VICTORIA UNIVERSITY
SYDNEY AUSTRALIA

NIT6150 Advanced Project

Project Evaluation Report

HealthCare Chatbot System

Team Leader:	Jitendra Shrestha (s8104215)
Team Member:	Pranish Acharya (s8100698)

Client:	Holroyd Private Hospital
Supervisor:	Fakhra Jabeen VU Sydney

Table of Contents

Version Control

Documentation and Source Code			
Modified Date	**Modified By**	**List of Modifications Made**	**Status**
Aug 9, 2024	Pranish	Created Repository Initial Proposal report file added Added Deliverable, schedule and budget	Completed
Aug 10, 2024	Jitendra	Added Project Allocation document and modified proposal part Finalized the Project Proposal	Completed
Aug 19, 2024	Pranish	Added System Analysis and Design report file added Added Functional requirements Created ER Diagram Created Sequence Diagram Created Class Diagram	Completed
Aug 20, 2024	Jitendra	Created Meeting Minutes Report Created System Version Control Created Wireframes for chatbot UI Created Login UI Created overall Use case Created Flow Chart Diagram	Completed
Aug 21, 2024	Jitendra	Created Initial Project Folder which will be used for project development	Completed
Aug 22, 2024	Pranish	Started setting up database Created Models	Completed
Aug 24, 2024	Pranish	API Development	Completed
Aug 24,2024	Jitendra	Frontend Development Setup	Completed

		Created App healthcare chatbot system	
Aug 25, 2024	Jitendra	UI Development	Completed
Aug 29, 2024	Pranish	Completed Chatbot logic in Shell	Completed
Aug 29, 2024	Jitendra	UI for personal health management system completed	Completed
Sep1, 2024	Jitendra	Presentation and initial frontend added	Completed
Sep 4, 2024	Jitendra	Added Final Draft documentation added	Completed
Sep 4, 2024	Pranish	Eposter Added Updated chatbot logic	Completed
Sep 6, 2024	Jitendra Pranish	Chatbot Integration	Completed
Sep 7, 2024	Jitendra	Added page navigation Updated system version control document Updated User manual	Completed
Sep 8, 2024	Jitendra	Updated User manual Created Technical Documentation Created Unit testing and UI testing Updated Version Control	Completed
Sep 11, 2024	Pranish	Added Quality Assurance Report	Completed
Sep 11, 2024	Jitendra	Added Project Evaluation Report	Completed

Workload Allocation

Team Member	Contribution
Jitendra Shrestha	Database Management UI Design Frontend Development Frontend Integration with chatbot Unit Testing and UI Testing Documentation
Pranish Acharya	Chatbot Logic Implementation using NLP Data Classification Train Chatbot Dialogue Management API Development for chatbot Documentation

1. Overview

A healthcare chatbot system is an artificial intelligence (AI) technology that functions as a virtual assistant, offering prompt answers to medical questions, symptom evaluations, and tailored guidance to aid users in managing their health. It improves accessibility to healthcare services by understanding and interpreting user input using Natural Language Processing (NLP) to give pertinent information and help. This makes it an invaluable tool for anyone looking for general health advice in between appointments with medical professionals.

2. Objective

The objectives of the system are as follows:

- To provide response on health queries
- To diagnose disease based on symptoms provided based on yes/no statement
- To provide way to maintain patient health record

3. Features

Healthcare chatbot system has many functions included in it which are explained below:

3.1 Symptoms checker

Users can check their ongoing health conditions using chatbot. With the provided data system will provide the home remedies for the related disease. In Case if it is not found it will recommend to the online portals.

3.2 Book an appointment

Users can book the appointment with related specialized doctors by signing up. Also, they can check the prescription given by doctor with the follow-up date if required.

3.3 User-Friendly Interface

All users can quickly navigate through the application and grasp our website in an efficient manner. From registering, using the log-in feature, signing documents, and exporting the file. These elements are all presented in a logical manner to help our client have a seamless transaction.

3.4 Roles Management

System consists of different roles which will limit the user to perform the tasks. One is Admin user who can manage doctors and view doctors. Doctors can add prescription to the booked appointment by the user. Patients can book appointments for the doctors choosing their specialization.

3.5 User Management

All users need to login for using the features except only for the chatbot feature

3.6 Data Privacy

We have not recorded the chat history when using the chatbot.

