MACHAKOS UNIVERSITY

SCHOOL OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMMUNICATION AND INFORMATION TECHNOLOGY

BSC COMPUTER SCIENCE

WECARE: MENTAL HEALTH CHATBOT FOR SUICIDE DETECTION, AND SUPPORT SYSTEM

DENNIS KIBET

J17-5580-2020

THIS PROJECT PROPOSAL IS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF BACHELOR OF SCIENCE IN COMPUTER SCIENCE AT MACHAKOS UNIVERSITY

2023

Declaration

I, Dennis Kibet do hereby declare that this proposal is original and has not been published or submitted for any other degree award to any university before. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

Sign …………………….. Date………………………..

Supervisor

Sign…………………….. Date………………………….

Abstract

In recent times, global concern over mental health issues, particularly related to suicide prevention, has significantly escalated. Integrating technology, specifically Machine Learning (ML), into suicidal detection systems aims to enhance operational measures that have been deployed. This project addresses the challenges faced by individuals dealing with mental health crises by leveraging ML to develop a Mental Health Chatbot for suicidal detection and support. This project examines current gaps in mental health support systems, such as limited accessibility, delayed response times, and a lack of personalized assistance. The systems currently in use often struggle to identify potential suicide risks in a timely manner. This project highlights the importance of technology in bridging these gaps, offering real-time accessibility and personalized support to individuals in distress. Notably, a significant portion of the youth population already owns smartphones, indicating the potential feasibility of implementing the chatbot. This project emphasizes the critical role of ML-driven chatbots in revolutionizing mental health support, specifically in suicide prevention. By addressing deficiencies in existing systems, the project offers a promising avenue for improving mental health outcomes. Ultimately, this represents a potential breakthrough in mental health, offering hope and paving the way towards a more effective responsive support system.

Dedication

I dedicate my dissertation work to my family and friends. A special feeling of gratitude to my loving parents Mr. and Mrs. William Kiprotich whose words of encouragement and push for tenacity ring in my ears.

Acknowledgement

My greatest gratitude goes to God for making this research proposal possible. I would like to acknowledge Machakos University, and the entire staff for their support throughout this study. Special gratitude to my graduate colleagues for their support and academic guidance throughout the research proposal.

Table of Contents

List of figures

List of tables

List of Abbreviations

MLA- Machine Learning Algorithm.

RF- Random Forest algorithm.

API - Application Programming Interface.

URL - Uniform Resource Locater

SDLC – Software Development Life Cycle

NLTK – Natural Language Toolkit

ELECTRA - Efficiently Learning an Encoder that Classifies Token Replacements Accurately

BERT - Bidirectional Encoder Representations from Transformers

NLP – Natural Language Processing

Chapter One: Introduction

1.1 Introduction

This chapter shows the background research on the project concept

1.2 Background of the study

Mental health is a critical concern in Kenya, where the burden of mental disorders is substantial and growing. According to the World Health Organization (WHO), mental health disorders are estimated to affect over 4 million people in Kenya, constituting about 10% of the population (WHO, 2017). Despite the prevalence of mental health issues, there is a significant lack of accessible and affordable mental health services, exacerbating the problem. The Mental Health Task Force Report in Kenya (2015) highlighted the need for innovative approaches to address the mental health challenges, including leveraging technology.

In recent years, advancements in technology have paved the way for innovative solutions to tackle mental health issues. Chatbots, artificial intelligence-driven programs designed to simulate human conversation, have shown promise in providing mental health support. They offer an accessible and stigma-free platform for individuals to seek guidance and assistance. Dr. Mary Mwangangi, a mental health advocate in Kenya, emphasizes the potential of digital solutions: "Chatbots have the potential to bridge the mental health care gap in our country by providing immediate support and information to those in need" (Mwangangi, 2021).

The WeCare project aims to contribute to the mental health landscape in Kenya by developing a chatbot that offers emotional support and resources to individuals struggling with mental health issues. Through the integration of culturally sensitive approaches and evidence-based practices, WeCare aims to enhance mental health awareness, reduce stigma, and provide accessible support to those in need. This initiative aligns with the Kenyan government's commitment to addressing mental health challenges and promoting well-being within the population (Kenya Vision 2030).

1.3 Problem statement

Mental health remains a pressing issue in Kenya, with a significant gap in accessible and affordable mental health support. The World Health Organization (WHO) has emphasized the need for better mental health services globally, and in Kenya, mental health disorders affect approximately 10% of the population, yet a substantial portion remains without adequate assistance (WHO, 2017). Stigma, lack of awareness, and limited resources further exacerbate this challenge. In Kenya, like in many parts of the world, accessing mental health services can be challenging due to various barriers such as cost, stigma, and limited availability of mental health professionals. Dr. Susan Njoroge, a mental health advocate, underscores this issue: "Many Kenyans face challenges in accessing mental health care due to financial constraints and stigma associated with seeking help" (Njoroge, 2020). These barriers deter individuals from seeking timely and appropriate support for their mental health needs.

The WeCare project aims to address this problem by developing a mental health chatbot that can provide immediate support and guidance to individuals struggling with mental health issues. By leveraging advancements in artificial intelligence and natural language processing, WeCare aims to create a user-friendly platform that offers information, coping strategies, and a listening ear. This aligns with the Kenyan government's commitment to improving mental health services and reducing the mental health burden on the population (Kenya Ministry of Health, 2019). By creating an accessible and stigma-free environment through the WeCare chatbot, we intend to bridge the gap in mental health support and provide an avenue for individuals to seek help, guidance, and information regarding their mental well-being. The development of WeCare seeks to contribute to the improvement of mental health outcomes and enhance the overall quality of life for individuals in Kenya.

1.4 Research Objectives

1.4.1 Overall Objectives

The main objective of this study was to design, develop and implement a tool that automatically detects depression and suicidal intents which requires less human intervention for the society.

1.4.2 Specific Objectives

1. To review the literature on suicidal detection models.
2. To design the suicidal intents detection machine learning model.
3. To develop a system utilizing the machine learning model.
4. To test and implement suicidal and DialogGPT chatbot model to the system.

1.5 Scope of the Study

The study focused on the development of a mental health chatbot utilizing machine learning, specifically the ELECTRA transformer model, to identify depression and suicidal intents. Implementation was carried out through Python FLASK, React.js for web development, and Flutter for mobile platforms. The most effective model was selected based on evaluation results.

1.6 Limitation of the Study

1. Hardware and Computing Resource Limitations: Due to restricted hardware capabilities and computing resources, the full potential of model training was hindered, which impacted the overall efficiency and speed of the study.
2. Accuracy Challenges in Suicidal Intent Detection: The machine learning models employed may not consistently deliver accurate results in detecting suicidal intents, indicating a need for further refinement and improvement.
3. Limited and Small Datasets: The availability of datasets for training the models was severely constrained, with the existing datasets being small in size. This limitation affected the comprehensiveness and reliability of the models.
4. Language and Demographic Constraints: The study was limited to English speakers, as the accessible datasets are predominantly in English. This restricted the applicability and reach of the mental health chatbot to a specific linguistic demographic.
5. Noisy Inputs and Dataset Cleaning: The dataset used may have contained noisy or unstructured data, necessitating a cleaning process. Noisy inputs could have negatively impacted the performance and accuracy of the models, requiring additional preprocessing efforts.

1.7 Justification

This study addressed the pressing need for effective mental health support in an increasingly digital world. By developing a chatbot leveraging machine learning, it aimed to mitigate limited human interaction in mental health discussions, ensuring round-the-clock accessibility. Additionally, it strived for enhanced privacy, reduced stigma, and tailored assistance, making mental health resources more approachable and comfortable for individuals seeking help.

Chapter Two: Literature Review

2.1 Introduction

This chapter explored the existing literature on mental health chatbots and the techniques utilized in automated text analysis to deepen the understanding of this concept and investigate the research problem. It described the current tools available and highlighted gaps, particularly in the context of mental health chatbot support within the machine learning domain. The chapter examined the nature of mental health chatbot interactions and the current processes in place to monitor and enhance the effectiveness of mental health support through chatbots.

2.2 Related Concepts

2.2.1 Suicidal and Depression Detection in Kenya

Like many countries, Kenya faces a concerning issue regarding suicide and depression detection. The World Health Organization (WHO) reported in 2016 that suicide is a significant public health concern, with an estimated 4.6 suicides per 100,000 population. Efforts are being made to improve detection and intervention. According to Dr. Rachna Patel, a mental health expert, "Early recognition of warning signs and accessible mental health services are crucial in preventing suicides" (Patel, 2019). In Kenya, organizations such as the Kenyan Mental Health Taskforce have been advocating for improved mental health services and awareness. Their report, published in 2015, emphasizes the need to integrate mental health services into the primary healthcare system and increase mental health literacy in communities. Additionally, initiatives like the Africa Mental Health Research and Training Foundation (AMHRTF) work towards enhancing mental health research and training in the country to aid in early detection and effective intervention (AMHRTF, n.d.). Collaborative efforts from both governmental and non-governmental sectors are essential to address this critical issue and create a supportive environment for those affected by depression and suicidal thoughts.

2.3 Machine Learning Algorithms

2.3.1 Convolutional Neural Network (CNN)

As this project attempts to classify text data, the sequence of words plays a part in contributing to the connotation of a sentence. According to Ce and Tie (2020), the Convolutional Neural Network (CNN) was proposed as an efficient way to classify text data as it is able to achieve decent prediction accuracy and consume lesser computational resources.

Model Architecture

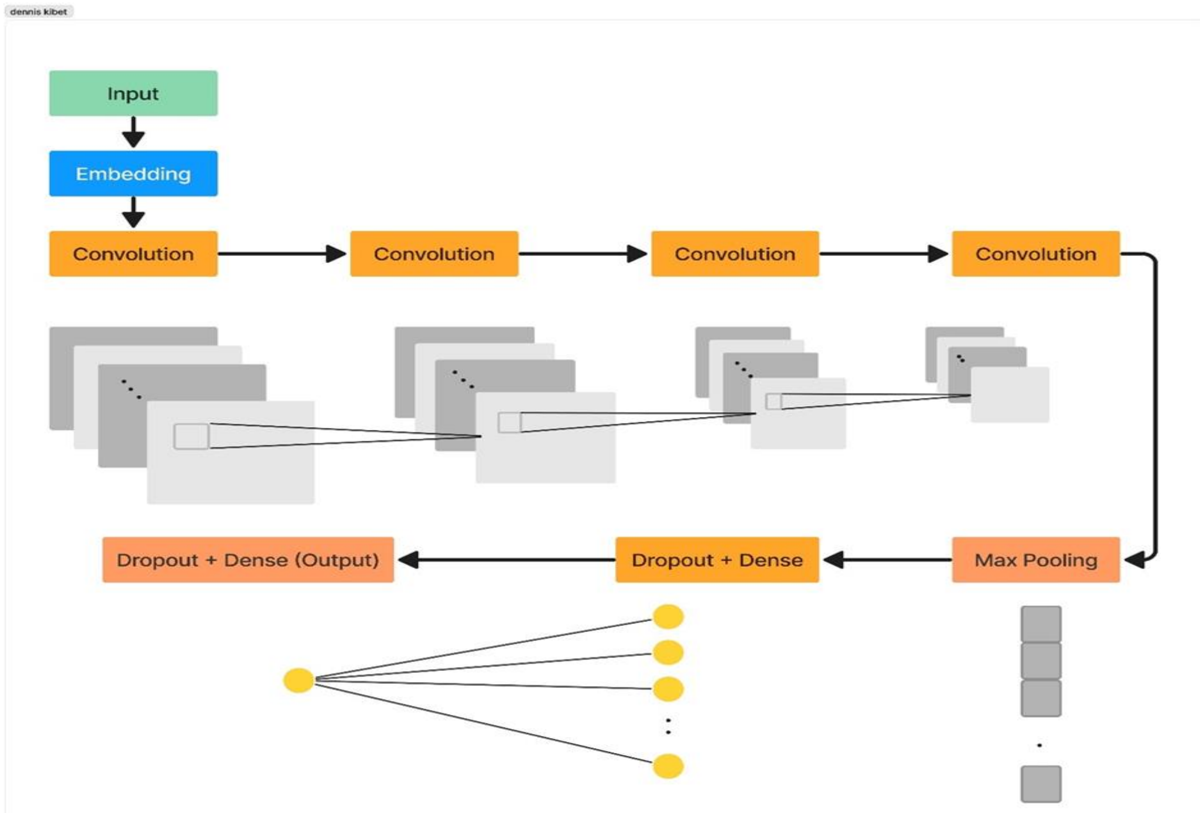

Figure 1. CNN Model Architecture

The CNN model architecture consisted of the following layers: an embedding layer, 4 convolutional layers, a pooling layer, a dropout layer, and a fully connected layer, as shown in Figure 1 above.

