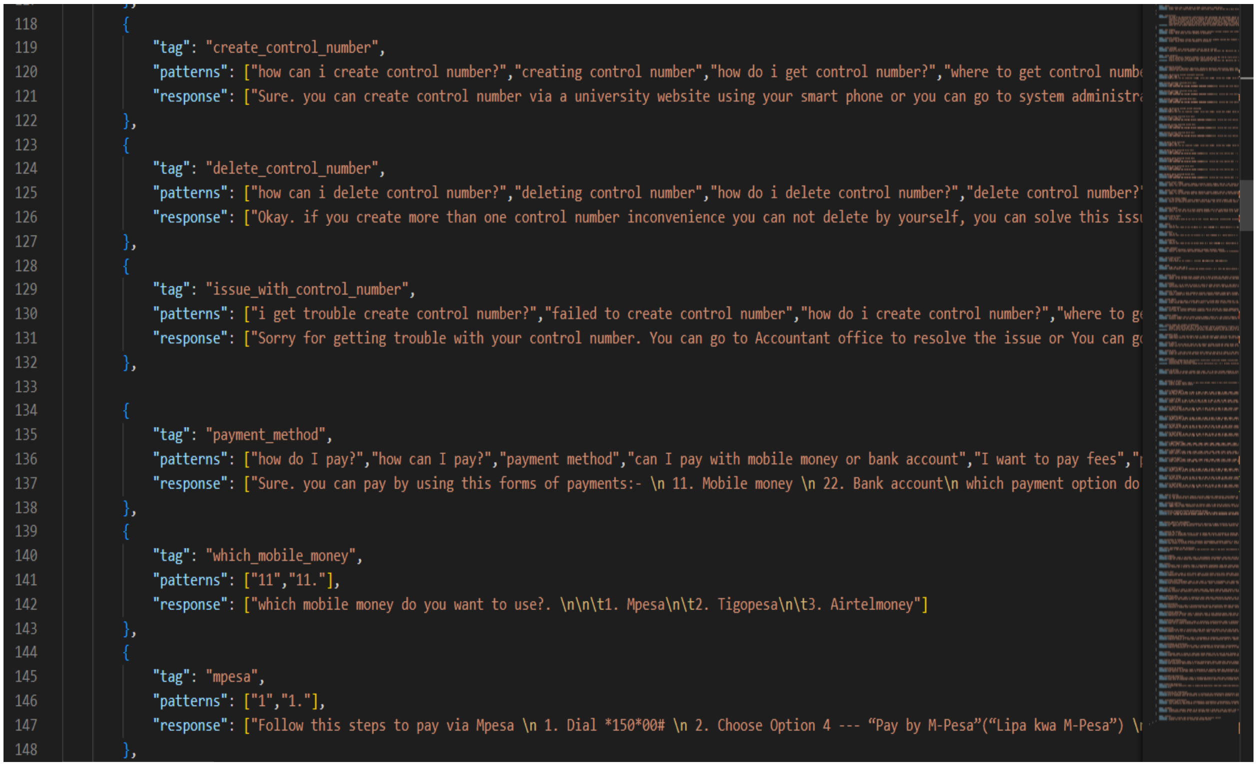

e Edit Selection View Go Run Terminal Help
intents.json - final project - Visual Studio Code
intents.json
traintest.txt
train.py 2
chatbot.py 1
intents.json > [ ] intents

```json
{
    "intents":[
        {
            "tag": "greetings",
            "patterns": ["hello","helo","hellow","helow","hey","hi","how are you","whats up","hellow there","hi there","how is it going"],
            "response": ["Hi, can I help you?","Need help?","Hi too, can i help you?","hellow, need help?"]
        },
        {
            "tag": "help",
            "patterns": ["yes","yap","yes i need help","I need","sure","sure i need help"],
            "response": ["okay, how can i help you?","how can i help you?","okay, tell me how can I help you"]
        },

        {
            "tag": "registration",
            "patterns": ["I need to do registration","registration","get registered","be registered","being registered","been registered",
                        "register","registering","I want to complete my registration process","requirements for registration",
                        "registration process","I want to register","I want to register in this semester","I want to do registration",
                        "how can i register","which are the steps do i follow to register","what is required for registration process",
                        "requirements for registration process","information needed for registration","information required for registration"],
            "response": ["Are you a new student admitted as first year or continue student.\n New student at MUST.\nContinue student at MUST."]
        },
        {
            "tag": "new_student",
            "patterns": ["new student","first year","admitted student","fresher","fresher student","new student","first year student"],
            "response": ["okay, which level are you? (bachelor or diploma)"]
        },
        {
            "tag": "continue_student",
            "patterns": ["continue student","second year student","third year student","fourth year student","continue student","last year stude
            "response": ["okay, which level are you? (bachelor or diploma)"]
```

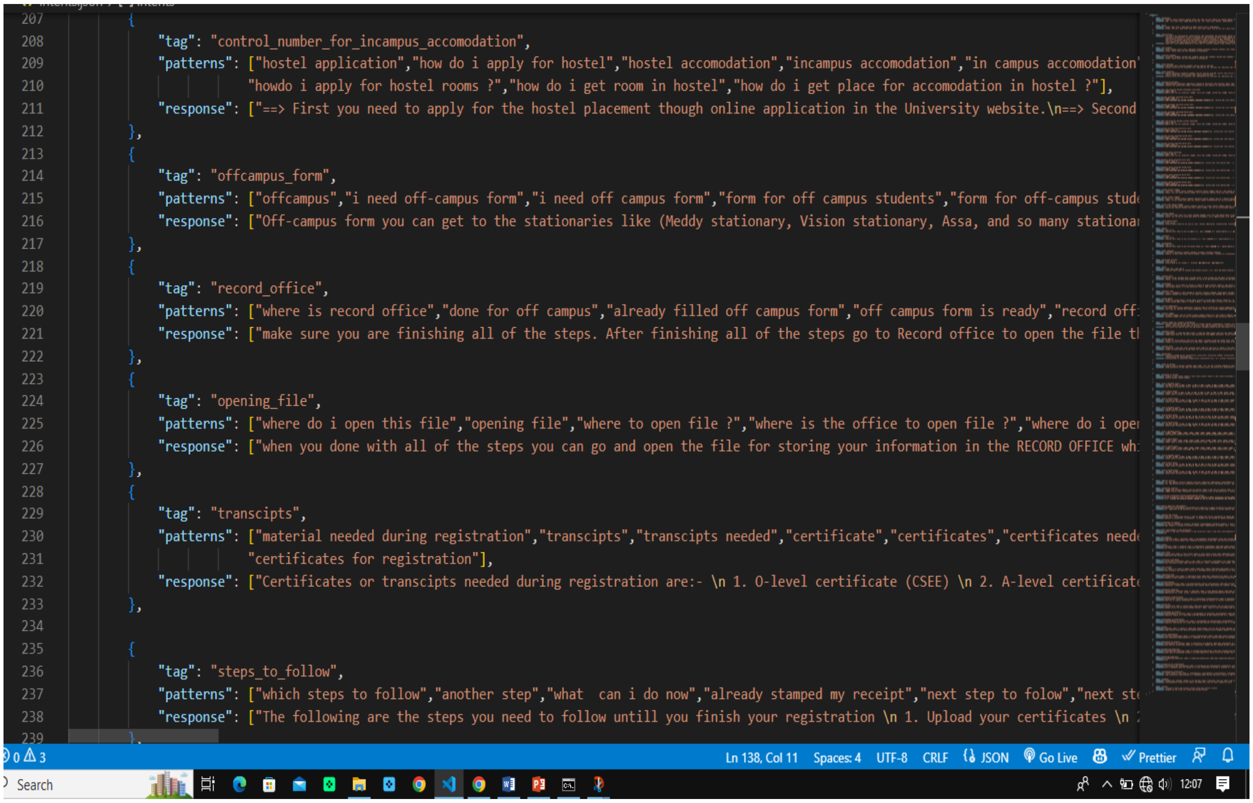
Figure 16. Image of intents. json file.



```json
        {
            "tag": "level_diploma",
            "patterns": ["diploma","dip","diploma student","nta 4","nta 5","nta 6"],
            "response": ["which course are you studying? (\n1. Engineering,\n2. Business,\n3. Education,\n4. Technical education,\n5. Agricultu
        },
        {
            "tag": "level_bachelor",
            "patterns": ["bach","bachelor","bachelor student","uqf 6","uqf 7A","uqf 7B","uqf 8","uqf 8 first year",
                        "uqf 8 second year","uqf 8 third year","uqf 8 fourth year"],
            "response": ["which course are you studying? (\n1. Engineering,\n2. Business,\n3. Education,\n4. Technical education,\n5. Agricultu
        },

        {
            "tag": "engineering_diploma",
            "patterns": ["Engineering","Business","Education","Technical education","Agriculture","fees","fee structure","bursaries","which amou
            "response": ["Select the appropriate number corresponding to your Sponsor as indicated below \n\n100. GOVERNMENT SPONSORED STUDENTS
        },
        {
            "tag":"fees_and_bursaries_diploma_government_sponsored",
            "patterns":["100","100."],
            "response":["Which year are you studying? (\n1001. first year,\n1002. second year,\n1003. third year)\n\n Select the appropriate nu
        },
        {
            "tag":"diploma_goverment_sponsored_first_year",
            "patterns":["1001","1001."],
            "response":["FEE STRUCTURE FOR DIPLOMA GOVERMENT SPONSORED STUDENTS\n\n1. Tuition fee (cost sharing fee)  = 130,000/=\n\nOther Exper
        },
        {
            "tag":"diploma_goverment_sponsored_second_year",
            "patterns":["1002","1002."],
            "response":["FEE STRUCTURE FOR DIPLOMA GOVERMENT SPONSORED STUDENTS\n\n1. Tuition fee (cost sharing fee)  = 130,000/=\n\nOther Exper
        },
```