4. Literature Review

4.1 Chatbot using AIML

Researchers Sameera A. Abdul-Kader and Dr. John Woods introduced techniques like AIML, parsing, pattern matching, SQL, and relational databases that are used in the development of chatbot systems in their research paper published on the "International Journal of Advanced Computer Science and Applications (IJACSA)". Chatbots are trained using the Natural Language Understanding Toolkit (NLTK) to interpret human language. In order to create a chatbot system that is an XML derivative, researchers have introduced the AIML approach in this work. With reference to a "stimulus-response" mechanism, the AIML language aims to modify conversational modeling tasks. (Woods, 2015)

4.2 An LSTM-based chatbot

Recurrent neural networks (RNNs) and long short-term memory (LSTM) are the two main components of the conversation modeling agent presented in a research paper presented at the Third International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things (ICETCE). The RNN algorithm can learn efficiently in circumstances where there is a small gap between the supplied information and the location where it is being used; in cases where there is a larger gap, the increased dependency prevents the system from efficiently reproducing the knowledge. So, the LSTM model was created to fully fill this void. The enhanced version of the recurrent neural network (RNN) is called LSTM. The LSTM model can eliminate unnecessary input by employing a forget gate that employs the sigmoid function. The sigmoid function makes this decision. (Vipasha Chandwani, 2020)

4.3 Chatbot with N-gram and TFID

Helping individuals with health information is the primary goal of the system suggested in the paper presented at the "8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)". Researchers have used Artificial Intelligence, Natural Language Processing, TFID, and N-gram technologies in the suggested system. This method uses a database to store data so that the chatbot can identify sentence similarities and respond to queries appropriately. Ranking and sentence similarity are computed using n-gram, TFIDF, and cosine similarity. For each, the user's question will be scored, and sentences that are more similar to the query will be found and supplied to the user (Lekha Athota, 2020).

4.4 Chatbot using K-nearest neighbor algorithm

In the paper that is titled “Proceedings of the Third International Conference on Trends in Electronics and Informatics (ICOEI 2019)” there is a suggestion to create the system that is to diagnose the diseases and suggest the treatment depending on the symptoms given by the user. For this purpose, a chatbot trained on pre-specified sets of data that the user can request is employed. In the proposed system, the K-nearest neighbor (KNN) is applied in developing a chatbot as a machine learning algorithm (Rohit Binu Mathew, 2019).

4.5 Critical Analysis to the Reviewed Literature

Studying all the research papers, we found that in any of the approaches that we choose in developing them, all the chatbots will become more personalized and relatable to humans. Researchers across the globe are using different approaches that are developing different forms of chatbot systems. Far more reliable chatbot systems are in the process of implementation through Artificial Neural Networks (ANN), Natural Language Understanding (NLU), and Natural Language Processing (NLP). There is another technique, which has not been discussed yet in this paper, and this is the Artificial Neural Network (ANN) that is used to develop the artificial intelligence chatbots. It calculates the result based of the inputs of the user through the use of weighted connections sourced from numerous data training processes. Some of the frequently employed components and strategies in these study consist of natural language processing, natural language understanding, pattern matching, Support Vector Machine (SVM), Naïve Bayes, Random Forest (Anupam Mondal, 2018), and K-nearest algorithm. The methods used to capture the datasets include tokenizing, vectorizing and cleaning. While other methods are employed to create basic small talk chatbot that will only reply to a query based on trained databases, the LSTM model can be employed to create an advanced Artificial Intelligence chatbot systems that will be able to store previous CHAT logs and analyze them in order to provide an appropriate reply to a query. Similar to this, emotion detection is accomplished through the use of three popular deep learning classifiers: Such as CNN, RNN, and HAN among which the optimal recommendation algorithm is CNN (Falguni Patel, 2019). SVM based chatbot system developed by the researcher achieved a maximum accuracy of 98 percent. 39 % while for the experimental accuracy of the proposed LSTM model it was 79. 96% (Ming-Hsiang Su, 2017). There are few data sets that have been defined beforehand in this kind of model as compared to other models that have been proposed. CNN's accuracy rate was also however very high if one compared it to other models (Siddhant Rai, 2018), the results came in at a 98. 39% accuracy which was similar to the accuracy of the SVM model though it was tested in a large number of data sets. Thus, according to all the articles I could collect and based on which the selected approach was defined, the LSTM model (Vipasha Chandwani, 2020) is the future technology in the creation of chatbots that use previous conversations as a knowledge source when answering questions posed by the user. As in this model's case, it emulates responses from previous examples while enhancing the capabilities of future outcomes.

5. Implementation

For the project implementation process we have worked on all steps sequentially which are explained below.

i. **Initiation**: Specify the goals, parameters, and deliverables of the project. Determine the important parties and compile the healthcare chatbot's first specifications. To describe the project's objectives and vision, draft a project charter.

ii. **Planning and Design**: Create a thorough project plan with deadlines, goals, and resource allocation. Wireframes and prototypes should be made to design the chatbot's user interface and experience.

iii. **Development**: Construct the chatbot's front end, back end, and components. Django is used to construct the backend while HTML, CSS, and JavaScript are used to implement the user interface. Set up the database using Django ORM and integrate the Rasa Framework for the chatbot.

iv. **Testing and Quality Assurance**: To make sure the chatbot functions properly and satisfies quality requirements, do functional, performance, and security testing. Find and address any defects or problems found during testing.

v. **Refinement and Finalization**: Examine and improve the chatbot in light of user input and testing results. Complete the functionality and design to make sure the application satisfies all project criteria and is prepared for deployment.

vi. **Deployment and Project Closure**: Install and run the chatbot program in a live setting. Make sure users can access and the application is hosted correctly. Finalize project documentation, carry out a project review, and get input from relevant parties. Ascertain that all deliverables are fulfilled and offer any training or resources that may be required. Archive the project's documentation, then declare it closed.

