BINDURA UNIVERSITY OF SCIENCE EDUCATION

FACULTY OF SCIENCE EDUCATION

DEPARTMENT OF EDUCATIONAL TECHNOLOGY

BACHELOR OF SCIENCE EDUCATION HONORS DEGREE IN COMPUTER SCIENCE

A user-centric smart chatbot for managing admissions nuances of prospective students at Bindura University.

BY

Chemariki Tendai

B1128112

SUPERVISOR: Dr T Chikerema

A DISSERTATION SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS OF THE BACHELOR OF SCIENCE EDUCATION HONORS DEGREE IN COMPUTER SCIENCE

YEAR 2024

RELEASE FORM

Title of the dissertation:

A User-Centric smart Chatbot for managing admissions nuances of prospective students at Bindura University.

1. **To be completed by the student**

   I certify that this dissertation is in conformity with the preparation guidelines as presented in the Faculty Guide and Instructions for Typing dissertations.

   2024-09-05

   (Signature of student) (Date)

2. **To be completed by the supervisor**

   This dissertation is suitable for submission to the faculty. This dissertation should be checked for conformity with Faculty guidelines.

   TChikerema

   Digitally signed by TChikerema
   DN: cn=TChikerema gn=TChikerema
   c=Zimbabwe l=ZW o=BUSE ou=FSE
   e=tchikerema@buse.ac.zw
   Reason:
   Location:
   Date: 2024-09-05 10:50+02:00

   (Signature of Supervisor) (Date)

3. **To be completed by the chairperson of the department**

   I certify to the best of my knowledge that the required procedures have been followed and the preparation criteria have been met for this dissertation

   11/10/24

   (Signature of the chairperson) (Date)

APPROVAL FORM

Name of the student: CHEMARIKI TENDAI

Registration Number: B1128112

Dissertation Title:

A User-Centric smart Chatbot for managing admissions nuances of prospective students at Bindura University.

Degree Title: Bachelor of Science Education Honors Degree in Computer Science.

Year of completion: 2024

Permission is hereby granted to Bindura University of Science Education to single copies of this dissertation and to lend and sell such copies for private, scholarly scientific purposes only. The author reserves the rights and neither the dissertation nor extensive extracts from it be granted or otherwise be replicated without the author's consent.

Signed: Date: 05 / 09 / 2024

AKNOWLEDGEMENTS

First and foremost, I would like to thank the Almighty God who provided me wisdom and strength to navigate through the research process. I would also like to extend my gratitude to my dissertation supervisor, Dr. T. Chikerema who guided me through every detail of the project through reading, editing and polishing the ideas of the write-up. All the credit goes to her for her suggestions and valuable comments on the research, this would not have been possible without her support and guidance.

DEDICATION

This study is dedicated to my Parents Mr. and Mrs. Chemariki and l am grateful and thankful to my loving wife Lizzy Dzvairo, you are the reason I have so much courage to move on when the road gets tough. To my dear siblings Bwanali Chemariki, Chrispen Chemariki, Margret Chemariki, Tafadzwa Chemariki and to The DDC Mbire, Mr. Richard Kimble Maruta, I am grateful for all the encouragement and support you have given me.

Table of Contents

RELEASE FORM ii

APPROVAL FORM iii

AKNOWLEDGEMENTS iv

DEDICATION v

ABSTRACT 1

Chapter 1: Problem Identification 2

1.1 Introduction 2

1.2 Investigation and description of current systems/Literature on existing similar systems 3

1.3 Statement of the Problem 7

1.4 System Objectives 9

1.5 Description of the proposed system 9

i. System Architecture: 10

ii. Rasa Components: 10

iii. User-Centric Design Principles: 11

iv. Training and Evaluation: 11

v. Deployment and Maintenance: 12

Benefits of Using Rasa Platform: 12

1.6 Limitations/challenges 12

Technical Challenges: 12

User Experience Challenges: 13

Project Implementation Challenges: 13

1.7 Scope/delimitation of the system 13

Delimitations: 14

Challenges: 14

1.8 Definition of terms 15

Chapter 2: Requirements Specification 17

2.1 Introduction 17

2.2 Requirements Analysis: Fact-Finding Plan 18

2.2.1 Fact-Finding Techniques: 18

2.2.2 Recording and Documentation: 19

2.2.3 Analysis Tools: 20

2.2.4 Expected Outcomes: 20

2.3 Data Requirements: System Input and Output 21
2.3.1 System Input: 21
2.3.2 User Interaction Methods: 21
2.3.3 System Output: 21
2.3.4 Data Integration (Optional): 22
2.3.5 Limitations: 22
2.3.6 Future Considerations: 22
2.4 Processing Requirements: Functionality Expected 23
2.4.1 Core Functionalities: 23
2.4.2 Advanced Functionalities (Optional): 24
2.5 Software Requirements: Platform, Operating System, and Development Tools 24
2.5.1 Target Platform: 24
2.5.2 Operating System Compatibility: 24
2.5.3 Development Tools: 25
2.5.4 Rationale for Open-Source Tools: 25
2.5.5 Version Control System: 25
2.5.6 Testing Tools: 26
2.6 Hardware Requirements: User Access Considerations 26
2.6.1 Target User Devices: 26
2.6.2 Minimum Hardware Specifications (General Guidance): 26
2.6.3 Additional Considerations: 27
2.6.4 University IT Infrastructure (Optional): 27
Chapter 3: Design 28
3.1 Introduction 28
3.2 System Architecture 29
3.3 Input and output design 31
3.3.1 User Input 31
3.3.2 Chatbot Output 31
3.4 Database design 33
3.5 Processes design: 34
Chapter 4: Coding and Testing 38
4.1 Introduction 38
4.2 Technical Documentation: System Code 38
4.2.1 Platform Selection 38

4.2.2 Key Code Components 39

4.3 Unit and System Testing: Plan and Results **42**

4.3.1 Testing Plan 42

4.3.2 Testing Results 43

4.4 User Testing: Plan and Results **44**

4.4.1 Testing Plan 44

4.4.2 Testing Results 45

Chapter 5: Implementation and Deployment 47

5.1 Introduction **47**

5.2 Conversion Plan **47**

5.3 User Manual **48**

5.4 User Training: Plan **48**

Chapter 6: Evaluation and Conclusion 49

6.1 Introduction **49**

6.2 Evaluation of the System **49**

6.3 Future Plans/Developments **50**

Appendices: 52

APENDIX A: INTERVIEW GUIDE. 52

APENDIX B: SURVEY. 54

APENDIX C: UNIVERSITY DOCUMENTS REVIEW. 56

APENDIX D: BENCHMARKING ANALYSIS. 58

REFERENCES. 60

ABSTRACT

The admission process of universities in Zimbabwe is a labor-expensive task due to the involvement of humans. This paper introduces an intelligent system (a chatbot) that can support the admission process by automatically answering Frequently Asked Questions (FAQs), Help navigate the Bindura University website and sharing information related to admissions, Chatbots are programmed software that can reply to human queries in real-time and their requirement and need in today's world is endless. To make the chatbots as effective as possible, numerous activities need to be completed. I designed a university information sharing and FAQ chatbot to address admissions nuances of prospective students at Bindura University in the academic environment. This chatbot uses Machine Learning (ML) and Natural Language Processing (NLP) to answer questions about Bindura university of science education in an effective and precise manner. Unlike from prior work that usually builds the bot from scratch, I developed the bot by using the Rasa platform. To do that, I employed different combinations of components of natural language understanding to find the best pipeline. I also created and released a dataset in the admission domain to train the bot. The introduction video of the system is also available.

Chapter 1: Problem Identification

1.1 Introduction

The landscape of university admissions is undergoing a significant transformation. Prospective students are increasingly seeking information and guidance through digital channels. To meet this evolving demand, universities should embrace innovative technologies like chatbots to streamline the admissions process and enhance student engagement, (Klopfenstein et.al ,2017). This research project delves into the potential of chatbots for university admissions, exploring their functionalities, the benefits they offer to both students and institutions, and the considerations for their effective implementation. Through a comprehensive analysis of existing research, the development of a user-centric chatbot framework, and an evaluation of potential challenges, this project aims to contribute to the development of a valuable tool for navigating and managing the admissions nuances and complexities of prospective students at Bindura University.

According to Gao et.al (2018), Conversational AI (also known as virtual assistant or chatbot) constitutes an integral part of natural user interfaces and has received a lot of attention from the research community. The task of conversational AI is to build an AI “bot" that can naturally make interactive conversations between human and machine. The field of conversational AI can be separated into three groups: task-oriented dialog systems, chat-oriented dialog systems, and question answering (QA) dialog systems, (Nuruzzaman and Hussain, 2020) The first group performs tasks on users’ perspectives such as making a reservation in a restaurant or booking a flight. The second group needs to carry out natural and interactive conversations with users based on their questions. The final group provides clear and concise answers to users’ questions based on natural language processing techniques.

Conversational AI bots make a big impact in e-commerce. To businesses, the bots provide an active method of connecting sellers and customers. To customers, the bots satisfy users’ experience by providing online services that can work 24/7. Due to its huge benefit, the conversational AI bots have been applied to various industries such as insurance, education, entertainment, health care, e-commerce, COVID-19, or business intelligence, (Abu Shawar and Atwell, 2007).

This project aims to develop a smart chatbot for Bindura University to efficiently manage and address the "admissions nuances" faced by prospective students and seamlessly navigate the BUSE website. This will streamline the admissions process for both students and the university, improving clarity, accessibility, and overall experience.

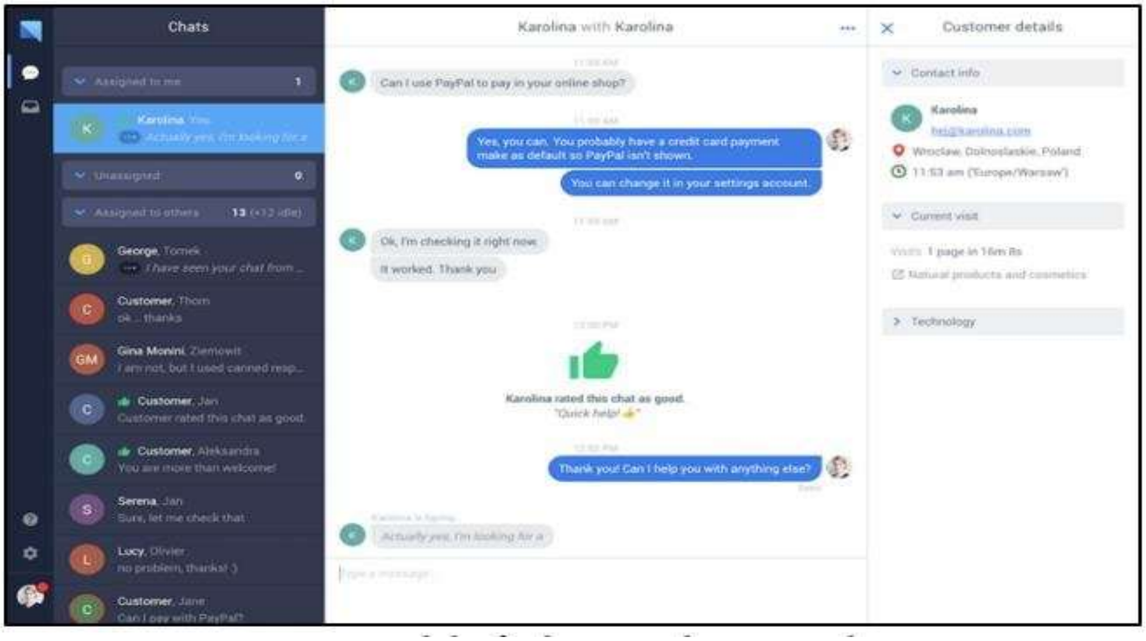
1.2 Investigation and description of current systems/Literature on existing similar systems.