In the CNN model, text data was tokenized, converting words to integers. The embedding layer mapped each word to a fixed-size vector, creating a space where similar words were close. Convolutional layers extracted context, with earlier ones capturing basic information and later ones identifying key features and sentiments. Max-pooling focused on vital sentence features, regardless of word order. Dropout was applied to prevent overfitting, and the final fully connected layer, followed by a sigmoid activation, classified text as suicidal or non-suicidal.

2.3.2 Long Short-term Memory Network (LSTM)

Although CNN models are useful in text classification, they detect local and position-invariant key phrases and are unable to detect a long-range semantic dependency like Recurrent Neural Network (RNN) models (Minaee et al., 2021). RNNs works better than CNN with sequential data such as text, however, they suffer from the problem of vanishing gradients (Zhao et al., 2019).

LSTM was capable of taking in longer sequences and remembering longer dependencies in a sequence. This was achieved due to its capability of retaining memory of relevant information and forgetting irrelevant information. More importantly, LSTM alleviated the vanishing gradient problem faced by RNNs through a memory cell in the LSTM layer to remember values over time intervals.

Model Architecture

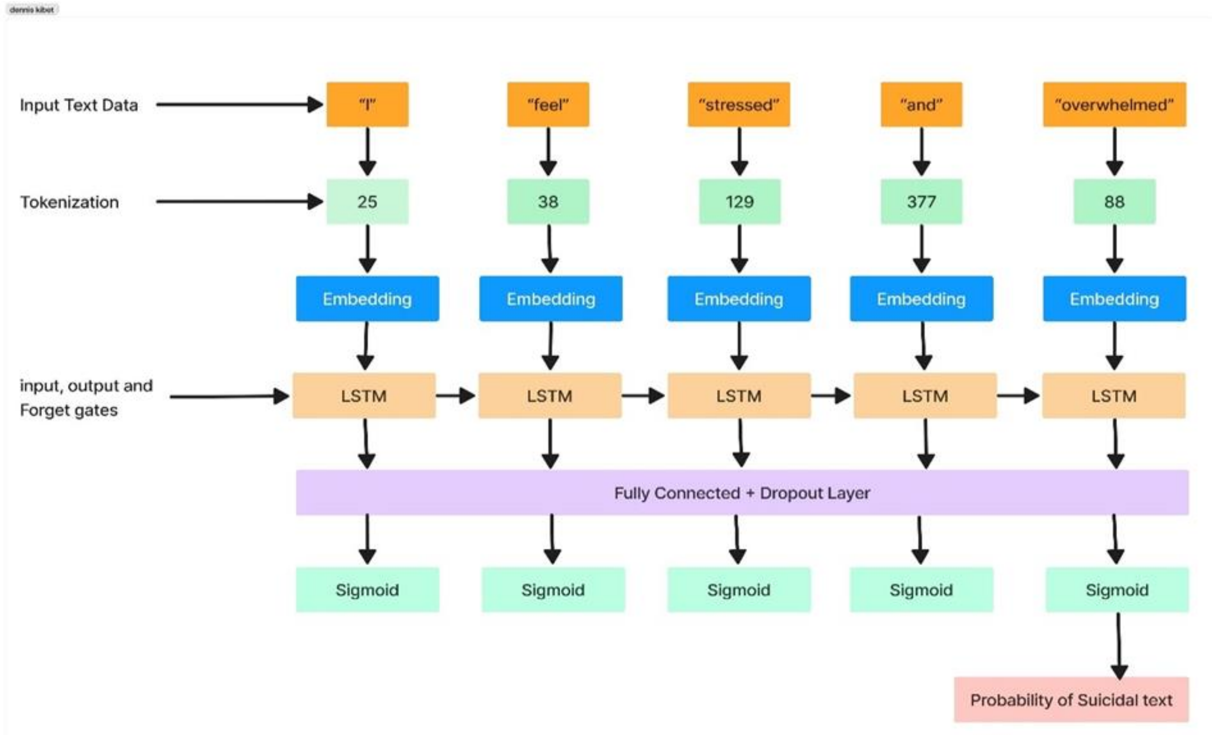

Figure 2. LSTM Model Architecture

The LSTM network architecture consisted of 5 layers as shown in Figure 2 above. Before the first layer, the data was tokenized to convert the words into integers. The first layer was the embedding layer, where the integer word tokens were converted into embedding vectors. The second layer was the LSTM layer where information was passed through, and the gates within the LSTM model decided the information to forget, store, or output. The third layer was a fully connected layer responsible for mapping the output of the LSTM layer to our output size of 1. The fourth layer contained the sigmoid activation function, classifying whether the text was suicidal or not suicidal. The final layer was the output layer where the output was received after the last time step and reshaped such that the number of predictions equaled the batch size.

2.3.3 Transformers Algorithms

2.3.3.1 BERT

BERT, also known as Bidirectional Encoder Representations from Transformers, utilized the encoder structure of a transformer for language modelling and was developed by Google in 2018 (Devlin et al., 2018).

Model Architecture

BERT was pre-trained on 2 tasks, namely Masked Language Model (MLM) and Next Sentence Prediction (NSP). The MLM technique enables bidirectional training in contrast to previous models which looked at a text sequence from a single-direction or combined bidirectional training.

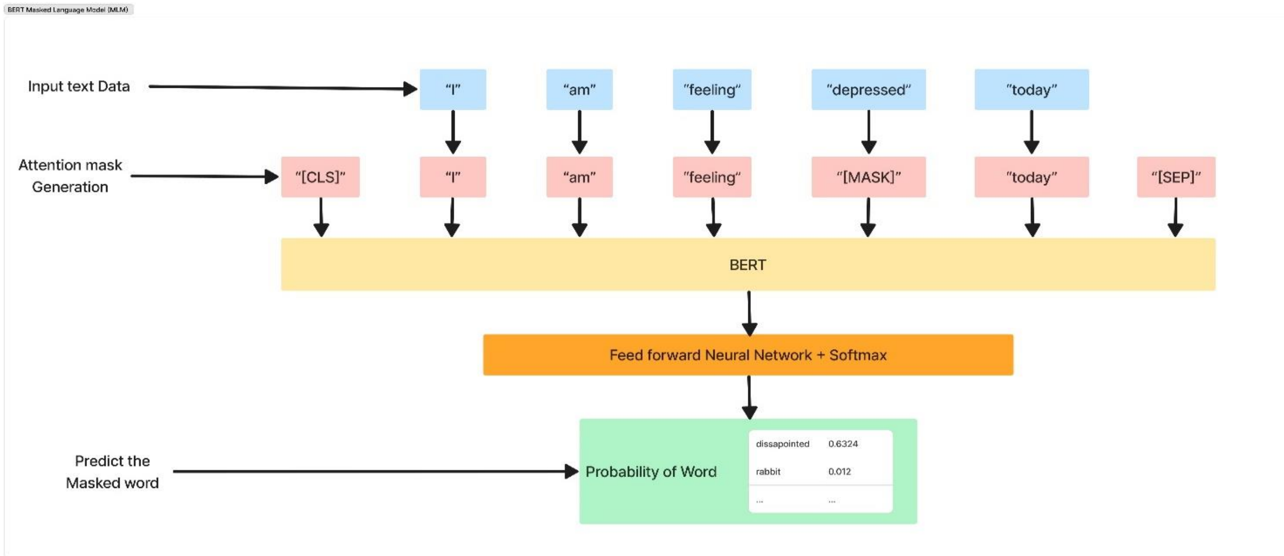

Figure 3. BERT Masked Language Model (MLM)

From Figure 3 above, the input sentence "I am feeling depressed today" was transformed into "I am feeling [MASK] today" after masking. The model was then trained to replace the masked tokens with the correct word, which then allowed the model to learn more accurate representations with the attention mechanism. MLM replaces 15% of the words in the sequences with a "[MASK]" token for training purposes.

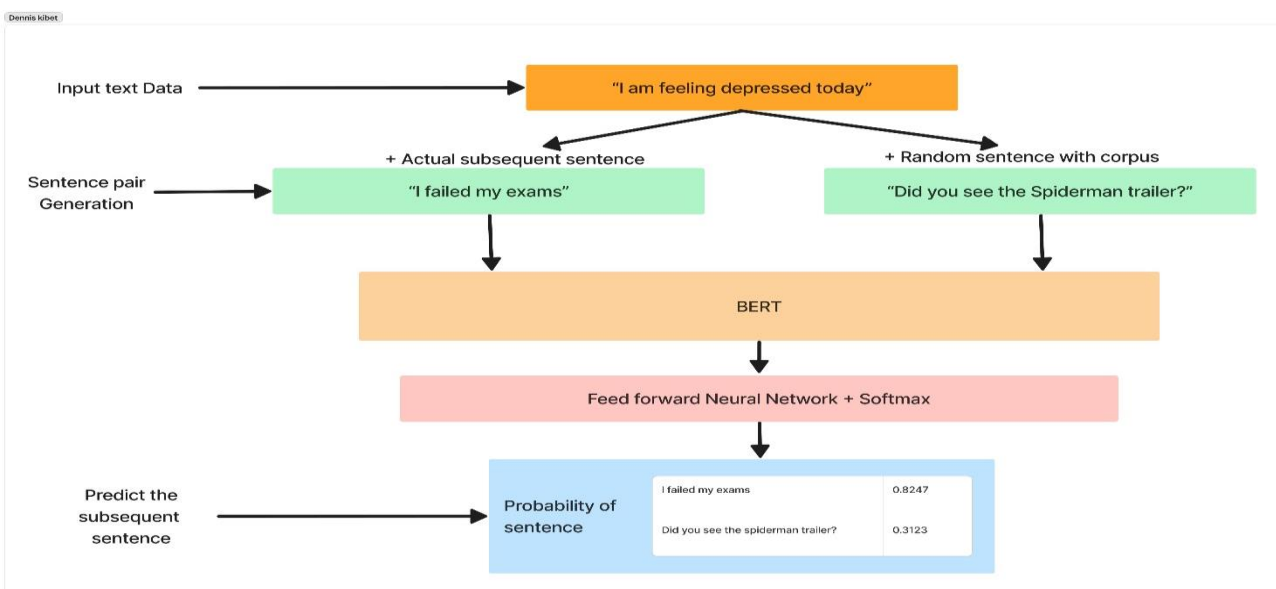

Figure 4. BERT Next Sentence Prediction (NSP)

MLM captured sentence features at a token level but did not capture sentence-level information, which was achieved by incorporating the NSP task. In the training process, the model received pairs of sentences as input and learned to predict if the second sentence in the pair was the subsequent sentence. From Figure 4 above, it can be observed that "I failed my exams" was the actual subsequent sentence for the input text “I am feeling depressed today”, while “Did you see the Spiderman trailer?” was a random sentence within the corpus. For each sentence, one real and one fake input were used as the training data. Additional tokens “[CLS]” and “[SEP]” were used to denote the sentence boundary at the start and the end, respectively (Figure 3 above). BERT was pre-trained on the English Wikipedia and Books Corpus data, containing 3.3 billion words, and made available in two variations — base and large.

2.3.3.2 ELECTRA

ELECTRA, also known as Efficiently Learning an Encoder that Classifies Token Replacements Accurately, one of the latest pre-trained transformer models released by Google in 2020 (Clark et al., 2020). Unlike other enhanced transformer models like BERT, such as RoBERTa and XLNet, which required larger datasets demanding greater computational power and longer training times, ELECTRA outperformed these models on several benchmark datasets with less than a quarter of the required computation power.

Model Architecture

ELECTRA adds a new way called Replaced Token Detection (RTD) to address the limitations of the masked language Model (MLM) technique used in BERT, which involved replacing input tokens with masked tokens.