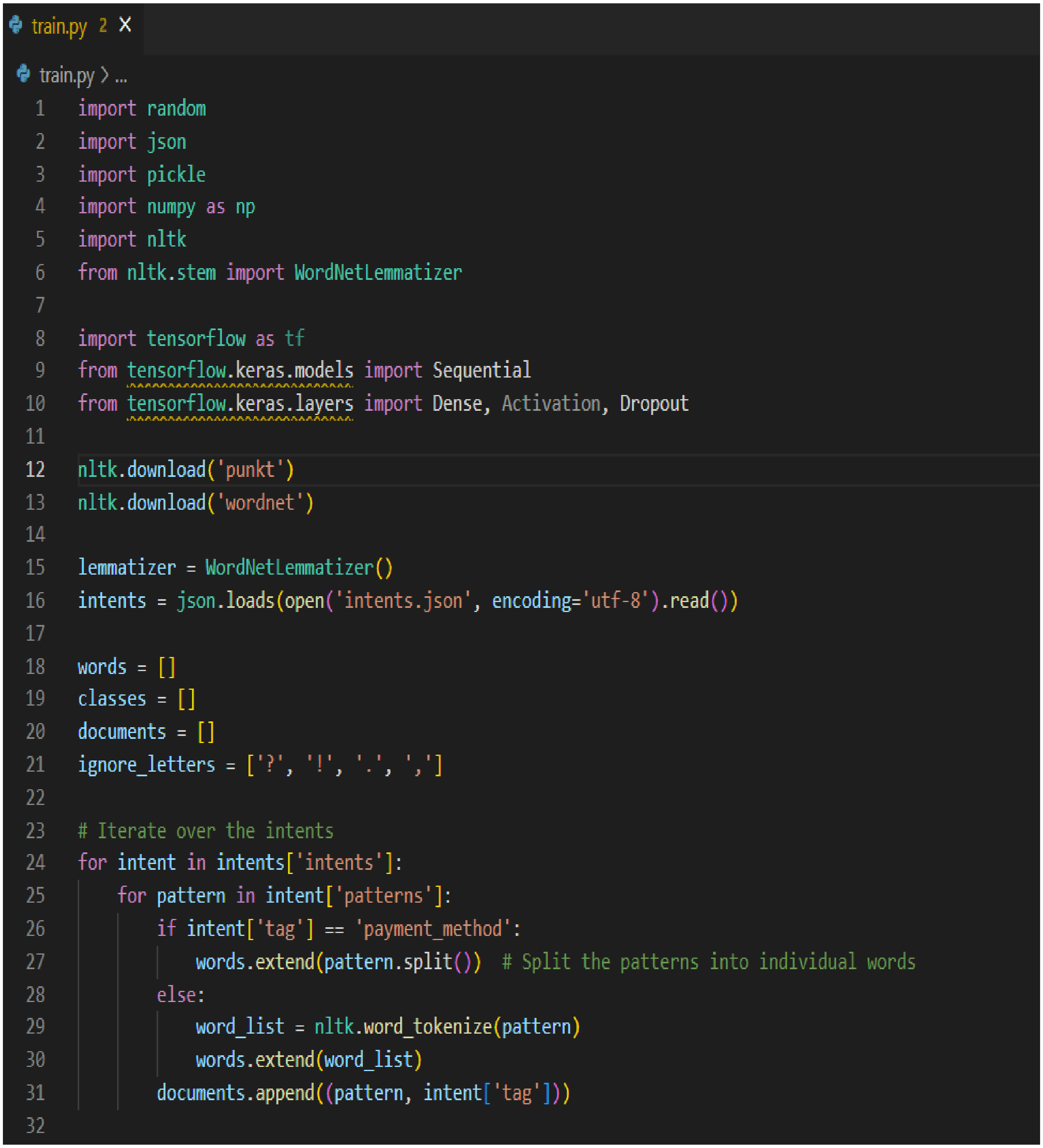


```
    },
    {
        "tag": "create_control_number",
        "patterns": ["how can i create control number?","creating control number","how do i get control number?","where to get control numbe
        "response": ["Sure. you can create control number via a university website using your smart phone or you can go to system administra
    },
    {
        "tag": "delete_control_number",
        "patterns": ["how can i delete control number?","deleting control number","how do i delete control number?","delete control number?'
        "response": ["Okay. if you create more than one control number inconvenience you can not delete by yourself, you can solve this issu
    },
    {
        "tag": "issue_with_control_number",
        "patterns": ["i get trouble create control number?","failed to create control number","how do i create control number?","where to ge
        "response": ["Sorry for getting trouble with your control number. You can go to Accountant office to resolve the issue or You can go
    },

    {
        "tag": "payment_method",
        "patterns": ["how do I pay?","how can I pay?","payment method","can I pay with mobile money or bank account","I want to pay fees","
        "response": ["Sure. you can pay by using this forms of payments:- \n 11. Mobile money \n 22. Bank account\n which payment option do
    },
    {
        "tag": "which_mobile_money",
        "patterns": ["11","11."],
        "response": ["which mobile money do you want to use?. \n\n\t1. Mpesa\n\t2. Tigopesa\n\t3. Airtelmoney"]
    },
    {
        "tag": "mpesa",
        "patterns": ["1","1."],
        "response": ["Follow this steps to pay via Mpesa \n 1. Dial *150*00# \n 2. Choose Option 4 --- “Pay by M-Pesa”(“Lipa kwa M-Pesa”) \r
    },
```



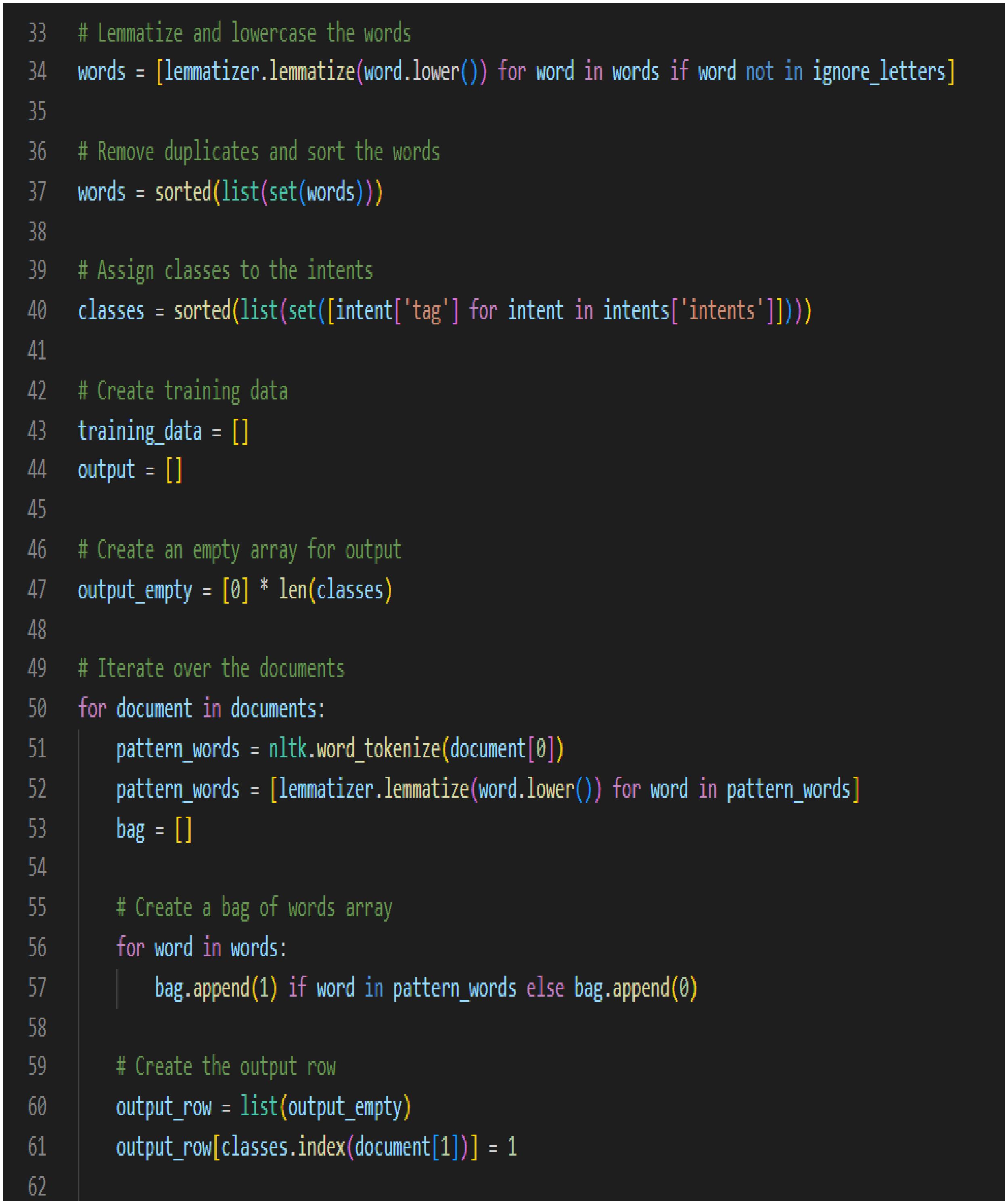
```
    {
        "tag": "control_number_for_incampus_accomodation",
        "patterns": ["hostel application","how do i apply for hostel","hostel accomodation","incampus accomodation","in campus accomodation'
                    "howdo i apply for hostel rooms ?","how do i get room in hostel","how do i get place for accomodation in hostel ?"],
        "response": ["==> First you need to apply for the hostel placement though online application in the University website.\n==> Second
    },
    {
        "tag": "offcampus_form",
        "patterns": ["offcampus","i need off-campus form","i need off campus form","form for off campus students","form for off-campus stude
        "response": ["Off-campus form you can get to the stationaries like (Meddy stationary, Vision stationary, Assa, and so many stationar
    },
    {
        "tag": "record_office",
        "patterns": ["where is record office","done for off campus","already filled off campus form","off campus form is ready","record off
        "response": ["make sure you are finishing all of the steps. After finishing all of the steps go to Record office to open the file th
    },
    {
        "tag": "opening_file",
        "patterns": ["where do i open this file","opening file","where to open file ?","where is the office to open file ?","where do i oper
        "response": ["when you done with all of the steps you can go and open the file for storing your information in the RECORD OFFICE wh
    },
    {
        "tag": "transcipts",
        "patterns": ["material needed during registration","transcipts","transcipts needed","certificate","certificates","certificates neede
                    "certificates for registration"],
        "response": ["Certificates or transcipts needed during registration are:- \n 1. O-level certificate (CSEE) \n 2. A-level certificate
    },

    {
        "tag": "steps_to_follow",
        "patterns": ["which steps to follow","another step","what  can i do now","already stamped my receipt","next step to folow","next ste
        "response": ["The following are the steps you need to follow untill you finish your registration \n 1. Upload your certificates \n
    },
```

Secondly, after having the data for training the model, now we need to feed this data to the Machine learning model for training purpose so as it can understand. Here we need to create the model for training.

