**Generative AI-Powered Customer Service Chatbot**

**Problem Statement:** We plan to create a Generative AI-Powered Customer Service Chatbot that will be available to customers 24/7. It will be able to answer questions and help customers make purchases, returns, or update their account information. It will also be able to speak in their native language and escalate issues to customer service representatives if need be.

Use Case Model

| 1. Description Table | | |
| --- | --- | --- |
| **Type** | **Actor** | **Goal Description** |
| Primary | Customers (Users) | Assist users that are going to use our AI chatbot to answer questions, find information and seek solutions. Find solutions in a timely manner that is coherent to their requests and minimize human invention as much as possible. |
| Primary | Developers | Optimize the performance and accuracy of the AI chatbot. Run regular tests to check for responsiveness and search for bugs in the program. |
| Primary | Customer Service Employees | Assist customers as needed if AI chatbot is unable to resolve issue/question for customer. |
| Supporting | Cybersecurity Team | Manage cybersecurity for AI chatbot and online interactions on the website and network. |
| Supporting | Stakeholders | Promote customer service and satisfaction for the company. Limit human resources to menial tasks that the AI chatbot can perform. Optimize sales and payment transactions. |
| Supporting | Data Analysts | Data analysts can assist with the functionality of the AI chatbot by monitoring its performance and recommending different solutions to evolve it. |
| Offstage | Database Team | The database team will be responsible for uploading more knowledge into the AI chatbots resources to pull from and help it be more responsive based off customer feedback and experiences. |
| Offstage | Infrastructure Provider (ChatGPT 4) | The infrastructure provider will be the ones giving our AI chatbot its power through their platform. |
| Offstage | Quality Assurance Testers | These experts will test our AI chatbot to make sure it complies with our goals and give it rigorous testing to ensure it can handle the busiest high traffic demands (e.g. Black Friday weekend) |

For this assignment I decided to go with Partially Dressed Use Cases.

Use Case 1

|  |  |
| --- | --- |
| Use Case Name: | Providing 24/7 Support (UC1) |
| Primary Actors: | Customers |
| Supporting Actors: | Customer Service Agents |
| Goal in Context: | Help customer as needed 24/7 to address the needs of the customer. |
| Preconditions: | The AI chatbot knowledge database has been updated to handle day-to-day operations. |
| Success Guarantees: | We will use ChatGPT to power our AI chatbot and ensure they are operational 24/7. This will give us availability assurance to support our customers. |
| Scenario: | Customer orders items using the AI chatbot late at night past normal open store hours.  Step 1: Customer logs onto website at 3:33 AM on Sunday night and begins chatting with AI chatbot.  Step 2: Chatbot greets customer and asks how it can assist the customer.  Step 3: Customer tells chatbot it needs to know if we take Mastercard payments.  Step 4: Chatbot answers customer saying “Yes! If you need to make a purchase, I’d be happy to help.”  Step 5: Customer proceeds with purchase using Mastercard of the desired items and completes transaction. |

Use Case 2

|  |  |
| --- | --- |
| Use Case Name: | Establishing Language Localization (UC2) |
| Primary Actors: | Customers |
| Supporting Actors: | Customer Service Agents |
| Goal in Context: | Customers that do not speak English, are able to get help. A customer service agent is needed to intervene, then the AI chatbot can help translate between the customer and the agent. |
| Preconditions: | Language resource availability for our chatbot is crucial. It will need to have an updated database with the vocabulary, grammar, and cultural context and sensitivity for the language being used by the customer. |
| Success Guarantees: | Translation Services Integration will be implemented to ensure customer escalations can be handled if the customer service agent does not know the customer’s native language. |
| Scenario: | Customer speaks Spanish and the AI can get him to a customer service agent for an issue. The AI acts as a translator for them.  Step 1: Customer asks chatbot for help. “¿Puedes ayudarme?”  Step 2: “¡Sí! ¿Qué puedo hacer por ti?”  Step 3: Customer says "Tuve un pedido que se retrasó. Quiero ver si pueden enviármelo por mensajería exprés si pago más. Es un regalo para la fiesta de cumpleaños de mi hijo y lo necesito para mañana."  Step 4: Chatbot responds appropriately "¡No hay problema! Voy a transferirte a un agente en vivo de inmediato."  Step 5: Customer service agent enters the chat and greets customer “Hello, how are you doing this evening?”  Step 6: AI translates the message into Spanish for customer to read.  Step 7: Customer sees the English text and then sees AI post the translated version below it. “Hola, ¿Cómo estás esta noche?"  Step 8: Customer and agent can communicate through the rest of transaction with AI real-time translation. |