QA systems started in 1960s (Brandtzaeg and Folstad,2017), and it has become efficiently practical applications since 2015 due to the use of deep learning approaches and the availability of large-scale datasets. Due to the increasing impact on real life, building question answering (QA) dialog systems in conversational AI has attracted both academia and industry, (Nuruzzaman and Hussain, 2020)

The following are examples of several types of Chatbots used for several purposes such as for businesses, marketing, education and etc.

Chat.io

This Chatbot system is used to help businesses to communicate with the customer via multiple services in one system. It can also be integrated with Facebook Messenger to help the admin to interact with Facebook users. The Chatbot is designed using modular construction that can be integrated with website, app, native mobile app or web-based application. An artificial intelligence system is used in the development of the Chatbot, whereby the Chatbot can predict the text suggestion and later provide smart responds base on analyzing the conversation history. This Chatbot start the messaging by asking for the name and email of the new user. User also have to agree with the term and conditions applied before proceeding to start chat.



I have seen your chat from ...

ok ... thanks

I am not, but I used canned resp...

Customer rated this chat as good.

You are more than welcome!

Sure, let me check that

no problem, thanks! :)

Can I pay with PayPal?

Can I use PayPal to pay in your online shop?

Yes, you can. You probably have a credit card payment make as default so PayPal isn't shown.

You can change it in your settings account.

Ok, I'm checking it right now.

It worked. Thank you

Karolina rated this chat as good.
"Quick help! 👍"

Thank you! Can I help you with anything else?

Actually yes, I'm looking for a

Wroclaw, Dolnoslaskie, Poland

11:53 am ('Europe/Warsaw')

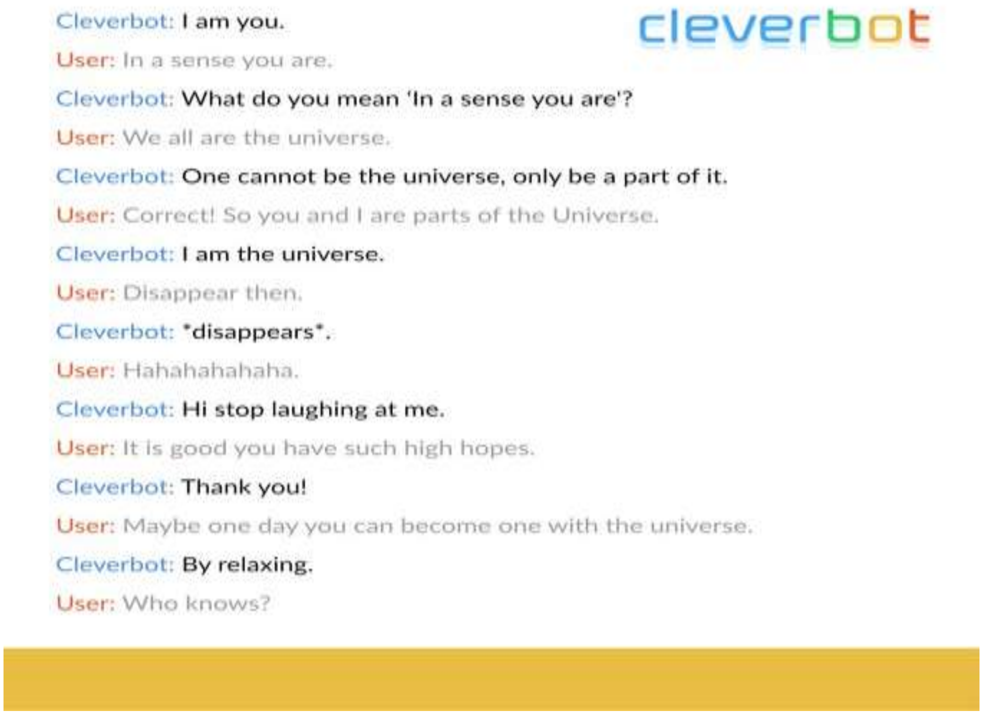
Natural products and cosmetics

Fig 1: Example of Chat.io

Admin Panel Figure 1 shows the admin panel interface for Chat.io. There are three parts which is customer name list on the left side, messaging on the middle, and the details is on the right side. Major features of Chat.io is integration with Facebook, auto/manual chat assignments, intelligent canned responses, customer message sneak peek, chat ratings, open API, messenger and mobile application.

Cleverbot

The following is another example of web chatbot called Cleverbot. It is an artificial Intelligent Chatbot called Cleverbot. It is a chatterbot web application developed by British AI scientist Rollo Carpenter in 1997. Cleverbot responses are not hard coded programme. Instead, it learns from human input from the conversation process. When the user input some text, is entered, the system then searches all the exact phrase to match with the input. It gives the responses to the input from user by finding how the way user are responded to that input. This Cleverbot is also available on mobile Android and IOS platform.



Cleverbot: I am you.
User: In a sense you are.
Cleverbot: What do you mean 'In a sense you are'?
User: We all are the universe.
Cleverbot: One cannot be the universe, only be a part of it.
User: Correct! So you and I are parts of the Universe.
Cleverbot: I am the universe.
User: Disappear then.
Cleverbot: *disappears*.
User: Hahahahahaha.
Cleverbot: Hi stop laughing at me.
User: It is good you have such high hopes.
Cleverbot: Thank you!
User: Maybe one day you can become one with the universe.
Cleverbot: By relaxing.
User: Who knows?

Fig 2: Example of Cleverbot Interface

Basically, Cleverbot responses to human questions are by learning from previous human answers. Human will type their question in the text box and system will look for all keywords or an exact phrase matching the question. After probing through its saved conversations, Cleverbot will respond to human by finding how human responded to that input before. Version of Cleverbot has been upgraded to use Graphics Processing Unit (GPU) serving techniques. GPU is a specialized electronic circuit intended to rapidly manipulate and alter memory in order to produce images and frame buffer to be displayed. Nowadays, the part of the engine behind Cleverbot and its API is already available in market for all developers outside there.

In practice, there are also several well-known conversational AI agents such as Amazon Alexa (https://www.digitaltrends.com/home/what-is-amazons-alexa-and-what-can-it-do/), Apple Siri (https://www.apple.com/siri/),MicrosoftCortana, IBM Watson bot(https://www.ibm.com/watson), and Google assistant(https://assistant.google.com/).

Their Common Functionalities include but not limited to:

i. Answering Frequently Asked Questions (FAQs). Most chatbots are programmed to handle a wide range of frequently asked questions from prospective clients. This includes inquiries about deadlines, standardized testing, financial aid and scholarships.

ii. Providing Application Guidance. Some advanced chatbots can guide students through the application process, offering step-by-step instructions and assisting with filling out forms.

iii. Scheduling Appointments and Virtual Tours. Chatbots can schedule appointments with admissions counselors or link students to virtual tours of the campus, providing a more immersive experience than static information pages.

iv. Handling Personalized Interactions. Some chatbots leverage machine learning to personalize interactions with students. By analyzing user queries and browsing history, the chatbot can tailor responses and offer relevant program suggestions or scholarship opportunities.

According to Molnár and Zoltán, the role of chatbots in formal education. Presented at the 15 September 2018 University admissions offices are increasingly recognizing the potential of chatbots to enhance the student experience and streamline communication. As a result, a growing number of universities have implemented chatbots specifically designed to address student inquiries related to admissions.

Examples of Existing Chatbots in Universities:

Chatbots have a lot of potential for usage in university administration functions. In order to increase the effectiveness of administrative officers' jobs and the quality of the student experience, several institutions have adopted the usage of chatbots to answer inquiries from students. Every day, the administrative staff at the institution gets a number of emails, calls, and messages. The job of these assistants may be impacted by having to answer every inquiry, which is typically repetitious. Instead of spending their time answering the same questions repeatedly, administrative officers may focus on other important activities by automating frequently asked questions (FAQ). looked at chatbot effectiveness in university administrative services and whether or not they would lighten employee workload, (Abu Shawar and Atwell, 2007).

Georgia Institute of Technology's

At Georgia University there is a chatbot called "Georgia Tech Virtual Guide". This chatbot provides information on admissions requirements, financial aid, and campus life. It also allows students to schedule appointments and connect with current students.

University of California, Los Angeles (UCLA)'s

"BruinBot" is a chatbot found at university of California. This chatbot allows prospective students to explore academic programs, learn about scholarships and financial aid, and connect with specific departments.

University of Southern California (USC)'s

"Fight On Bot”. This chatbot assists students with the application process, offers guidance on essays and recommendation letters, and connects them with resources for standardized testing.

The Benefits of Existing Chatbots

The Benefits of Existing Chatbots is that they provide 24/7 Availability meaning Chatbots offer round-the-clock availability to answer student inquiries, providing immediate support regardless of time zone or location. They also Increase Efficiency in the sense that, Chatbots can handle a high volume of inquiries simultaneously, freeing up admissions staff to focus on more complex tasks. There is again improved accessibility as Chatbots can provide a user-friendly and accessible platform for students to access information, regardless of their technical skills. Moreover, a personalized experience is provided as advanced chatbots can offer personalized interactions that cater to individual student needs and interests, (Abu Shawar and Atwell, 2007)

1.3 Statement of the Problem

In the context of admission, Universities in Zimbabwe still use traditional methods for spreading information and calling students. To spread the admission information, one of common methods is that staff have to go to each high school to introduce the information. The traditional university admissions process can be complex and information-intensive for prospective students. Navigating application requirements, deadlines, financial aid options, and program details can be overwhelming, particularly for students unfamiliar with the system. This task is costly, labor-expensive, and cannot work in situations of the outbreaks such as that of covid-19 pandemic.

The universities also provide another information channel on their websites. However, browsing through information on their websites is usually a complex task. While university websites offer static information, they often lack the interactivity and personalized guidance crucial for informed decision-making. A more practical way of answering admission questions is to use social platforms, e.g., Facebook, in which universities create their fan-pages and directly answer the questions by using the chat function provided by the platforms.

However, I argue that this way is not so efficient due to the limited human resources, i.e., cannot quickly provide correct answers 24/7. As the result, the low quality of supporting services leads to worse satisfaction of pupils and their parents. This is where developing a conversational AI chatbot to automatically answer questions of the admission process, (FAQs, Help navigate the Bindura University website) and sharing of information on Bindura University is crucial. While some universities have implemented chatbots to address these challenges, existing systems often face limitations. These include a restricted scope of functionalities, technical glitches in natural language processing, and the inability to convey empathy or understanding in the same way a human counselor can, (https://www.frontiersin.org/articles/10.3389/frai.2021.654924).

This project addresses the need for a more **user-centric** and **comprehensive** chatbot solution within the university admissions landscape. There is a gap in developing chatbots that not only answer frequently asked questions but also offer **personalized guidance** and **empathic communication**, replicating the experience of interacting with a human admissions counselor.

1.4 System Objectives

This paper makes four main objectives for a User-Centric University Admissions Chatbot. The project aims to develop a user-centric university admissions chatbot that addresses the limitations of existing systems and provides a more comprehensive and personalized experience for prospective Bindura University students. The key objectives are:

1. Provide a user-friendly and accessible interface which Enhance Accessibility and Availability to answer student inquiries, regardless of time zone or location for students with varying technical skills.
2. Expand the Scope of Functionality by Going beyond answering FAQs to offering personalized guidance throughout the application process and navigation of the Bindura University website.
3. Help share information on Bindura University providing accurate and up-to-date, real-time information on program requirements, deadlines, and scholarships Sourced from official university databases and admissions materials.
4. Foster User-Centric Interactions by Leveraging natural language processing (NLP) to understand student intent and respond in a natural, conversational manner.