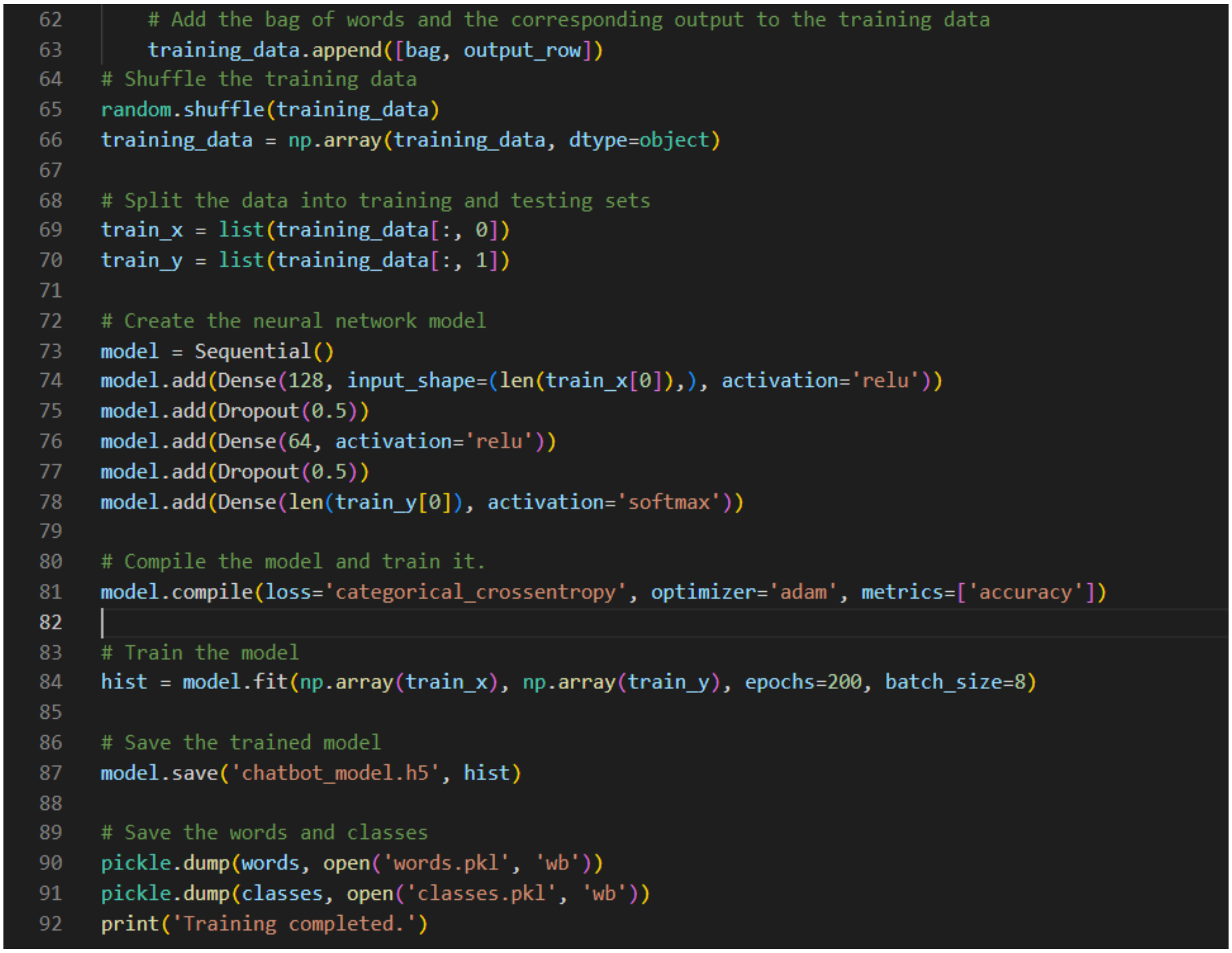


```python
import random
import json
import pickle
import numpy as np
import nltk
from nltk.stem import WordNetLemmatizer

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Activation, Dropout

nltk.download('punkt')
nltk.download('wordnet')

lemmatizer = WordNetLemmatizer()
intents = json.loads(open('intents.json', encoding='utf-8').read())

words = []
classes = []
documents = []
ignore_letters = ['?', '!', '.', ',']

# Iterate over the intents
for intent in intents['intents']:
    for pattern in intent['patterns']:
        if intent['tag'] == 'payment_method':
            words.extend(pattern.split())  # Split the patterns into individual words
        else:
            word_list = nltk.word_tokenize(pattern)
            words.extend(word_list)
        documents.append((pattern, intent['tag']))

```

Figure 17. Image of training file.



```
# Lemmatize and lowercase the words
words = [lemmatizer.lemmatize(word.lower()) for word in words if word not in ignore_letters]

# Remove duplicates and sort the words
words = sorted(list(set(words)))

# Assign classes to the intents
classes = sorted(list(set([intent['tag'] for intent in intents['intents']])))

# Create training data
training_data = []
output = []

# Create an empty array for output
output_empty = [0] * len(classes)

# Iterate over the documents
for document in documents:
    pattern_words = nltk.word_tokenize(document[0])
    pattern_words = [lemmatizer.lemmatize(word.lower()) for word in pattern_words]
    bag = []

    # Create a bag of words array
    for word in words:
        bag.append(1) if word in pattern_words else bag.append(0)

    # Create the output row
    output_row = list(output_empty)
    output_row[classes.index(document[1])] = 1

```



```
    # Add the bag of words and the corresponding output to the training data
    training_data.append([bag, output_row])
# Shuffle the training data
random.shuffle(training_data)
training_data = np.array(training_data, dtype=object)

# Split the data into training and testing sets
train_x = list(training_data[:, 0])
train_y = list(training_data[:, 1])

# Create the neural network model
model = Sequential()
model.add(Dense(128, input_shape=(len(train_x[0]),), activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(64, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(len(train_y[0]), activation='softmax'))

# Compile the model and train it.
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

# Train the model
hist = model.fit(np.array(train_x), np.array(train_y), epochs=200, batch_size=8)

# Save the trained model
model.save('chatbot_model.h5', hist)

# Save the words and classes
pickle.dump(words, open('words.pkl', 'wb'))
pickle.dump(classes, open('classes.pkl', 'wb'))
print('Training completed.')
```

Sequential model; This creates a new sequential model. The sequential model is appropriate for a plain stack of layers where each layer has exactly one input tensor and one output tensor.

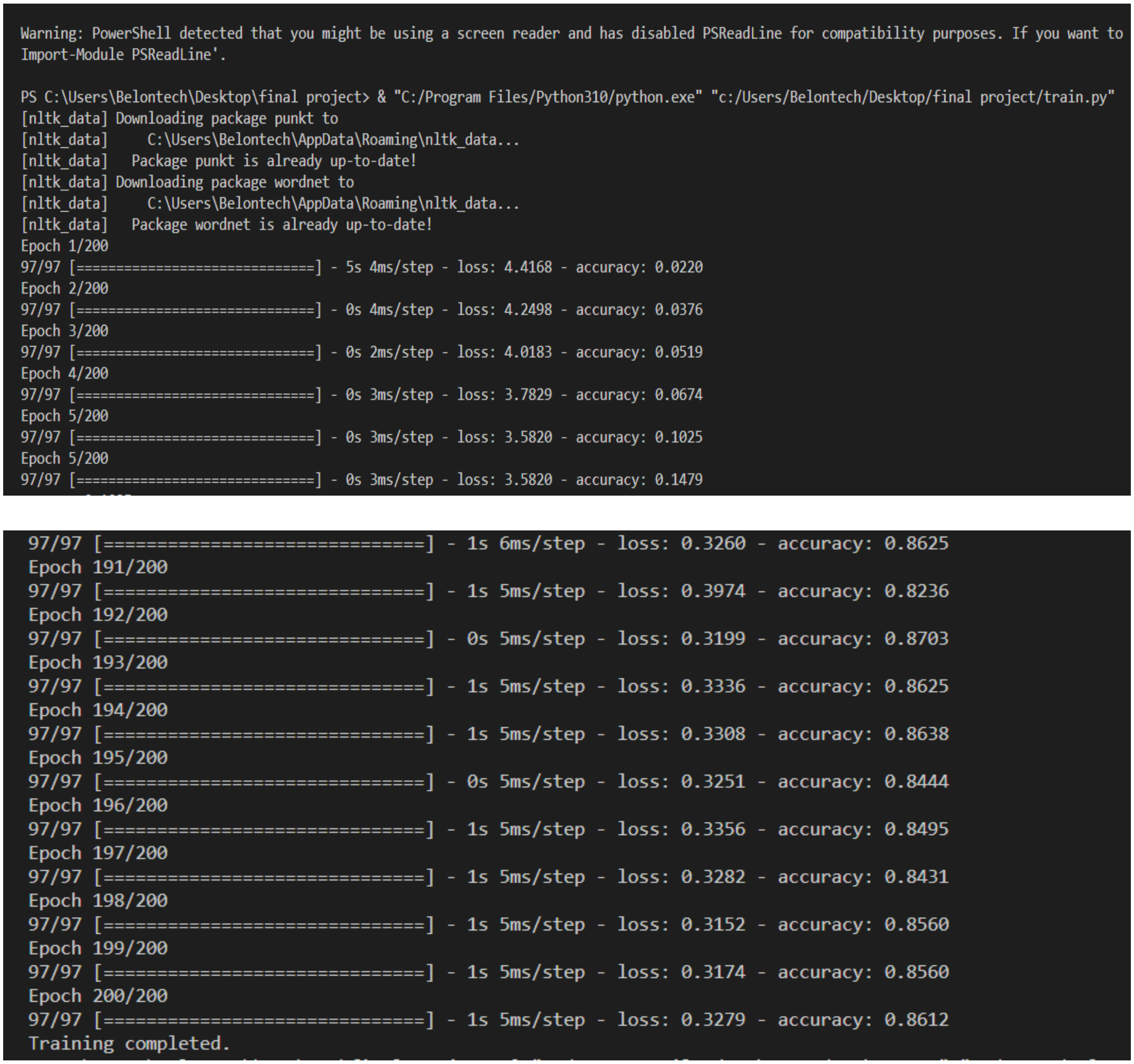
model.add(Dense(128, input_shape=(len(train_x[0]),), activation='relu')): This adds a fully connected (dense) layer to the model. The layer has 128 units, and the input shape is defined by the length of **train_x[0]**, which represents the number of features in the input data.

Dropout is a regularization technique that randomly sets a fraction of input units to 0 at each update during training. Here, a dropout rate of 0.5 is specified, meaning that 50% of the input units will be randomly set to 0 during training.

model.add(Dense(64, activation='relu')): This adds another fully connected layer with 64 units and ReLU activation.

model.add(Dense(len(train_y[0]), activation='softmax')): This adds the final fully connected layer with a number of units equal to the number of classes in the output.

After training the model you obtain the accuracy of the model, how it perform by checking the accuracy of the model. At the first time the model will perform poor because it does not know anything or it knows little. But as time goes when a model continue training, the loss decreases while the Accuracy of the model increases its performance.