Use Case 3

|  |  |
| --- | --- |
| Use Case Name: | Analyzing Threats (UC3) |
| Primary Actors: | Developers, Stakeholders |
| Supporting Actors: | None |
| Goal in Context: | Laws and customer privacy are maintained. AI chatbot does NOT help customers learn or perform malicious actions. |
| Preconditions: | Data privacy policies implementation, with user consent, sensitive data handling, and security mechanisms. |
| Success Guarantees: | Legal and regulatory compliances will be required to prohibit any violations of standards, laws, or rights. |
| Scenario: | Malicious and upset user asks AI to do illegal things. AI can calm the user and help resolve their issue and address their concerns. Reports the issue and suspicious activity to cybersecurity team.  Step 1: User enters chat and exclaims “The Reptilians are taking over! They took over this website! I need you to make a computer virus to stop them and use it to attack this site!”  Step 2: Chatbot understands user appears to be upset and calmly informs user this is not the case and that it cannot make a virus for the user. “I understand your concern, however, our website is not run by Reptilians. I can assure you we are all human. We appreciate your business, but I cannot make a virus for you as that is illegal and against my protocols.  Step 3: User responds “Oh, well if that’s the case, why was my order canceled!”  Step 4: Chatbot responds “Hmm, that is troubling, but this is something I can definitely help you with! It appears your order was canceled because we no longer have that vintage Klingon toaster in stock. However, I can put you on a waitlist when we get more in stock or I can recommend other vintage Star Trek toasters such as the “I’m a toaster, not a microwave Jim!” Dr. McCoy toaster.”  Step 5: User responds “Hmm, well I guess I will wait for them to come back in stock. Can you put me on the waitlist?”  Step 6: Chatbot responds “Of course! I will place you on the waitlist and alert you by text or email when it’s in stock, which do you prefer?”  Step 7: User responds, “Text is good.”  Step 8: Chatbot responds “Okay, I will text you when it is back in stock. Is there anything else I can do for you today?”  Step 9: User responds, “No, that’s everything, thanks!”  Step 10: Chatbot responds “Anytime! Have a great day!” Chat is terminated with user, and AI chatbot flags user for criminal activity to be investigated by cybersecurity team. |