6. Technical Documentation

Healthcare chatbot system is developed with subsystem of user management system, personal health management system and chatbot. Each system is developed alone and integrated at the end.

6.1 Tools and Technologies

S.N.	Category	Tool/Technology	Description
1.	Programming Language	Python (3.8.0)	Overall language used for the system
2.	Frameworks	Django (4.2.16)	Web framework for building application’s backend and handling requests
		RASA (3.6.20)	Open-source framework for building conversational AI and chatbots
3.	Testing	Unit Testing	Built-in python module
		Selenium (4.24.0)	Tool used for web browsers to perform UI testing and ensure the UI behaves correctly
4.	Web Technologies	HTML, CSS, JavaScript	Creating webpages
5.	Libraries	jQuery	Fast JavaScript library to simplify the DOM manipulation and event handling
		Bootstrap	For responsive design
6.	Database	SQLite	Default Database used by Django
7.	Version Control	Git	Version Controlling all the project and documentation
8.	Other Tools	Trello	We have used Trello for project management. Assigning tasks and tracking the deadlines. Divided the tasks for each member
		MindView(9.0)	Tools used for work breakdown structure

7. Code Documentation

7.1 Frontend Development

7.1.1 Functions Explanation

S.N.	Function Name	Purpose
1.	homepage	Renders the website homepage
2.	patienthome	Renders the patient homepage
3.	aboutpage	Renders the patient about us page
4.	loginpage	Renders the login form page where user can provide username and password to login into the system
5.	createaccountpage	To create the new account for patients
6.	chatbot_response	Renders the chatroom page
7.	updatepassword	Renders the update password page to change the password of the logged in user
8.	adminaddDoctor	Renders the page to add doctor
9.	adminviewDoctor	Renders the page to view doctor
10.	admin_delete_doctor	To delete the doctor
11.	patient_delete_appointment	To delete the appointment by patient
12.	adminviewAppointment	Renders the appointment page for admin
13.	Logout	Logout from the system
14.	Logout_admin	Logout admin from the system
15.	AdminHome	Renders the homepage of admin
16.	Home	Renders the home according to the user category.
17.	profile	Renders the profile according to the user category
18.	MakeAppointments	Renders the page to book the appointments
19.	viewappointments	Renders the page to view the appointments
20.	viewhealthrecords	Renders the page to view the past appointments
21.	contactus	Renders the contact us page

22.	get_available_time_slots	This function will generate the time slot for everyday according to the date chosen , also checks in the system that chosen date has available time slot or not This can be only selected if there is time slot in that date and chosen date is in weekday.

7.2 Backend Developement

7.2.1 Database Structure

S.N.	Model Name	Fields	Description
1.	Doctor	name	Stores the name of max characters 50
		email	Stores the email and unique
		licenseNo	Stores the license number of authorized doctor
		gender	Stores the gender either male or female
		phonenumber	Stores the phone number of characters 10
		address	Stores the address of characters up to 100
		specialization	Specifies the specialization of doctor
2.	Patient	name	Stores the name of max characters 50
		email	Stores the email and unique
		gender	Stores the gender either male or female

		phonenumber	Stores the phone number of characters 10
3.	Contact	name	Stores the name of max characters 50
		email	Stores the email and unique
		phonenumber	Stores the phone number of characters 10
		message	Stores the message provided by the user
4.	Appointment	doctorname	Stores the name of max characters 50
		doctoremail	Stores the email of doctor
		patientname	Stores the patient name
		patientemail	Stores the patient email
		appointmentdate	Stores the date of the appointment
		followupdate	Stores the date of follow up
		symptoms	Stores the symptoms provided by patient when booking appointment
		status	Specifies means the doctor has completed prescription or not
		prescription	Prescription for the client by doctor
		appointment_time	Appointment time slot for the chosen date

7.2.2 API

URL Path	View Function	Description
''	views.homepage	Main landing page
'home/'	views.home	User's home page after login
'about/'	views.aboutpage	Information about the healthCare
'contact/'	views.contactus	Contact information or form
'profile/'	views.profile	User profile page
'chatroom/'	chatviews.chatroom	Chatbot feature
'login/'	views.loginpage	User login page
'logout/'	views.logout	User logout functionality
'adminlogout/'	views.logout_admin	Admin-specific logout
'createaccount/'	views.createaccountpage	New patient registration
'patienthome/'	views.patienthome	Dashboard for patients
'doctorhome/'	views.doctorhome	Dashboard for doctors
'adminhome/'	views.adminhome	Dashboard for administrators
'adminadddoctor/'	views.adminadddoctor	Admin interface to add new doctors
'adminviewdoctor/'	views.adminviewdoctor	Admin interface to view doctor list
'admindeletedoctor/int:pk/str:email/'	views.admin_delete_doctor	Admin interface to remove a doctor
'adminviewappointment/'	views.adminviewappointment	Admin interface to view all appointments
'makeappointment/'	views.makeappointments	Interface for booking appointments
'viewmypatients/'	views.viewmypatients	Doctor's interface to view their patients
'viewallrecords/'	views.viewallrecords	View all medical records (likely admin or doctor)
'PatientDeleteAppointment/int:pid/'	views.patient_delete_appointment	Allow patients to cancel appointments

