This app is a finite state machine, and so the app has almost all initiative in the dialogue which does not closely mimic general human conversation, which might be an assumed goal of the ideal meet-booking app.
We could give the user some more control of the dialogue flow by adding states and prompt the user at more stages of the dialogue. This would make the dialogue slightly more mixed-initiative which would make it closer to a natural conversation than previously.

Another issue is that the user can only give information when prompted, so the dialogue is quite rigid. Based on my understanding of xstate it would not be possible to turn this finite state machine into a slot-filling machine where the user can answer several questions that have not been asked yet, so the rigidity of the dialogue is not possible to solve.

The turn taking is rigid as well, the user cannot give information until asked and anything said outside of the given “speaking slots” in the app will be ignored. This is not solvable within this type of system as far as I am aware. Other limitations include the fact that the users cannot give more information than asked at any one point, or ask counter questions.

Except for the intents part of the app, the grammar is fixed and so only takes certain strings as input, the app cannot extract the intent of the user. This could be solved by adding NLU intents and choosing steps based on the intent instead of the grammar, but doing this would mean recreating each transition, so I will skip this step.

The improvements I have made are the following:
In most states where a question is asked, I will add the option of returning to the previous state when the user utters anything in the newly created goBackGrammar by uttering words like “previous” or “go back”.

In this way the dialogue is moved in a certain direction but with more input from the user than if for example the user were to say some name wrong, they can stop the machine, go back and repeat the previous step so the user do not have to go through the entire state chart when they know that it has gone wrong somewhere.

Another issue is that the persons, times and days available for booking are not part of the information available to the user, it would be preferable to have that information available. I created a function that will tell the user what their options are so that the user can better decide when and who to meet.

The speech recognition itself is not great, but is not really related to the construction of the app, I could make a slight improvement by adding “yes” and “no” to intents instead of a grammar, but it probably would not make a big difference since the various common ways there is to say “yes” or “no” is not that many, compared to the different ways to request a meeting. To slightly improve the speech recognition, I extended the grammar to include some phrases like “at seven” “seven” and “seven o clock” for recognizing the time of day.