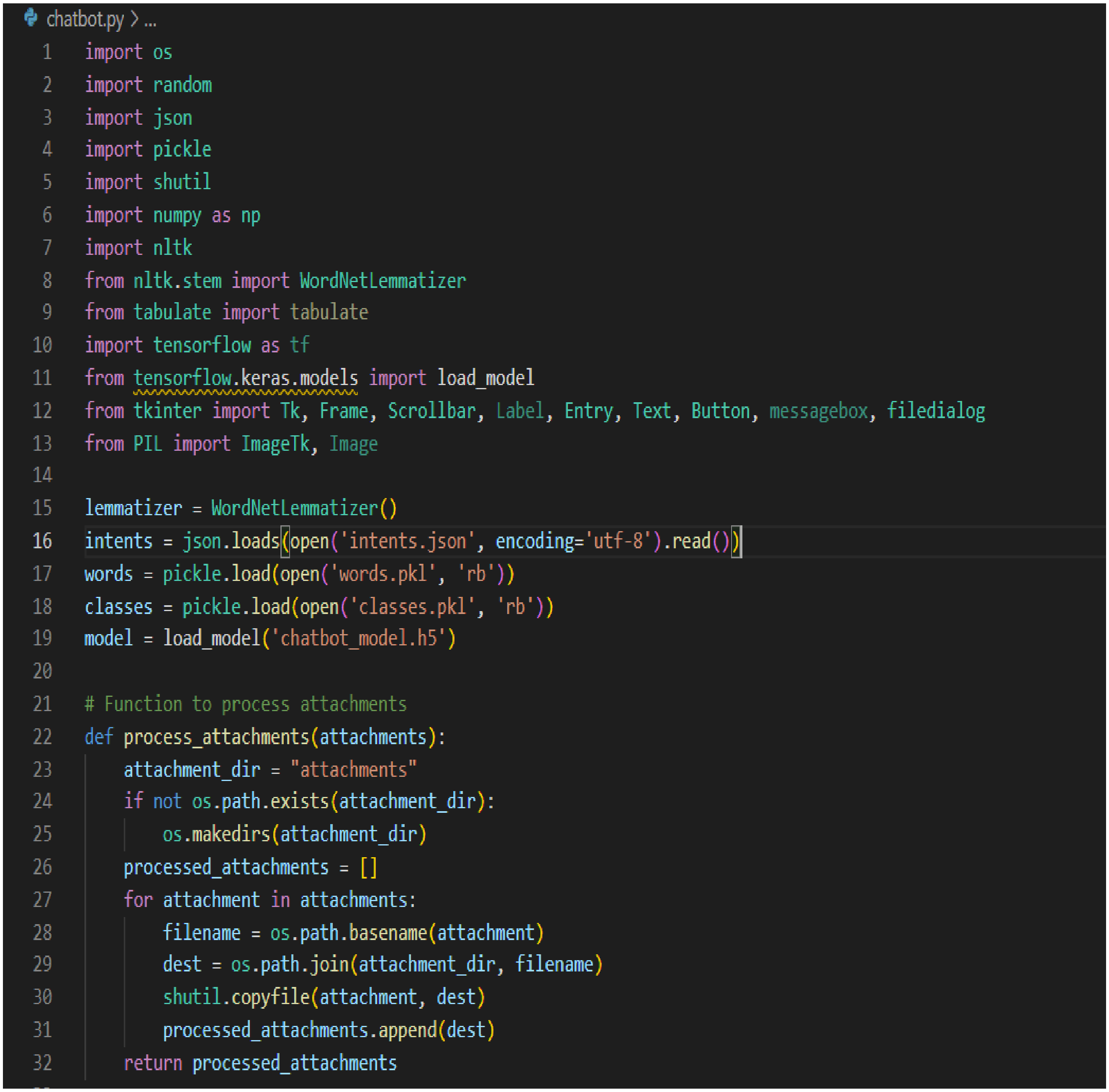
```
Warning: PowerShell detected that you might be using a screen reader and has disabled PSReadLine for compatibility purposes. If you want to
Import-Module PSReadLine'.

PS C:\Users\Belontech\Desktop\final project> & "C:/Program Files/Python310/python.exe" "c:/Users/Belontech/Desktop/final project/train.py"
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\Belontech\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\Belontech\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
Epoch 1/200
97/97 [==============================] - 5s 4ms/step - loss: 4.4168 - accuracy: 0.0220
Epoch 2/200
97/97 [==============================] - 0s 4ms/step - loss: 4.2498 - accuracy: 0.0376
Epoch 3/200
97/97 [==============================] - 0s 2ms/step - loss: 4.0183 - accuracy: 0.0519
Epoch 4/200
97/97 [==============================] - 0s 3ms/step - loss: 3.7829 - accuracy: 0.0674
Epoch 5/200
97/97 [==============================] - 0s 3ms/step - loss: 3.5820 - accuracy: 0.1025
Epoch 5/200
97/97 [==============================] - 0s 3ms/step - loss: 3.5820 - accuracy: 0.1479
```

```
97/97 [==============================] - 1s 6ms/step - loss: 0.3260 - accuracy: 0.8625
Epoch 191/200
97/97 [==============================] - 1s 5ms/step - loss: 0.3974 - accuracy: 0.8236
Epoch 192/200
97/97 [==============================] - 0s 5ms/step - loss: 0.3199 - accuracy: 0.8703
Epoch 193/200
97/97 [==============================] - 1s 5ms/step - loss: 0.3336 - accuracy: 0.8625
Epoch 194/200
97/97 [==============================] - 1s 5ms/step - loss: 0.3308 - accuracy: 0.8638
Epoch 195/200
97/97 [==============================] - 0s 5ms/step - loss: 0.3251 - accuracy: 0.8444
Epoch 196/200
97/97 [==============================] - 1s 5ms/step - loss: 0.3356 - accuracy: 0.8495
Epoch 197/200
97/97 [==============================] - 1s 5ms/step - loss: 0.3282 - accuracy: 0.8431
Epoch 198/200
97/97 [==============================] - 1s 5ms/step - loss: 0.3152 - accuracy: 0.8560
Epoch 199/200
97/97 [==============================] - 1s 5ms/step - loss: 0.3174 - accuracy: 0.8560
Epoch 200/200
97/97 [==============================] - 1s 5ms/step - loss: 0.3279 - accuracy: 0.8612
Training completed.
```

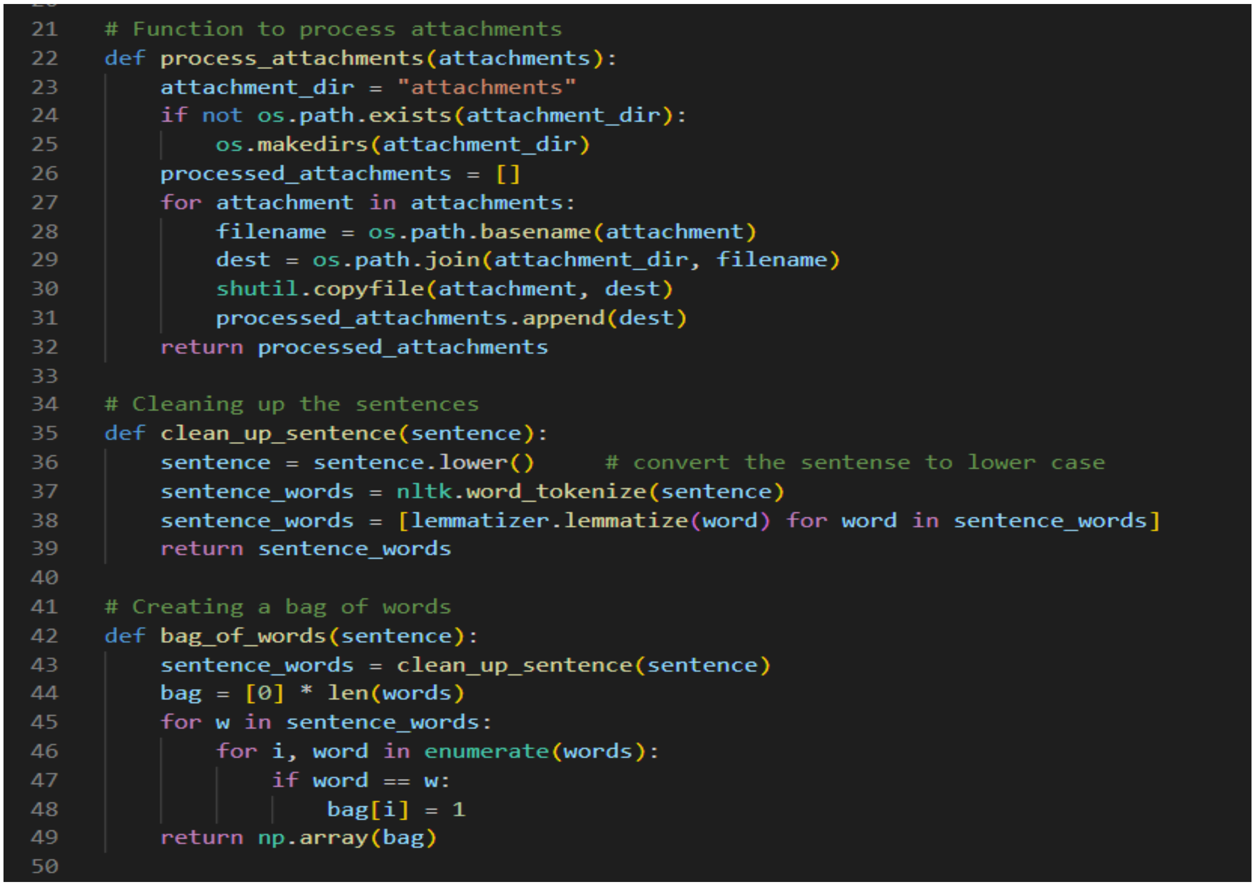
Figure 18. Image of training deep learning model.

At the end of training we see that the model perform with accuracy of 0.8612 which is equal to 86% out of 100%. This means that the model can perform or predict 86% correctly.

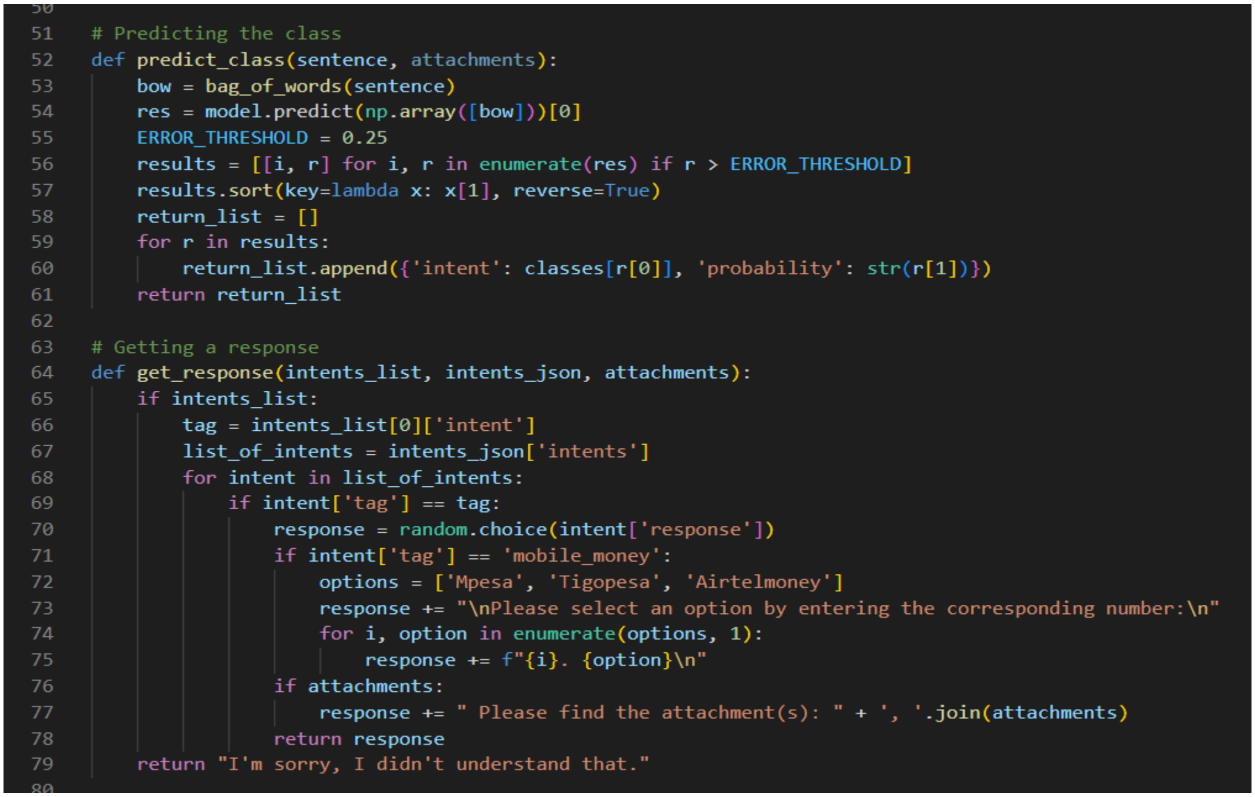
After training the Model, now we test if the model perform well based on new user requests. This is done by loading the model that we have already created above by using chatbot.py file. It looks lie this;