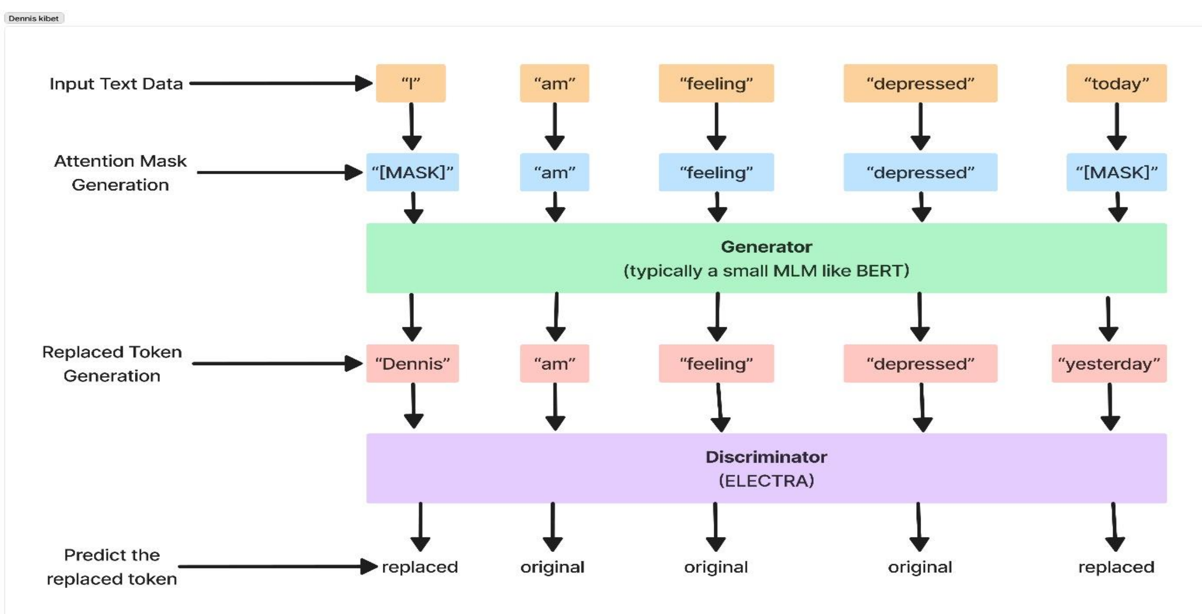

Figure 5. ELECTRA Replaced Token Detection (RTD)

ELECTRA utilized RTD to train a bidirectional model where tokens were replaced with incorrect but plausible fakes instead of “[MASK]” tokens. For instance, the input sentence “I am feeling depressed today” could be transformed into “Dennis am feeling depressed yesterday” (Figure 5 above). Compared to sentences replaced with “[MASK]” tokens, the replaced sentence made more sense but still did not entirely fit within the context.

The discriminator's primary task was to detect tokens that had been substituted with others, inspired by generative adversarial networks (GANs), to distinguish between real and fake input data. This binary classification objective promoted a more precise learning of data representations across all input positions, as the model needed to acquire a precise understanding of the data to perform this task effectively. ELECTRA demonstrated an advantage in requiring less training data to achieve equivalent model performance due to its ability to receive more information per example.

The tokens used for replacement were generated by a generator, typically a smaller model like BERT in the context of a masked language model (MLM). The generator and discriminator were co-trained while sharing the same input word embeddings.

2.4 Model Selection and Evaluation

After extensive research on the best machine learning models, the ELECTRA transformer model was selected due to its performance in classifying suicidal texts, making it efficient and effective for the chatbot. Other compared models included Convolutional Neural Network (CNN), Long Short-term Memory (LSTM), and Bidirectional Encoder Representations from Transformers (BERT). The transformer model ELECTRA (Efficiently Learning on Encoder that Classifies Token Replacements Accurately) was built.

The model was trained using the train dataset and then fine-tuned using the validation dataset. It was subsequently tested on the test dataset. The model's performance on the test dataset was evaluated based on four evaluation metrics:

i. Accuracy
ii. Precision
iii. Recall
iv. Final Best Score (F1 score).

More focus was placed on the final best score for the use case to ensure a better model in handling the downstream tasks in the application. False negatives were not ideal, and the F1 score provided a more balanced perspective of the model's performance compared to recall.

2.5 Related Systems

2.5.1 Arif Chatbot project in Kenya

A collaboration between Zindi, Mtoto News, Basic Needs Basic Rights Kenya, and Qhala resulted in the pioneering of a mental health chatbot, powered by machine learning, designed to support young people in Kenya, particularly university students. The application of machine learning and data science was a breakthrough in Africa's efforts to address mental health challenges (Zindi, 2019). Through a targeted Tech4MentalHealth challenge organized by Basic Needs Basic Rights Kenya and Zindi in 2019, a text-based dataset was curated, focusing on key phrases describing mental health states, laying the foundation for the chatbot's development (Loki, 2019). The objective was to categorize users based on mental health indicators, enabling appropriate support mechanisms (Loki, 2019). The resulting chatbot, named Arif, was launched on Telegram, marking a critical milestone in leveraging technology to redirect users to suitable support systems and counselors within their schools (Loki, 2019). The initiative, although challenging, received positive feedback, demonstrating its effectiveness in interpreting user interactions and aiding in directing affected individuals to appropriate professionals for support (Loki, 2019). As mental health concerns escalate globally, there is a growing need for proactive solutions, emphasizing the importance of further development and expansion of the chatbot platform to address mental health and other critical issues within Africa (Loki, 2019).

2.5.2 Vera: A WhatsApp Chatbot Revolutionizing Healthcare Access in Kenya

Zuri Health, a health tech startup in Kenya, created 'Vera,' a WhatsApp chatbot designed to bridge the gap between patients and healthcare services in Kenya. In a Kenya where a 97% of internet users are active on WhatsApp, the integration of Vera represents a strategic move to democratize healthcare access. The founder of Zuri Health, Ikechukwu Anoke, highlights the necessity of diversifying communication channels to extend healthcare accessibility. Vera operates 24/7, offering real-time responses and efficiently connecting patients to vital healthcare services and products. Notably, Vera accommodates multiple inquiries simultaneously, reducing the need for physical hospital queues. Supporting multilingual capabilities in English, Swahili, and French, Vera facilitates a seamless healthcare experience for users, enabling them to schedule appointments, purchase medication, and book essential lab tests. To engage with Vera, users simply initiate a conversation by sending a greeting to the provided WhatsApp number or via a designated link, kickstarting their journey toward accessible healthcare services (Zuri Health).

2.6 Research Gap

While mental health chatbots have a potential solution for providing support and resources to individuals experiencing mental health challenges, there exists a significant research gap in understanding the complexities of their effectiveness and ethical considerations surrounding their implementation. Studies generally focus on technical functionalities and user satisfaction, neglecting several critical aspects like Personalization, Clinical Validation, Ethical Considerations, User Engagement and Long-term Adoption, Integration with Existing Mental Health Services and Cultural and Socioeconomic Considerations. The ELECTRA model used in this study addressed these research gaps, and thus contributing to a more comprehensive understanding of the potential of mental health chatbots as a tool for offering support, leading to more effective and ethically sound interventions in the field of mental health care.

2.7 Conceptual Framework

The approach in this project was grounded in established psychological theories such as the Interpersonal Theory of Suicide and the Stress-Diathesis Model, providing a foundation for understanding the risk factors and triggers for suicidal thoughts and behaviors. The project employed the application of natural language processing and machine learning techniques. The ELECTRA model was employed for text analysis, sentiment detection, and intent recognition within chat conversations, based on the theoretical framework of transfer learning and pre-trained models for text classification.

The chatbot's conversational abilities were powered by the DialoGPT-large language model, leveraging theoretical foundations in dialogue generation and understanding within NLP. The process involved data collection from individuals seeking support, preprocessing, sentiment analysis, and identification of suicidal intent. When such intent was recognized, the chatbot provided structured support based on cognitive-behavioral principles and offered crisis helpline numbers, connecting users with professionals and crisis services.

Continuous evaluation of the ELECTRA model's performance and adherence to ethical guidelines further contributed to the conceptual and theoretical framework of this innovative approach, ensuring its alignment with best practices in the field of mental health support and technology.

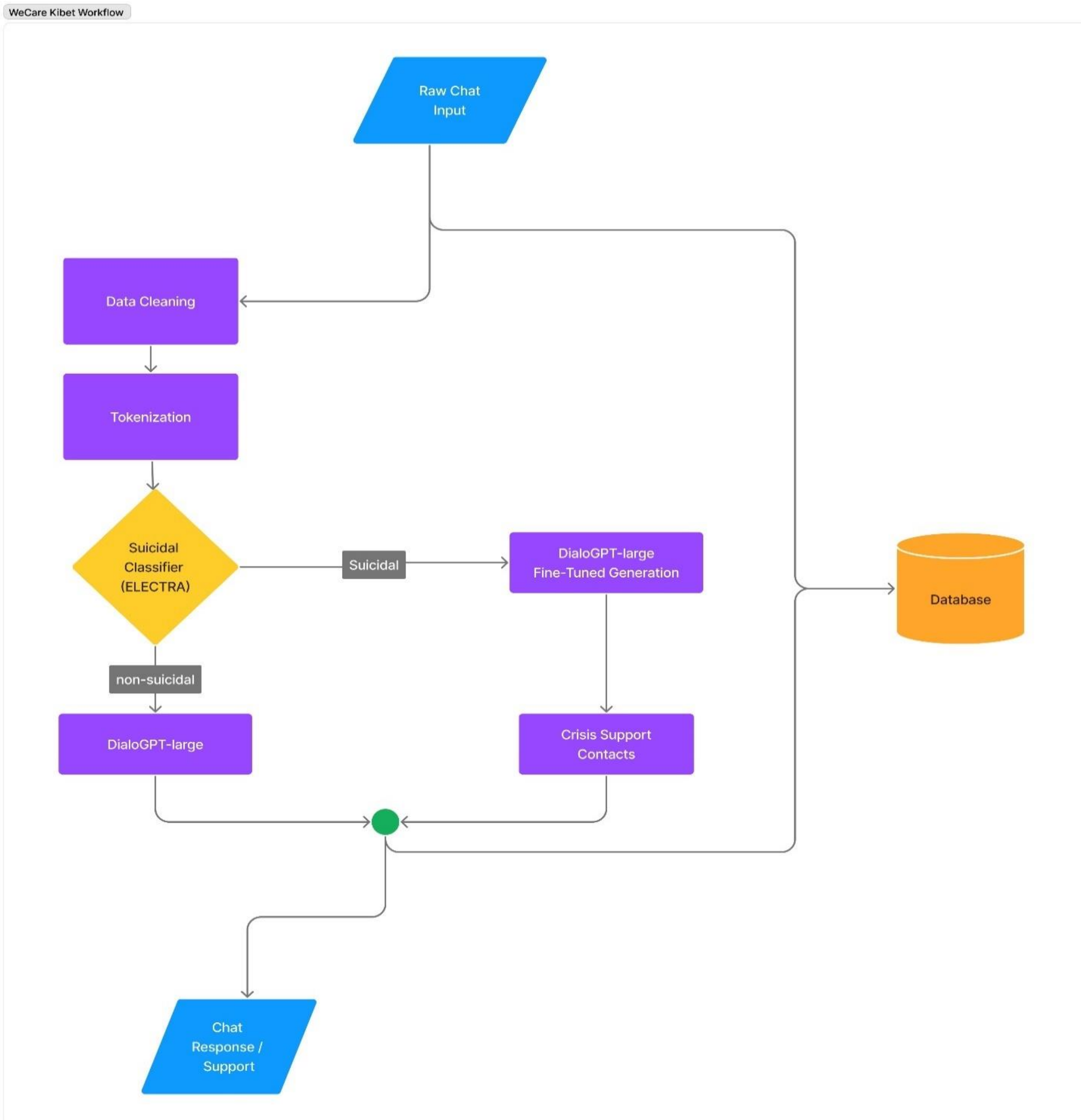

Figure 6. WeCare Workflow

Chapter Three: Research Methodology

3.1 Introduction
This chapter outlines the chosen methodology, detailing the approaches to data collection, analysis and research approaches.

3.2 Research Design
The research problem was text categorization, primarily yielding qualitative results. From a computer science perspective, the research leaned towards mathematical methods for automatically detecting underlying patterns in text to generate a model. Consequently, a mixed-method strategy was adopted, utilizing both qualitative and quantitative methods. The study was guided by an approach involving a series of experiments that examined various feature combinations and machine learning algorithms to identify the most effective features and optimal classification algorithm for detecting mental health concerns in chatbot interactions. Given the empirical nature of this research, a technique that was easier to compute was chosen. This technique was well-suited for the research, as it promoted the use of repeatable and generalizable scientific methods for knowledge generation.

3.3 Sampling Methods
In contrast to earlier research that relied on web crawlers to collect a large volume of data based on specific pre-defined keywords on social media, this approach doesn't adhere to conventional sampling methods. Collecting a substantial dataset on mental health discussions from various social media sources proved to be impractical and ineffective. Instead, this study employed a straightforward random sampling technique to construct a representative sample for annotation within the extensive Kaggle dataset. Utilizing convenience sampling, data from the Kaggle dataset which consists of post from the social media platform reddit. The dataset consisted of 2 columns where “text” indicated the post content and “class” indicated the label of the posts. A total of 232,074 posts were scraped from 2 subreddits, namely SuicideWatch and teenagers to build the dataset. Posts from SuicideWatch were labelled as suicidal, while the posts collected from teenagers were labelled as non-suicidal, thereby simplifying the collection process.

3.4 Data Collection Methods
This project used a machine learning approach which heavily relied on data to train the models that was eventually used to classify the suicidal and non-suicidal texts from the user’s inputs. The ELECTRA model, utilized in this project was trained using sample training corpora that was a subset of the entire dataset collected from Kaggle. In this project, interviews, questionnaires and language bias based on culture, habits and conventions was used as part of a mixed-method research approach (combination of qualitative and quantitative).

