**REPORT – lab 2 - Basic dialogue management – "the appointment"**

*Write a report (max 1 page) which describes errors and limitation of your app.Fix a couple of them and briefly describe your solution in the report. You don't have to fix all the limitations.*

Limitations:

1) If the user say some extra word and doesn't say exactly the same word stored in the grammar, the code will stop and it will ask to repeat recursevily in loop.
   a. user cannot say "ON" or "AT" when refers to days and times and cannot specify different times from 1, 2, 3 etc. So I fixed them: (FIX NUMBER 1)

```
// Days
  monday: { day: "Monday" },
  "on monday": { day: "Monday" },  // ✅ Synonym

  // Times
  "0": { time: "00:00" },
  "at 0": { time: "00:00" },

  "midnight": { time: "00:00" },
  "at midnight": { time: "00:00" },

  "1": { time: "1 AM" },
  "at 1": { time: "1 AM" },

// Half-hour times
  "0:30": { time: "00:30" },
  "1:30": { time: "01:30" },
  "2:30": { time: "02:30" },
```

With this new implementation there could be a grammatical syntax error at the end. Machine could say "Do you want to create me an appointment with Marco on ***ON MONDAY*** ... etc. ? So I fixed also this: (FIX NUMBER 2)

```
          entry: {
            type: "spst.speak",
            params: ({ context }) => ({

              utterance: `Do you want me to create an appointment with
${getPerson(context.person?.[0]?.utterance)} on
${getDay(context.day?.[0]?.utterance)} at ${getTime(context.time?.[0]?.utterance)}
?`,

            }),
          },
```

   b. it is very bad to understand very short words (like No) and pronunciation is fundamental (e.g. if you try to say “Pippo”, it always understand “People”, also because is set English language).
   c. in general, in debugging, I noticed that the longer is the phrase the easier is the error of misunderpretation. This is due not only to different interpretation of what the user says, but also on the fact that when you speak, machine decides where to put the punctuation like “.” “,” and this can lead to “not in the grammar case”. A better implementation would be to remove the punctuation.

2) If the user say a wrong word and the machine recognises another, you cannot change it anymore (if the word was in the grammar) and the process goes on.
3) If the user has changed his mind and wants to go back to the previous step, she/he is not allowed
   a. user cannot cancel the appointment
   b. user cannot go back
4) User/speaker cannot add new people, times, dates etc. in the grammar list (this can be done only by operating on the code by hand)
5) User cannot have at the same time contemporarely an appointment with more of one person. Same for days etc.
6) Everytime machine says if the word exists in the grammar or not (if it has correspondence). This can be annoying and all the process in general is very slow. It can be improved a lot and also shortcuts could be added.
7) It overwrites everytime all the appointments (all the data)