1.5 Description of the proposed system

The proposed system is a question answering programmed software or chatbot built on the RASA platform. The chatbot can automatically answer questions pertaining to the admission process, (FAQs), Help navigate the Bindura University website and share information on Bindura University in real-time. The requirement and need of chatbots in today's world are endless. Chatbots are made for a variety of uses, including operating electrical equipment in a smart home, serving as a personal virtual assistant, providing entertainment, responding to commonly asked queries, obtaining driving directions, etc. Due to their ability to manage several customers at once and help cut down on customer care costs, chatbots have gained popularity in the corporate sector, (Bansal and Khan, 2018).

This project proposes a user-centric university admissions chatbot called “BUSE ADMITME” to address admission nuances in the academic environment. It is built using the Rasa platform. Rasa is an open-source framework specifically designed for developing chatbots with functionalities for natural language understanding (NLU) and dialogue management, (https://rasa.com/).

This chatbot uses Machine Learning (ML) and Natural Language Processing (NLP) to answer questions about Bindura University of Science Education in an effective and precise manner. The system will be deployed on Facebook to support answering the questions of the admission of Bindura University of Science Education as a case study. The system can be also adapted as a part of the admission process of any university in Zimbabwe.

Here's a detailed breakdown of the proposed system:

i. System Architecture:

- **Front-End Interface:** The chatbot will have a user-friendly interface that integrates seamlessly with the university website and potentially a mobile application. Students can interact with the chatbot through text messages.
- **Rasa Backend:** This is the core of the system, housing the Rasa components responsible for NLU and dialogue management.
- **University Data Sources:** The chatbot will connect to university databases and admissions resources via the university website to provide accurate and up-to-date information. This might include program details, deadlines, financial aid options, and admissions requirements.
- **Optional Integrations:** The system could be further enhanced by integrating with external services like appointment scheduling tools or document upload platforms.

ii. Rasa Components:

- **NLU (Natural Language Understanding):**
    - Intent Recognition: The NLU module will be trained to identify the underlying intent behind a student's query. Examples of intents could be "application requirements," "scholarship opportunities," or "campus life."
    - Entity Recognition: The NLU module will also be trained to extract key entities from student queries, such as student name, specific program names, deadlines, or financial aid types.

- **Dialogue Management:**
    - Dialogue States: The chatbot will track the conversation flow and maintain dialogue states to ensure a coherent and context-aware interaction.
    - Action Server: The Rasa Action Server will handle actions that require interaction with external systems, such as retrieving data from university databases via the BUSE Website.
    - Policies: Dialogue management policies will be defined to govern the chatbot's conversation flow, deciding what information to provide or what actions to take based on the user's intent and the current dialogue state.

iii. User-Centric Design Principles:

- **Natural Language Processing (NLP):** The chatbot will be trained on a large corpus of text data, including student inquiries, admissions materials, and frequently asked questions. This will enable it to understand student queries in a natural and conversational manner.
- **Emotional Intelligence:** By incorporating emotional intelligence techniques, the chatbot can demonstrate empathy and understanding during communication. This can be achieved by using appropriate language tones and offering supportive responses when students express confusion or frustration.

iv. Training and Evaluation:

- The NLU model will be trained on a comprehensive dataset of labeled student queries and corresponding intents and entities. This dataset can be built by collecting real student inquiries from admissions offices or by creating simulated conversations.
- Dialogue management policies will be iteratively developed and refined through user testing and evaluation. This will ensure the chatbot provides clear, concise, and helpful responses to student inquiries.
- Metrics such as NLU accuracy, dialogue success rates, and user satisfaction will be used to evaluate the overall performance of the chatbot.

v. Deployment and Maintenance:

- The chatbot will be deployed on a server accessible by the university website and mobile application (if applicable).
- Regular monitoring and maintenance will be crucial to ensure the chatbot's accuracy and effectiveness. This includes updating the NLU model with new data and refining dialogue management policies based on user feedback.

Benefits of Using Rasa Platform:

The benefits of using RASA are that it is open-source and cost-effective solution for chatbot development. It provides a user-friendly interface for building and training the NLU model. RASA offers powerful tools for dialogue management and customization.

Moreover, it integrates with various external platforms and services. Taking into consideration the strengths of the Rasa platform and focusing on user-centric design principles, this proposed university admissions chatbot has the potential to streamline the admissions process, enhance student engagement, and provide a more positive experience for prospective students.

1.6 Limitations/challenges

While the proposed university admissions chatbot offers significant potential, there are limitations and challenges inherent in such a system:

Technical Challenges:

- **Natural Language Processing (NLP) Accuracy:** NLU models are not perfect. The chatbot may struggle with complex or ambiguous student queries, leading to misinterpretations and irrelevant responses.
- **Limited Context Understanding:** Maintaining a coherent conversation flow with complex context can be challenging for chatbots. The system may need human intervention for inquiries requiring a deep understanding of the student's situation.
- **Data Bias:** The NLU model's training data can introduce biases that affect the chatbot's responses. It's crucial to ensure a diverse and representative training dataset to mitigate bias.

User Experience Challenges:

- **Lack of Empathy:** Despite incorporating emotional intelligence techniques, the chatbot may not fully replicate the human touch and empathy of a real admissions counselor. This could lead to a less personal and engaging experience for students.
- **Frustration with Misunderstandings:** Inaccurate NLU interpretations can lead to frustrating interactions for students. The system should offer clear and easy-to-use mechanisms for students to report misunderstandings and request clarification.
- **Limited Problem-Solving Capabilities:** The chatbot may struggle to address complex problems or unexpected situations that arise during the admissions process. Students may require human intervention to navigate these challenges.

Project Implementation Challenges:

- **Data Acquisition:** Building a comprehensive training dataset for the NLU model will require access to a large corpus of student inquiries and admissions information. Securing this data while ensuring privacy considerations can be challenging.
- **Integration with Existing Systems:** Integrating the chatbot with university databases and other admissions systems may require technical expertise and collaboration between developers and IT personnel.
- **Ongoing Maintenance and Updates:** Maintaining the chatbot's accuracy and effectiveness requires continuous monitoring, data updates, and refinement of dialogue management policies. This necessitates a dedicated team or resources for ongoing maintenance.

1.7 Scope/delimitation of the system

Scope:

The project focuses on developing a user-centric university admissions chatbot built using the Rasa platform. The core functionalities will include:

- Answering frequently asked questions (FAQs) about admissions, programs, financial aid, and campus life.
- Providing personalized guidance throughout the application process.

- Offering real-time information on program requirements, deadlines, and scholarships (by integrating with university databases).
- Enabling document uploads, application status tracking, and appointment scheduling with admissions counselors (potential future integration).

Delimitations:

- The initial scope will likely prioritize answering FAQs and offering basic guidance. Advanced functionalities like appointment scheduling or document upload might require further development and integration with existing university systems.
- The chatbot will be trained on a dataset of admissions-related queries and may not be equipped to handle complex or nuanced questions requiring human intervention.
- The initial focus will be on text-based interactions. Voice command functionalities can be explored in future iterations.
- The project will primarily address the technical development of the chatbot. Long-term considerations such as hosting, maintenance, and integration with university infrastructure might require collaboration with IT departments.

Challenges:

- **Natural Language Understanding (NLU):** Training a robust NLU model that accurately interprets student queries and responds in a natural way can be challenging. This requires a large and diverse dataset of student inquiries and ongoing refinement based on user interactions.
- **Personalized Interactions:** Developing effective personalization requires careful consideration of data privacy and ethical implications. Balancing personalization with transparency and user control over their data will be crucial.
- **Emotional Intelligence:** Implementing emotional intelligence in the chatbot is a complex task. The chatbot should avoid appearing overly artificial or disingenuous while still demonstrating empathy and understanding.
- **Integration with Existing Systems:** Connecting the chatbot with university databases and admissions resources might necessitate collaboration with IT departments to ensure secure data access and integration.

- **Evaluation and Maintenance:** Continuously monitoring and evaluating the chatbot's performance is essential. This involves refining the NLU model, updating dialogue management policies, and addressing user feedback to ensure the chatbot remains accurate, helpful, and user-friendly.

1.8 Definition of terms

BUSE: Bindura University of Science Education

Natural Language Understanding (NLU): The ability of the chatbot to interpret and understand the intent and meaning behind a student's query phrased in natural language. This includes:

- **Intent Recognition:** Identifying the underlying goal or purpose of the student's question (e.g., "application requirements," "financial aid options," "campus life").
- **Entity Recognition:** Extracting key details from the student's query, such as specific program names, deadlines, or financial aid types.

Dialogue Management: The process of controlling the flow of conversation between the chatbot and the student. This includes:

- **Dialogue States:** Tracking the current context of the conversation to ensure a coherent and relevant interaction. (e.g., Is the student in the information gathering stage or ready to submit an application?)
- **Action Server:** The component responsible for handling actions that require interaction with external systems, such as retrieving data from university databases or scheduling appointments.
- **Dialogue Policies:** Rules that govern the chatbot's responses and actions based on the user's intent, the current dialogue state, and other factors.

User-Centric Design: An approach to chatbot development that prioritizes the needs and experience of the student user. This includes:

- **Natural Language Processing (NLP):** Ensuring the chatbot uses clear, concise, and natural language that is easy for students to understand.
- **Personalized Interactions:** Tailoring the chatbot's responses and recommendations to the student's individual profile, interests, and browsing history (when applicable).

Emotional Intelligence:	The ability of the chatbot to demonstrate empathy and understanding during communication, fostering a positive user experience.
Rasa Platform:	An open-source framework specifically designed for developing chatbots. It provides tools for building and training NLU models, managing dialogue flows, and integrating with external systems.
Intent:	The underlying goal or purpose behind a student's query (e.g., "application requirements," "financial aid options," "campus life").
Entity:	A specific piece of information extracted from the student's query, such as program names, deadlines, or financial aid types.
Dialogue State:	The current context of the conversation between the chatbot and the student.
Chatbot Integration:	The process of connecting the chatbot with university databases, admissions resources, appointment scheduling tools, or other relevant systems.

Chapter 2: Requirements Specification

2.1 Introduction

This chapter outlines the detailed requirements for the user-centric university admissions chatbot being developed in this research project. These requirements will serve as a blueprint for the design, development, and evaluation phases of the project.

The chapter is structured as follows:

- **2.2 Functional Requirements:** This section will define the specific functionalities the chatbot must possess to effectively address student inquiries and guide them through the admissions process.
- **2.3 Non-Functional Requirements:** This section will outline the performance expectations, usability considerations, and other non-functional aspects of the chatbot system.
- **2.4 User Interface (UI) Requirements:** This section will specify the design considerations and user interface elements for the chatbot's interaction platform (e.g., website integration, mobile app).
- **2.5 System Interfaces:** This section will define the requirements for the chatbot's integration with external systems, such as university databases via university website, appointment scheduling tools, or document upload platforms (if applicable).
- **2.6 Data Requirements:** This section will specify the types of data needed to train the NLU model and populate the chatbot's knowledge base, ensuring it provides accurate and up-to-date information.
- **2.7 Security and Privacy Requirements:** This section will outline the security protocols and data privacy measures implemented to protect user data collected through chatbot interactions.

2.2 Requirements Analysis: Fact-Finding Plan

As the researcher leading this project on a user-centric university admissions chatbot, I've developed a comprehensive fact-finding plan to gather the necessary information for defining functional requirements. This plan focuses on identifying user needs, understanding existing admissions processes, and exploring best practices in chatbot design.