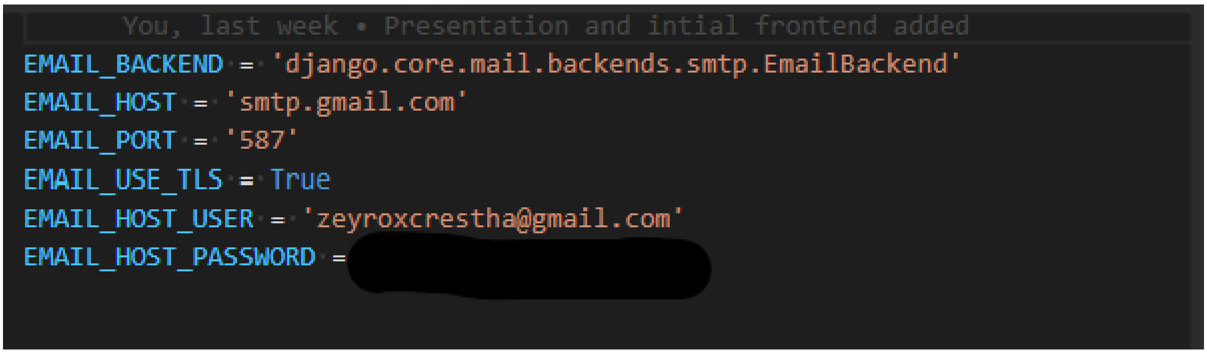
'get-available-time-slots/'	views.get_available_time_slots	API to fetch open appointment slots
'updatepassword/'	views.updatepassword	Interface for users to change password
'reset_password/'	auth_views.PasswordResetView.as_view()	Initiate password reset process
'reset_password_sent/'	auth_views.PasswordResetDoneView.as_view()	Confirmation of reset email sent
'reset/<uidb64>/<token>/'	auth_views.PasswordResetConfirmView.as_view()	Page to enter new password
'reset_password_complete/'	auth_views.PasswordResetCompleteView.as_view()	Confirmation of password reset

7.2.3 Folder Structure

- System
  - Frontend
  - HealthBot
  - Healthcarechatbotsystem
  - Manage.py

7.2.4 Password Recovery

This function is implemented by using Django SMTP email backend. For now, we have used our own email address by generating the app password. Later on, we can replace with the client one. We have use Gmail so set the host as GMAIL.



```
      You, last week • Presentation and intial frontend added
EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'
EMAIL_HOST = 'smtp.gmail.com'
EMAIL_PORT = '587'
EMAIL_USE_TLS = True
EMAIL_HOST_USER = 'zeyroxcrestha@gmail.com'
EMAIL_HOST_PASSWORD = [redacted]
```

7.3 Training the Chatbot

We have used RASA Framework for the chatbot. For the chatbot we have divided the file to train the chatbot.

S.N.	Name	Description
1.	nlu.yml	This file contains the user intents that means the input expected from the user and categorized into single topic. Example: - intent: thank_you examples: | - thank you - thanks - thnx - thank you very much - awesome - wow By this we can know user can say thank you in many ways. Like this we have created more intents.
2.	rules.yml	Rules contains specific criteria whenever we get something unique than expected. Example: - rule: Say goodbye anytime the user says goodbye steps:

		- intent: goodbye - action: utter_goodbye By this we can see any time user say bye it will response by responses in between chats also.
3.	stories.yml	This file contains the user story where user can go through the patterns one after another. Example: - story: gas stomach steps: - intent: stomach_symptom - action: utter_day - intent: day - action: utter_stomach_types - intent: affirm - action: utter_gas_response - action: utter_did_that_help By this we can see intent represent input from user and action is for response from bot. If anything matches it will select from multiple stories.
4.	domain.yml	This file contains the collections of intents, responses and text message for the user .

7.4. Integration

As we have separately developed the sub systems. We have integrated all using the available servers running on different port. Below is the detail breakdown of the integration between servers.

S.N.	**Host URL**	**Description**
1.	Localhost:8000	From this server the Django frontend and backend both is handled

2.	Localhost:5005	From this server the RASA framework will be running and replying to the response to the user.

8. Troubleshooting Issues

For issues related to Django it will be seen in the logs through the terminal from where it is being start up.

For issues related to chatbot it will be seen in the logs through the terminal from where it is being start up. Also, we can enable debug logs for chatbot from which every step for AI.

