

## Lab Report:

### -Limitation 1:

My hour system can be improved, primitive solution can be to assign lots of grammar to cover all the possible combinations to talk about a certain hour. A better solution would be to import an ASR model to cover these utterances and capture the hour system. There is probably a more efficient solution since this limitation is not very complex. Regardless, as I will explain in the following bullet points, I find limited but clear & concise grammar to be more effective in systems that aim for a service and quality of life boost; such as these kind of appointment systems. Transparency about the functioning of the system in such services are quite important in my opinion. People not being aware that what system can do is one of the most probable reasons why voice based systems are way unpopular than text based systems.

### -Limitation 2:

Which is somewhat related to limitation 1 in a sense that, unusual utterances and performance products of the language tends to lead to an unsuccessful ASR. Such as uttering a certain word; a person that has a phonetic inventory of an unrelated language to English (or to the target language) consistently cause a high WER. I am very curious about an efficient solution to this problem. But for a certain model that has a certain purpose with a certain grammar, such as creating appointments and recognizing the speech related to that; I would define various utterances for that limited grammar to minimize WER. As a temporary solution, strangely, changing the TTS voice to a Swedish speaker showed an increasing success on recognising names such as Christine and Staffan. (Not sure how default TTS voice is related to ASR but maybe it was just my experience, not facts.)

-Limitation 3: In my opinion, informing the user about the optimal usage of the system is very important for overall satisfaction of the dialogue system. Therefore creating variables to fetch all the possible utterances in different categories in the grammar, then `spst.speak` with that variable makes the program more sustainable since addition and removal of possible time slots / scholars and in a more realistic setting: filled time slots and removal of the filled combinations are some of the realistic opinions to make the system work.

In addition, storing the already made combinations to inform the user that "this appointment is not available" can be a good idea. If the system is going to be used from different devices but not via cloud or a website domain, syncing the aforementioned combinations' variable to a cloud system may be a good idea.

This system can be integrated to make the outcome of the appointment details into a .csv file then via JavaScript, importing them to a cloud calendar service may be realistically useful.

-Limitation 3: This is a quite big one and my coding skills on TypeScript are not good enough to make the code as unrepeatitive as possible. That is, I would have to over-utilize AGI/AI services, good ones, which would lead me to use their cloud services that I avoid if there is no other option due to ethical and cyber security reasons. A good solution would be setting the structure in a way that the child states would refer to each other and similar structured states that are similar in form may be coded beneath each other. My coding here is not efficient at all and making the structure one big complex state and making the states refer to each other in an efficient way could be the answer but I have tried and failed to do that, especially in the `CheckGrammar` states. Keeping the readability of the code but also making it efficient was not in my capability for this project.

-Limitation 4: I could have implemented substates to check if the user has meant to say a certain word, which I find very lacking in popular voice assistants today (maybe they do not prefer to do it to make the system execute much faster -but inaccurate-). Especially if the project aims to interact with executables in a system. But for this project, I have overcome this -rather found it useless- since our grammar is very limited and similar words and utterances are non-existent. Grammar check states are good enough for our purpose for now.