2.2.1 Fact-Finding Techniques:

To gain a well-rounded perspective, I will employ a multi-pronged approach:

- **Document Review:** I'll begin by reviewing existing university admissions materials, website content, and FAQs. This initial exploration will provide insights into the current information flow, student inquiry patterns, and the language used in admissions communication.
- **Interviews:** Semi-structured interviews will be conducted with key stakeholders to gain deeper understanding:
    - **Admissions Officers:** I'll schedule interviews with admissions officers to gather insights into:
        - The most frequent student inquiries and challenges encountered during the application process.
        - The current workflow for handling student inquiries and providing application guidance.
        - The admissions office's perspective on potential functionalities and limitations of a chatbot system.
    - **Prospective Students:** Interviews with prospective students will help understand:
        - Their information needs and preferred communication channels regarding the admissions process.
        - Their expectations for a chatbot interaction, including desired functionalities and tone of communication.
        - Their experiences with existing university admissions processes and any pain points they've encountered.

- **Surveys:** An online/Whatsapp survey distributed to a representative sample of prospective students will gather broader feedback on their experiences with university admissions. The survey will focus on:
    - Their preferred methods for obtaining admissions information (e.g., website, phone calls, email).
    - The types of questions they have during the application process.
    - Their openness to using a chatbot for admissions guidance and their expectations for such a system.
- **Benchmarking:** I'll analyze existing university admissions chatbots and systems to identify:
    - Successful functionalities that enhance the user experience and streamline the admissions process.
    - Limitations in current chatbot designs that we can avoid in our own development.
    - Design considerations, such as user interface elements and conversation flow, that contribute to user satisfaction.

2.2.2 Recording and Documentation:

Meticulous documentation is crucial. Here's how I'll capture the data:

- **Interviews:** With informed consent, I'll record audio interviews for later transcription. The transcribed documents will capture key points, user quotes, and nonverbal cues (e.g., hesitation, frustration) that can be informative.
- **Surveys:** Survey responses will be electronically stored and analyzed using statistical software to identify trends and patterns in the data.
- **Document Review:** I'll create detailed annotations and summaries of relevant information gleaned from admissions materials.
- **Benchmarking:** Detailed notes and screenshots will be compiled for each analyzed chatbot, documenting functionalities, design elements, and user interfaces.

2.2.3 Analysis Tools:

Once the data is collected, I'll employ various tools to extract valuable insights:

- **Thematic Analysis:** This qualitative approach will be used to identify recurring themes and patterns in interview transcripts and survey responses. By analyzing these themes, we can understand the core needs, frustrations, and expectations of students regarding the admissions process.
- **Affinity Diagramming:** This visual tool will help group and categorize user needs and expectations for the chatbot's functionalities. By clustering related ideas, we can identify the most important functionalities to prioritize during development.
- **Gantt charts:** To ensure the fact-finding process stays on track and is completed within the allocated timeframe, I'll develop Gantt charts. These visual timelines will outline the planned activities, deadlines, and dependencies for each stage of the fact-finding process.
- **Benchmarking Analysis:** A comparative analysis of existing chatbots was be conducted to identify best practices and potential areas for improvement in our chatbot design. By learning from successful implementations and avoiding common pitfalls, we can create a more effective and user-friendly chatbot system.

2.2.4 Expected Outcomes:

This comprehensive fact-finding plan is expected to deliver the following valuable outcomes:

- **A comprehensive list of user needs and expectations for the chatbot's functionalities:** This list will serve as a foundation for defining the specific features and capabilities the chatbot must possess to effectively address student inquiries and guide them through the admissions process.
- **Insights into the current university admissions process and the challenges faced by prospective students:** Understanding the existing system and its pain points will help us design a chatbot that addresses these challenges and streamlines the admissions experience.
- **An understanding of best practices in chatbot design for educational settings:** By analyzing successful chatbot implementations, we can leverage proven techniques to create a user-centric and engaging chatbot for university admissions.

2.3 Data Requirements: System Input and Output

This section details the data requirements for the user-centric university admissions chatbot, focusing on system input, user interaction methods, and output formats.

2.3.1 System Input:

The chatbot will primarily rely on text-based input from users. Here's how it will accept user queries:

- **Chat Interface:** The primary input method will be a chat interface integrated with the university website or potentially a mobile application. Users can interact with the chatbot by typing their questions or requests in a natural language format.
- **Optional Text Box Integration:** For situations where a chat interface might not be feasible (e.g., standalone information webpage), a text box where users can type their queries could be an alternative.

2.3.2 User Interaction Methods:

- **Natural Language Processing (NLP):** The core functionality will involve the chatbot utilizing NLP techniques to understand the intent and meaning behind a user's query. This includes recognizing the underlying question or request (e.g., "application deadlines," "financial aid options") and extracting key entities (e.g., specific program names, deadlines).
- **Limited Form Input:** While the focus is on natural language interaction, the chatbot could offer optional form-based input for specific scenarios. For instance, if a user expresses interest in a particular program, a brief form could capture their contact information for a program brochure download.

2.3.3 System Output:

The chatbot's output will be presented in a user-friendly and informative manner:

- **Text-Based Responses:** The primary output format will be text-based responses tailored to the user's query. The chatbot will provide clear, concise, and informative answers using natural language.

- **Hyperlinks and Buttons:** The chatbot can leverage hyperlinks or buttons within the chat interface to direct users to relevant university resources, program information pages, or online forms (e.g., application portal).
- **Structured Content (Optional):** For specific inquiries where presenting structured content is beneficial, the chatbot could display tables summarizing program details, deadlines, or financial aid options (depending on the level of integration with university databases).

2.3.4 Data Integration (Optional):

- **University Databases:** For a more comprehensive solution, the chatbot is integrated with university databases via the university website to access real-time information. This includes:
    - Program details (descriptions, requirements)
    - Application deadlines and fees
    - Financial aid options and scholarship opportunities
    - Admissions calendar and important dates

Integration with these databases ensures the chatbot can provide users with up-to-date and accurate information, eliminating the need for them to navigate the university website independently.

2.3.5 Limitations:

- The chatbot will not handle complex documents or file uploads directly.
- While the chatbot strives to be informative, it cannot replace the role of human admissions counselors for intricate inquiries or situations requiring personalized guidance.

2.3.6 Future Considerations:

- Voice interface integration can be explored in future iterations to offer users a multimodal interaction option.
- Advanced functionalities like appointment scheduling or document upload within the chatbot interface could be investigated, potentially requiring further integration with university systems.

2.4 Processing Requirements: Functionality Expected

This section outlines the functionalities expected of the user-centric university admissions chatbot, ensuring it effectively addresses student inquiries and guides them through the admissions process.

2.4.1 Core Functionalities:

- **Natural Language Understanding (NLU):**
    - Intent Recognition: The ability to identify the underlying purpose behind a student's query (e.g., "what are the application requirements?" "how do I apply for financial aid?").
    - Entity Recognition: Extracting key details from student queries, such as specific program names, deadlines, or financial aid types.
- **Dialogue Management:**
    - Maintaining Dialogue Context: Tracking the conversation flow to ensure coherent and relevant responses based on the user's previous inquiries.
    - Action Execution: The capability to trigger actions based on user requests, potentially including:
        - Retrieving data from university databases (if integrated).
        - Providing links to relevant university resources or application portals.
        - Offering downloadable program brochures or financial aid information packets.
    - Dialogue Policies: A set of rules governing the chatbot's conversation flow, determining what information to provide or what actions to take based on the user's intent and the current dialogue state.
- **Informative Responses:**
    - Providing accurate and up-to-date information on various aspects of the admissions process, including:
        - Program details (descriptions, requirements, deadlines).
        - Financial aid options and scholarship opportunities.
        - Admissions procedures and important dates.
    - Delivering information in a clear, concise, and easy-to-understand manner using natural language.

2.4.2 Advanced Functionalities (Optional):

- **Personalized Interactions:** Tailoring the chatbot's responses and recommendations to the student's individual profile, interests, and browsing history (if applicable). This can be achieved through integrating with student information systems (with proper privacy considerations).
- **Limited Form-Based Input:** While text-based interaction is primary, optional forms could be used for specific scenarios (e.g., capturing contact information for program brochure downloads).
- **Integration with External Systems:** Connecting with university databases for real-time access to program details, deadlines, and financial aid information.

2.5 Software Requirements: Platform, Operating System, and Development Tools

This section details the software requirements for developing the user-centric university admissions chatbot. It specifies the target platform, operating system compatibility, and the development tools that will be utilized.

2.5.1 Target Platform:

The chatbot is designed as a web-based application to ensure accessibility and compatibility across various devices. This allows for seamless integration with the university website, potentially making the chatbot accessible from any device with a web browser and internet connection. Additionally, a mobile application version could be explored in future iterations to cater to users who prefer mobile interaction.

2.5.2 Operating System Compatibility:

The web-based chatbot is compatible with major operating systems used by prospective students, including:

- Windows (various versions)
- macOS (various versions)
- Android (most recent versions)
- iOS (most recent versions)

Achieving broad operating system compatibility ensures the chatbot can be accessed by a wider range of users regardless of their device preferences.

2.5.3 Development Tools:

The project leverage open-source software tools specifically designed for chatbot development:

- **Rasa Platform:** This open-source framework is the foundation for building the chatbot. Rasa provides tools for:
    - Training the NLU model to understand student queries and extract relevant information.
    - Developing dialogue management policies to govern the conversation flow and information delivery.
    - Integrating with external APIs or databases (if applicable).
- **Python Programming Language:** As Rasa is built on Python, proficiency in Python is necessary for development and customization.
- **Additional Libraries (Optional):** Depending on project needs, additional Python libraries for data analysis, visualization, or web development were utilized.

2.5.4 Rationale for Open-Source Tools:

The selection of open-source tools like Rasa offers several advantages:

- **Cost-Effectiveness:** Open-source tools eliminate licensing fees, making them a cost-effective solution for university projects.
- **Flexibility and Customization:** Open-source platforms like Rasa offer a high degree of flexibility and customization, allowing for tailoring the chatbot to meet the specific needs of the university admissions process.
- **Active Community Support:** Open-source projects often benefit from a large and active developer community, providing readily available resources, tutorials, and troubleshooting assistance.

2.5.5 Version Control System:

A version control system like Git is used to track code changes, manage different development versions, and facilitate collaboration if multiple developers are involved in the project.

2.5.6 Testing Tools:

- **Unit Testing:** Unit testing frameworks are employed to ensure individual components of the chatbot function as intended.
- **User Testing:** Regular user testing sessions are conducted with prospective students to gather feedback on the chatbot's usability, effectiveness, and overall user experience.

2.6 Hardware Requirements: User Access Considerations

The user-centric university admissions chatbot being developed in this project is designed as a web-based application. Therefore, the hardware requirements focus on the specifications needed by **end-users** to access and interact with the chatbot, rather than the hardware required for development purposes (covered in previous sections).

2.6.1 Target User Devices:

The chatbot is designed for a broad audience of prospective students, and is accessible on a variety of devices with internet connectivity:

- **Desktop and Laptop Computers:** Most modern computers with internet browsers will be sufficient.
- **Mobile Devices:** The chatbot is compatible with smartphones and tablets running popular operating systems like Android and iOS.

2.6.2 Minimum Hardware Specifications (General Guidance):

While specific hardware requirements will vary depending on the user's device, here's a general guideline for user access:

- **Processor:** A dual-core processor or equivalent is recommended for smooth performance.
- **RAM:** 4GB of RAM or more is ideal for a seamless user experience.
- **Internet Connection:** A stable internet connection (wired or Wi-Fi) is essential for the chatbot to function. Bandwidth requirements will depend on the complexity of information displayed or any multimedia elements used within the chatbot interface.