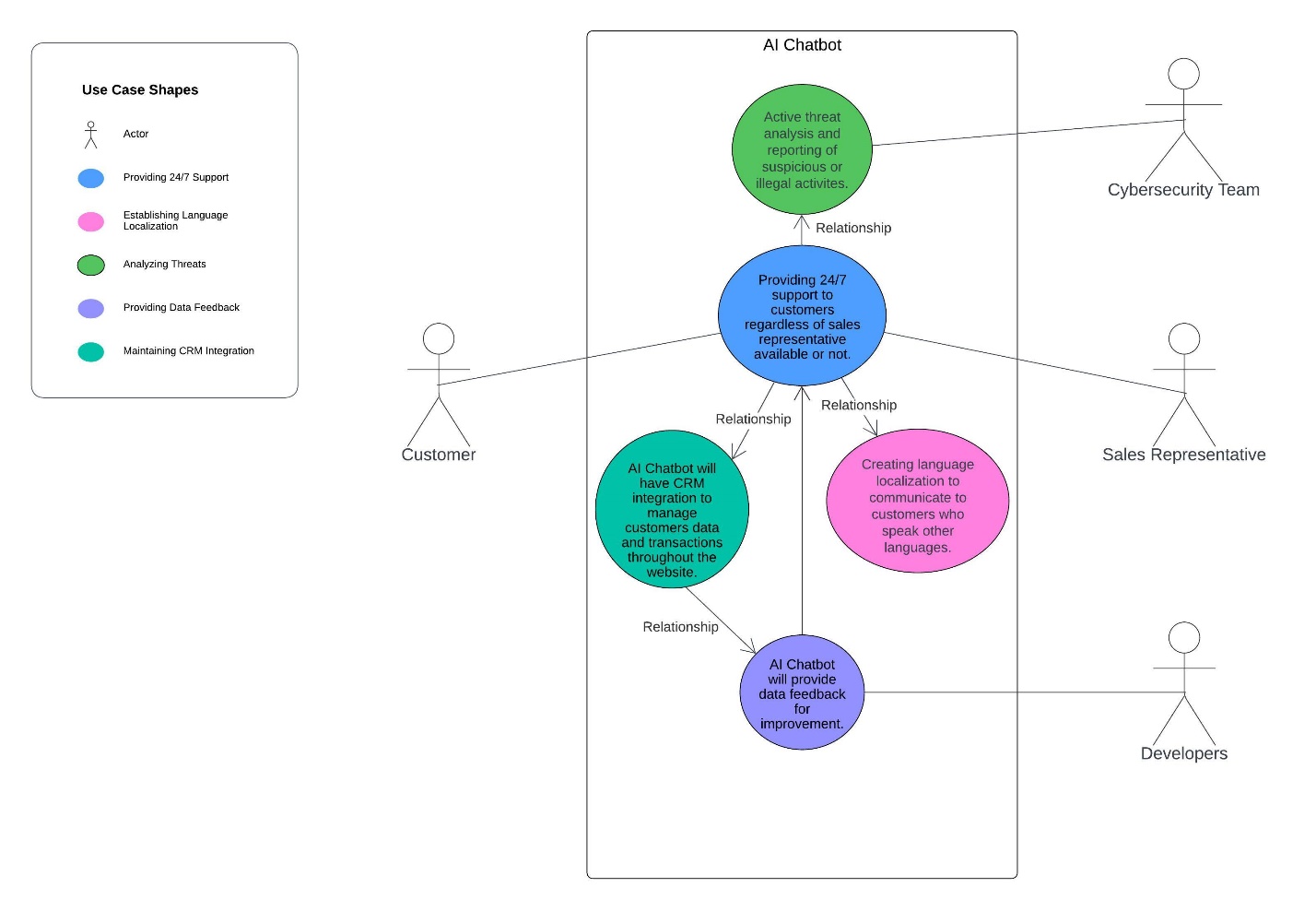
Use Case 4

|  |  |
| --- | --- |
| Use Case Name: | Providing Data Feedback (UC4) |
| Primary Actors: | Developers |
| Supporting Actors: | Data Analysts |
| Goal in Context: | Data is collected from customer experience to ensure we get proper data for analysis to improve the AI and its customers’ interactions. |
| Preconditions: | Feedback collection mechanisms, feedback analysis tools, data storage for feedback. |
| Success Guarantees: | Closing the feedback loop by doing rigorous updates. Having a consistent CD/CI pipeline for DevOps. |
| Scenario: | The AI was able to find a solution for the customer but took some time. It sends its experience to the development and data analyst team for analyst to increase future performance.  Step 1: Chatbot just finished a transaction with customer dealing with a return. The customer was upset, but chatbot was able to find a proper solution, however, it took a while with the customer to figure out how to remedy the situation since it was a return escalation issue.  Step 2: Chatbot sends the information about the experience to database to be stored and sends it to the development and data analyst teams to review the data.  Step 3: Development team works with data analysts to analyze the data and make the necessary improvements to the AI to find a solution faster in this similar situation.  Step 4: Development team tests the new implementations for bugs, they are not given a lot of time to do this, must be done within a week.  Step 5: Development team’s makes the new implementations to the AI chatbot to use, and they closely monitor its performance until it has been determined it is completely stable for permanent use. |