3.4.1 Observation
The social and organizational needs used was determined using an observational technique. Once the data had been sorted into two categories: negative and positive. It was examined to determine whether the classifier had done so correctly.

3.4.2 Questionnaires

Questionnaires offer a structured approach to gather specific information from a diverse pool of respondents. It facilitates scalability, allowing distribution to a large audience, thus ensuring a comprehensive dataset.

3.4.2.1 Findings From Questionnaire

From the issued questionnaire analysis, several key findings emerged regarding participants' attitudes and experiences related to suicide detection chatbots and mental health support. Out of the 12 responsdents 35% reported that they have previously interacted with a system designed for suicidal detection or aid. while participants demonstrated varying degrees of familiarity with chatbots, with 30% indicating a moderate level of familiarity and 20% indicating a high level, it is evident that chatbots are a familiar concept to a considerable portion of the surveyed population. Additionally, participants commonly utilized messaging apps, social media platforms, and email for communication, underscoring the potential avenues through which suicide detection chatbots could be integrated into existing communication channels. Furthermore, the majority of participants expressed a belief in the potential of technology to address mental health challenges, with 90% acknowledging the significant role that technology can play in providing access to resources and support. Moreover, participants overwhelmingly emphasized the importance of privacy and data security when utilizing suicide detection chatbots, with 80% rating these aspects as extremely important. These findings collectively underscore the potential of suicide detection chatbots as a valuable tool in mental health support while highlighting the need for robust privacy measures and user-centric design considerations.

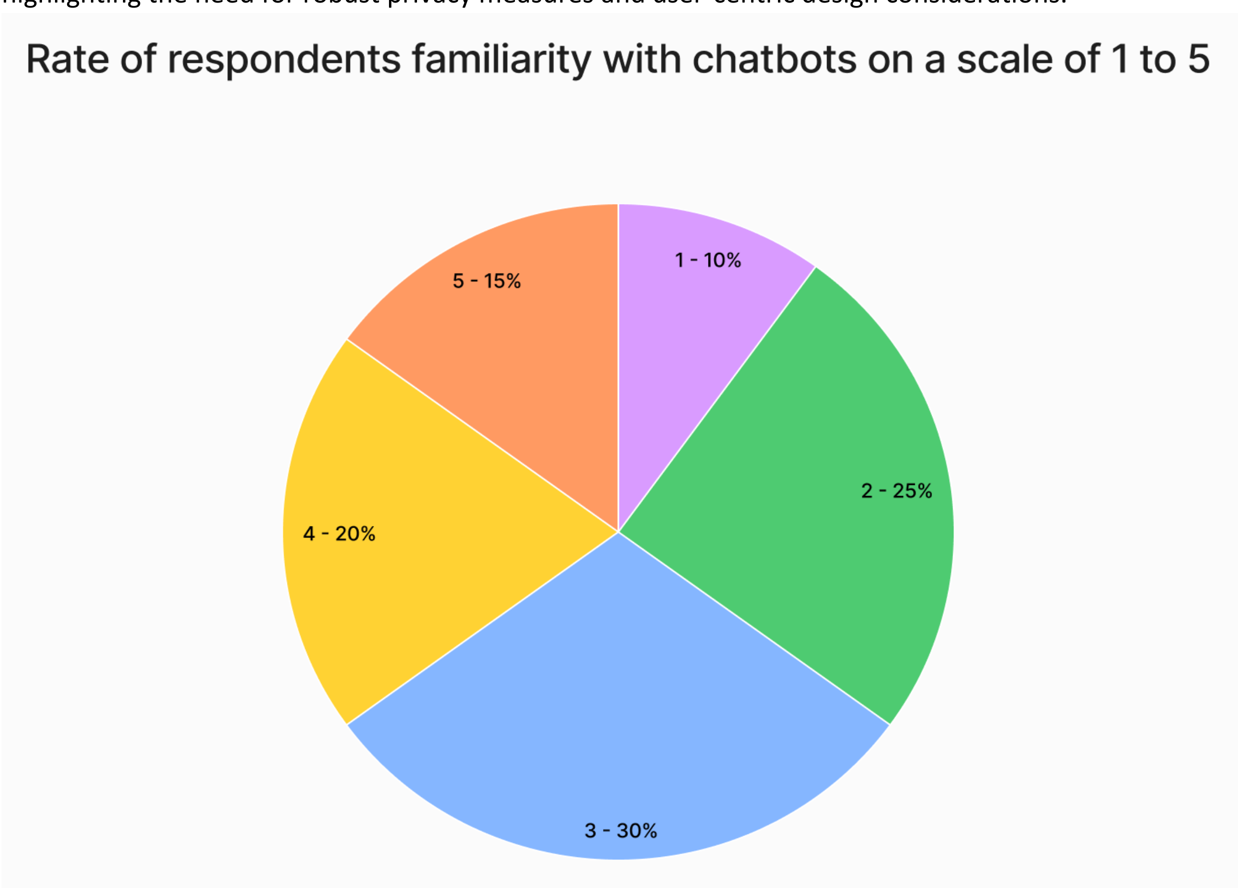

Figure 7. Chatbots Familiarity

3.5 Data Classification and Analysis

Data classification and analysis efforts involved the training of classifier, incorporating features derived from both an initial training corpus and a supplementary dataset containing frequently encountered mental health-related language. To establish a robust foundation for this chatbot, a specialized vocabulary of the training data sets and relevant corpus, forms the core of the classification model. Each text message, whether included in the testing phase or not, contributed to fine-tuning the classifier. In this particular instance, the ELECTRA model served as a tool for handling the nuances of mental health conversations, detecting suicidal intents. The ultimate objective of this process was to employ the trained ELECTRA model to classify and analyze text data with suicidal content, enabling the chatbot to provide timely and empathetic responses to individuals in crisis.

3.6 Collecting and Labelling Data

Text containing suicidal intent from the Kaggle dataset which itself was scrapped from reddit thread containing suicidal intent was the primary source of data for this study's first phase of data collection. Data on Kenyan suicidal text was scarce and loosely documented, making it challenging to readily assemble a comprehensive dataset suitable for model training purposes. The collected dataset had 2 columns, "text" containing content of user posts while the "class" basically was the target variable.

	text	class
0	Ex Wife Threatening SuicideRecently I left my ...	suicide
1	Am I weird I don't get affected by compliments...	non-suicide
2	Finally 2020 is almost over... So I can never ...	non-suicide
3	i need helpjust help me im crying so hard	suicide
4	I'm so lostHello, my name is Adam (16) and I'v...	suicide

Figure 8. Suicide Detection Raw Dataset

3.7 Preprocessing Data

Preprocessing aimed to obtain clean data to improve the detection process's accuracy. The text data requires preprocessing to prepare the data into suitable formats for the subsequent model building. Social media data tends to be more unstructured and require more customized preprocessing and cleaning processes. The data was cleaned with the following steps:

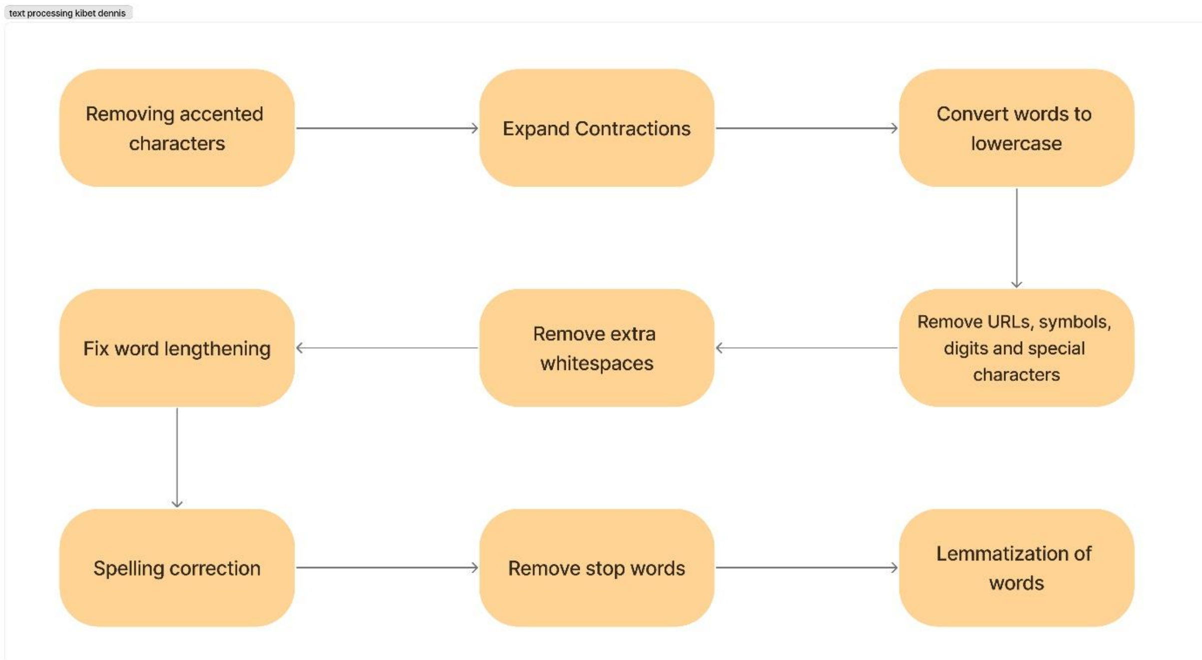

Figure 9. Data Processing Steps

3.8 Process Model (Agile Methodology)

This study project employed the Agile method due to its iterative and flexibility when demands or requirements change. Agile method's responsiveness to evolving requirements, aligns closely with the dynamic nature of the research objectives. The primary objective of this model was focused to develop an application without comprehensively knowing all specifications and requirements (Sommerville, 2009). The Agile software development life cycle contains six phases: concept, inception, iteration, release, maintenance, and retirement.

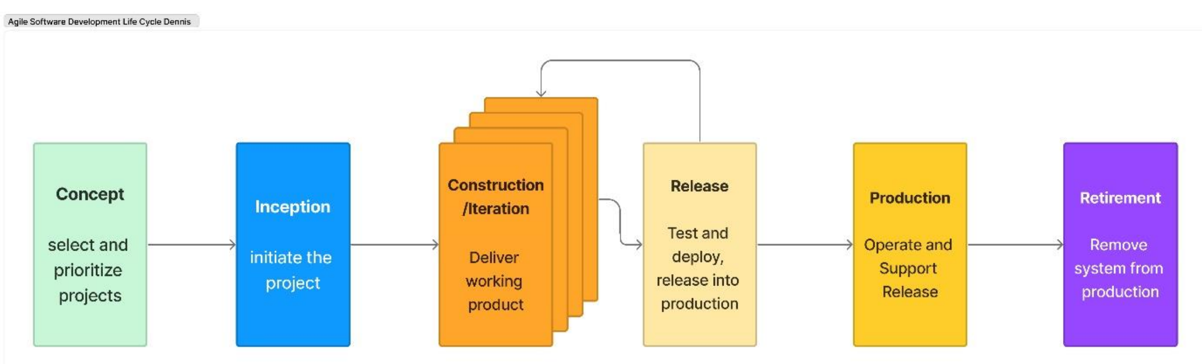

Figure 10. Agile Software Development Life Cycle

3.8.1 Concept

There was a critical challenge in detecting depression and identifying suicidal tendencies among the youth in Kenya. To address this issue, a solution was developed involving the creation of a mobile chatbot application specifically designed to cater to the mental health needs of young individuals who had access to mobile devices with internet connectivity. The primary focus of this solution was the early identification of suicidal thoughts, accompanied by essential support services. The mobile application not only detected early signs of distress but also provided crisis intervention contacts and platforms for emotional support. The overarching goal of this initiative was to make a significant contribution to suicide prevention among Kenya's youth, fostering a community of more resilient and mentally healthy young individuals.

3.8.2 Inception

In this stage, the technical and functional requirements for the mental health chatbot project were outlined. Specifically, the implementation details of the functionalities outlined in the functional specifications were detailed. Subsequently, a feasibility study was conducted to assess the project's viability, considering current technical capabilities, economic feasibility, and legal considerations. This study also involved identifying any gaps in the project and proposing adaptable solutions by exploring a range of tools and technologies.

3.8.3 Construction / Iteration

During the construction stage, the core functionality of the chatbot was built, ensuring that it could provide meaningful support and guidance to users in distress. Iteration, on the other hand, involved continuous refinement and enhancement of the chatbot based on feedback from users, mental health professionals, and ongoing research. It allowed adaptation to changing circumstances and user requirements, ultimately delivering a more robust and empathetic tool for addressing mental health challenges. The iterative nature ensured that the chatbot remained dynamic, adaptive, and capable of making a positive impact on those who needed it most.

