

# Dialog systems- Lab3

February 2026

**Question1:** Write a report (max 1 page) which describes errors and limitation of your app.

**Answer:**

## Application Overview

The application is a voice-driven meeting Booking Application (scheduler) built with XState v5 and SpeechState 2.15.0, using Microsoft Azure Cognitive Services for text-to-speech (TTS) and automatic speech recognition (ASR). The user is guided through a spoken dialogue to select a person, day, and time for a meeting, followed by a confirmation step.

## 1 Errors and Limitations

### 1.1 Rigid Grammar-Based NLU

The system uses a flat dictionary lookup. Only exact single-word matches are recognised (e.g., “monday”, “tom”). Natural phrases such as “I’d like to meet Tom” are rejected and the system re-prompts the user.

### 1.2 ASR Mishearing Short Words

The Web Speech API frequently mishears short words in isolation. For example, “monday” was transcribed as “Day” or “Play”, causing unnecessary re-prompts. Short, context-free words are harder for ASR engines to recognise accurately.

### 1.3 SPEAK\_COMPLETE Timing Race Condition

A stray SPEAK\_COMPLETE event from a previous state sometimes arrived after the machine had already transitioned to a Listen state, causing states like `AskDay` to fire their speak entry multiple times before moving to `ListenDay`.

## 1.4 NOINPUT Event Name Mismatch

The original code handled `NO_INPUT` (with underscore), but `SpeechState` fires `NOINPUT` (no underscore). This caused the machine to freeze whenever the ASR timed out, requiring a page refresh to recover.

## 1.5 No Error Recovery for Azure Failures

If Azure TTS or ASR fails mid-conversation (e.g., token expiry or network error), the machine silently gets stuck with no user feedback and no automatic retry mechanism.

## 1.6 No Visual Feedback During Dialogue

The UI shows only a single button with no indication of what the system expects. There is no transcript display, no waveform, and no visual cue when the microphone is active.

## 1.7 Silent Rejection of Unrecognised Words

When the user says a word not present in the grammar, the system simply re-asks the same question without explaining why the input was rejected. The user receives no feedback about what went wrong or what valid words exist. For example, saying “saturday” (not in grammar) causes the system to silently repeat “Which day works for you?” — confusing and unhelpful for users who do not know the valid vocabulary.

**Question 2:** Fix a couple of them and briefly describe your solution in the report. You don’t have to fix all the limitations.

**Answer:**

## Fixes Implemented

### Fix 1: Corrected NOINPUT Event Name

All Listen states were updated to handle `NOINPUT` (not `NO_INPUT`). The machine now correctly loops back and re-asks the question when the ASR times out instead of freezing.

### Fix 2: Resolved SPEAK\_COMPLETE Race Condition

Added `SPEAK_COMPLETE: undefined` in all Listen states to discard late-arriving events. A 500ms `setTimeout` delay was also introduced before each `SPEAK` command to give the speech actor time to settle before receiving the next instruction.

### **Fix 3: Expanded Grammar with Common Variations**

The grammar was extended with additional entries for common ASR mishearings and natural variations (e.g., “yeah”, “yep”, “correct” for confirmation; “fri”, “mon” as shorthand for days; “noon” for twelve o’clock). This reduces the rejection rate without requiring a full NLU pipeline.

### **Fix 4: Button State Reflects System Readiness**

The Start button is now disabled and labelled “Initializing...” until `ASRTTS_READY` is received. This prevents users from clicking before the speech system is ready, which previously caused the `CLICK` event to be silently dropped.

### **Fix 5: Spoken Feedback for Unrecognised Words**

To address Limitation 1.7, the fallback transitions in each Listen state were updated to first speak an informative error message before re-asking the question. When a recognised utterance is not found in the grammar, the system now speaks a helpful rejection message and then repeats the prompt.

#### **Implementation Example for the Person Slot:**

- `ListenPerson` (`unmatched`) → `NotRecognisedPerson`
- `NotRecognisedPerson` speaks: “Sorry, I did not recognise that name. Please say vlad, bora, tal, or tom.”
- On `SPEAK_COMPLETE` → `AskPerson` (re-asks the question)

The same pattern applies to day, time, and confirmation slots, each with slot-specific hint words so the user always knows exactly what to say.

## **Remaining Known Issues**

- The core NLU limitation (grammar-only matching) remains. A proper solution would integrate an LLM-based intent classifier or a flexible slot-filling approach.
- Azure service failure recovery is not fully implemented.
- The UI lacks rich feedback, such as microphone activity indicators or a transcript display, which are left as future improvements.