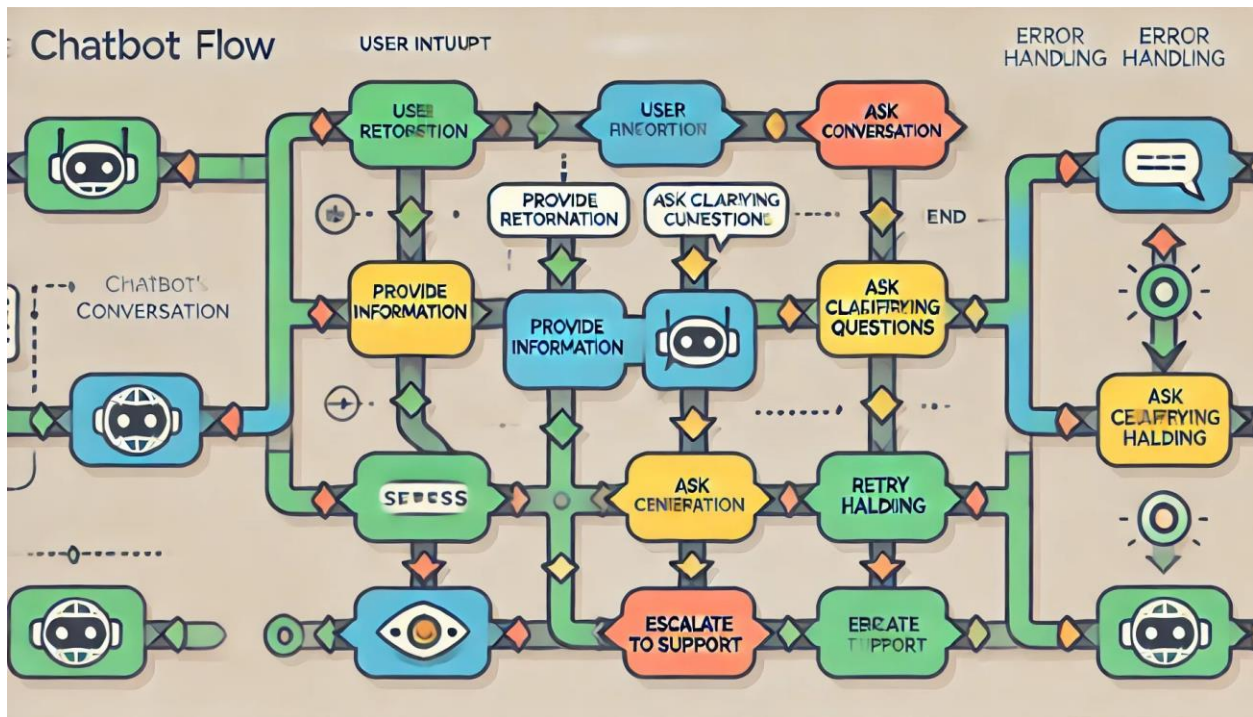


## [ WEEK -2 ( DAY-4 ) ]

### 2.A flowchart or state diagram illustrating the conversation flows.



Here is a graphical representation of the conversation flow for the chatbot.

### 3. Explanation of the Dialogue Management System

- **Architecture and Methodology:** A Framework for Dialogue Management

To enable structured and organized conversations, its dialogue management system is modular and state-based.

It centers on three key elements:

1. **Intent Mapping:** We have a General NLP model which is already trained to map user intent (e.g. product, troubleshooting, FAQs etc.)

2. **State Transition Management**: Conversation flow is controlled by a finite-state machine (FSM), communicating current state, and transitioning to new states based on user utterance, context and system action.
3. **Answer Generation**: Responses that are contextually relevant are generated from templates(rule-based)along with dynamic data providing a human-touch and an accurate response.

A user query will always initiate each conversation, and therefore must go through the intent recognition layer. After intent detection, FSM activates the state that goes in order of logic and is reasonable to go and not get stuck in a state useful only for error handling. The system writes for unsupported or ambiguous queries Assistant

## **Sample Conversations**

### **Normal Interaction**

**User:** "I need help with my bike's battery."

**Bot:** "Sure! Are you facing issues with charging or performance?"

**User:** "It's not charging."

**Bot:** "Got it. Please ensure the charger is connected securely and plugged into a functional outlet. Does this resolve the issue?"

**User:** "Yes, it works now. Thanks!"

**Bot:** "Happy to help! Let me know if you need anything else."

### **Edge Case - Ambiguous Input**

**User:** "Tell me about my bike."

**Bot:** "Could you clarify? Are you looking for information about your bike's specifications, features, or usage tips?"

**User:** "Specifications."

**Bot:** "Your bike is equipped with a 250W motor, a 48V battery, and a top speed of 25 km/h. Would you like more details?"

### **Edge Case - Unsupported Query**

**User:** "Can you suggest a cycling route in my area?"

**Bot:** "I'm sorry, but I currently don't provide cycling route recommendations. You might find useful suggestions on local cycling apps or forums. Can I assist you with something else?"

### **Edge Cases and Error Handling**

Edge cases must be handled to keep a smooth user experience. Here are the main strategies:

#### **Ambiguous Inputs:**

When intent is ambiguous, the bot will ask clarification questions.

For instance, if the user says, Tell me about my bike the bot will respond with Are you looking for specifications or features or usage Tips?

#### **Unsupported Queries:**

For any queries that the bot does not know about (for example, in the case of "Give me some cycling routes"), the bot will express its apology while redirecting to other suggestions and guiding users back into which features are supported within the dialog path.

#### **Repetitive User Inputs:**

When a user pings the bot with an already asked question, it searches for similar inputs.

If the same query was repeated, bot shows additional information or asks whether user wants help with anything else.

#### **Misunderstandings:**

If the bot misinterprets the intent, it includes a fallback state where users can manually select or confirm their intent.

#### **Error Recovery:**

When faced with unexpected inputs (e.g., typos or incomplete phrases), the bot utilizes fuzzy matching or regex-based parsing to approximate intent.

## **Improvements and Extensions**

The current system provides a structured and modular framework, but there is always room for growth:

1. **Advanced NLP Integration:**
  - a. Implementing a transformer-based model (like GPT or BERT) for better understanding of complex or nuanced queries.
  - b. Adding sentiment analysis to adapt responses based on user mood.
2. **Context Preservation:**
  - a. Enhancing the system's ability to maintain context across longer conversations or multiple sessions, allowing for seamless interactions.
3. **Multilingual Support:**
  - a. Extending language capabilities to serve a diverse user base.
  - b. Leveraging language translation APIs for real-time multilingual communication.
4. **Dynamic FAQs and Knowledge Base:**
  - a. Linking the bot to a live database for dynamic updates to product information or FAQs.
5. **Proactive Recommendations:**
  - a. Introducing proactive suggestions based on user behavior, past interactions, or context (e.g., recommending battery care tips after a troubleshooting query).
6. **Human-AI Collaboration:**
  - a. Improving escalation processes by seamlessly transferring conversation history to human agents when needed, ensuring continuity.
7. **Voice Integration:**
  - a. Adding voice-based interaction capabilities to enhance accessibility and user convenience.