3.8.4 Release

There are several important aspects toward the release

3.8.4.1 Testing and Quality Control

At this stage, the objective was to assess the system's compliance with all specified requirements and to determine whether it met quality standards. Independent testers who were not involved in the creation of the tool were chosen to assure genuine system testing. A setting closely resembling production was used for testing.

3.8.4.2 Prototyping and review

The chatbot was assessed to identify potential flaws and improvements. At this stage, the specifications remained unchanged, and no new ones were added.

3.8.4.3 Finalize system and user documentation

At this stage, the completion and incorporation of the documentation created during the Construction/Iteration stage will be undertaken. To provide firsthand information to system administrators and general users of the tool, documentation of the utility was produced during this phase.

3.8.4.4 Training

To receive training, the users were enrolled. Only three users were trained at a time based on the tool's intended use.

3.8.5 Deploy the system

At this stage the chatbot system was put to use. It transitioned from development to practical use, where the system became operational and available for the intended users.

3.9 Time Schedule

Below is a structured roadmap showing how events and tasks unfolded over time. It offers clarity on how specific activities were done and helped in overall project planning. This helped me estimate timelines, deadlines and a sense of accountability.

Project Activities	Duration in Weeks
Planning and Analysis	3
Designing	4
Development	9
Testing	2
TOTAL	18

Table 1. Estimated Time

3.10 Gantt Chart

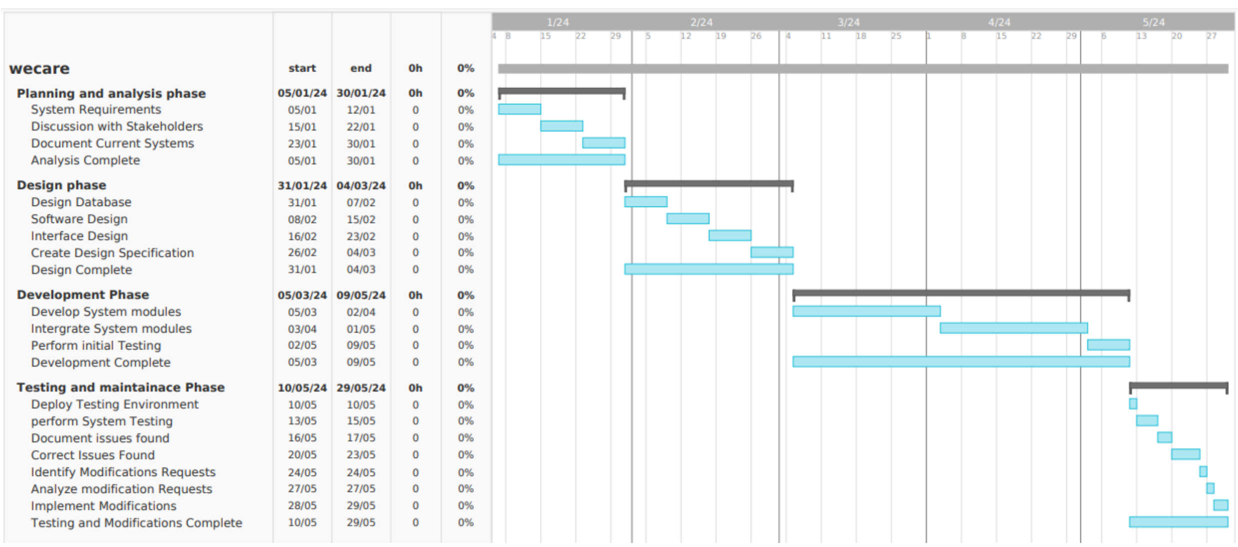

Figure 11. Gantt Chart

3.11 Budget

The project was anticipated to cost an approximate of 85,000 Kenyan Shillings. Table below shows the broken down expenses in detail

Expenses	Cost in Shillings
Laptop	50,000
Internet Subscription	5,000
Travel Expenses	10,000
Food	3,000
Domain name	2,000
Hosting	15,000

Table 2. Estimated Budget

Chapter Four: System Analysis and Design

4.1 Introduction
This chapter discusses methods of natural language processing (NLP) using NTLK that were applied in development of the tool to perform analysis of suicidal texts on the Chatbot. This section provides a comprehensive logical and process architectural overview of the developed tool and the components that make up the core functions of the tool.

4.2. Data Analysis Results
The discussion of what constituted suicidal intent was the main objective of this phase. I examined various definitions from dictionaries, legal documents, and user policy agreements on digital platforms that address suicidal content. To conduct content analysis, I observed the similarities and differences in the criteria for identifying suicidal text. An intriguing discovery was that divergent perspectives of individuals expressing suicidal thoughts and the concerns of those trying to detect and prevent such content seemed to influence all categories, with legal constraints playing a pivotal role.

4.2.2 Findings
Perspectives on suicidal intent describes a consistent pattern. It encompassed both spoken and nonverbal expressions that hinted at, discussed, or conveyed suicidal thoughts. Defining this complex aspect involves understanding the many ways it may manifest. Suicidal intent can be expressed both explicitly and subtly, often evident through verbal or written statements, such as direct expressions of a desire to end one's life. Legal constraints play a pivotal role, highlighting the delicate balance between safeguarding individuals and respecting privacy. These findings emphasize the challenge of establishing universal guidelines and stress the need for comprehensive strategies in addressing suicidal content in the digital domain.

4.3 Requirements Analysis
The goal of this study was to build a chatbot and a model for tracking suicidal intent and providing support. Based on this objective, this section outlines the various requirements to be provided for by the proposed solution.

4.3.1 Functional Requirements
1. The application should allow the user to create an account and login to his account
2. The application should allow the user to input chat messages and send to the bot.
3. The application model should perform text sentiment analysis on users's messages.
4. The application should generate responses to user's chats
5. The application should provide supportive messages and helpline messages on suicidal texts

4.3.2 Non-Functional Requirements
Usability

The suicide detection chatbot prioritizes usability with an intuitively designed interface, promoting ease for users in distress. Emphasis on accessibility ensures a compassionate interaction, fostering a supportive environment for those seeking help.

Scalability

Addressing future growth, the chatbot will be engineered for scalability, capable of seamlessly handling increased user loads. This design ensures sustained performance and responsiveness, even as the user base expands, maintaining reliable suicide detection capabilities.

Persistent Storage

Robust persistent storage mechanisms safeguard sensitive user data and chat histories, enhancing the chatbot's ability to recognize patterns. This secure repository enables comprehensive analysis and effective risk assessment, contributing to the chatbot's overall efficacy in suicide detection.

4.4 System Architecture

The system architecture shows the general layout of WeCare Suicide Detection system and the components it's made up of. After logging in to the system the user interacts with the system by creating a bot, then he/she can start sending chat messages to the bot. The bot will be analyzing each received message by the suicide detection model. In the case of a suicidal text is detected the suicidal model will generate comforting supportive message and will provide help information. In the case of a normal message the suicide model will flag it as a normal message and DialogGPT-large will generate a normal message to continue the conversation. The user's message are then saved to a MongoDB database and the response is sent to the user.

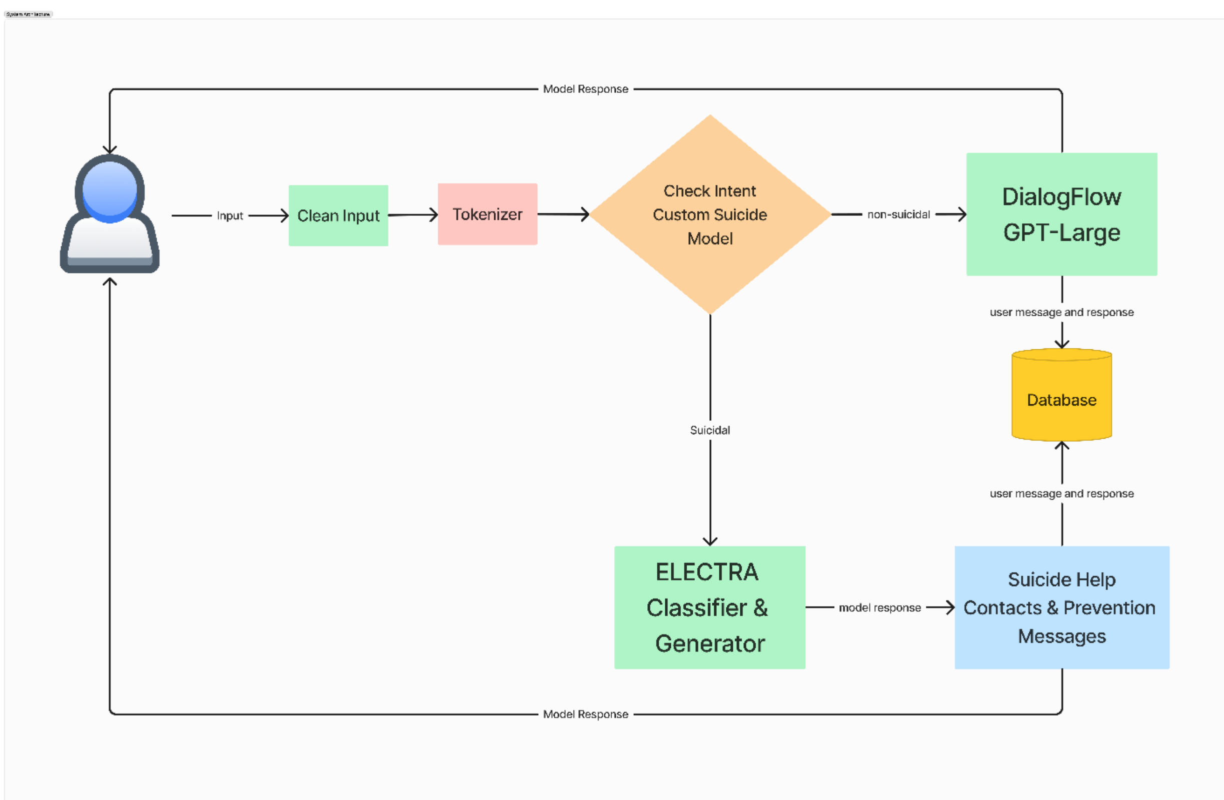

Figure 12. System Architecture

4.5 Use Case

Use case diagrams are used to illustrate interaction between actors and the system. These interactions are between the various actors and the chatbot system prototype. The diagram also depicts the functionality that the proposed system should have.

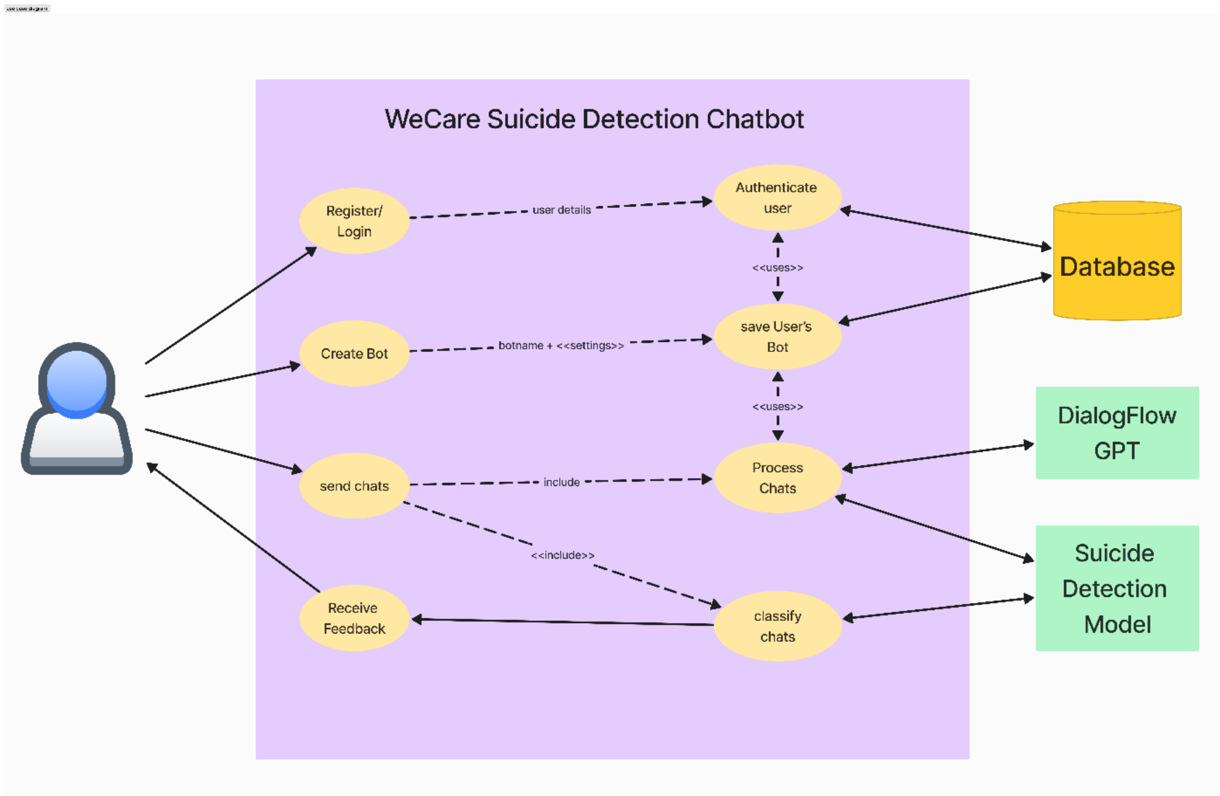

Figure 13. Use case Diagram

4.5.1 Detailed use case description

This section provides comprehensive descriptions of the use case in the Figure above.