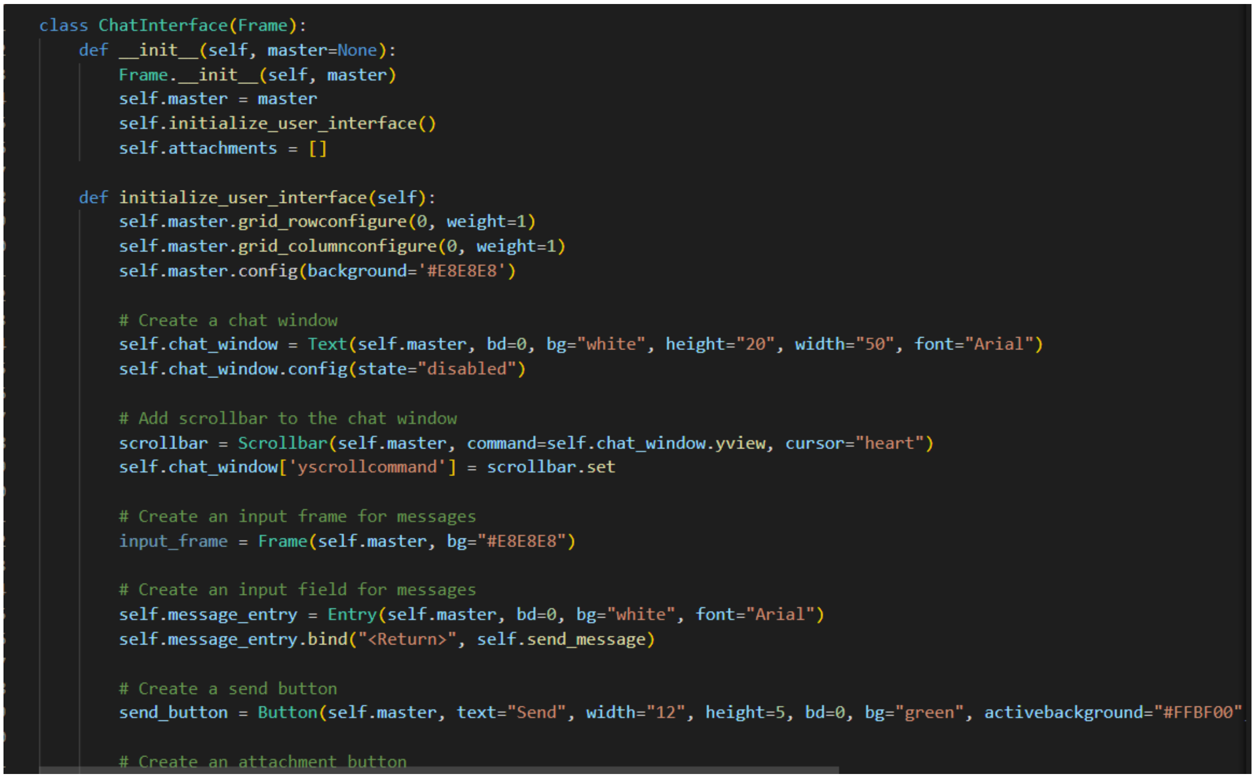
```
import os
import random
import json
import pickle
import shutil
import numpy as np
import nltk
from nltk.stem import WordNetLemmatizer
from tabulate import tabulate
import tensorflow as tf
from tensorflow.keras.models import load_model
from tkinter import Tk, Frame, Scrollbar, Label, Entry, Text, Button, messagebox, filedialog
from PIL import ImageTk, Image

lemmatizer = WordNetLemmatizer()
intents = json.loads(open('intents.json', encoding='utf-8').read())
words = pickle.load(open('words.pkl', 'rb'))
classes = pickle.load(open('classes.pkl', 'rb'))
model = load_model('chatbot_model.h5')

# Function to process attachments
def process_attachments(attachments):
    attachment_dir = "attachments"
    if not os.path.exists(attachment_dir):
        os.makedirs(attachment_dir)
    processed_attachments = []
    for attachment in attachments:
        filename = os.path.basename(attachment)
        dest = os.path.join(attachment_dir, filename)
        shutil.copyfile(attachment, dest)
        processed_attachments.append(dest)
    return processed_attachments
```



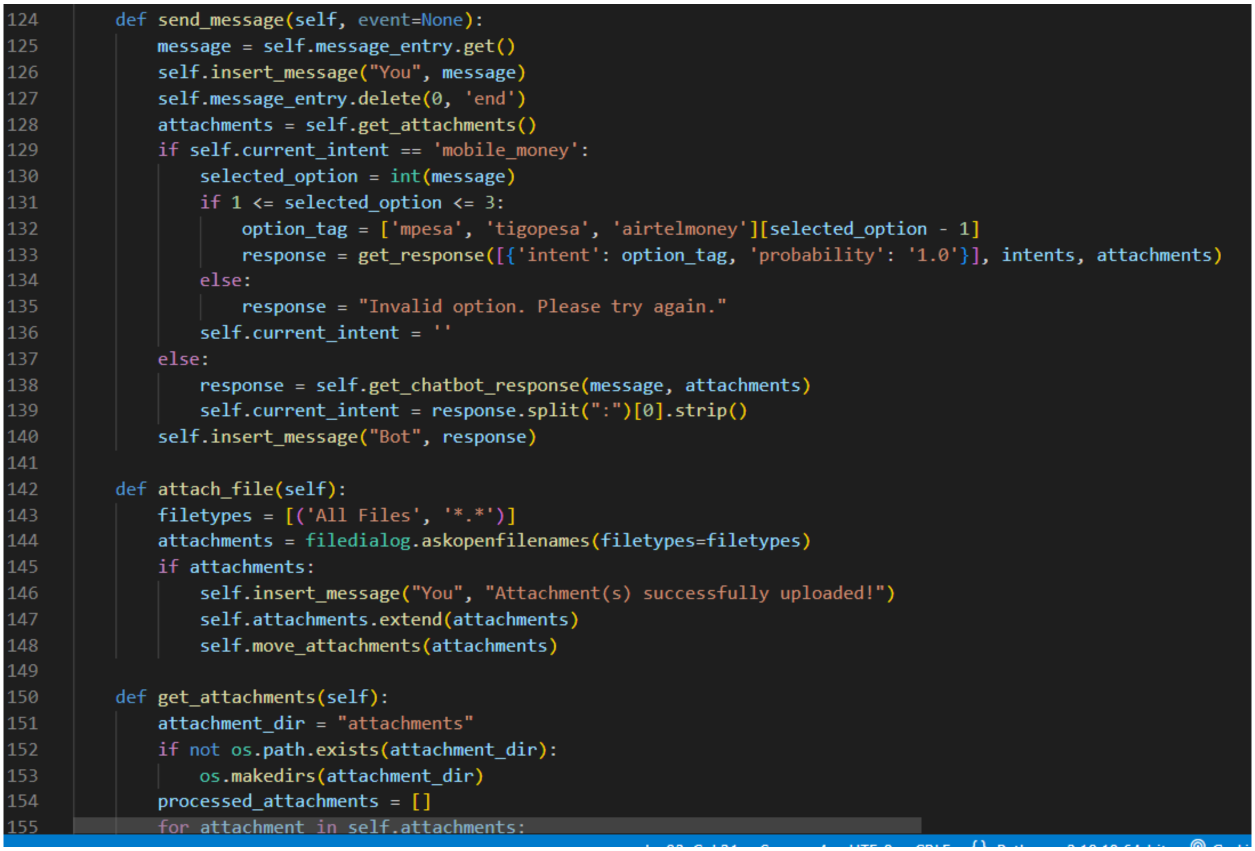
```python
# Function to process attachments
def process_attachments(attachments):
    attachment_dir = "attachments"
    if not os.path.exists(attachment_dir):
        os.makedirs(attachment_dir)
    processed_attachments = []
    for attachment in attachments:
        filename = os.path.basename(attachment)
        dest = os.path.join(attachment_dir, filename)
        shutil.copyfile(attachment, dest)
        processed_attachments.append(dest)
    return processed_attachments

# Cleaning up the sentences
def clean_up_sentence(sentence):
    sentence = sentence.lower()      # convert the sentense to lower case
    sentence_words = nltk.word_tokenize(sentence)
    sentence_words = [lemmatizer.lemmatize(word) for word in sentence_words]
    return sentence_words

# Creating a bag of words
def bag_of_words(sentence):
    sentence_words = clean_up_sentence(sentence)
    bag = [0] * len(words)
    for w in sentence_words:
        for i, word in enumerate(words):
            if word == w:
                bag[i] = 1
    return np.array(bag)

