**Use Case:**

**Support Copilot for Issue Lifecycle Management**

**1. Project Overview**

The goal is to develop an AI-powered Support Copilot that assists support executives throughout the entire issue lifecycle. The copilot will analyze incoming support issues, provide actionable insights, recommend communication templates, and monitor issue status to ensure timely resolution. The system will integrate seamlessly with existing support portals via API endpoints, ensuring real-time responsiveness (max 15 seconds delay).

**2. Functional Requirements**

**2.1 Issue Intake & Analysis**

* **Trigger:** When a new issue is raised on the support portal.
* **Actions:**
  + Retrieve the number of past issues raised by the same customer.
  + Search the historical issue database for similar issues and their resolutions.
  + Analyze the severity of the issue and assign a criticality tag: *Low*, *Normal*, or *High*.
  + Detect if any critical issues from the same customer or related product remain unattended for more than 24 hours and flag them.

**2.2 Support Executive Guidance**

* Generate recommended message templates for each incoming user message.
* Recommendations should consider:
  + The issue description.
  + Customer history.
  + Similar past issues and their resolutions.
  + Severity and criticality tags.

**2.3 Conversation Summarization**

* Automatically generate a concise summary of the entire conversation for historical review and knowledge base updates.

**2.4 Integration & Performance**

* All functionalities must be exposed through RESTful API endpoints.
* API response time must not exceed 15 seconds to ensure real-time assistance.
* Solution should be hosted in cloud preferably over AWS.

**3. Non-Functional Requirements**

* **Scalability:** Support increasing number of users and issues without performance degradation.
* **Reliability:** Ensure high availability and fault tolerance.
* **Security:** Secure APIs and data storage to protect customer and issue data.
* **Maintainability:** Modular architecture to facilitate updates and improvements.
* **Compliance:** Adhere to relevant data privacy laws (e.g., GDPR).

**4. Deliverables:**

* **Technical Architecture:** Detailed architecture diagram and data flow diagram of the whole solution.
* **Explanation:** Reason behind each component used and process details.
* **Code Snippets:** To support any process you can add Python code and MySQL queries. Will help in understanding the coding capabilities.

**Note:**Please feel free to use the Large Language Model (LLM) as a reference; however, avoid directly copying and pasting solutions from it, as such actions can be detected.