Use case: Register/Login, Create Bot, Send Chats, Provide User with Feedback.

Primary Actors

User

Suicide Detection Model

DialogFlow GPT Model

Preconditions

Models have to be loaded in memory

User has access to the chatbot system

Postconditions

Analyze the user message and classify it as suicidal or non-suicidal
Generate an appropriate response to the user

Save user's message and the response to database

Main Success Scenarios

Action intention	**System Responsibility**
User sends message which initiates classification	Preprocess the user message to clean and remove irrelevant symbols and texts
	User message is tokenized and the model transformation is applied
	Suicide Model classifies the message as suicidal or non-suicidal

4.6 Sequence Diagram

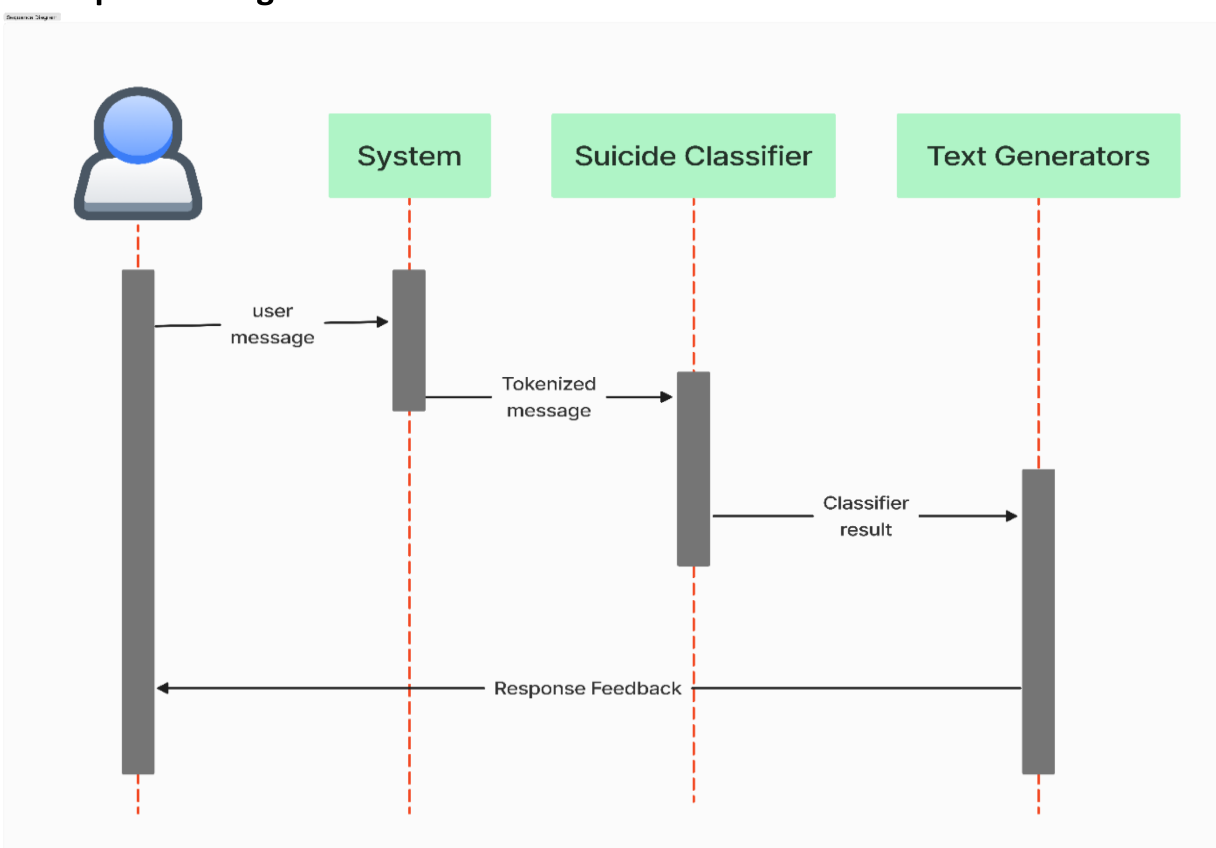

Figure 14. Sequence Diagram

4.7 Data Flow Diagram

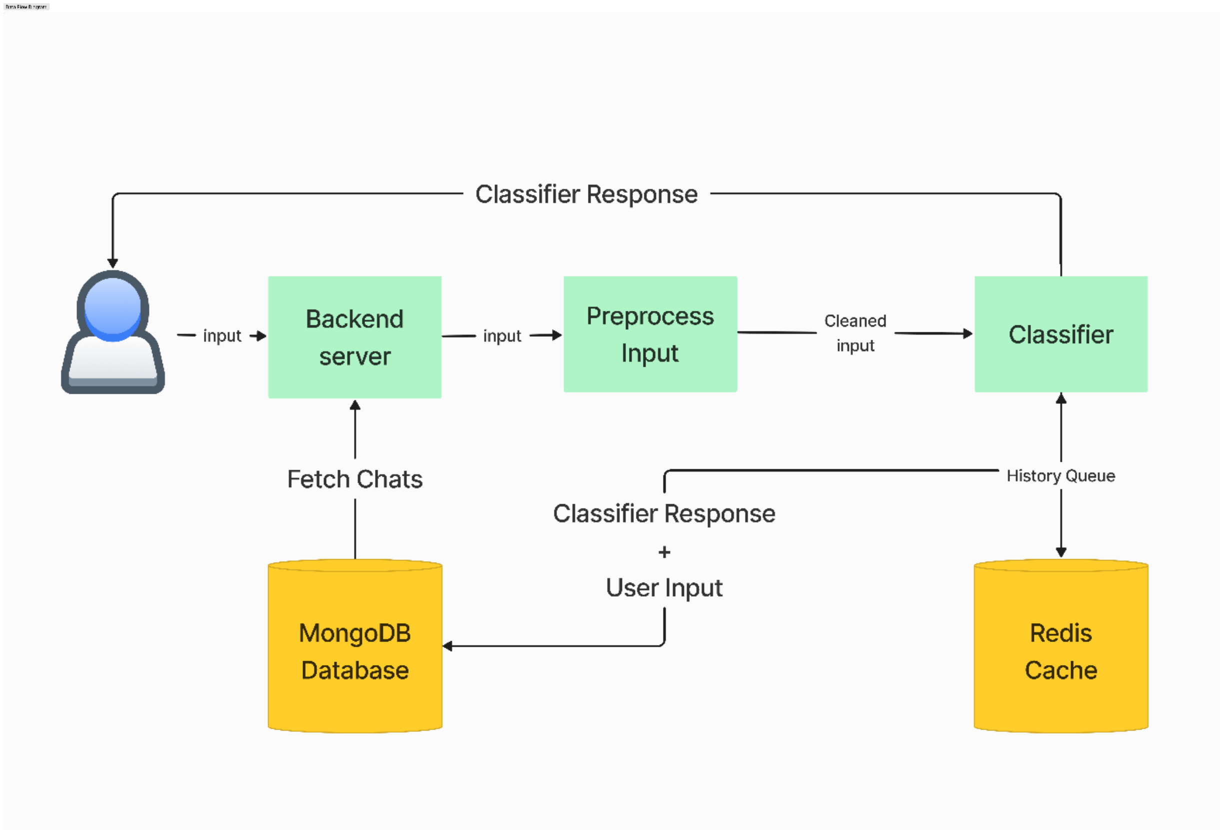

Figure 15. Data Flow Diagram

4.8 Physical Design

The application database is implemented in MongoDb structure with mongoose establishing the connection between the database and the Python Environment runtime. The database schema is designed with mongoose and the relevant relationships enforced. The primary collections in the database are users, bots, botchats. The user collection stores user-related information, while the bot collection contains details about the chatbots associated with users. The botchats collection stores information about individual chats, such as names, last messages, and timestamps. This physical design leverages MongoDB's document-oriented structure, allowing for efficient storage and retrieval of chat-related information while maintaining relationships between entities through [table_name]_id references.