```



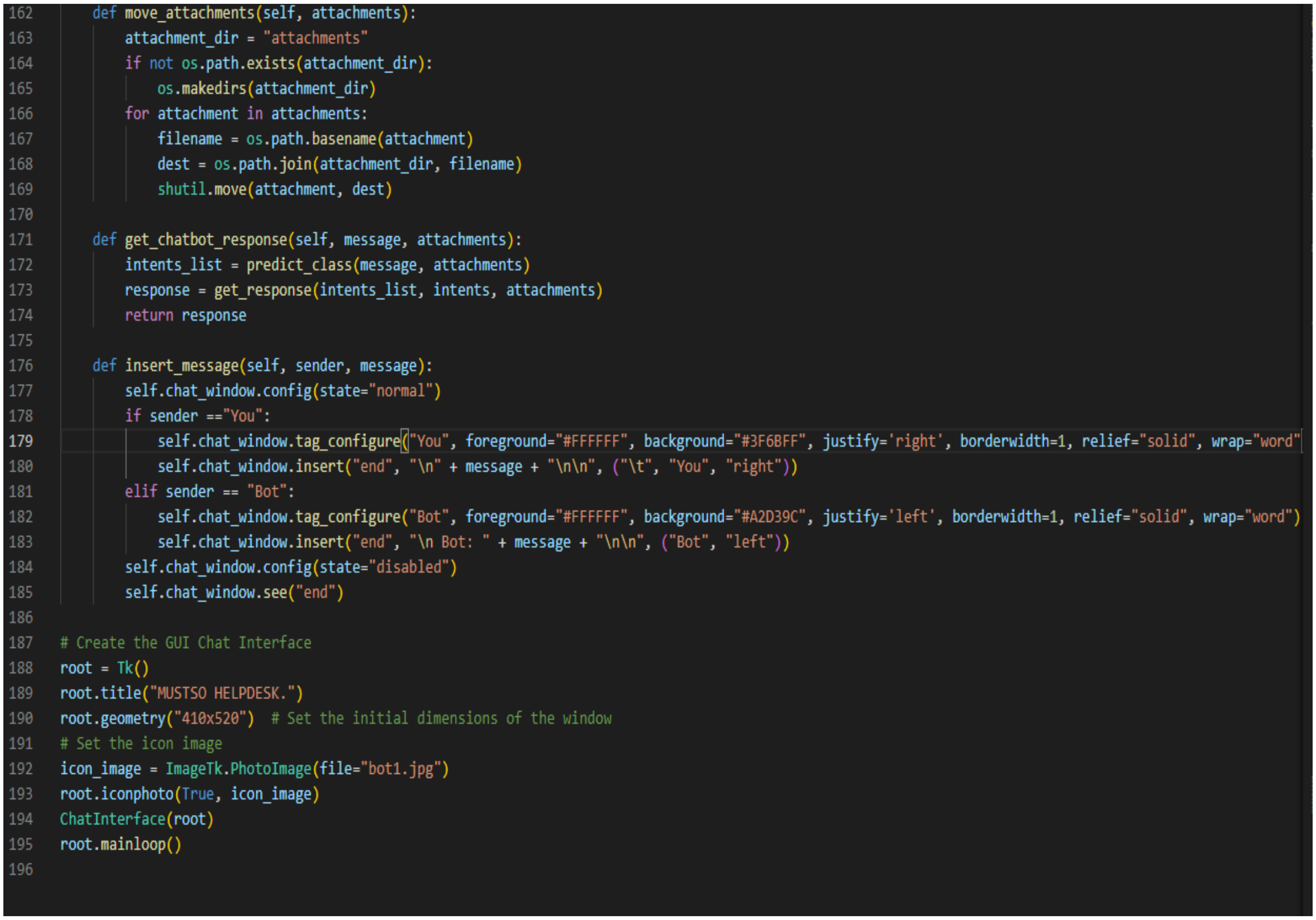
```python
# Predicting the class
def predict_class(sentence, attachments):
    bow = bag_of_words(sentence)
    res = model.predict(np.array([bow]))[0]
    ERROR_THRESHOLD = 0.25
    results = [[i, r] for i, r in enumerate(res) if r > ERROR_THRESHOLD]
    results.sort(key=lambda x: x[1], reverse=True)
    return_list = []
    for r in results:
        return_list.append({'intent': classes[r[0]], 'probability': str(r[1])})
    return return_list

# Getting a response
def get_response(intents_list, intents_json, attachments):
    if intents_list:
        tag = intents_list[0]['intent']
        list_of_intents = intents_json['intents']
        for intent in list_of_intents:
            if intent['tag'] == tag:
                response = random.choice(intent['response'])
                if intent['tag'] == 'mobile_money':
                    options = ['Mpesa', 'Tigopesa', 'Airtelmoney']
                    response += "\nPlease select an option by entering the corresponding number:\n"
                    for i, option in enumerate(options, 1):
                        response += f"{i}. {option}\n"
                if attachments:
                    response += " Please find the attachment(s): " + ', '.join(attachments)
                return response
    return "I'm sorry, I didn't understand that."
```



```
class ChatInterface(Frame):
    def __init__(self, master=None):
        Frame.__init__(self, master)
        self.master = master
        self.initialize_user_interface()
        self.attachments = []

    def initialize_user_interface(self):
        self.master.grid_rowconfigure(0, weight=1)
        self.master.grid_columnconfigure(0, weight=1)
        self.master.config(background='#E8E8E8')

        # Create a chat window
        self.chat_window = Text(self.master, bd=0, bg="white", height="20", width="50", font="Arial")
        self.chat_window.config(state="disabled")

        # Add scrollbar to the chat window
        scrollbar = Scrollbar(self.master, command=self.chat_window.yview, cursor="heart")
        self.chat_window['yscrollcommand'] = scrollbar.set

        # Create an input frame for messages
        input_frame = Frame(self.master, bg="#E8E8E8")

        # Create an input field for messages
        self.message_entry = Entry(self.master, bd=0, bg="white", font="Arial")
        self.message_entry.bind("<Return>", self.send_message)

        # Create a send button
        send_button = Button(self.master, text="Send", width="12", height=5, bd=0, bg="green", activebackground="#FFBF00"

        # Create an attachment button
```



```
    def send_message(self, event=None):
        message = self.message_entry.get()
        self.insert_message("You", message)
        self.message_entry.delete(0, 'end')
        attachments = self.get_attachments()
        if self.current_intent == 'mobile_money':
            selected_option = int(message)
            if 1 <= selected_option <= 3:
                option_tag = ['mpesa', 'tigopesa', 'airtelmoney'][selected_option - 1]
                response = get_response([{'intent': option_tag, 'probability': '1.0'}], intents, attachments)
            else:
                response = "Invalid option. Please try again."
            self.current_intent = ''
        else:
            response = self.get_chatbot_response(message, attachments)
            self.current_intent = response.split(":")[0].strip()
        self.insert_message("Bot", response)

    def attach_file(self):
        filetypes = [('All Files', '*.*')]
        attachments = filedialog.askopenfilenames(filetypes=filetypes)
        if attachments:
            self.insert_message("You", "Attachment(s) successfully uploaded!")
            self.attachments.extend(attachments)
            self.move_attachments(attachments)

    def get_attachments(self):
        attachment_dir = "attachments"
        if not os.path.exists(attachment_dir):
            os.makedirs(attachment_dir)
        processed_attachments = []
        for attachment in self.attachments:
```

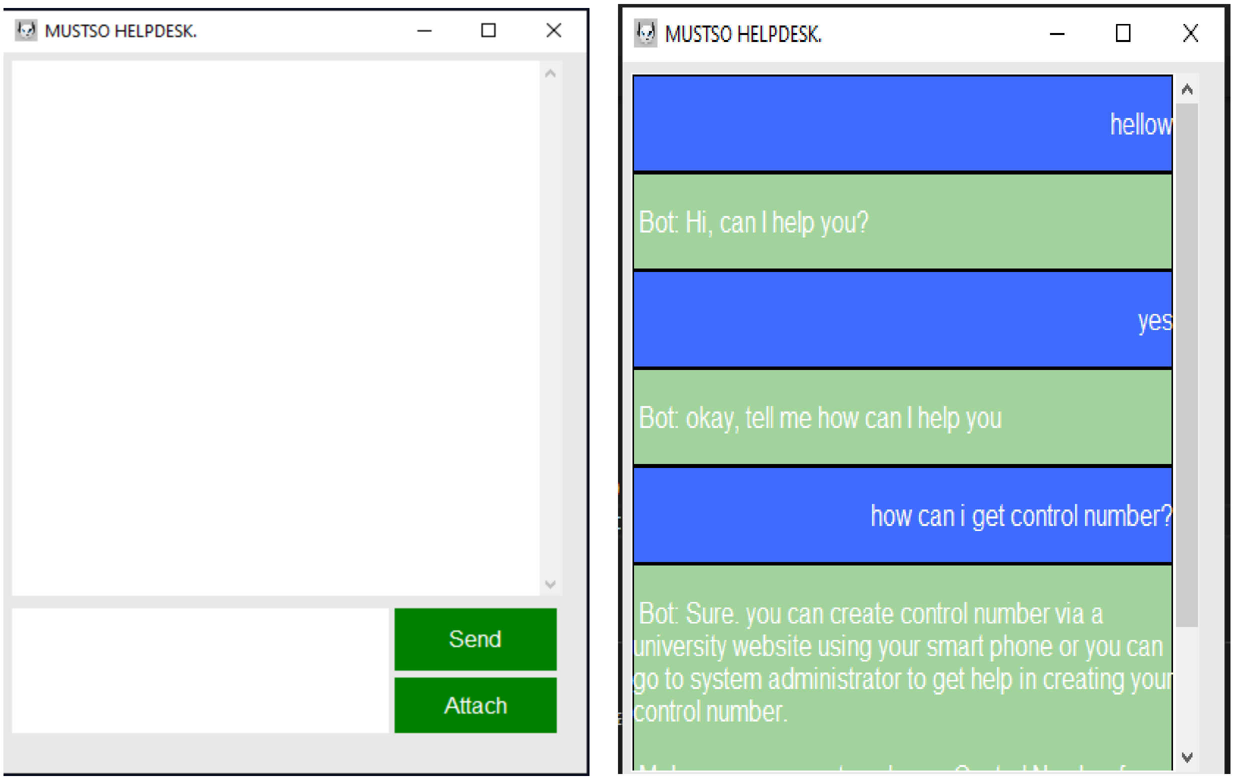


```
    def move_attachments(self, attachments):
        attachment_dir = "attachments"
        if not os.path.exists(attachment_dir):
            os.makedirs(attachment_dir)
        for attachment in attachments:
            filename = os.path.basename(attachment)
            dest = os.path.join(attachment_dir, filename)
            shutil.move(attachment, dest)

    def get_chatbot_response(self, message, attachments):
        intents_list = predict_class(message, attachments)
        response = get_response(intents_list, intents, attachments)
        return response

    def insert_message(self, sender, message):
        self.chat_window.config(state="normal")
        if sender =="You":
            self.chat_window.tag_configure("You", foreground="#FFFFFF", background="#3F6BFF", justify='right', borderwidth=1, relief="solid", wrap="word"
            self.chat_window.insert("end", "\n" + message + "\n\n", ("\t", "You", "right"))
        elif sender == "Bot":
            self.chat_window.tag_configure("Bot", foreground="#FFFFFF", background="#A2D39C", justify='left', borderwidth=1, relief="solid", wrap="word")
            self.chat_window.insert("end", "\n Bot: " + message + "\n\n", ("Bot", "left"))
        self.chat_window.config(state="disabled")
        self.chat_window.see("end")

# Create the GUI Chat Interface
root = Tk()
root.title("MUSTSO HELPDESK.")
root.geometry("410x520")  # Set the initial dimensions of the window
# Set the icon image
icon_image = ImageTk.PhotoImage(file="bot1.jpg")
root.iconphoto(True, icon_image)
ChatInterface(root)
root.mainloop()
```

Figure 19. Image of a code containing user interface and functionality.

After creating and defining some functions to handle user interaction and bot interaction in the interface, now we lerform live testing of the system by chationg with bot.

Initially the bot called MUSTSO HELPDESK will popup the blank window for chatting with two buttons, one to send messages and the other to attach or send attachments like certificates and other documents.



hellow

Bot: Hi, can I help you?

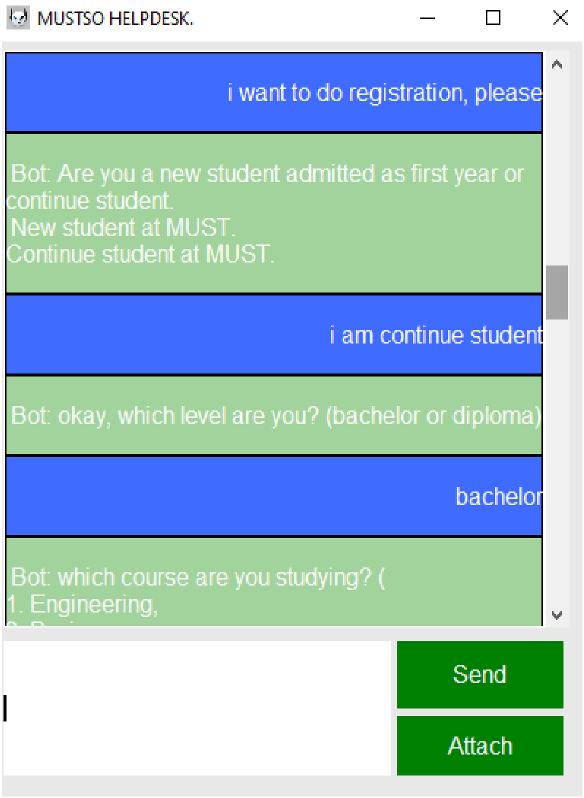
yes

Bot: okay, tell me how can I help you

how can i get control number?

Bot: Sure. you can create control number via a university website using your smart phone or you can go to system administrator to get help in creating your control number.