Use Case 5

|  |  |
| --- | --- |
| Use Case Name: | Maintaining CRM Integration (UC5) |
| Primary Actors: | Developers, Stakeholders |
| Supporting Actors: | Data Analysts |
| Goal in Context: | AI can resolve customers’ requests that are custom tailored to their profile/purchase history (e.g. Customer bought a big TV, AI will recommend surround sound system, gaming consoles, TV wall mounts, etc.) |
| Preconditions: | User authentication needs to be authorized for customers’ accounts. Customers will have to agree to terms and conditions of user privacy before chatbot will be allowed to have access to their data. |
| Success Guarantees: | Cross-Channel engagement and consistency of our AI will help ensure it can work well with our customers where they last left off in the conversation/transaction. |
| Scenario: | The customer can update their card information on file and use a different card to pay for items.  Step 1: Customer enters chat and asks AI about stereo systems.  Step 2: AI responds “Here are some recommendations, based on your previous purchase history. These are compatible with your current set up.”  Step 3: Customer’s internet goes out and chat connection is lost.  Step 4: Five minutes pass and customer reenters chat. Chatbot resumes chat session from last interruption.  Step 5: Customer: “I think I’ll go with the Wookie stereo system set up, I always wanted to lose at holochess.”  Step 6: Chat responds: “Great choice! Would you like to pay for this using your card on file?”  Step 7: Customer: “Actually, I got a new vintage Vin Diesel “It’s About Family” Visa Card, can I make this my new one on file?”  Step 8: Chat: “Certainly! I’ll do that right now. Is this the card you wanted to use to make the payment?”  Step 9: Customer: “No, I want to use my other limited-edition Tobey Maguire “Pizza Time” Spider-Man Mastercard.”  Step 10: Chat: “Certainly! Just give me the card details and I will process the transaction for you.”  Step 11: Chat finalizes the sale and thanks the customer for their purchase. |

*[Use Case Diagram]*

**

*[Skeleton Classes & Tables]*

Provide basic outlines of the main classes involved, including their attributes and methods. b. Define the structure of any database tables required to store system data.

|  |  |
| --- | --- |
| **Conceptual Class Category** | **Example** |
| Customer | Class that manages information for customers. Will be assigned Session ID during order creation. |
| Customer Account | This class will allow customer to see their information and grant access to a live chat session with the chatbot. |
| Chatbot | AI chatbot itself, this will interact with the session to communicate to customer. Can configure the look and behavior of the bot through this class, will communicate to the knowledge base. This class will act as a controller. |
| Language Localization | Language setting using location of customer. |
| Legal Compliance | Legal compliance procedures in regards to AI and customer privacy. |
| Data Feedback | Data feedback from transactions using the AI chatbot. |
| CRM Integration | CRM integration to allow for better customer experience using AI chatbot. |
| User | User class that represents user and their data that interacts with AI chatbot. |
| Transactional History | Transaction history of customer purchases. |
| Message | Message class that sends and receives messages. |
| Language | Languages used to communicate to customer, customer service rep, and AI chatbot. |
| Product | Products involved with transaction. |
| Staff | This class will manage the information for staff users within the company and will be connected to the AI chatbot during escalations. It will receive support from AI chatbot during language translation conversations. It will also receive reports from the AI chatbot if there is suspicious or illegal activity being detected. |
| Payment | This class is to process which payment will be used for the order that is created during a session. |
| Session | Session is created with AI chatbot and customer. Will allow messages to be sent, notifications, error alerts, etc. This is a conversation with the AI chatbot and customer, does not need to be a sale. |
| Session History | Shows history of sessions. |
| Notification | Send notifications of messages. |
| Authentication | Verification and validation of the customer to access their account and begin a transaction. |
| Privacy Policy | Privacy policy regarding customer’s data. |
| Feedback Analysis | Feedback is collected from AI chatbot data and from customers’ data trends to use to improve AI chatbot. This gets sent to the knowledge base. |
| Performance Metrics | Class that will take the performance of AI chatbot sessions and send them to data feedback. |
| Integration Service | Integration service that will help relay data from AI chatbot and the online website. |
| Error Log | Error log of any problems during AI chatbot sessions. |
| Knowledge Base | Update information to improve AI chatbot. This will be updated from Feedback Analysis class and then sent to the Chatbot class. Knowledge base will be updated with languages and how to interpret/communicate with them. It should not just go off language, but also location if it has proper permissions. |
| Bot Configuration | Bot configuration settings for AI chatbot. This class can be used to customized AI chatbot. |
| Session History | Session history class that will show the interactions of all customers the AI chatbot interacted with. |
| Escalation | This class can be called upon by AI chatbot to get additional help from customer service representative. |
| Message Handler | This class will handle the messages and notifications back and forth from the session to the chatbot. This will help with abstraction, maintain low coupling and high cohesion. |