4.9 Entity Relation Diagram

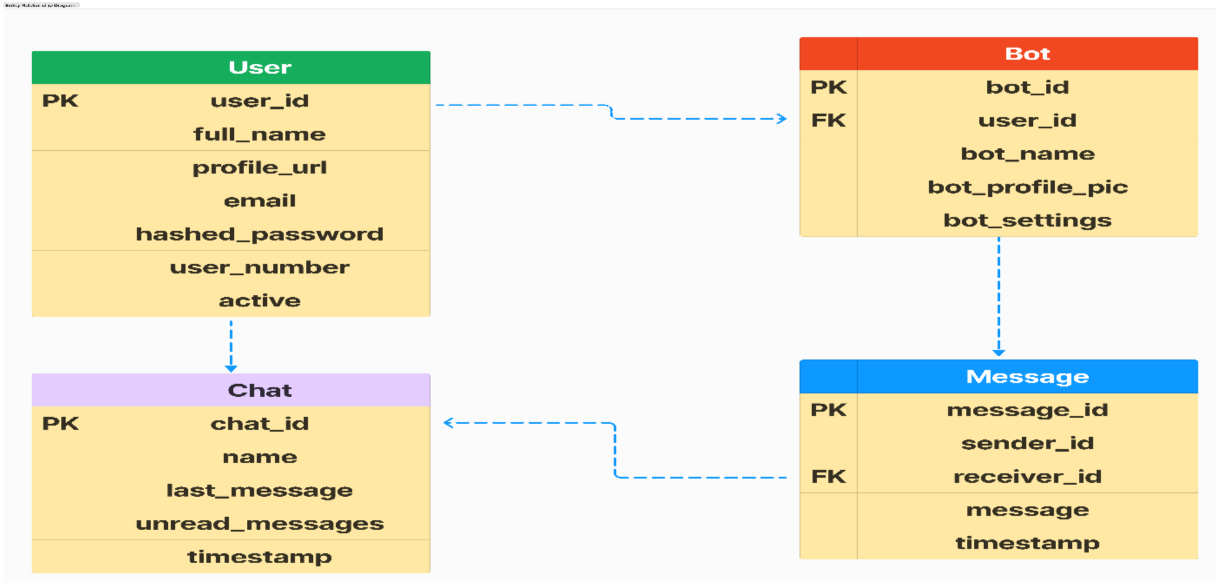

Figure 16. Entity Relationship Diagrams

4.10 UML Class Diagram

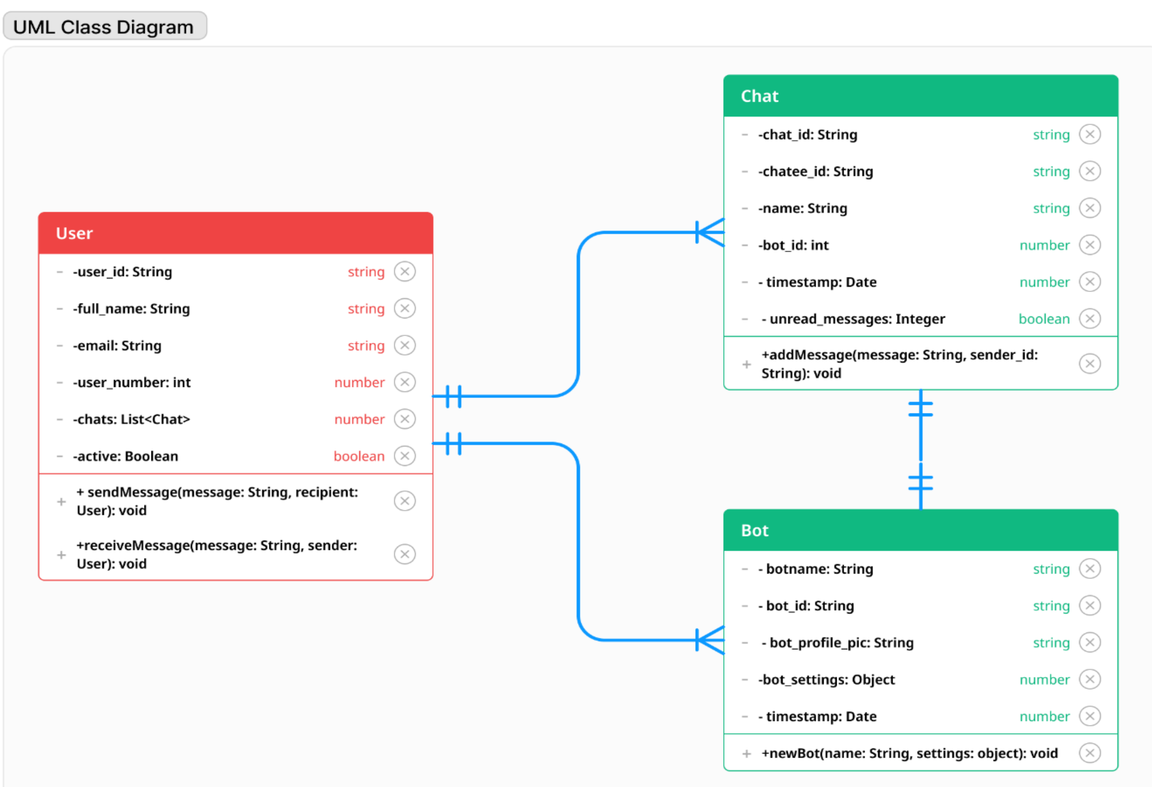

Figure 17. UML class diagram

Chapter Five: System Implementation and Testing

5.1 Introduction
This chapter describes how the system was implemented, tested and validated.

5.2 Development Tools
Front-End

React with Typescript – The web interface is built using React, a JavaScript library for building user interfaces. Typescript is used as a superset of JavaScript to enhance type safety during development. This ensures a more robust and maintainable codebase. Styling is achieved using SCSS (Sass), a popular CSS preprocessor. SCSS allows for the use of variables, nested rules, and other features, providing a more maintainable and organized stylesheet.

Backend

Python Flask Rest API – The server-side logic and API endpoints are implemented using Python with Flask. Flask is a lightweight web framework that simplifies the process of building RESTful APIs and web applications. The backend serves as a RESTful API, allowing communication between the front-end and the database. RESTful principles ensure a scalable and stateless architecture. Flask handles routing and defines endpoints for various functionalities, enabling communication with the React front-end.

Database

MongoDB - MongoDB is chosen as the database to store and manage data for the application. MongoDB is a NoSQL database that stores data in a flexible, JSON-like format. The Flask backend communicates with MongoDB via mongoose adapters to perform CRUD (Create, Read, Update, Delete) operations, ensuring seamless data flow between the front-end and the database.

5.3 Corpus Construction
The dataset for this project was sourced from Kaggle dataset which consists of post from the social media platform reddit. The dataset consists of 2 columns where “text” indicates the post content and “class” indicates the label of the posts. A total of 232,074 posts were scraped from 2 subreddits, namely SuicideWatch and teenagers to build this dataset. Posts from SuicideWatch were labelled as suicidal, while the posts collected from teenagers were labelled as non-suicidal. The data is classified as a binary classification. Therefore the final model should predict a text input as either suicidal or non-suicidal.

5.3.1 Cleaning and Processing
The initial dataset collected from Kaggle was in an unstructured format had empty columns and rows which is unsuitable for the training process. Due to this it was necessary to clean and preprocess the dataset to ensure higher accuracy of the model. Chat Messages can contain different types of information such as names, dates, Numbers, text, URLs, emojis and many more. This research was only intested with the text part which would be used to build the corpora and the rest are removed. The data cleaning and preprocessing involved spell checking using Symspell library, removing extra white spaces

from text, removing accented characters from text, removing urls, removing symbols and digits, removing special characters and fixing word lengthening (characters are wrongly repeated).

```
nlp = spacy.load("en_core_web_sm")
sym_spell = SymSpell(max_dictionary_edit_distance=2, prefix_length=7)
dictionary_path = pkg_resources.resource_filename("symspellpy", "frequency_dictionary_en_82_765.txt")
bigram_path = pkg_resources.resource_filename("symspellpy", "frequency_bigramdictionary_en_243_342.txt")
sym_spell.load_dictionary(dictionary_path, term_index=0, count_index=1)
sym_spell.load_bigram_dictionary(bigram_path, term_index=0, count_index=2)

deselect_stop_words = ['no', 'not']
for w in deselect_stop_words:
    nlp.vocab[w].is_stop = False

def text_preprocessing(text, accented_chars=True, contractions=True, convert_num=True,
                       extra_whitespace=True, lemmatization=True, lowercase=True,
                       url=True, symbols_digits=True, special_chars=True,
                       stop_words=True, lengthening=True, spelling=True):
    if accented_chars:
        text = unidecode.unidecode(text)
    if contractions:
        text = contract.fix(text)
    if lowercase:
        text = text.lower()
    if url:
        text = re.sub(r'http\S+', '', text)
    if symbols_digits:
        text = re.sub('[^a-zA-Z\s]', ' ', text)
    if special_chars:
        text = text.replace("\r", " ").replace("\n", " ").replace("    ", " ").replace('"', '')
    if extra_whitespace:
        text = " ".join(text.strip().split())
    if lengthening:
        text = re.sub(r"(.)\1{2,}", r"\1\1", text)
    if spelling:
        text = sym_spell.lookup_compound(text, max_edit_distance=2)[0].term

    doc = nlp(text)
    clean_text = [token.lemma_ if lemmatization and token.lemma_ != "-PRON-" else token.text
                  for token in doc if not (stop_words and token.is_stop and token.pos_ != 'NUM')
                                      and not (convert_num and token.pos_ == 'NUM')]

    return " ".join(clean_text)
```

Figure 17. data cleaning and preprocessing

5.4 Exploratory Data Analysis

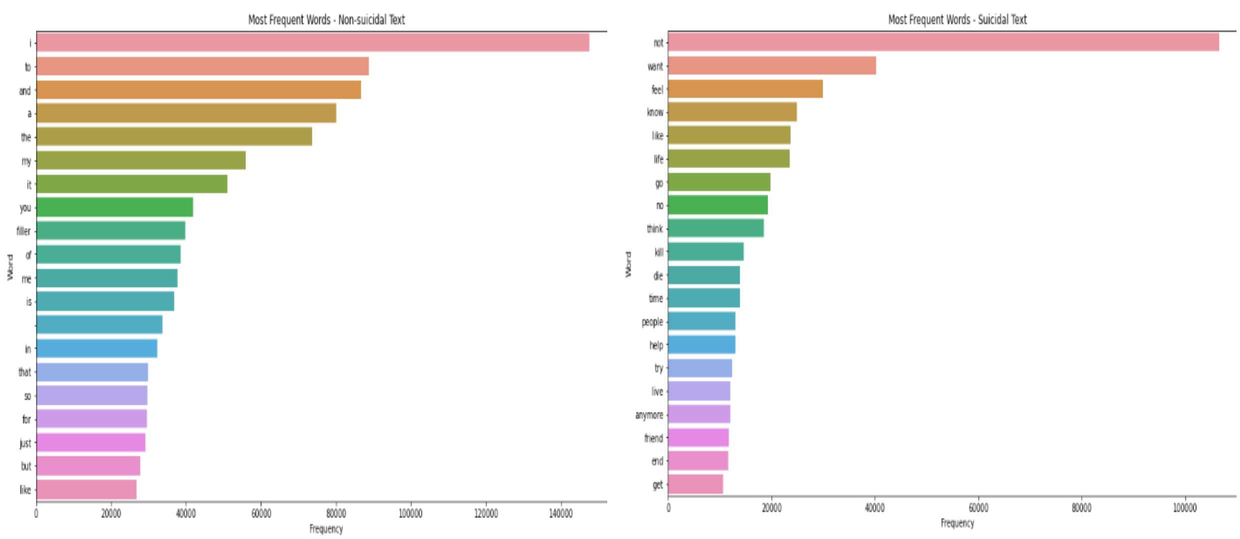

Figure 18. Most Frequent words in the Corpus

Polarity Distribution Score

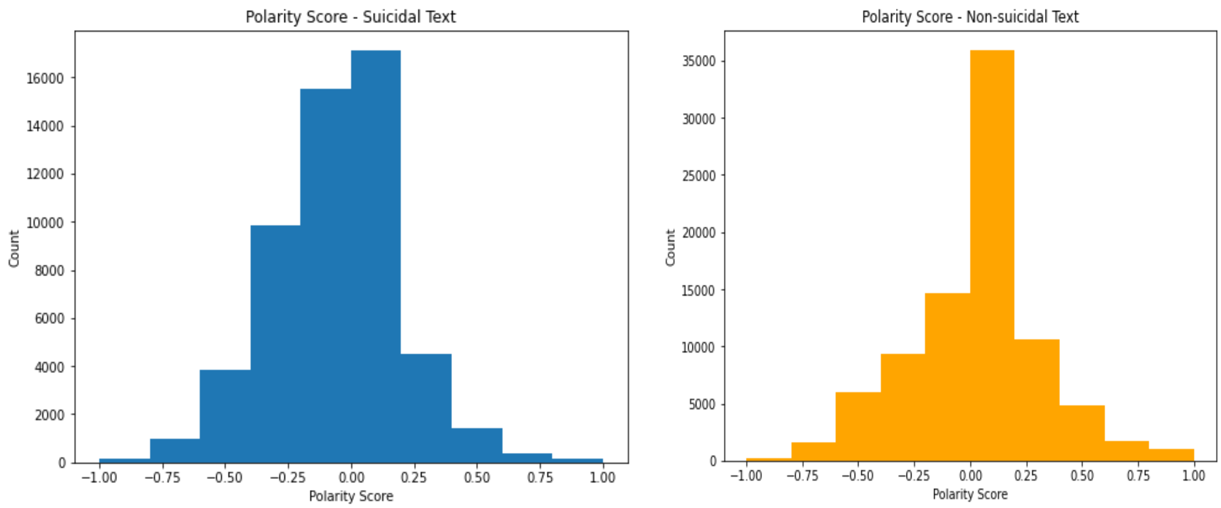

Figure 19. Polarity Distribution

I explored the polarity score of texts which ranges from -1 to 1, where -1 represents negative, 0 stands for neutral, and +1 represents positive sentiment. The distributions of polarity scores obtained are seen in the histograms in the Figure above.

Suicidal texts have more data points on the left side of the distribution, representing more negative sentiment posts. On the other hand, most non-suicidal texts are classified as neutral. This observation is consistent with our initial intuition that suicidal text has a more negative sentiment.

Bigrams

Using pointwise mutual information, I created bigrams to find related words since individual word frequencies was informative but in practice some of the tokens are part of longer phrases that should be treated as a single unit.

Bigram	Count
('feel', 'like')	3291
('year', 'old')	1299
('want', 'talk')	1248
('sub', 'edit')	1198
('min', 'craft')	1144
('need', 'help')	1121
('good', 'friend')	939
('look', 'like')	865
('want', 'know')	844

Bigram	Count
('feel', 'like')	10405
('want', 'die')	5398
('want', 'kill')	2365
('want', 'end')	1872
('commit', 'suicide')	1653
('want', 'live')	1636
('suicidal', 'thought')	1560
('end', 'life')	1392
('need', 'help')	1381

Figure 20. bigrams

5.4.4 Model Building

Once the dataset was preprocessed and cleaned, training of the model could proceed. As I had stated in previous chapters I built an ELECTRA based model. To build the model I first loaded the preproccesed and cleaned dataset into a pandas DataFrame. I then split the corpus into training, validation and test using train_test_split() method from the scikit-learn library in the ratio of 40:40:20. I then converted the dataframe to a dataset and tokenized the corpus using google/electra-base-discriminator. I then created the model by importing the base ELECTRA-base pretrained model with the parameter num_labels=2. I used the wandb platform to save my checkpoints as the trainig process is compute intesive and time consuming. This allowed me to pause the training and fallback incase of errors. I then defined the compute metrics like accuracy, recall, precision, the final average score, and loss. I also defined the training arguments for the model. I the used the trainer.train() function to train the model using the dataset and the set parameters. After the training completed I saved the new model.

```
def compute_metrics(eval_pred):
    metric_acc = load_metric("accuracy")
    metric_rec = load_metric("recall")
    metric_pre = load_metric("precision")
    metric_f1 = load_metric("f1")
    loss_fn = torch.nn.CrossEntropyLoss()

    logits, labels = eval_pred
    predictions = np.argmax(logits, axis=-1)

    # Compute loss
    logits_tensor = torch.from_numpy(logits)
    labels_tensor = torch.from_numpy(labels)
    loss = loss_fn(logits_tensor, labels_tensor).item()

    # Compute metrics
    accuracy = metric_acc.compute(predictions=predictions, references=labels)["accuracy"]
    recall = metric_rec.compute(predictions=predictions, references=labels)["recall"]
    precision = metric_pre.compute(predictions=predictions, references=labels)["precision"]
    f1 = metric_f1.compute(predictions=predictions, references=labels)["f1"]

    return {"accuracy": accuracy, "recall": recall, "precision": precision, "f1": f1, "loss": loss}
```

Figure 21. model metrics

5.4.5 Testing and Evaluating the Model

The 20 percent test set was used to test the model.