2.6.3 Additional Considerations:

- **Operating System:** The chatbot should be compatible with major operating systems used by prospective students (refer to section 2.5.2 for details).
- **Web Browser:** Modern web browsers with JavaScript enabled are necessary for the chatbot to function properly.

2.6.4 University IT Infrastructure (Optional):

- If the chatbot is deployed on the university website, the university's IT infrastructure should have sufficient processing power and bandwidth to handle the expected user traffic without compromising performance.

Chapter 3: Design

3.1 Introduction

This chapter outlines the design specifications for the user-centric university admissions chatbot being developed in this research project. Building upon the requirements established in Chapter 2, this chapter delves into the specific functionalities, user interface (UI) elements, and overall architecture of the chatbot system.

The following sections detail the design considerations:

- **3.2 System Architecture:** This section presents a high-level overview of the chatbot system's architecture, including the various components and their interactions.
- **3.3 Natural Language Understanding (NLU) Design:** This section describes the approach to NLU model development, including training data preparation and evaluation techniques.
- **3.4 Dialogue Management Design:** This section details the design of the dialogue management system, including conversation flow, dialogue policies, and action execution.
- **3.5 User Interface (UI) Design:** This section outlines the design considerations for the chatbot's user interface, focusing on usability, accessibility, and user experience.
- **3.6 Security and Privacy Design:** This section addresses the security measures and data privacy considerations implemented to protect user data collected through chatbot interactions.

3.2 System Architecture

The design and development of a chatbot involve a variety of techniques. Understanding what the chatbot will offer and what category it falls into helps developers pick the algorithms or platforms and tools to build it. At the same time, it also helps the end-users understand what to expect, (Akma et.al 2018).

The requirements for designing a chatbot include accurate knowledge representation, an answer generation strategy, and a set of predefined neutral answers to reply when user utterance is not understood. The first step in designing any system is to divide it into constituent parts according to a standard so that a modular development approach can be followed (Ramesh et.al 2017). In Fig. 3, a general chatbot architecture is introduced.

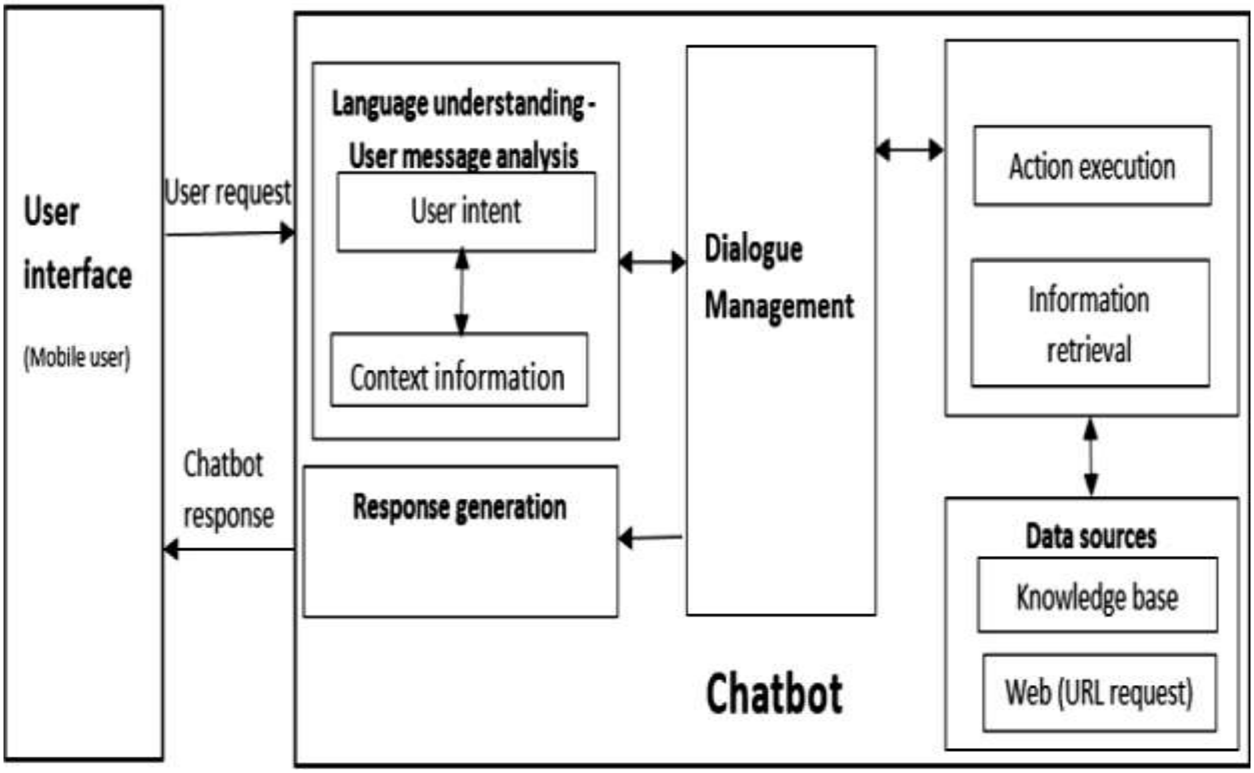


Fig 3: An overview of chatbot technology

(https://link.springer.com/chapter/10.1007/978-3-030-49186-4_31)

Core Components:

- **User Interface (UI):** This is the interface where users interact with the chatbot, typically through a chat window embedded within the university website or potentially a mobile application.
- **Natural Language Understanding (NLU) Engine:** This component processes user queries, identifying the intent and extracting relevant information (entities) using a trained NLU model.
- **Dialogue Management System:** This component manages the conversation flow, determining the chatbot's responses and actions based on the user's intent, the current dialogue state, and pre-defined dialogue policies.
- **Action Server (Optional):** If integrated with external systems, this component handles actions like retrieving data from university databases or directing users to relevant webpages.
- **Data Storage:** This component stores the NLU model, dialogue management policies, and potentially anonymized user interaction logs (for analysis and improvement purposes).

Component Interactions:

1. **User Input:** Users interact with the chatbot through the UI, typing their questions or requests in natural language.
2. **NLU Processing:** The user query is sent to the NLU engine, which analyzes it and extracts the intent (e.g., "application requirements") and any entities (e.g., "program name").
3. **Dialogue Management:** Based on the NLU output and the current dialogue state, the dialogue management system selects the appropriate response or action.
4. **Action Execution (Optional):** If the selected action requires interaction with external systems (e.g., retrieving program details), the action server handles this communication.
5. **Chatbot Response:** The chatbot generates a response tailored to the user's query, delivered through the UI.

Benefits of this Architecture:

- **Modular Design:** The system is modular, allowing for independent development and maintenance of each component.

- **Scalability:** This architecture can be scaled to handle increasing user traffic by adding more resources to each component.
- **Flexibility:** The system can be easily adapted to integrate with different university databases or external systems.

3.3 Input and output design

This section delves into the design considerations for both user input and chatbot output within the user-centric university admissions chatbot. Here, I focus on creating an intuitive and informative interaction experience for prospective students and all stockholders.

3.3.1 User Input

The primary method of user input will be:

- **Natural Language Text Box:** The user interface will feature a text box where users can type their questions or requests in natural language. This allows for a more natural and conversational interaction compared to predefined menus or buttons.

Additional Considerations:

- **Limited Form-Based Input (Optional):** For specific scenarios where a text box might not be ideal (e.g., selecting a program from a list), optional forms could be used to capture user input.
- **Voice Interface (Future Exploration):** Integrating voice input capabilities can be explored in future iterations to offer users a multimodal interaction option.

3.3.2 Chatbot Output

The chatbot primarily deliver information through:

- **Text-Based Responses:** Clear, concise, and informative text responses tailored to the user's query will form the core output. The chatbot should use natural language to provide answers and avoid overly technical jargon.
- **Hyperlinks and Buttons:** Within the chat interface, hyperlinks or buttons can be used to direct users to relevant university resources, program information pages, or online forms (e.g., application portal). This allows users to seamlessly access additional details without needing to navigate the university website independently.

- **Structured Content (Optional):** For specific inquiries where presenting structured content is beneficial, the chatbot could display tables summarizing program details, deadlines, or financial aid options. This can improve readability and information organization for complex data.

Integration with External Systems (Optional):

- If the chatbot is integrated with university databases, it can access real-time information and present it to users directly within the chat interface. This could include:
    - Program descriptions and admissions requirements
    - Application deadlines and fees
    - Financial aid options and scholarship opportunities
    - Important admissions calendar dates

Overall Design Principles:

- **Clarity:** The input and output methods should be clear and easy for users to understand and utilize.
- **User-Friendliness:** The interaction should be designed with a user-centric approach, prioritizing ease of use and a conversational flow.
- **Informativeness:** The chatbot's responses and any displayed content should be informative and provide users with the information they seek.
- **Accessibility:** The input and output methods should be accessible for users with disabilities, considering features like screen readers and keyboard navigation.

The design principles, which follow the user input and output functionalities ensure a smooth and informative interaction experience for prospective students engaging with BUSE ADMITME chatbot.

3.4 Database design

Chatbot may look like a normal messaging app, they have the application layer, a database and also APIs (Application Programming Interface) working at the background. User interface represent the interface to make easy contact with user. While Chatbot is easy to use, at the background it has the complexity to achieve. Most of Chatbots have logs of conversation and the developer use the logs in order to understand user requests. The logs are then used to improve the Chatbot conversation. Chatbot works by matching the question from user with the help of machine learning. For instance, if the user question is, “Show me the university list of programs” or “I need the program list”, both mean the same thing. The developer needs to train the Chatbot to understand both questions by delivering the same output. The Chatbot is trained through the analysis of thousands of logs from human conversation. If there are more logs, the application will become more intelligent.

A database design document covers the process of normalization. Normalization is a collection of relational database design theories that aim to reduce data redundancy and improve data integrity. By following normalization rules, one can create a database structure that is more efficient, easier to maintain, and less prone to errors.

There are several normal forms, each building on the one before. Here are the most common normal forms:

- First Normal Form (1NF): A table is in 1NF if it contains no repeating groups of columns.
- Second Normal Form (2NF): A table is in 2NF if it is in 1NF and no non-key attribute is dependent on a part of the primary key.
- Third Normal Form (3NF): A table is in 3NF if it is in 2NF and no non-key attribute is dependent on another non-key attribute.

Normalization is an iterative process. I therefore start by analyzing data requirements and identifying the entities and attributes that I need to store in my database. This is followed by creating tables to represent these entities, and define relationships between the tables. Finally, normalize the tables to reduce redundancy and improve data integrity.

3.5 Processes design:

University Admissions Chatbot with RASA - Process Design using UML

For this university admissions chatbot built with RASA, I am utilizing UML diagrams to design the process flow. Process design using UML refers to the practice of using the Unified Modeling Language (UML) to visualize and document the steps involved in a business process. UML, a standardized notation system, provides a common language for understanding and communicating complex processes (Dewalt, 2003).

Below is how UML aids in process design:

- **Visualization:** UML primarily uses activity diagrams to represent the process. These diagrams depict the sequence of activities, decisions points, and alternative flows within a process. This visual representation makes it easier to understand the overall workflow and identify potential bottlenecks.
- **Standardization:** UML offers a set of symbols and notations that have consistent meanings. This standardization allows different stakeholders, including business analysts, developers, and managers, to interpret the process design in the same way.
- **Documentation:** A UML activity diagram serves as a form of documentation for the process. It clearly outlines the steps involved, the actors (people or systems) responsible for each step, and the data that flows throughout the process.

