Task 3. Improvements

1) Errors and limitation of the app

During the development and testing of the application, several errors and limitations became evident as reported as follows:

Errors identified

- Inaccurate speech recognition. The application occasionally misinterpreted the user's responses, especially in the presence of background noise or when the utterance did not match the expected grammar;
- Delayed reaction when speech is not recognized. Sometimes the user speaks, but the system does not immediately detect that the input was not understood. Instead of triggering a no match or no input response right away, the system remains in the listening state for too long. This creates confusion for the user, who may think the system is not responding or malfunctioning;
- Inconsistent handling of "no input" and "no match" events. Events such as ASR_NOINPUT and ASR_NOMATCH were handled differently across multiple states, and in some cases repeated unnecessarily, leading to unpredictable behavior and duplicated logic;
- Incorrect assignment of recognized values (dates, times, yes/no answers). Some invalid inputs were still being stored in the context, causing the system to enter an inconsistent or incomplete state and preventing the conversation from progressing correctly.

Limitations identified

- Limited grammar and vocabulary. The recognizer only accepts specific keywords. Any variation beyond the predefined grammar results in an error or requires the user to repeat their answer;
- Basic contextual understanding. The system cannot understand more complex or indirect responses (e.g., "Maybe in the afternoon"), limiting the naturalness of the interaction;
- State structure that is not easily scalable. The presence of many similar states creates redundancy and makes it more difficult to expand the logic without introducing more duplicated transitions or inconsistencies.

2) Some solutions to the errors and limitation found

Several adjustments were made to reduce the errors as reported as follows:

- Centralized error handling. The ASR_NOINPUT and ASR_NOMATCH events were moved from the individual "Ask" states to their parent states. This reduced code repetition and ensured a more coherent and predictable handling of recognition errors;
- Better input validation. Additional helper functions (such as isYes, isNo, getTime, and getPerson) were used more consistently to validate user responses. This prevented invalid values from being saved in the context;
- Clearer and more targeted prompts. Error messages were simplified and made more specific ("Please answer yes or no", "Try Monday, Tuesday…") to help users respond in a format that the system can correctly recognize;

- Reduction of duplicated logic. By refactoring the states and relocating repeated transitions, the application now behaves more consistently and has fewer unexpected transitions.