9. Testing

9.1 Unit Testing

- We have written and executed unit tests for each function using built-in python module. These tests ensure that individual components of our code are functioning correctly in isolation.
- Each unit test is designed to verify specific functionality and edge cases, helping us identify and fix issues at the function level before integration.

9.2 UI Testing

- For user interface testing, we utilized Selenium, a powerful tool for automating web browsers. Selenium allowed us to create automated test scripts that interact with the application's user interface.
- Our UI tests cover various scenarios, including form submissions, navigation, and user interactions, to ensure that the application behaves as expected from an end-user perspective.
- These tests help us verify that the user interface is both functional and user-friendly, catching any issues that may arise in real-world usage.

Together, these testing strategies provide a comprehensive approach to verifying the reliability and quality of our project, from individual functions to the overall user experience.

10. Project Completion

To finish the healthcare chatbot project quickly and with minimal data, concentrate on creating a Minimum Viable Product (MVP) that has basic symptom screening, typical health queries, and rapid medical advice. Furthermore, incorporate the chatbot with the personal health management system so that users may schedule doctor appointments. All users can be managed using the user management system, which guarantees smooth communication and efficient platform functionality. This method delivers essential value to users while guaranteeing timely project completion.

11. Timeline Comparison

Index:

Green *(Represents on track)*

Red *(Represents overspent)*

S.N.	Project Stages	Time Allocated	Time Spent
1.	**Initiation**	7 days	7 days
	• Kick off meeting	1 day	1 day
	• Requirement Gathering	4 days	4 days
	• Team Setup	2 days	2 days
2.	**Planning and Design**	6 days	6 days
	• Project Plan	1 day	1 day
	• **System Architecture Design**	4 days	4 days
	○ Wireframes Design	2 days	2 days
	○ UI Design	2 days	2 days
	• Approval of Design	1 day	1 day

3.	**Development**	16 days	**21 days**
	• **Backend Development**	9 days	14 days
	○ Integration Setup	2 days	5 days
	○ API Development	5 days	8 days
	○ Database Setup	2 days	1 day
	• **Frontend Development**	7 days	7 days
	○ Initial Testing	2 days	2 days
	○ Chatbot Integration	2 days	2 days
	○ UI Development	3 days	3 days
4.	**Testing and Quality Assurance**	13 days	13 days
	• Unit Testing	4 days	4 days
	• System testing	5 days	5 days
	• UAT (User Acceptance Testing)	4 days	4 days
5.	**Refinement and Finalization**	3 days	3 days
	• Bug Fixing	1 day	1 day
	• Feature Refinement	1 day	1 day
	• Documentation	1 day	1 day
6.	**Deployment and Project Closure**	3 days	3 days
	• Deployment	1/2 day	1/2 day
	• Training	1/2 day	1/2 day
	• Project Handover	1/2 day	1/2 day
	• Post-Deployment Support	1/2 day	1/2 day
	• Project review and Closure	1 day	1 day
	Total Duration	**43 days**	**43 days**

Refer to the Appendix for proposed Gantt Chart and Project Schedule

12. Challenges and Problems

During the implementation and testing of the healthcare chatbot and personal health management system, we faced several challenges that required creative solutions.

Data Limitations: One significant hurdle was the limited data available for the chatbot, which made it difficult to provide accurate health advice. To address this, we relied on publicly available health data and consulted with medical professionals to validate the chatbot’s responses.

Integration Issues: Integration issues arose when trying to connect the chatbot with the existing health management system, particularly around user management and appointment scheduling. We tackled this by conducting thorough API testing and using middleware to ensure smooth communication between the systems.

Natural Language Processing (NLP) Accuracy: The accuracy of the chatbot’s NLP was another challenge, especially with the limited training data. To improve this, we created more intents and rules which covers the basic chatbot features.

By addressing these challenges thoughtfully, we were able to deliver a robust and reliable healthcare chatbot system that met users’ needs effectively.

13. Strength and Limitations

13.1 Strength

User-Centric Design: The chatbot is designed to give users quick and easy access to basic health information, making it super convenient for anyone looking for some initial health advice.

Integration with Health Management System: By connecting with a health management system for booking appointments and managing user roles, the chatbot offers a one-stop solution that makes managing health easier and more organized.

Fallback Mechanism: If the chatbot can’t handle a question, it directs users to healthcare portals. This way, users get accurate help and feel secure knowing there’s always a backup.

13.2 Limitations

Data Dependency: The chatbot’s effectiveness really depends on the quality and amount of data it has. Without enough good data, it might not always give the best advice.

NLP Accuracy: The chatbot might struggle with complex or unclear questions due to its NLP limitations. It might need better training or updates to handle a broader range of user inputs.

System Integration: Making sure the chatbot and the health management system work smoothly together can be tricky. Integration issues might affect how well everything works.

14. Future Improvements and Lessons Learned

14.1 Future Improvements

Expand Data Sources: Adding more data sources and keeping the chatbot’s knowledge up to date will help it provide more accurate and relevant health advice.