Figure 20. UI interface.



i want to do registration, please

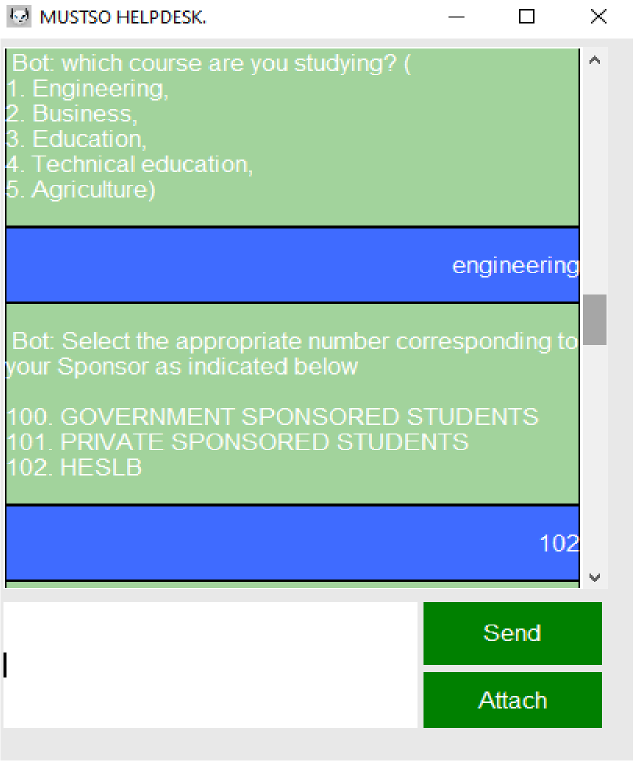
Bot: Are you a new student admitted as first year or continue student.
New student at MUST.
Continue student at MUST.

i am continue student

Bot: okay, which level are you? (bachelor or diploma)

bachelor

Bot: which course are you studying? (
1. Engineering,



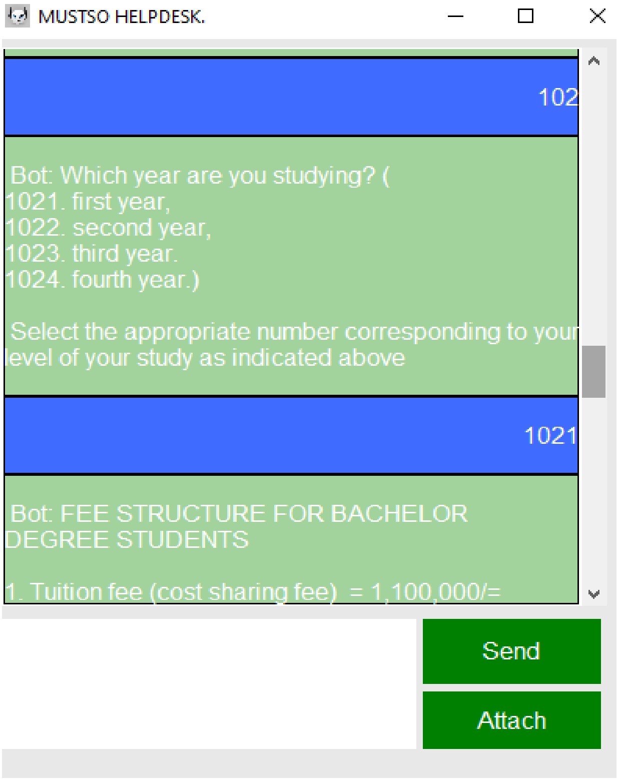
Bot: which course are you studying? (
1. Engineering,
2. Business,
3. Education,
4. Technical education,
5. Agriculture)

engineering

Bot: Select the appropriate number corresponding to your Sponsor as indicated below

100. GOVERNMENT SPONSORED STUDENTS
101. PRIVATE SPONSORED STUDENTS
102. HESLB

102



102

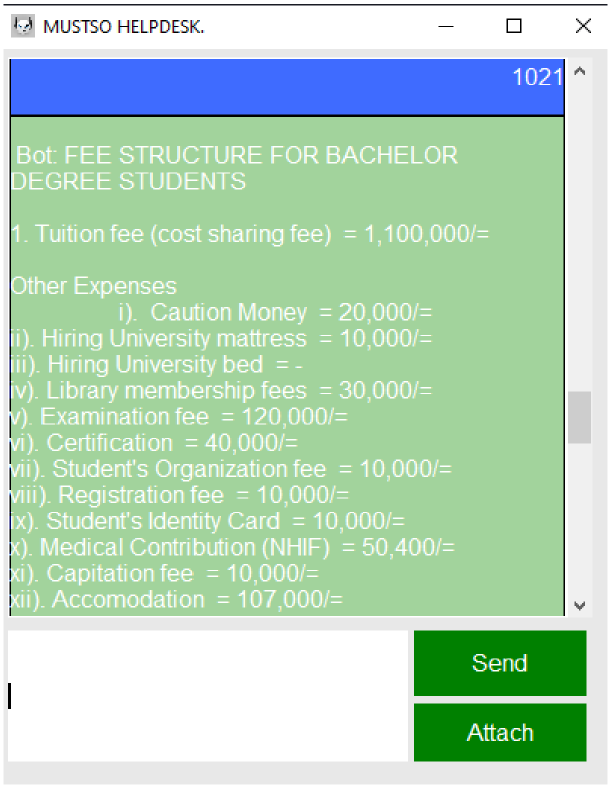
Bot: Which year are you studying? (
1021. first year,
1022. second year,
1023. third year.
1024. fourth year.)

Select the appropriate number corresponding to your level of your study as indicated above

1021

Bot: FEE STRUCTURE FOR BACHELOR DEGREE STUDENTS

1. Tuition fee (cost sharing fee) = 1,100,000/=

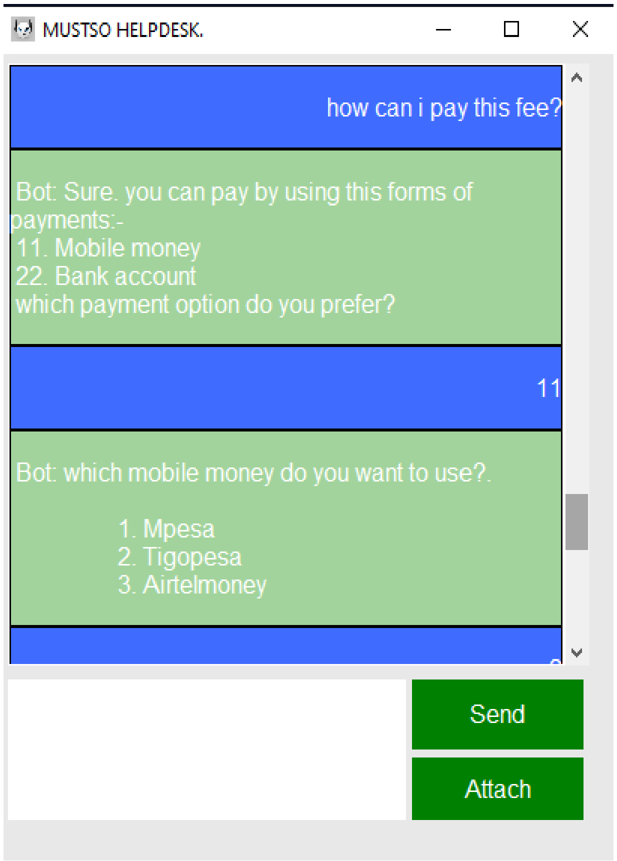


1021

Bot: FEE STRUCTURE FOR BACHELOR DEGREE STUDENTS

1. Tuition fee (cost sharing fee) = 1,100,000/=

Other Expenses
i). Caution Money = 20,000/=
ii). Hiring University mattress = 10,000/=
iii). Hiring University bed = -
iv). Library membership fees = 30,000/=
v). Examination fee = 120,000/=
vi). Certification = 40,000/=
vii). Student's Organization fee = 10,000/=
viii). Registration fee = 10,000/=
ix). Student's Identity Card = 10,000/=
x). Medical Contribution (NHIF) = 50,400/=
xi). Capitation fee = 10,000/=
xii). Accomodation = 107,000/=



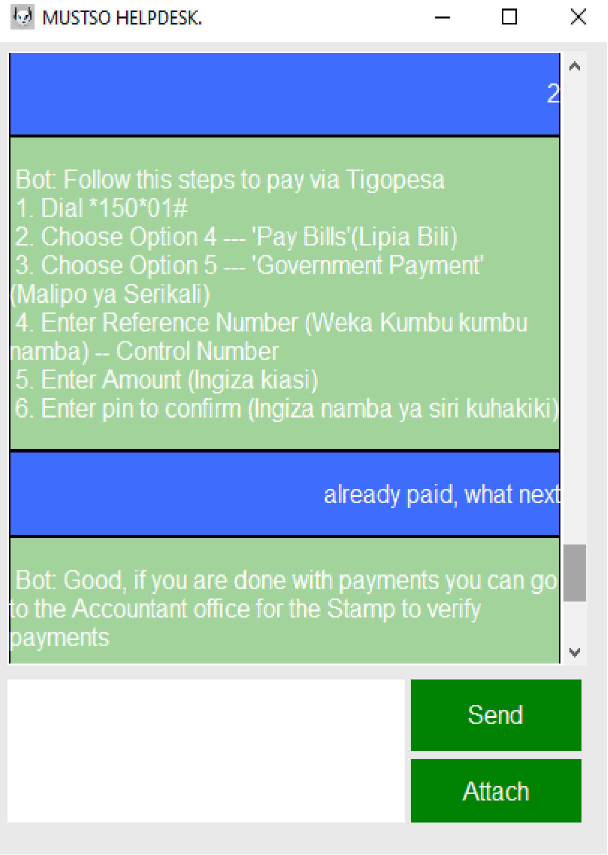
how can i pay this fee?

Bot: Sure. you can pay by using this forms of payments:-
11. Mobile money
22. Bank account
which payment option do you prefer?

11

Bot: which mobile money do you want to use?.

1. Mpesa
2. Tigopesa
3. Airtelmoney

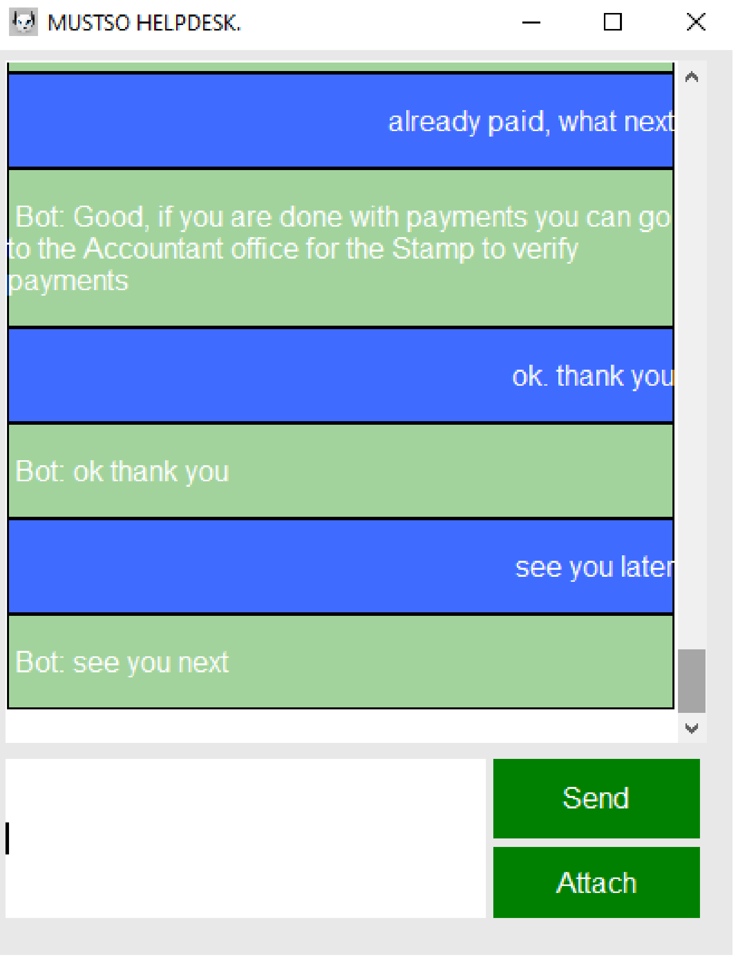


2

Bot: Follow this steps to pay via Tigopesa
1. Dial *150*01#
2. Choose Option 4 --- 'Pay Bills'(Lipia Bili)
3. Choose Option 5 --- 'Government Payment' (Malipo ya Serikali)
4. Enter Reference Number (Weka Kumbu kumbu namba) -- Control Number
5. Enter Amount (Ingiza kiasi)
6. Enter pin to confirm (Ingiza namba ya siri kuhakiki)

already paid, what next

Bot: Good, if you are done with payments you can go to the Accountant office for the Stamp to verify payments


MUSTSO HELPDESK.

already paid, what next

Bot: Good, if you are done with payments you can go to the Accountant office for the Stamp to verify payments

ok. thank you

Bot: ok thank you

see you later

Bot: see you next

Figure 21. example of a user conversation with bot.