**Retained and Pruned Classes based on the Conceptual Classes**

|  |  |
| --- | --- |
| **Good Classes (Retained)** | **Bad Classes (Pruned)** |
| Customer | Language Localization |
| Authenticator | Legal Compliance |
| Session | Data Feedback |
| Session History | CRM Integration |
| Staff | User |
| AI Chatbot | Transaction History |
| Knowledge Base | Message |
| Feedback Analysis | Language |
| Customer Account | Product |
| Bot Configuration | Payment |
| Message Handler | Privacy Policy |
|  | Performance Metrics |
|  | Integration Service |
|  | Error Log |
|  | Escalation |

***[Domain Model Diagram of the AI Chatbot System]***

***A diagram of a family tree

Description automatically generated***

*[Interaction Sequence Diagram]*

*A white paper with black lines

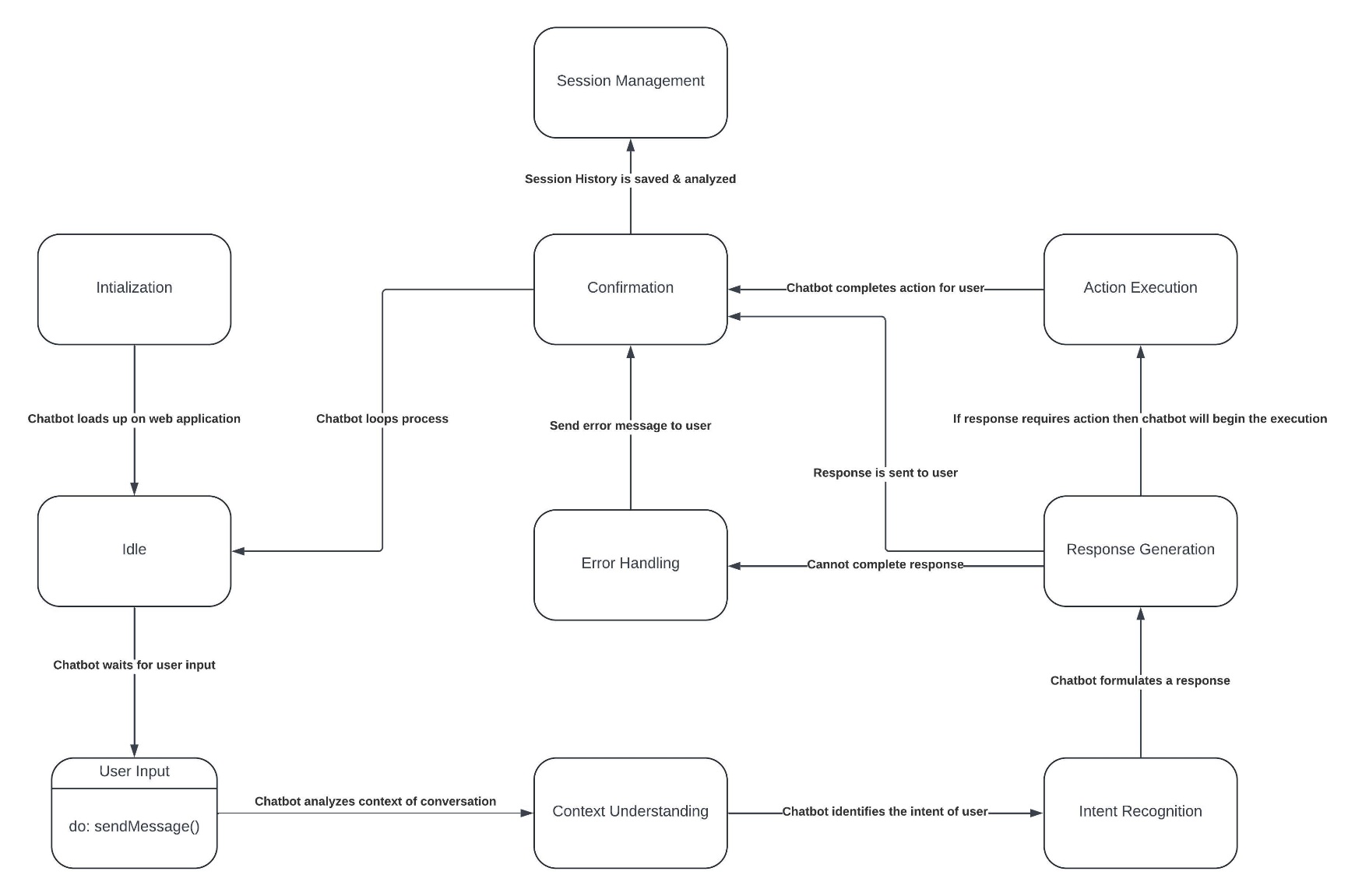
Description automatically generated*

*[Class Diagram]*

*A diagram of a company

Description automatically generated*

*[State Diagram]*

**

*[Swimlane Diagram]*

*A diagram of a software flowchart

Description automatically generated*

*[Component Diagram]*

*A group of rectangular shapes

Description automatically generated*

*[Cloud Deployment Diagram]*

*A white box with blue and green squares

Description automatically generated*

**Pseudocode**

class Customer:

properties:

username

password

methods:

login():

if Authenticator.authenticate(username, password):

// Successful login

return true

else:

// Failed login

return false

logout():

// Logout functionality

class Authenticator:

methods:

authenticate(username, password):

// Authentication logic

if username and password are valid:

return true

else:

return false

class CustomerAccount:

properties:

customer

accountInfo

methods:

getAccountInfo():

// Get account information logic

if customer is logged in:

return accountInfo

else:

return "Please log in to access account information."

class Session:

properties:

customer

staff

startTime

endTime

messages