Enhance NLP Capabilities: Investing in better NLP technology and expanding the training data can improve the chatbot’s ability to understand and respond to a wider variety of questions.

Optimize System Integration: Working on a smoother integration process and fixing compatibility issues will ensure everything runs more seamlessly.

14.2 Lessons Learned

Importance of User Feedback: Researching and getting feedback from user is crucial. It helps catch issues and make the chatbot more user-friendly.

Need for Robust Testing: Thorough testing is key to catching potential problems before they affect users. This includes checking APIs, performance, and security.

Scalability Considerations: Planning for future growth from the start helps avoid performance issues as the user base expands. Keeping the system responsive and efficient is important.

15. Potential Privacy Issues

Since personal health information is sensitive, controlling privacy issues is essential in healthcare chatbot projects. Unauthorized access to data, data breaches, regulatory noncompliance, incorrect data retention, insufficient user permission, and insufficient data anonymization are some major hazards. To solve them, put strong access restrictions and encryption in place to stop unwanted access, carry out frequent security assessments to find weaknesses, and make sure that laws like HIPAA and GDPR are being followed by conducting regular audits and training. To safely handle information, establish explicit regulations for the preservation and deletion of data, offer clear permission procedures that educate users about the use of their data, and employ data anonymization techniques for any study or analysis to safeguard user privacy. This may protect sensitive data, uphold regulatory compliance, and foster user confidence by proactively addressing these threats.

16. Professional ethics Issues

A healthcare chatbot must consider several ethical considerations, including protecting user privacy by disclosing the chatbot's limitations, maintaining accuracy and dependability through evidence-based information, preventing bias using diverse data, and handling data ownership and usage transparently. The data privacy and confidentiality seem to be the major one so described below clearly.

Data Privacy and Confidentiality: For a healthcare chatbot, maintaining data privacy and confidentiality is essential. To comply with laws like HIPAA and GDPR, this entails protecting data with robust encryption, enforcing stringent access restrictions, getting explicit user consent, and performing frequent security assessments. By taking these precautions, users' confidence is preserved, and sensitive data is protected. (Kosinski, 2018)

17. Project Meeting Minutes

Project Members:

- **Jitendra Shrestha**
- Pranish Acharya

Date:	31st July, 2024	**Time**:	15:00	**Location**:	Inside Classroom

1. Kickoff Meeting

Agenda:

- Introduction
- Project Overview
- Project Scope and deliverables
- Work division of project proposal

Attendance

Present	Absent
Both	Nil

Meeting Brief Summary

Introduction:

- Self Introduction
- Bring topics and get confirmed by supervisor
- Find out each members strength on the chosen topic
- Search the project on scholar and discussed on those
- Discussed on SDLC

Project Overview:

- Discussed the selected chatbot system topic
- Discussed types of chatbot system
- Finalized the objectives and purpose

Project Scope and Deliverables:

- Defined the scope which we will be working on it
- Searched for the deliverables required for project

Work division of project proposal:

- Introduction, Background and client profile is completed on surface level
- Purposes and Objectives, Scope and Exclusion, Assumptions and constraints will be done by Jitendra

- Deliverables, Schedule, budget will be done by Pranish
- Will merge our contribution and review once before submission by both

AOB:

- Nil.

Next Meeting

Date:	7th August, 2024	**Time**:	14:00	**Location**:	Inside Classroom

This meeting minute was prepared by Jitendra.

Date:	7th August, 2024	**Time**:	14:00	**Location**:	Inside Classroom

2. Requirement Gathering

Agenda:

- Progress of Project Proposal
- Any challenges in proposal
- Feasibility Study

Attendance

Present	**Absent**
Both	Nil

Meeting Brief Summary

Progress of Project Proposal:

- Introduction, Purposes and Objectives, Scope and Exclusion are completed
- Deliverables and schedule are completed

Any challenges in proposal:

- Discussed the challenges might be on real data feeding for chatbot
- Also, for the budget, estimated the real project cost

Feasibility Study:

- Can be developed using python and NLP
- Will be working on limited data so every area will not be covered

AOB:

- Nil.

Next Meeting

Date:	14th August, 2024	**Time**:	18:00	**Location** :	Outside Classroom

This meeting minute was prepared by Pranish.

Date:	14th August, 2024	**Time**:	18:00	**Location**:	Outside Classroom

3. Design Phase 1 Meeting

Agenda:

- Development Methodology
- Functional and Non-Functional Requirements
- Use Case Diagrams
- User Interfaces
- Tools discussion for creating diagrams
- Task Allocation

Attendance

Present	Absent
Both	Nil

Meeting Brief Summary

Development Methodology:

- Using Agile Approach
- Testing at same time while development

- Breaking down the features and combine

- Issue can be resolved and modify the features

Functional and Non-Functional Requirements:

- Functional Requirements

- Create ER diagram
- Create Functional Decomposition Diagram
- Define the data (User, Appointment, Doctor)