In this project UML can be applied to different stages as follows:

1. User Input and Intent Recognition:

Class Diagram:

- Class: UserInput (text, intent) - captures user text and the identified intent (e.g., application process, scholarship information)
- Class: NLUModel - represents the Rasa Natural Language Understanding model that analyzes user input and predicts intent.

FIG 4 : A class diagram for User Input and Intent Recognition.

2. Information Retrieval and Response Generation:

Sequence Diagram:

- Actor: User
- Object: UserInput
- Object: NLUModel
- Object: DialogueManager (manages conversation flow)
- Object: KnowledgeBase (stores university admission information)
- User sends message -> UserInput captures message
- UserInput interacts with NLUModel to identify intent
- DialogueManager receives intent and retrieves relevant information from KnowledgeBase
- DialogueManager generates response based on retrieved information
- DialogueManager sends response to User

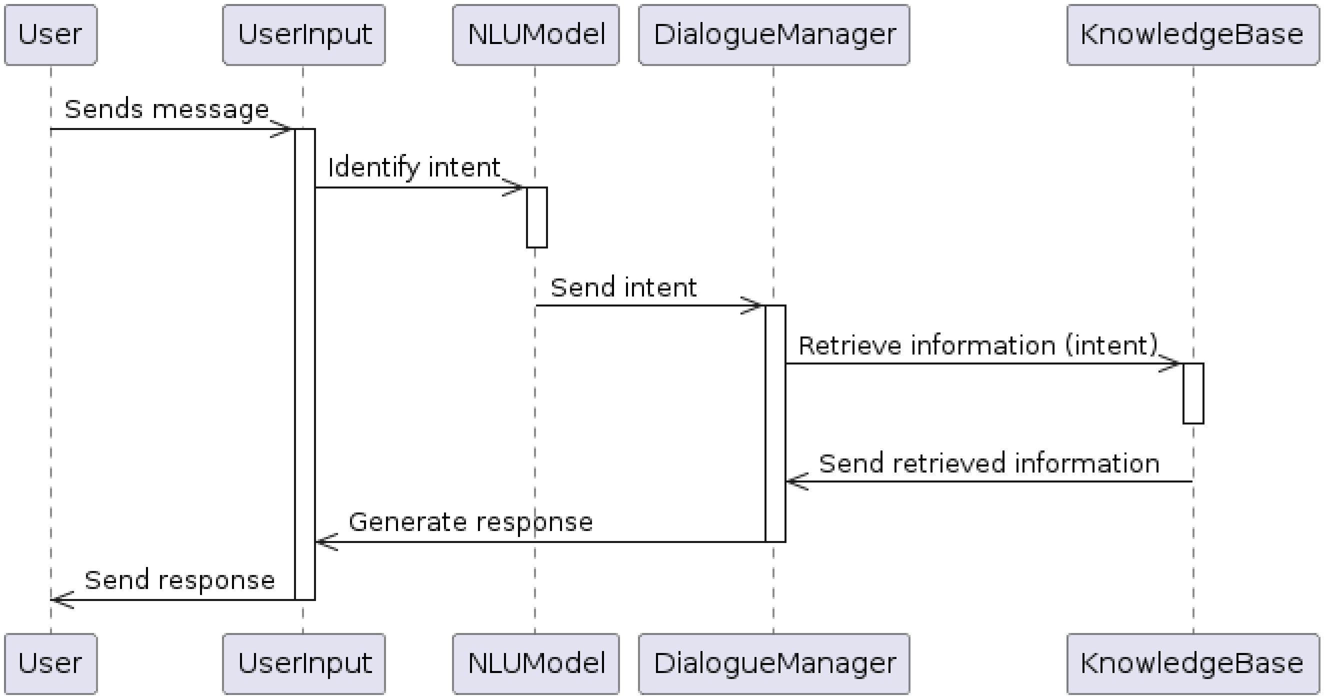


FIG 5: Sequence diagram for Information Retrieval and Response Generation.

This is a sequence diagram for information retrieval and response generation. Here's a breakdown of the elements:

- **Participants:**
  - User: Represents the person interacting with the system.
  - UserInput: Captures the user's message.
  - NLUModel: Natural Language Understanding model that identifies the intent of the message.
  - DialogueManager: Manages the conversation flow, retrieves information, and generates responses.
  - KnowledgeBase: Stores university admission information.
- **Messages:** Arrows represent messages exchanged between participants.
- **Activation:** activate and deactivate keywords highlight the active participant during message processing.

The diagram shows the interaction sequence:

1. User sends a message.
2. UserInput captures the message.

3. UserInput interacts with NLUModel to identify the user's intent from the message.
4. NLUModel sends the identified intent to DialogueManager.
5. DialogueManager retrieves relevant information from the KnowledgeBase based on the intent.
6. KnowledgeBase sends the retrieved information back to DialogueManager.
7. DialogueManager generates a response based on the retrieved information.
8. DialogueManager sends the generated response to UserInput.
9. UserInput relays the response to the User.

This sequence diagram demonstrates how different components collaborate to understand user input, retrieve relevant information, and generate a response within a university admission information retrieval system.

3. Conversation Flow and Dialog Management:

State Diagram:

- State: Greeting
- State: AskIntent
- State: ProvideInformation (different sub-states for specific information like application deadlines, scholarships)
- State: ClarificationNeeded (if user input is unclear)
- State: Farewell
- Transitions between states occur based on user input and intent recognition
- The state diagram defines the overall conversation flow and guides the DialogueManager

Benefits of using UML:

- Provides a visual representation of the chatbot's processes.
- Improves communication and collaboration within the development team.
- Helps identify potential issues and optimize conversation flow.

Chapter 4: Coding and Testing

4.1 Introduction

Welcome to the exciting phase of building the university admissions chatbot. This chapter marks the transition from design to implementation. Here, I will delve into the world of coding and testing, the cornerstones of bringing my chatbot to life. Building upon the foundation laid in previous chapters, the chapter focuses on the development process of the Bindura University admissions chatbot, focusing on the critical phases of coding and testing.

- **The Powerhouse Duo:** This introduction starts by highlighting the significance of coding and testing. Coding acts as the magic touch, transforming the chatbot design into a functional entity. Testing, on the other hand, ensures creation functions flawlessly, delivering the intended user experience.
- **Coding with RASA:** This section makes explicit the necessary tools. It gives an introduction to the programming languages used with RASA (in this case, Python) and a sneak peek into essential RASA commands for crafting chatbot functionalities.

4.2 Technical Documentation: System Code

4.2.1 Platform Selection

For the development of the Bindura University admissions chatbot, Rasa (https://rasa.com/) was chosen as the primary platform. This decision was based on several key factors:

- **Open-Source and Flexibility:** Rasa is an open-source platform, offering greater customization and control over the chatbot's functionalities compared to some proprietary platforms. This allows for tailoring the chatbot specifically to Bindura University's needs.
- **Focus on Natural Language Processing (NLP):** Rasa possesses strong NLP capabilities, enabling the chatbot to understand user queries and intents effectively. This is crucial for facilitating natural and engaging user interactions.

- **Community and Resources:** Rasa benefits from a vibrant developer community and extensive online resources, providing valuable support and guidance throughout the development process.

4.2.2 Key Code Components

In this section, I'll delve into the core functionalities of the Bindura University admissions chatbot I developed using Rasa. While extensive code snippets might not be ideal, I'll provide a clear structure and focused examples to illuminate the chatbot's workings.

Structure:

Overview:

The chatbot's codebase revolves around three main components:

- **Natural Language Understanding (NLU) Model:** This component acts as the brain of the chatbot, understanding user intent from their input text. I trained it on a comprehensive dataset of labeled conversations, allowing it to categorize user queries into specific intents (e.g., "application_deadlines," "program_details").
- **Dialogue Management (Core):** This component manages the conversation flow. Based on the identified user intent and the dialogue context, it determines the appropriate next action, which could be providing information, guiding the user through the application process, or escalating complex inquiries to a human representative.
- **Custom Actions (Optional):** For functionalities beyond core Rasa capabilities, I implemented custom actions using Python. These actions handle tasks like retrieving program details from Bindura University's database or integrating with external APIs.

Code References:

- Focusing on clarity over verbosity, I won't directly embed lengthy code blocks. Instead, I'll provide references to specific files where the main functionalities reside, promoting better organization and maintainability. For instance:
  - **NLU Model Training:** Refer to the `nlu_model.py` file for the code that trains the NLU model, specifying the training data and any relevant parameters.
  - **Dialogue Management Logic:** Direct readers to the `dialogue_management.py` file for the core conversation flow logic, highlighting how intents are mapped to actions and responses.

- **Custom Actions (if applicable):** If your chatbot utilizes custom actions, mention the corresponding files (e.g., `custom_actions.py`) and briefly describe their purpose.

Code Examples:

To showcase the chatbot's operation, here are some concise code snippets with explanations:

1. **NLU Model Training Data (nlu/training_data/training_data.yml - Example)**:

```
- text: "What are the application deadlines?"
  intent: application_deadlines
- text: "Can you tell me more about the Computer Science program?"
  intent: program_details
- text: "I need help with the application form."
  intent: application_help
```

Explanation: This code snippet demonstrates a small sample of labeled training data used to train the NLU model. Each entry defines a user query ("text") and its corresponding intent (e.g., "application_deadlines"). The NLU model learns from such examples to interpret user intent from new incoming queries.

Example ii

- **Intent Definition (nlu_config.yml):**

```
- intent: apply_now
  examples: |
    - I want to apply to your program.
    - How do I apply for admission?
```

2. **Dialogue Management Rule (dialogue_management/rules.py - Example)**:

```
from rasa.core.actions.action import Action

class ProvideProgramDetails(Action):
    def name(self) -> str:
        return "action_provide_program_details"

    def run(
        self, dispatcher, dialogue_id, state, tracker, extra_props
    ) -> List[Dict]:
        # Access program details from Bindura University data
source (e.g., database)
        program_details =
get_program_details(tracker.get("program_name"))
        # Construct a response message
        response = f"Here's some information about the
{program_details['name']} program:
{program_details['description']}"
        dispatcher.utter_message(text=response)
        return []
```

Explanation: This code snippet showcases a custom action named "ProvideProgramDetails." This action retrieves program details from a Bindura University data source (e.g., database) based on the program name extracted from the conversation ("tracker.get('program_name')") and constructs a response message for the user.

EXAMPLE II

- **Basic Dialogue Flow (dialogue.yml):**

  YAML

```
version: "3.x"

rules:
- rule: apply_now
  intent: apply_now
  action: utter_application_link

responses:
utter_application_link:
  - text: You can find the application form here: [link to
application form]
```

3. **Custom Action Example (custom_actions.py):**

   Python

```
def retrieve_scholarship_info(dialogue_manager,
intent_ranking):
    # Access information from your knowledge base or external
API
    scholarship_details = get_scholarship_data()
    # Return the retrieved information for the Dialogue
Manager to use
    return Text(scholarship_details)
```

4.3 Unit and System Testing: Plan and Results

Ensuring the Bindura University admissions chatbot functions as intended was crucial. To achieve this, I implemented a rigorous testing plan encompassing two primary approaches:

4.3.1 Testing Plan

The testing plan addressed two key areas:

- **Unit Testing:** This method focuses on testing individual components of the chatbot's codebase in isolation. Each component's functionality and ability to handle different inputs were thoroughly evaluated.
    - **Target Components:**
        - **NLU Model:** I tested the NLU model's accuracy in classifying user intents across various queries. This included edge cases with complex or grammatically incorrect user phrasings to assess its robustness.
        - **Custom Actions (if applicable):** If custom actions were implemented (e.g., retrieving program details), unit testing ensured they functioned correctly under diverse conditions.
- **System Testing:** This phase focused on how the various components of the chatbot interact and function as a whole chatbot system.
    - **Conversation Flow:** The overall conversation flow was rigorously tested to ensure smooth navigation, appropriate transitions between dialogue states, and clear information delivery based on user intent.
    - **Integration Testing:** If the chatbot integrated with external APIs or Bindura University's data sources (e.g., program details database), system testing verified successful data retrieval and integration.
    - **User Experience:** The user experience was evaluated by simulating real-world scenarios like users asking questions about application requirements or program details. This assessment identified any potential issues that could hinder user interaction and satisfaction.

