# ANNA UNIVERSITY REGIONAL CAMPUS COIMBATORE

# CHATBOT DEPLOYMENT WITH IBM CLOUD

# WATSON ASSISTANT

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

1. **Define Objectives:**

Clearly outline the objectives and goals of your virtual guide. What specific information or assistance will it provide? Who is your target audience?

1. **Platform Selection**

**:** Make sure you understand the requirements and capabilities of your chosen messaging platforms (Facebook Messenger and Slack) to ensure compatibility and effective integration.

1. **IBM Watson Assistant Setup:**
   * Set up an IBM Cloud account if you don't have one.
   * Create a Watson Assistant instance within IBM Cloud.
   * Configure the assistant with basic conversation flows and responses. Initially, you can start with a simple FAQ-style chatbot.
2. **NLU Implementation:**

Implement Natural Language Understanding (NLU) for more accurate user intent recognition. Watson Assistant provides NLU capabilities, allowing your chatbot to understand the context and meaning behind user queries.

1. **Conversation Design:**

Design conversational flows that guide users through their interactions. You can use IBM Watson Assistant's dialog nodes to structure conversations.

1. **Content Development:**

Populate your virtual guide with relevant content, FAQs, and responses. This content should be informative and up-to-date. You can use Watson Assistant's content management features to organize this data.

1. **User Engagement:**

Make the conversation engaging and user-friendly. Use friendly language and provide options for users to navigate through the information efficiently.

1. **Multilingual Support:**

Consider providing multilingual support, especially if your audience is global.

1. **Integration with Messaging Platforms**

**:** Integrate your virtual guide with Facebook Messenger and Slack. This may involve creating chatbot apps or configuring the necessary APIs.

Continuously test your virtual guide to ensure it performs well and accurately understands user intents. Gather feedback from users and make iterative improvements.

1. **Security and Privacy:**

Pay attention to data security and privacy. Ensure that any user data collected is handled in compliance with relevant regulations (e.g., GDPR) and that sensitive information is protected**Scalability:**

Plan for scalability to handle increased user interactions as your chatbot gains popularity.

1. **Analytics and Insights:**

Implement analytics to track user interactions and gather insights. This data will help you improve the chatbot's performance over time.

1. **Marketing and Promotion:**

Promote your virtual guide to attract users. Social media, your website, and existing communication channels can be used for this purpose.

1. **Feedback Mechanism**

**:** Implement a feedback mechanism within the chatbot so users can provide feedback and suggestions for improvement.

1. **Maintenance and Updates:**

Regularly update the chatbot's content to keep it current. Stay informed about changes in the messaging platforms and adjust your chatbot as needed.

1. **Compliance and Regulations:**

Ensure that your virtual guide complies with all relevant laws and regulations, especially those related to data protection and user privacy.

1. **User Training:**

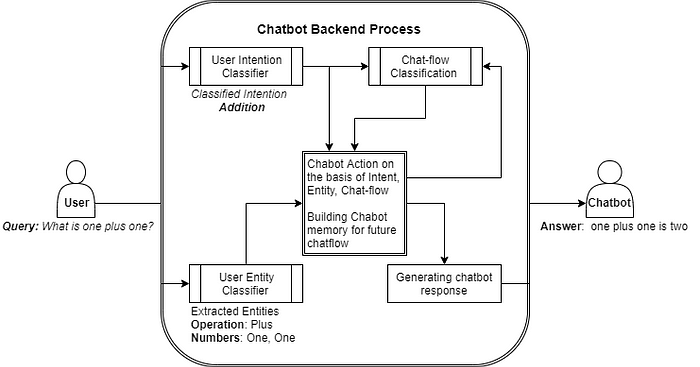
Educate users on how to use the virtual guide effectively. Provide clear instructions and promote its capabilities.

1. **User Support:**

Have a system in place for user support and escalation if the chatbot is unable to answer a user's query.

**CHATBOT FLOW:**

The chatbot flow is based on three main components Intention, Keywords, Chatflow.



1. **Welcome Message**:
   * Start the conversation with a friendly welcome message.
   * You can also set the tone of the conversation here, whether it's formal, casual, professional, etc.
2. **User Greeting and Identification**:
   * Ask the user for their name or any other relevant information to personalize the conversation.
   * If the user has interacted with the chatbot before, you can greet them by name if you have that information.
3. **User Input**:
   * Allow the user to provide their query or request.
   * Use open-ended questions to encourage natural language responses.
4. **Processing User Input**:
   * Analyze and interpret the user's input.
   * Use natural language processing (NLP) to understand the user's intent.
   * Recognize keywords and context to determine the appropriate response.
5. **Intent Recognition**:
   * Based on the user's input, identify the user's intent, which could be to get information, make a reservation, ask a question, etc.
6. **Action Selection**:
   * Determine the appropriate action to take based on the recognized intent.
   * This might involve querying a database, providing information, connecting to a live agent, or executing specific functions within your application.
7. **Response Generation**:
   * Craft a clear and concise response to the user's input.
   * Personalize the response if you have user information.
   * Be sure to provide the information or action the user requested.
8. **Follow-up Questions**:
   * If more information is needed, ask follow-up questions to clarify the user's request.
9. **Error Handling**:
   * Handle situations where the chatbot doesn't understand the user's input or intent. Politely request clarification or offer alternatives.
10. **Task Completion**:
    * Once the user's request is fulfilled, confirm task completion or ask if there's anything else the user needs.
11. **Goodbye**:
    * End the conversation with a polite closing message.
    * Optionally, ask for feedback or invite the user to return in the future.
12. **Fallback Mechanism** (Optional):
    * If the chatbot can't understand the user's input or intent after multiple attempts, offer a more general service or transfer to a human agent.
13. **User Assistance**:
    * Provide assistance to users who are not sure how to interact with the chatbot. You can offer suggestions or a list of common commands.
14. **User Engagement** (Optional):
    * You can incorporate elements like greetings, jokes, or personalized content to make the conversation more engaging.
15. **Analytics and Learning**:
    * Collect data on user interactions to improve the chatbot's performance over time. Analyze user feedback and adjust the flow accordingly.
16. **Multimodal Support** (Optional):
    * Consider supporting multiple communication channels, such as text, voice, and visual interfaces, if applicable.
17. **Continuity**:
    * Ensure the chatbot can maintain context and continue the conversation seamlessly if the user leaves and returns later.

## Chatbot data

Let's get training data to chatdata.json in which the intent and its example questions are given. In which greet is intent and hi, hello are questions.

{  
 "greet": [  
 "hi",  
 "hello",  
 "hey",  
 "hola"  
 ],  
 "goodbye": [  
 "bye",  
 "goodbye",  
 "good bye"  
 ]  
}

The response is predefined in response.json in which the intent and response are given.

{  
 "greet": "hello there...!",  
 "goodbye": "bye see you later",  
}

## Conclusion

Throughout we have learned how simple chatbot is created using Random Forest. We learned an Intent-response based chatbot.

For fun, you can add funny conversations and this code to the web applications and create chatbot APIs.