- Non-Functional Requirements

- Look and Feel Requirements
- Performance Requirements
- Maintainability and support Requirements
- Security Requirements
- Cultural and Political Requirements
- Legal Requirements

Use Case Diagrams:

- Overall use case Diagram

- Deep dive into the functionality on each function from overall use case

User Interfaces:

- Discussion on the wireframes for the chatbot

- Discussion on the wireframes for booking an appointment

- Discussion on Login Page wireframe

Tools discussion for creating diagrams:

- Draw.io will be used for diagrams (https://app.diagrams.net/)

- WBS tool for Gantt chart (https://www.workbreakdownstructure.com/)

Task Allocation:

Jitendra	Pranish
• Creating wireframes for chatbot UI, login in UI • Non-Functional requirements • Creating Overall Use case and multiple level Use case • Flow chart diagram	• Functional Requirements • Create ER Diagrams • Sequence Diagram • Activity Diagram • Class Diagram

AOB: - Nil.					
Next Meeting					
Date:	16th August, 2024	**Time**:	13:00	**Location** :	Inside Classroom
This meeting minute was prepared by Jitendra.					

Date:	16th August, 2024	**Time**:	13:00	**Location**:	Inside Classroom
4. Design Phase 2 Meeting					
Agenda: - Progress on the assigned tasks - Issues on design - System Navigation					
Attendance					
Present			**Absent**		
Both			Nil		
Meeting Brief Summary					
Progress on the assigned tasks: - Completed the overall use case diagram - Completed wireframes for login page UI - Completed the Flow chart diagram, sequence diagram					
Issues on design: - Discussion on wireframes for chatbot - Go through the chatbot from different websites and checked the UI - Planned to implement the chatbot for quicker and correct response					

System Navigation:

- After creating wireframes, decided the system navigation
- Users doesn’t need to login for accessing chatbot
- Users need login to save their data and book appointment
- Doctors can add prescription after looking the patients
- Admin can manage the patients and doctors

AOB:

- Nil.

Next Meeting

Date:	21st August, 2024	**Time**:	13:00	**Location** :	Inside Classroom

This meeting minute was prepared by Jitendra.

Date:	21th August, 2024	**Time**:	13:00	**Location**:	Inside Classroom

5. Discussion on System analysis and Design

Agenda:

- The meeting was initiated by Pranish to discuss the ongoing project regarding system analysis and design for Health Care Chatbot system.
- The objective is to finalize the functional and non-functional requirements and to develop a high-level design for the new system.

Attendance

Present	**Absent**
Both	Nil

Meeting Brief Summary

Progress of System Analysis and Design:

- Current System Architecture.
- Functional requirements

- Non-functional requirements - Data Flow Diagram - Wireframes - Database design					
Any challenges in proposal: - Discussed the challenges might be on database design and data flow diagram - Also, for Architecture Design.					
AOB: - Nil.					
Next Meeting					
Date:	23th August, 2024	**Time**:	18:00	**Location**:	Outside Classroom
This meeting minute was prepared by Pranish.					

Date:	23th August, 2024	**Time**:	18:00	**Location**:	Outside Classroom
6. Discussion on System analysis and Design					
Agenda: - The meeting was initiated by Jitendra to discuss the ongoing progress regarding system analysis and design for Health Care Chatbot system. - The objective is to finalize the System Analysis and Design Part before the deadline.					
Attendance					
Present			**Absent**		
Both			Nil		
Meeting Brief Summary					
Progress of System Analysis and Design: - Current System Architecture - ERD - Data Dictionary					

<table>
<tr><td colspan="6">- Jitendra presented how data will be processed through the system.
- Pranish proposed an architecture based on LSTM model, SVM, explaining the use of specific technologies such as Rasa framework.
- Jitendra discussed mockups for the user interface. Feedback was given on improving use case diagram, gantt chart and ERD.
- Discussed on Project Cost Estimation, Data privacy</td></tr>
<tr><td colspan="6"><b>Any challenges in System analysis and design:</b>
- As the project progresses, new features which can lead to scope creep.
- A tight timeline may not allow enough time for thorough analysis or testing, leading to errors in the final system design
- Design complexities or unexpected changes can lead to cost overruns.</td></tr>
<tr><td colspan="6"><b>Feasibility Study</b>:
- Can be secure data using
- Will be working on limited data so every area will not be covered.</td></tr>
<tr><td colspan="6"><b>AOB:</b>
- Nil.</td></tr>
<tr><td colspan="6"><b>Next Meeting</b></td></tr>
<tr><td><b>Date</b>:</td><td>28th August, 2024</td><td><b>Time</b>:</td><td>18:00</td><td><b>Location</b>:</td><td>Outside Classroom</td></tr>
<tr><td colspan="6">This meeting minute was prepared by Pranish.</td></tr>
</table>