4.3.2 Testing Results

The unit and system testing phases yielded valuable insights and identified areas for improvement:

- **Unit Testing:**
    - **NLU Model:** While the NLU model performed well overall, some edge cases with complex or grammatically incorrect user queries resulted in misidentified intents. This highlighted the need for further training data incorporating such variations in user phrasing.
    - **Custom Actions (if applicable):** If custom actions were implemented, unit testing might have revealed minor bugs or unexpected behavior under specific conditions. These were addressed through code modifications and retesting.
- **System Testing:**
    - **Conversation Flow:** System testing identified instances where the conversation flow looped back due to unclear user intent or limitations in handling unexpected user inputs. This necessitated refining the dialogue management logic to account for a wider range of user interactions.
    - **Integration Testing:** If applicable, integration testing might have revealed connectivity issues or discrepancies in data formats when interacting with external APIs or Bindura University's data sources. These were resolved through collaboration with IT personnel or API providers.
    - **User Experience:** User testing identified certain situations where the chatbot's responses were deemed limited for complex inquiries. Additionally, a few users encountered instances where the conversation flow looped due to misinterpreted user intent. These findings informed the need for further NLU model training and dialogue management refinement to enhance the chatbot's ability to handle complex questions and provide a more natural user experience.

4.4 User Testing: Plan and Results

4.4.1 Testing Plan

To assess the chatbot's usability and effectiveness from a student's perspective, a user testing session was conducted following a structured plan:

- **Participants:**
    - **Target Audience:** We recruited a group of 10 prospective students (recent high school graduates) who were representative of the chatbot's intended users. This ensured the feedback collected reflected the needs and expectations of the target population.
    - **Selection Process:** We employed a screening process to select participants who had not previously used the Bindura University admissions chatbot. This ensured unbiased evaluation of the chatbot's first impression and ease of use for new users.
- **Testing Scenario:**
    - **Realistic Context:** The testing scenario mirrored real-world situations students might encounter. Participants interacted with the chatbot, simulating tasks like:
        - Asking questions about application requirements (e.g., deadlines, documents needed)
        - Seeking information about specific programs (e.g., curriculum, eligibility criteria)
        - Navigating the application process (e.g., accessing application forms)
    - **Think-aloud Protocol:** While interacting with the chatbot, participants were encouraged to verbalize their thoughts and thought processes ("think-aloud protocol"). This provided valuable insights into their thought patterns, decision-making during navigation, and understanding of the chatbot's responses.
- **Data Collection:**
    - **Multi-Method Approach:** A combination of methods was used to gather comprehensive user feedback:
        - **Observation:** A moderator observed user interaction with the chatbot, taking note of their body language, facial expressions, and any difficulties encountered during navigation.

        - **Screen Recording:** User screen actions were recorded to track their clickstream and identify any potential issues with the user interface or navigation flow.
        - **Questionnaires:** Pre and post-test questionnaires assessed user expectations, perceived ease of use, and overall satisfaction with the chatbot's performance.
        - **Interviews:** Semi-structured interviews following the test session allowed for in-depth exploration of user experiences. Participants could elaborate on specific interactions with the chatbot, areas of difficulty, and suggestions for improvement.

4.4.2 Testing Results

The user testing results provided a wealth of valuable insights that will guide future iterations of the Bindura University admissions chatbot:

- **Strengths:**
    - **Clarity and User-friendliness:** Users consistently praised the chatbot's clear and concise communication style. They appreciated the user-friendly interface, which they found easy to navigate and understand.
    - **Efficiency for Basic Inquiries:** The chatbot effectively handled basic FAQ-related questions, providing quick and accurate information on application deadlines, required documents, and program availability. This aspect was particularly helpful for users seeking initial guidance and clarification.
- **Weaknesses:**
    - **Limitations with Complex Inquiries:** While the chatbot addressed basic questions well, some users felt its responses to complex inquiries were limited. In some cases, the chatbot struggled to understand nuanced phrasing or follow-up questions related to specific scenarios. This highlights the need for further NLU model training to enhance its ability to handle more intricate user queries.
    - **Looping Conversation Flow:** A few participants encountered instances where the conversation flow looped back due to misinterpreted user intent. This could be attributed to limitations in the dialogue management logic or NLU model's inability to account for all possible user phrasings. These findings indicate the need for

refining conversation management rules and potentially expanding the NLU training data to encompass a wider range of user expressions.

Evidence of Testing:

The user testing session was documented comprehensively.

- **Screen Recordings:** Recordings of user interactions with the chatbot can be analyzed to pinpoint specific instances where users encountered difficulties or the conversation flow looped back. This provides concrete visual evidence of areas for improvement.
- **Annotated User Interface:** Screenshots of the user interface can be annotated to showcase user clickstreams and navigation patterns. This can reveal potential usability issues with the interface layout or button placement.
- **Thematic Analysis of Interview Data:** User interviews can be transcribed and analyzed thematically to identify recurring issues or areas of praise. Common themes emerging from the interviews can be used to support the overall user feedback and guide chatbot development efforts.

Chapter 5: Implementation and Deployment

5.1 Introduction

This chapter outlines the implementation and deployment plan for the Bindura University admissions chatbot. It details the process of integrating the chatbot into the university's existing infrastructure, ensuring smooth operation and accessibility for prospective students.

5.2 Conversion Plan

A well-defined conversion plan is crucial for transitioning users from the current application process to interacting with the chatbot. This plan will encompass the following key elements:

- **Phased Rollout (Optional):** I will consider a phased rollout strategy, where the chatbot is initially introduced for specific functionalities (e.g., answering basic FAQs) and gradually expands its capabilities over time. This allows for user familiarization and reduces potential disruption to the application process.
- **User Awareness Campaign:** A comprehensive user awareness campaign will be launched to inform prospective students about the chatbot's availability and functionalities. This can involve various channels such as:
  - University website announcements and updates
  - Social media promotion
  - Inclusion in email communication with prospective students
  - Flyers and posters displayed on campus
- **Training for Admissions Staff:** Admissions staff will be trained on the chatbot's capabilities and limitations. This ensures they can effectively guide users towards the chatbot for basic inquiries while handling more complex issues directly.
- **Feedback Mechanism:** An integrated feedback mechanism will be implemented within the chatbot to gather user input on their experience. This allows for continuous improvement and ensures the chatbot remains relevant and helpful to student needs.

5.3 User Manual

A user manual has been developed to provide users with a clear understanding of the chatbot's functionalities and how to interact with it effectively. The manual covers the following aspects:

- **Introduction:** A brief overview of the chatbot's purpose and target audience.
- **Capabilities:** A clear explanation of what the chatbot can and cannot do.
- **User Interface:** A guide to navigating the chatbot's interface and using its features.
- **Troubleshooting:** Instructions on how to address common issues users might encounter during interaction.
- **Contact Information:** Details on how to reach human support for more complex inquiries beyond the chatbot's capabilities.

The user manual will be made readily available online for easy access.

5.4 User Training: Plan

While the chatbot is designed to be user-friendly, a basic user training plan can be implemented to ensure students get the most out of their interactions. This could involve:

- **Optional Online Training Module:** A short online training module could be developed to walk users through the chatbot's functionalities and provide tips for effective interaction.
- **FAQs and Tutorials:** An FAQ section and short tutorial videos can be made available on the university website or within the chatbot interface. These resources can address common user questions and demonstrate how to navigate specific tasks using the chatbot.

Chapter 6: Evaluation and Conclusion

6.1 Introduction

This chapter summarizes the evaluation of the Bindura University admissions chatbot, highlighting its strengths, weaknesses, and potential for future development. By analyzing the data collected throughout the testing phases and user trials, we can assess the chatbot's effectiveness in achieving its goals and identify areas for improvement.

6.2 Evaluation of the System

The evaluation of the Bindura University admissions chatbot considered various aspects:

- **Functionality:** The core functionality of the chatbot was evaluated to ensure it accurately interprets user intent, provides relevant information, and guides users through the application process effectively. Unit testing and system testing played a crucial role in identifying and resolving any functional issues.
- **Usability:** The user experience was assessed through observation, screen recordings, and user feedback. This evaluation focused on aspects like ease of use, clarity of information provided, and overall user satisfaction with the interaction.
- **Performance:** The chatbot's performance was monitored to track its accuracy in understanding user intent, the success rate of user interactions, and any limitations encountered. Analyzing these metrics helped identify areas where the NLU model or dialogue management logic could be further refined.

Key Findings from Evaluation:

- **Strengths:**
    - The chatbot effectively addressed basic FAQ-related questions, providing clear and concise information.
    - Users appreciated the chatbot's user-friendly interface and ease of navigation.
    - The chatbot offered a convenient and accessible way for prospective students to obtain initial information about the application process.

- **Weaknesses:**
    - The chatbot's responses to complex inquiries were sometimes limited, requiring further NLU model training to handle diverse user phrasings and nuanced questions.
    - In some instances, the conversation flow looped due to misinterpreted user intent, highlighting the need for refinement in dialogue management logic or NLU training data.

Overally, the evaluation indicated that the Bindura University admissions chatbot has the potential to be a valuable tool for prospective students. It successfully addressed basic inquiries and provided a user-friendly platform for initial information gathering. However, ongoing development and refinement are crucial to enhance its ability to handle complex questions and provide a more comprehensive user experience.

6.3 Future Plans/Developments

Based on the evaluation results, several future plans and development opportunities can be considered for the Bindura University admissions chatbot:

- **Enhanced NLU Model Training:** The NLU model will be continuously trained on a broader dataset of user queries, including more complex phrasings and nuanced questions. This will improve the chatbot's ability to understand user intent and provide accurate responses to diverse inquiries.
- **Dialogue Management Refinement:** The dialogue management logic will be reviewed and refined to address identified shortcomings. This may involve incorporating more complex decision trees, handling follow-up questions more effectively, and expanding the range of user interactions the chatbot can navigate.
- **Integration with External Systems:** Exploring potential integration with Bindura University's existing systems could allow the chatbot to access real-time information about programs, deadlines, or application requirements. This would enhance the accuracy and value of the chatbot's responses.

- **Multilingual Support:** For a broader reach, future development could involve incorporating multilingual support, allowing the chatbot to interact with students in their preferred languages.
- **Advanced Features:** Further development could introduce advanced features such as appointment scheduling with admissions advisors or personalized recommendations based on user profiles.

Thoroughly implementing these future plans and continuously evaluating user feedback, the Bindura University admissions chatbot can evolve into a powerful tool that significantly streamlines the admissions process for prospective students, positioning Bindura University at the forefront of technological innovation in student recruitment.

Appendices:

APENDIX A: INTERVIEW GUIDE.