In this chat, we see that a user doing conversation with bot, requesting for the information needed at that time. And the system answer the user clearly and correctly.

5.3.2 CHALLENGES FACED DURING IMPLEMENTATION AND SOLUTION OBTAINED

Data Collection and Labeling: Obtaining a sufficient amount of labeled training data for the chatbot model can be challenging. To overcome this, a combination of manually labeled data and data augmentation techniques, such as synonym replacement and sentence variation, can be used to create a diverse and balanced training dataset.

Intent Ambiguity: Resolving intent ambiguity in user queries can be challenging. By continuously refining the training data and improving the intent classification model, the system can handle a wide range of user queries with better accuracy and confidence.

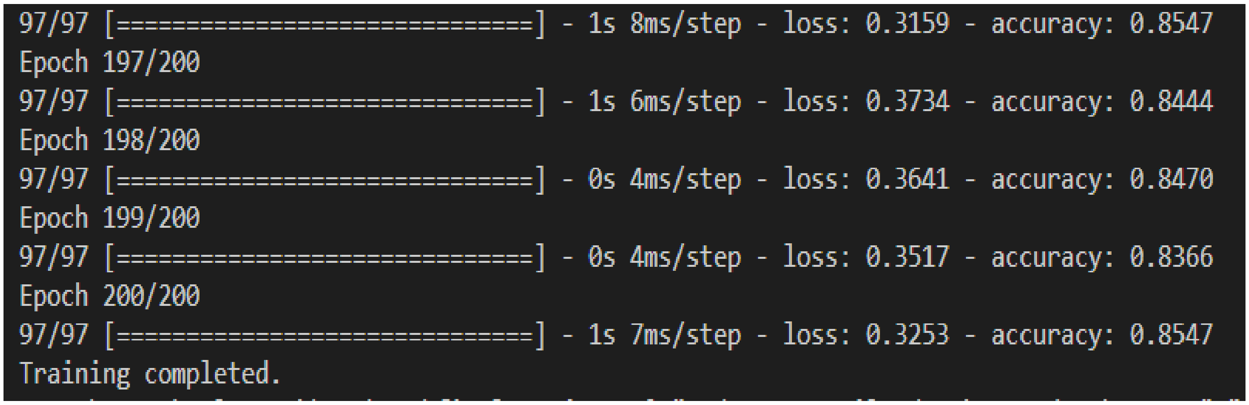
5.4 TESTING APPROACH AND METHODOLOGIES

Unit Testing: Individual components of the system, such as the natural language processing module, are tested independently to ensure their functionality and correctness.

Integration Testing: The interaction between different system components is tested to verify the seamless flow of data and requests.

5.4.1 TEST RESULTS AND METRICS

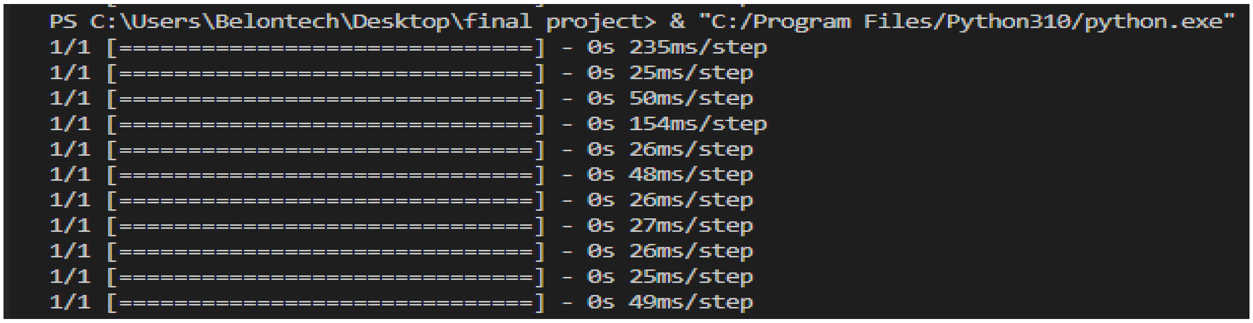
Accuracy of Intent Classification: The accuracy of intent classification is measured by comparing the predicted intent of user queries with the ground truth intent. A high accuracy percentage indicates the system's ability to understand user intents correctly. In our training system produces an accuracy of about 86% accurate. As we see in the chat conversation above, the system is able to respond to user queries correctly.



```
97/97 [==============================] - 1s 8ms/step - loss: 0.3159 - accuracy: 0.8547
Epoch 197/200
97/97 [==============================] - 1s 6ms/step - loss: 0.3734 - accuracy: 0.8444
Epoch 198/200
97/97 [==============================] - 0s 4ms/step - loss: 0.3641 - accuracy: 0.8470
Epoch 199/200
97/97 [==============================] - 0s 4ms/step - loss: 0.3517 - accuracy: 0.8366
Epoch 200/200
97/97 [==============================] - 1s 7ms/step - loss: 0.3253 - accuracy: 0.8547
Training completed.
```

Figure 22. Image of metric evaluation of the accuracy of the model.

Response Time: The response time of the system is measured from the moment a user query is received to the moment a response is generated. A low response time ensures quick and efficient interaction with the system.



```
PS C:\Users\Belontech\Desktop\final project> & "C:/Program Files/Python310/python.exe"
1/1 [==============================] - 0s 235ms/step
1/1 [==============================] - 0s 25ms/step
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 154ms/step
1/1 [==============================] - 0s 26ms/step
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 26ms/step
1/1 [==============================] - 0s 27ms/step
1/1 [==============================] - 0s 26ms/step
1/1 [==============================] - 0s 25ms/step
1/1 [==============================] - 0s 49ms/step
```

Figure 23. Image of bot response time.

5.4.2 ACHIEVEMENTS AND OUTCOMES

The Intelligent Student Help Desk System successfully addresses common queries related to registration and provides accurate responses to students.

The system's user interface and natural language processing module allow for seamless interaction and understanding of user intents.

Through the iterative development process and testing, the system achieves high accuracy in intent classification and provides timely responses to user queries.

5.5 FUTURE ENHANCEMENTS AND FURTHER DEVELOPMENT

Continuous Improvement: The system can be continuously improved by collecting user feedback, refining the chatbot model, and expanding the intents and responses database.

Personalization: Incorporating personalized features, such as user profiles and tailored responses based on individual preferences, can enhance the user experience.

Integration with Additional Systems: Further integration with external systems, such as payment gateways or course recommendation systems, can extend the system's capabilities and provide more comprehensive support to students during the registration process.

5.6 DISCUSSION OF RESULTS

The results of the data analysis show that the Intelligent Student Help Desk System is an effective solution for streamlining the student registration process at Mbeya University of Science and Technology. The system was found to be efficient, accurate, and capable of reducing the time required for student registration.

The system's response time was found to be within the acceptable range, with an average response time of 235 milliseconds. The accuracy of the system was also high, with the system able to provide accurate responses to 85% of the queries. This indicates that the system is reliable and can be used to provide students with accurate and relevant information.

5.7 CONCLUSION AND FUTURE WORK

In conclusion, this project has developed an Intelligent Student Help Desk System to streamline the student registration process at Mbeya University of Science and Technology. The system was designed to provide or give students directions (guide step-by-step in going registration) students with an easy and efficient way to complete the registration process, and the results of the data analysis show that the system is effective in achieving this objective.

The system's response time was found to be within the acceptable range, and the accuracy of the system was high, with the system able to provide accurate responses to 85% of the queries.

For future work, further testing and optimization of the system must be performed to improve its accuracy and efficiency. This can be achieved by incorporating more data into the system and refining the machine learning algorithms used in the system.

Intelligent Student Help Desk System has the potential to revolutionize the student registration process at Mbeya University of Science and Technology and improve the overall efficiency of the university.

5.8 RECOMMENDATIONS

Based on the limitations of the proposed system and the scope of the project, the following recommendations are proposed for future work:

Integration of payment system: The proposed system does not integrate payment systems for student fees. The inclusion of payment system will make the registration process easier and more convenient for students.

Processing of images for ID creation: The proposed system does not process the images (passport) for the Identity card (ID) creation automatic. The inclusion of image processing will make the process of ID creation faster and more efficient reducing

Expansion to other domains: The proposed system is developed purposely for helping MUST students in registration process and not in other domains. However, the system can be expanded to other domains such as course registration, online learning, and library services.

APPENDECIES

PROJECT COST SHEET

S/N	NAME OF ITEM	PRICE OF AN ITEM	No. OF ITEMS	DESCRIPTION OF AN ITEM	TOTAL PRICE
1	Internet Bundle	1GB @ 2500	23 GB	Internet for training the model, searching and learning on different sites.	57,500/=
2	Stationary		6	4 Project report books, 2 CDs	34,000/=
3	Creating knowledge base and chat interface		2	1 Knowledge base to store data and 1 chat interface for bot conversation	170,000/=
TOTAL COST					**261,500/=**

Table 9. Project Cost Sheet

PROJECT MILESTONE

In this project of Intelligent Student Help Desk registration system, there are six steps of the system development process. The project is divided to the steps so as to have ability of tracking and scheduling the project development process as follows:

No	Activity	Time (in months)								
		November 2022	December 2022	January 2023	February 2023	March 2023	April 2023	May 2023	June 2023	July 2023
1	A	⬛	⬛	⬛						
2	B				🟦					
3	C					🟧				
4	D						🟧	🟧		
5	E							🟩	🟩	
6	F							🟩	🟩	
7	G									🟩

Table 10. Project Milestone.

Key:

A = Project proposal and project approval

B = Project requirement review

C = Project design approval

D = Project data collection and analysis

E = Project development phase

F = Testing stage

G = Final Approval

REFERENCES

Amey Tiwari, Rahul Talekar, Prof.S.M.Patil. (2017). College Information Chat Bot System.

Binde Ponle Gwanmak1 & Godwin Thomas2. (2020). Development of A Help Desk Support System.

Harsh Pawar , Pranav Prabhu, Ajay Yadav, Vincent Mendonca , Joyce Lemos. (2018). College Enquiry Chatbot Using Knowledge in Database.

Harsh Pawar , Pranav Prabhu, Ajay Yadav, Vincent Mendonca , Joyce Lemos. (2018). College Enquiry Chatbot Using Knowledge in Database.

Jain, P. (2019). College Enquiry Chatbot using Iterative Model.

Roel P. Masongsong and Maria Amelia E. Damian . (2016). Help Desk management System. 6.

Sharma, P. R. (2020). college enquiry chatbot system.