5.5 Chatbot Integration

After building and saving the model, the next steps involved intergrating it to the main chatbot application. The model will be hosted in the backend of the chatbot application, where the frontend makes requests and the backend will be passing the required parameters to the model and handling responses. For faster response times the model needs to be loaded in memory initially before the conversation starts.

```python
def load_suicide_tokenizer_and_model(tokenizer="google/electra-base-discriminator", model="Models/electra"):
    global suicide_tokenizer, suicide_model
    suicide_tokenizer = AutoTokenizer.from_pretrained(tokenizer)
    suicide_tokenizer.padding_side = "left"
    suicide_model = AutoModelForSequenceClassification.from_pretrained(model)
    return suicide_tokenizer, suicide_model
```

Figure 22. loading suicide model to memory

```python
def check_intent(text):
    global suicide_tokenizer, suicide_model
    socketio.emit('typing', {'response': True})
    if suicide_tokenizer is None or suicide_model is None:
            socketio.emit('info', {'response': Markdown('Loading Suicide model...').data})
            suicide_tokenizer, suicide_model = load_suicide_tokenizer_and_model()
    tokenised_text = suicide_tokenizer.encode_plus(text, return_tensors="pt")
    logits = suicide_model(**tokenised_text)[0]
    prediction = round(torch.softmax(logits, dim=1).tolist()[0][1])
    return prediction
```

Figure 23. Function to check user message suicidal intent

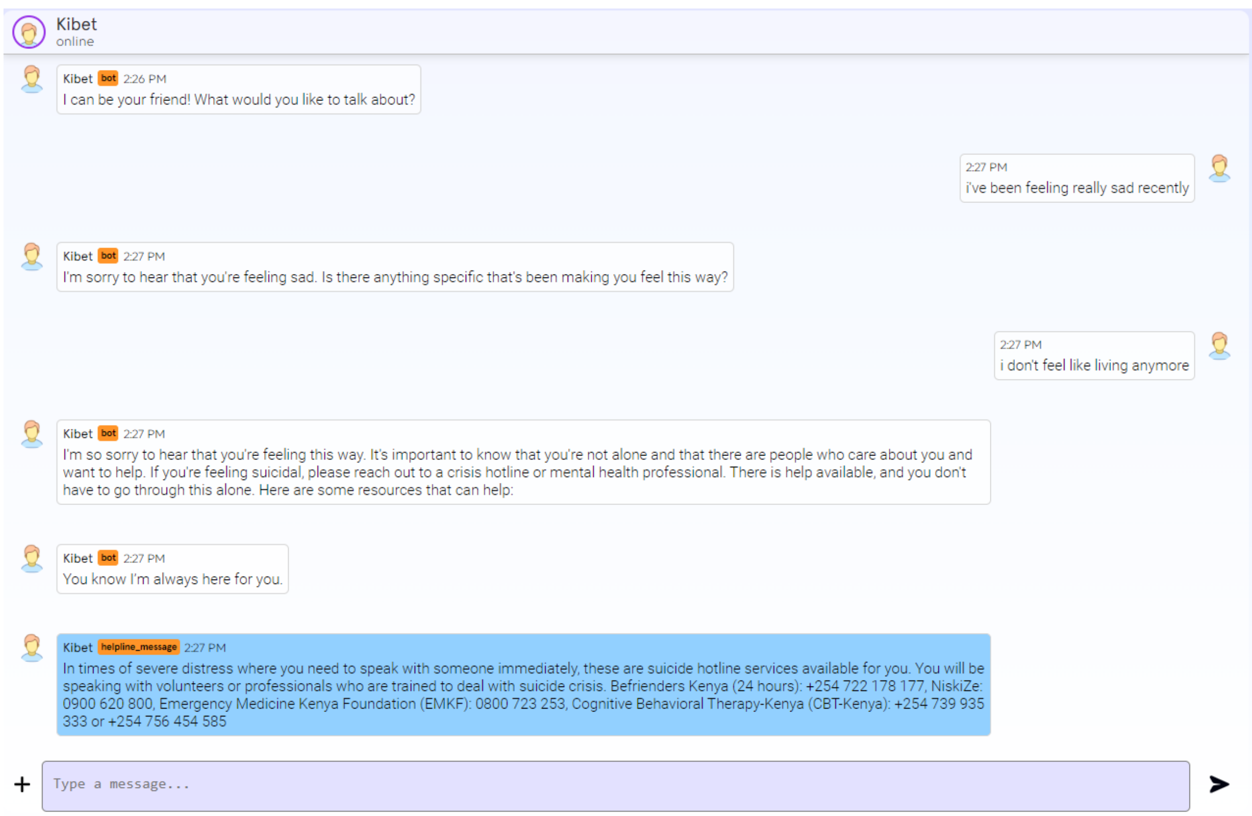

Figure 24. Chatbot conversation

Chapter Six: Discussion, Recommendations and Conclusion

6.1 Introduction

This chapter discusses results achieved in the previous chapters. It clearly highlights the results obtained in this study and how the objectives were met.

6.2 Discussion

The achieved results greatly met the objectives set out, as it led to the development of a suicidal detection chatbot, that is capable of analyzing user messages and providing support and even contact information. The effectiveness of the chatbot in detecting and potentially preventing suicide is notable. By using Artificial Intelligence, machine learning algorithms and natural language processing techniques, the chatbot demonstrates a high level of accuracy in identifying signs of depression and suicidal ideation. Its proactive approach in engaging users in meaningful conversations, coupled with timely interventions and resource recommendations, offers a valuable support system for individuals struggling with mental health issues. The successful fulfillment of the project objectives shows the importance of utilizing technology-driven solutions in addressing mental health challenges. The developed chatbot not only serves as a tool for early detection and intervention but also contributes to destigmatizing conversations around mental health, fostering a supportive and inclusive environment for those in need.

The model developed are well suited for this chatbot as it was trained from a labelled dataset of suicidal and non suicidal conversations, which makes the model very efficient in flagging suicidal messages from non suicidal ones. The model is based on the ELECTRA model architecture which allows it to infer new messages with old ones for context, this allows the model to be always on point with the generated responses.

6.3 Recommendation

Although the model's accuracy is outstanding, it is not 100 percent accurate and can sometimes detect false positives. This is mainly due to the limitations in the size of the dataset and training time. The model can be trained with more data and can be even tailored to support more languages by also training it with more diverse datasets to cover a wide range of languages. The model can be hosted on a more powerful computer with higher processing capability to allow fast responses from the model. The whole system can also be designed in a distributed way to allow scaling in the case the system receives higher load requests, this can involve techniques like distributing the backend and using a load balancer to efficiently balance the load.

6.4 Future Work

Increasing the number of tokens that the model can handle to effectively generate accurate responses. The system can also be equiped with the capability to parse emojis as sometimes they add context and emotion in user messages. The chatbot currently only supports English conversations and the incorporation of more languages can expand our reach to a wider international community. This could be achieved by training and fine-tuning our models on conversational data in different languages, which can be compiled by scraping various social media platforms such as twitter, reddit, facebook or threads.

6.5 Conclusion

I made two assumptions in this study. The application is web based, assuming that majority of the youth population in Kenya own mobile smart phones. Secondly, that ordinary users who have access to smartphones have access to internet connectivity and are actually able to use the web interface. However, the system is aimed to be easy and reliable to its users.

This study made the following conclusions based on the research objectives: the current methods of information collection are traditional and manual and are affected by human bias, which effectively hinders the victims from opening up. Developing a chatbot that can detect suicidal intentions and provide support is a promising approach to reducing suicide rates among young people in Kenya. The suggested system is appropriate in Kenya as the majority of the Kenyan youth own and use smartphones, with chat enabled features.

References

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin (2017). Attention is All You Need. NIPS. https://arxiv.org/pdf/1706.03762.pdf

Bukusi, D. E. (2015). The Mental Health Policy for Kenya. https://www.scribbr.com/citing-sources/cite-a-book/

Carol Chu, M.S., Jennifer M. Buchman-Schmitt (2017). The Interpersonal Theory of Suicide. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5730496/

Kenaco (2019). Mental Health Status in Kenya. https://kanco.org/mental-health-status-in-kenya/

Komati, N. (2021). A dataset that can be used to detect suicide and depression in a text. Kaggle. https://www.kaggle.com/datasets/nikhileswarkomati/suicide-watch

Marangu, E. M. (2018). Culture and Mental Health in Kenya. https://bing.com/search?q=APA+format+of+referencing+for+books

Mathai, M., Mbwayo, A. W., Mutavi, T., & Bukusi, D. (2019). Child and Adolescent Mental Health in Kenya: Do We Need a Child and Adolescent Mental Health Policy? In Child Behavioral Health in Sub-Saharan Africa. https://www.verywellmind.com/book-references-in-apa-format-2794850

Nation Africa, Wanja Gathu (2017). This is why teens in Kenya are committing suicide. https://nation.africa/kenya/life-and-style/dn2/this-is-why-teens-in-kenya-are-committing-suicide-448948

Raj, S. (2019). Building Chatbots with Python. http://noandishandp.ir/wp-content/uploads/2022/02/Building-Chatbots-with-Python-Using-Natural-Language-Processing-and-Machine-Learning-by-Sumit-Raj-preview.pdf

Standard Media, Brigid Chemweno (2015). Suicide: What is ailing Kenya's young people? https://www.standardmedia.co.ke/lifestyle/article/2000175681/suicide-what-is-ailing-kenya-s-young-people

Swan, T. (2018). The Anatomy of Loneliness: How to Find Your Way Back to Connection. https://www.easybib.com/guides/citation-guides/apa-format/how-to-cite-a-book-apa/

Zhang, Y., Sun, S., Galley, M., Chen, Y.-C., Brockett, C., Gao, X., Gao, J., Liu, J. & Dolan, B. (2019). DialoGPT: Large-Scale Generative Pre-training for Conversational Response Generation. https://arxiv.org/abs/1911.00536.

Appendix

Appendix A: Interview Guide

MACHAKOS UNIVERSITY

ENGINEERING AND TECHNOLOGY

BSC COMPUTER SCIENCE

Research Questionnaire

I am a graduate student at Machakos University, School of Engineering and technology. I am conducting research for my final year project in Computer Science. This research aims at developing a Suicide Detection Model that will be implemented as a chatbot. I am therefore kindly requesting you to fill this questionnaire. This Survey is strictly for academic purposes and will **NOT** be shared. Responding to this questionnaire is voluntary and the responses will be kept strictly confidential. Kindly don't write your name or any features that can be used to identify you for anonymity purposes.

Your co-operation will be highly appreciated.

Yours Faithfully:

Dennis Kibet

Date………………………………………… Questionnaire No……………………………………

Questions

1. Have you ever interacted with any system that is designed to detect and/or aid in suicidal detection?
   ……
   ……
2. On a scale of 1 to 5, how familiar are you with chatbots?
   ……
   ……
3. What platforms do you commonly use for communication? (e.g., messaging apps, social media, email)
   ……
   ……
4. How often do you engage in conversations with chatbots or virtual assistants?
   ……
   ……

5. What features do you believe are essential for an effective suicide detection model implemented as a chatbot?
……
……
……

6. How comfortable would you feel discussing sensitive topics like mental health with a chatbot?
……
……
……

7. Have you or someone you know ever experienced suicidal thoughts or behaviors?
……
……

8. What factors do you think contribute to the effectiveness of suicide prevention measures?
……
……
……

9. In your opinion, what role can technology play in addressing mental health challenges such as suicide prevention?
……
……

10. Would you prefer a suicide detection chatbot to be proactive in reaching out to users, or would you prefer users to initiate conversations when needed?
……
……

11. How important do you think it is for a suicide detection chatbot to provide resources and support services to users in crisis?
……
……

12. How would you rate the importance of privacy and data security when using a suicide detection chatbot?
……
……

13. What concerns, if any, do you have regarding the use of artificial intelligence in mental health applications such as suicide detection?
……
……
……

14. How do you think cultural differences may impact the effectiveness of a suicide detection chatbot?
……
……

……

15. Have you ever sought help or information regarding mental health or suicide prevention online?
……
……

16. How would you prefer to be notified if a chatbot detects potential suicidal behavior in your conversations?
……
……

17. Do you believe that integrating machine learning algorithms into suicide detection chatbots can improve their accuracy over time?
……
……
……