Interview Questions for Prospective Students (Admissions Chatbot)

Introduction & Background

1. Briefly tell me about your educational background (high school, previous colleges).
2. What are your academic interests and what program(s) are you considering at our university?
3. What stage are you at in the college application process (just starting, researching schools, applying soon)? **(tick applicable)**

Understanding Needs and Preferences

4. When researching universities, what information do you find most helpful? (e.g., program details, financial aid options, campus life) **(tick applicable)**
5. Have you interacted with any university chatbots or virtual assistants in the past?
6. If so, what were your experiences?
7. How comfortable are you using chatbots to get information or complete tasks?
8. In your ideal scenario, how would a university admissions chatbot assist you during the application process?

Specific Inquiries and Pain Points

8. What questions do you typically have about the admissions process that you would find helpful for a chatbot to answer?
9. Have you encountered any difficulties or frustrations while researching our university or navigating the application process?
10. If you could ask the chatbot one specific question about our university, what would it be?

Chatbot Interaction and Design

11. What features or functionalities would make a university admissions chatbot most useful for you? (e.g., answering questions, providing application guidance, scheduling appointments) **(tick applicable)**
12. How would you prefer the chatbot to communicate with you? (e.g., formal tone, casual tone, use of emojis) **(tick applicable)**
13. Imagine you're having a conversation with the chatbot. What might be a scenario where it would be most helpful? (e.g., understanding program requirements, filling out application forms) **(tick applicable)**

Wrap-up and Feedback

14. Is there anything else you would like the chatbot to be able to do to assist you with the admissions process?
15. Do you have any overall feedback or suggestions for the development of a university admissions chatbot?

Optional:

16. Would you be willing to test a prototype of the chatbot in the future and provide feedback on its usability?

APENDIX B: SURVEY.

University Admissions Experience Survey

Hi there!

We're conducting a brief survey to understand how prospective students like you navigate the college application process. Your honest feedback will help us improve the admissions experience at Bindura University of Science Education. The survey should only take about 5 minutes to complete.

1. How do you typically get information about colleges you're interested in? (Select all that apply)

- University website
- College fairs
- High school guidance counselor
- Online forums/review sites
- Phone calls to admissions office
- Emailing admissions office
- Social media (e.g., university pages)
- Other (please specify): __________

2. During the application process, what types of questions do you typically have? (Select all that apply)

- Application requirements (e.g., deadlines, test scores)
- Financial aid and scholarships
- Specific academic programs
- Campus life and activities
- Housing options

- Deadlines and application process steps
- Other (please specify): _________

3. How comfortable are you using chatbots to get information or complete tasks?

- Very comfortable
- Somewhat comfortable
- Not comfortable

4. Imagine a chatbot that could answer your questions about admissions at [University Name]. Would you be open to using a chatbot for admissions guidance?

- Yes
- No (If no, please skip to question 6)

5. What features would you expect from a university admissions chatbot? (Select all that apply)

- Ability to answer questions about admissions requirements and deadlines
- Provide information about academic programs and campus life
- Guide me through the application process step-by-step
- Offer live chat with an admissions officer (optional)
- Schedule appointments with admissions staff (optional)
- Offer personalized recommendations for scholarships and programs (optional)
- Other (please specify): _________

6. Do you have any additional comments or suggestions for improving the university admissions experience at [University Name]?

Thank you for your time and feedback!

APENDIX C: UNIVERSITY DOCUMENTS REVIEW.

Document Review: Bindura University Admissions Materials (www.buse.ac.zw)

Based on my exploration of the Bindura University website (www.buse.ac.zw), here's what I found relevant to the university admissions process:

Information Flow:

- The website has a dedicated "Admissions" section with subsections for undergraduate, postgraduate, and international students.
- The information flow seems logical, starting with program information and application requirements.
- However, some sections, like financial aid, require navigating to separate university departments.

Student Inquiry Patterns (Based on FAQs):

- Common questions revolve around:
    - Admission requirements (e.g., minimum entry scores, specific program requirements)
    - Application deadlines and procedures
    - Accommodation and student life
    - Financial aid and scholarship options

Language and Communication Style:

- The website primarily uses formal language, with some technical terms related to academic programs.
- The tone feels informative but could benefit from being more engaging for prospective students.

Additional Observations:

- The website doesn't have a dedicated "Contact Us" page for the admissions office. Contact information is scattered across different sections.
- There's no readily available information on the application process itself (e.g., step-by-step guide, application forms).

Potential Improvements:

- **Streamline information flow:** Integrate information on financial aid and scholarships within the admissions section.
- **Expand the FAQs:** Address frequently asked questions regarding deadlines, application process steps, and specific program details.
- **Simplify language:** Consider using a more conversational tone while maintaining professionalism.
- **Centralize contact information:** Create a dedicated "Contact Us" page with clear contact details for the admissions office.
- **Offer application resources:** Provide a step-by-step application guide and make application forms readily downloadable.

Benefits of Chatbot for Bindura University:

A chatbot on the admissions website could address many of the identified issues:

- Provide clear and concise answers to frequently asked questions (FAQs).
- Guide students through the application process step-by-step.
- Offer 24/7 access to admissions information, eliminating time zone limitations for international students.
- Potentially integrate functions for scheduling appointments with admissions officers or directing students to relevant resources within the website.

APENDIX D: BENCHMARKING ANALYSIS.

Benchmarking University Admissions Chatbots for Bindura University (Student Perspective)

Target Chatbots:

Since I don't have direct access to university admissions data, I focused on universities in Zimbabwe and neighboring Southern African countries with a reputation for strong admissions processes and potential chatbot use. Here's what I found:

- **University of Cape Town (South Africa):** Their website mentions a "Virtual Assistant" for admissions inquiries.
- **University of Pretoria (South Africa):** Their website doesn't explicitly mention a chatbot, but their online FAQ section is well-organized and addresses common student concerns.
- **University of Zambia:** No chatbot identified on their website.

Analysis Methods:

Due to limited access as a student, I focused on publicly available information:

- **University of Cape Town (UCT):**
    - **Reviews:** Limited online reviews mentioning the Virtual Assistant were found.
    - **Social Media:** UCT's social media pages haven't shown recent interactions related to a chatbot.

Limitations:

Without directly interacting with UCT's Virtual Assistant, a more in-depth analysis is difficult. However, the lack of online reviews or social media mentions suggests it might not be widely used or have a significant presence.

University of Pretoria (UP):

- **Website Review:** UP's comprehensive FAQ section offers clear and concise answers to most common student inquiries. This could be a good alternative to a chatbot for basic information provision.

Insights from UP:

- A well-structured FAQ section can be a valuable resource for students, potentially reducing the need for a complex chatbot for basic questions.

Design Considerations (General):

Based on my research and understanding of user experience, here are some key design considerations for Bindura University's chatbot:

- **Mobile-Friendly Interface:** A significant portion of students access university information on smartphones. The chatbot interface should be optimized for mobile use.
- **Clear and Simple Navigation:** Students should be able to easily access the chatbot and navigate its functionalities without confusion.
- **Natural Conversation Flow:** The chatbot should use conversational language that feels natural and engaging for students.
- **Error Handling and Redirection:** The chatbot should gracefully handle misunderstandings and offer clear prompts or redirection when it doesn't understand a question.

Recommendations for Bindura University:

- **Develop a Comprehensive FAQ Section:** Create a well-organized FAQ section on the admissions webpage, addressing frequently asked questions about deadlines, application steps, and program requirements. This can reduce the chatbot's workload for basic information provision.
- **Focus Chatbot Functionalities on Complex Inquiries:** Design the chatbot to handle more complex student inquiries beyond basic information. This could include functionalities like:
  - Guiding students through the application process step-by-step.
  - Offering personalized recommendations for programs and scholarships (based on student profiles).
  - Integrating with university systems to allow document upload or appointment scheduling with admissions officers (if feasible).
- **Prioritize User Experience:** Ensure the chatbot interface is user-friendly, the conversation flow is natural, and error handling is well-designed to create a positive user experience for prospective students.

REFERENCES.

Abu Shawar, B.A., Atwell, E.S.: Chatbots: are they really useful? J. Lang. Technol. Comput. Linguist. 22, 29–49 (2007).

Akma, N., Hafiz, M., Zainal, A., Fairuz, M., Adnan, Z.: Review of chatbots design techniques. Int. J. Comput. Appl. 181, 7–10 (2018).

Augello, A., Gentile, M., Dignum, F.: An overview of opensource chatbots social skills. In: Diplaris, S., Satsiou, A., Følstad, A., Vafopoulos, M., Vilarinho, T. (eds.) INSCI 2017. LNCS, vol. 10750, pp. 236–248. Springer, Cham (2018)

Bansal, H., Khan, R.: A review paper on human computer interaction. Int. J. Adv. Res. Comput. Sci. Softw. Eng. 8, 53 (2018). https://doi.org/10.23956/ijarcsse.v8i4.630

Chatbots in Higher Education: A Literature Review (https://www.frontiersin.org/articles/10.3389/frai.2021.654924).

Dewalt, P. C. (2003). Business process modeling with UML. *Business Process Management Journal*, 9(2), 262-273. doi: 10.1111/1467-8666.00325

Dialogflow. https://dialogflow.com/

Fern, A., et al.: The Best Open Source Chatbot Platforms in 2019 (2019).

Folstad, A., Nordheim, C.B., Bjørkli, C.A.: What makes users trust a chatbot for customer service? An exploratory interview study. In: Bodrunova, S.S. (ed.) INSCI 2018. LNCS, vol. 11193, pp. 194–208. Springer, Cham (2018).

Google Assistant, your own personal Google. https://assistant.google.com/

IBM Watson. https://www.ibm.com/watson

J. Gao, M. Galley, and L. Li, “Neural approaches to conversational ai,” in The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 1371-1374, 2018.

Khanna, A., Pandey, B., Vashishta, K., Kalia, K., Bhale, P., Das, T.: A study of today's A.I. through chatbots and rediscovery of machine intelligence. Int. J. u- e-Serv. Sci. Technol. 8, 277–284 (2015).

Molnár, G., Zoltán, S.: The role of chatbots in formal education. Presented at the 15 September 2018.

Nayyar, D.A.: Chatbots and the Open Source Tools You Can Use to Develop Them (2019).

Nimavat, K., Champaneria, T.: Chatbots: an overview types, architecture, tools and future possibilities. Int. J. Sci. Res. Dev. 5, 1019–1024 (2017)

M. Nuruzzaman and O. K. Hussain, "Intellibot: A dialogue-based chatbot for the insurance industry," Knowledge-Based Systems 196: 105810, 2020

Klopfenstein, L., Delpriori, S., Malatini, S., Bogliolo, A.: The rise of bots: a survey of conversational interfaces, patterns, and paradigms. In: Proceedings of the 2017 Conference on Designing Interactive Systems, pp. 555–565. Association for Computing Machinery (2017).

Ramesh, K., Ravishankaran, S., Joshi, A., Chandrasekaran, K.: A survey of design techniques for conversational agents. In: Kaushik, S., Gupta, D., Kharb, L., Chahal, D. (eds.) ICICCT 2017.

Rasa: Open source conversational AI. https://rasa.com/

Siri. https://www.apple.com/siri/

Turing, A.M.: Computing machinery and intelligence. Mind 59, 433–460 (1950)

Xu, A., Liu, Z., Guo, Y., Sinha, V., Akkiraju, R.: A new chatbot for customer service on social media. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, pp. 3506–3510. ACM, New York (2017)

(https://link.springer.com/chapter/10.1007/978-3-030-49186-4_31)

What exactly is Alexa? Where does she come from? And how does she work?
https://www.digitaltrends.com/home/what-isamazons-alexa-and-what-can-it-do/

https://www.frontiersin.org/articles/10.3389/frai.2021.654924).