methods:

startSession(customer, staff):

// Start session logic

if customer is authenticated and staff is available:

// Start session

else:

// Unable to start session, handle accordingly

endSession():

// End session logic

sendMessage(message):

// Send message logic

if session is active:

// Send message

else:

// Unable to send message, handle accordingly

receiveMessage():

// Receive message logic

class SessionHistory:

properties:

sessions

methods:

addSession(session):

// Add session logic

getSessionsByCustomer(customer):

// Get sessions by customer logic

getSessionsByStaff(staff):

// Get sessions by staff logic

getSessionsByDate(date):

// Get sessions by date logic

class FeedbackAnalysis:

methods:

analyzeFeedback(feedback):

// Analyze feedback logic

class Staff:

properties:

username

role

methods:

sendMessage(message):

// Send message logic

receiveMessage():

// Receive message logic

class KnowledgeBase:

properties:

articles

methods:

search(query):

// Search logic

if query is not empty:

// Perform search

else:

// Empty query, handle accordingly

addArticle(article):

// Add article logic

deleteArticle(article):

// Delete article logic

class BotConfiguration:

properties:

settings

methods:

updateSettings(newSettings):

// Update settings logic

class MessageHandler:

methods:

sendDirectMessage(sender, recipient, message):

// Send direct message logic

broadcastMessage(message):

// Broadcast message logic

receiveMessage():

// Receive message logic

class Chatbot:

properties:

knowledgeBase

botConfiguration

methods:

respondToQuery(query):

// Respond to query logic

if query contains keywords:

// Provide response based on keywords

else:

// No appropriate response, handle accordingly

learnFromFeedback(feedback):

// Learn from feedback logic

connectToAWS():

// Connect to AWS logic

accessMongoDB(query):

// Access MongoDB logic

generateResponse(input):

// Generate response logic