<table>
<tr><td><b>Date</b>:</td><td>28th August, 2024</td><td><b>Time</b>:</td><td>13:00</td><td><b>Location</b>:</td><td>Outside Classroom</td></tr>
<tr><td colspan="6"><b>7. Discussion on Development of project</b></td></tr>
<tr><td colspan="6"><b>Agenda</b>:
- Introduction to Django and Rasa Frameworks
- Define Project Deliverables and Work Division
- Setup Development Environment and Tools</td></tr>
<tr><td colspan="6"><b>Attendance</b></td></tr>
<tr><td colspan="3"><b>Present</b></td><td colspan="3"><b>Absent</b></td></tr>
</table>

Both	Nil

Meeting Brief Summary

Progress of System Development:

- Discussion of Django’s role as the web framework and Rasa as the NLP engine for chatbot responses.
- Pranish will handle the integration of Rasa with Django for conversational flows.
- Jitendra will develop user authentication, manage patient data storage, and oversee database security.
- Pranish will focus on training the Rasa chatbot for symptom checking.
- Environments: Python 3.8, Django 4.0, Rasa 3.0.

Any challenges in Development:

- Challenge integrating real-time feedback from Rasa into Django UI for chatbot.
- Resolved by using Django Channels for WebSocket integration.

Feasibility Study:

- Can be developed using rasa and Django chatbot.
- Will be working on limited data so every area will not be covered.

AOB:

- Nil.

Next Meeting

Date:	4th September, 2024	**Time**:	12:00	**Location** :	Inside Classroom

This meeting minute was prepared by Pranish.

Date:	4th September, 2024	**Time**:	12:00	**Location**:	Inside Classroom

8. Discussion on Development and Testing and

Agenda:

- System finalization
- Testing strategies and procedures
- Bug fixing and improvements

Attendance

Present	Absent
Both	Nil

Meeting Brief Summary

Progress of System Development:

- The healthcare chatbot system is fully integrated with Rasa and Django.
- All essential features such as appointment scheduling, symptom checking, user authentication, and patient-doctor interaction have been implemented.
- Both manual and automated testing have been completed.
- Unit tests covered Django models and views, while Rasa chatbot underwent functional tests for intent recognition and conversational flow accuracy.
- Bugs found in edge cases of patient data input were fixed.

Any challenges in Development:

- None

AOB:

- Nil.

Next Meeting

Date:	10th September, 2024	**Time**:	14:00	**Location** :	Outside Classroom

This meeting minute was prepared by Jitendra.

Date:	10th September, 2024	**Time**:	14:00	**Location**:	Outside Classroom

2. Discussion on System analysis and Design

Agenda:

- Finalizing system documentation
- User manual preparation
- Project report completion

Attendance	
Present	**Absent**
Both	Nil
Meeting Brief Summary	
Progress of System Development: - All technical aspects, including system architecture, APIs, database schema, and user interaction flows, have been documented. - User manual includes instructions for using the chatbot, scheduling appointments, and managing profiles. - The final project report is almost complete, including project objectives, development phases, and challenges faced. - Documentation will be submitted before the deadline.	
Any challenges. - None	
AOB: - Nil.	
This meeting minute was prepared by Jitendra.	

References

Anupam Mondal, M. D. (2018). Chatbot: An automated conversation system for the educational domain. *2018 International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)*,

Ayanouz, S., Abdelhakim, B.A. and Benhmed, M. (2020 March). A smart chatbot architecture based NLP and machine learning for health care assistance. *In Proceedings of the 3rd international conference on networking, information systems & security*, 1-6.

Falguni Patel, R. T. (2019). Combating Depression in Students using an Intelligent ChatBot: A Cognitive Behavioral Therapy. Rajkot, India: IEEE.

Kosinski, Y. W. (2018). Deep Neural Networks Are More Accurate Than Humans at DetectingSexual Orientation From Facial Images. *Journal of Personality and Social Psychology, 114*, 246-257.

Kurup, G. and Shetty, S.D. (2022). AI conversational chatbot for primary healthcare diagnosis using natural language processing and deep learning. *In Computational Intelligence in Pattern Recognition: Proceedings of CIPR 2021*, 259-272.

Lekha Athota, V. K. (2020). 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO). Noida, India: IEEE.

Ming-Hsiang Su, C.-H. W.-Y.-B.-M. (2017). A chatbot using LSTM-based multi-layer embedding for elderly care. Singapore: IEEE.

Rohit Binu Mathew, S. V. (2019). Chatbot for Disease Prediction and Treatment. Kottayam, Kerala, India: IEEE.

Santosh Gore, D. D. (2023). Leveraging BERT for Next-Generation Spoken Language Understanding with Joint Intent Classification and Slot Filling. Mumbai, India: IEEE.

Siddhant Rai, A. R. (2018). Darwin: Convolutional Neural Network based Intelligent Health Assistant. Coimbatore: IEEE.

Vipasha Chandwani, S. K. (2020). Long Short-Term Memory based Conversation. Jaipur, India, India: IEEE .

Woods, S. A.-K. (2015). Survey on Chatbot Design Techniques in Speech. *(IJACSA) International Journal of Advanced Computer Science and Applications*, 72-80.