The obvious limitations of my app are of course the minimized grammar. One can always add more grammar to the machine but it will never cover all the possibilities a user might utter to show affirmation or negation. This alone shows again how humans still expect to interact differently with a program than with another human. One main issue was that the program didn't understand my words due to accent or something else. I assumed that it also was connected to the more difficult names I chose that had a lot of vowels with many possible pronunciations. My quick fix for testing the program was adding a simpler name, here 'Andreas'. Many testings later I managed to pronounce most names in a machine-understandable manner.
I started off by creating states for person, day, time and confirmation adding for each more states like xxxAsk and xxxCheckGrammar. Copying my states over and over again the code became very long and I decided to try to put it all in a parent state. This was a very troublesome process. For the longest time my machine got stuck after recognizing the person and didn't move to the state of asking the time. In the end the problem was that I used functions like 'entry' and 'always' that caused my machine to move on without waiting for the event to end. Debugging this cost me a lot of time which is why there are still some missing features in my machine:

1. If I decline at the confirmation, the question just gets repeated instead of choosing a different way (either just a goodbye message or a return to the beginning).

2. I didn't work on the style sheet at all so the look of my app doesn't fit its purpose.

3. There are surely ways to improve the understanding abilities of the machine, here I could try to play around with the settings made in const settings.