FINAL REPORT

‘TALK SLEEPY TO ME’

GROUP JKAY (GROUP 3) BREAKDOWN:

- Yining Chen: 25%
- Andrew Huynh: 30%
- Kelly Lee: 15%
- Josh Lim: 30%

PROBLEM & SOLUTION OVERVIEW

The problem that we are addressing with ‘Talk Sleepy To Me’ is the inability of users to go to sleep at the time they want/need to at night. Although many of the functions that can help users are available through GUIs of various applications, the light from the screens actually can make it more difficult for users to sleep. ‘Talk Sleepy To Me’ offers options for listening to music and white noise, as well as stepping through breathing exercises, and offers the user the option of journaling. This is all accessed through voice, so it eliminates the problem of the screen light. If the user at any point wants to access their ‘favorites’ or transcribed journal entries, then they can go to the GUI.

TASKS

1. WRITING A JOURNAL ENTRY

- The user can tell Alexa what they did during the day, and the journal entry will be added as a value in a table of journal entries, with the key being the date. They can signal this intent by saying, “Write journal,” “write in journal,” “write journal entry,” “write a journal entry,” etc. The sentences are processed by Alexa one at a time, and each sentence can be up to 10 to 12 words.

2. READING A JOURNAL ENTRY

- The user can have Alexa read them a previously written journal entry. Upon saying “read journal on [date],” “read on [date],” “read from [date],” etc. Alexa will take the

value at the date key and read it back to the user. The user can also access their transcribed journal entry on the GUI.

3. EXERCISES

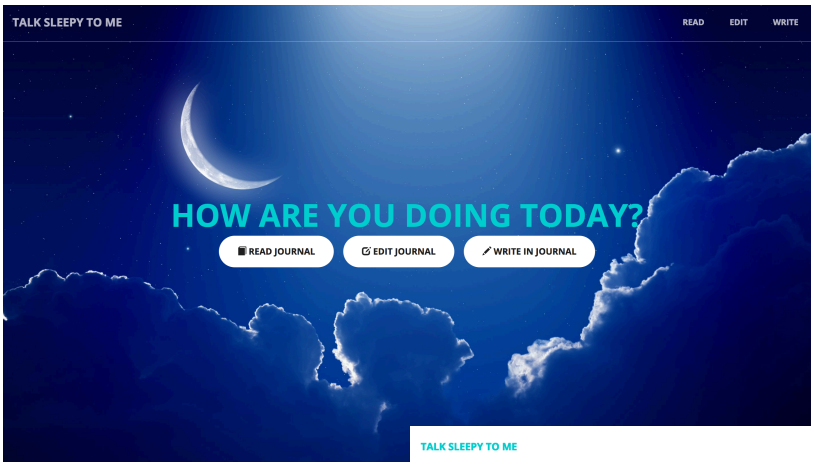
- The user can go through breathing exercises (there are 4 that have been implemented), in which they are stepped through the exercises, saying “I’m done,” when they are ready for the next step. There are also anti-snoring exercises available, and the user can differentiate between these upon intent.

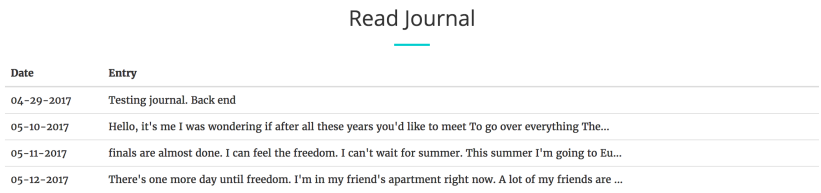
VIDEO

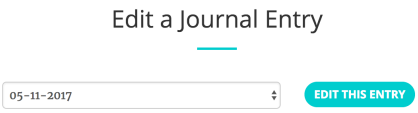
Link:

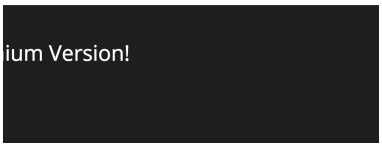
https://youtu.be/CKoecjKgeAA

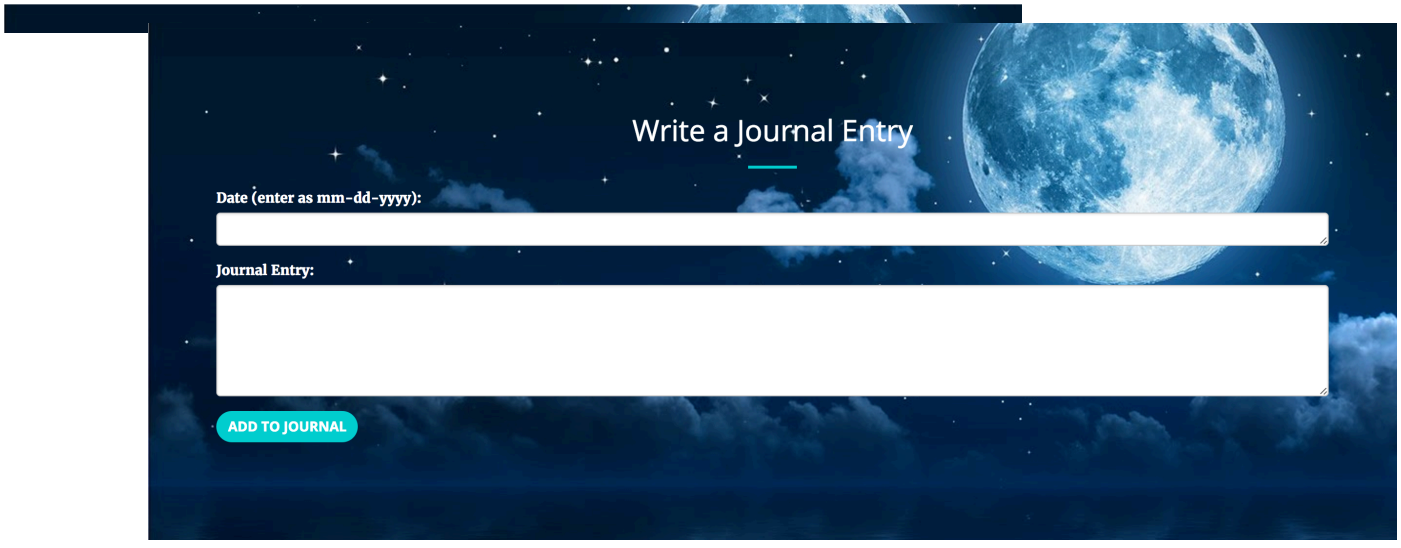
REVISED INTERFACE DESIGN











The reason we switched from a mobile UI to a web UI did not change with our final iteration. The idea behind the app is encourage the user not to use her phone during bedtime. Having a mobile UI felt like it was going against that idea, so we switched it to a web interface. As we worked on the web interface, our reasons for switching were further proven to be a good idea. The purpose of having the GUI is not to help the user fall asleep, but an extra function that lets the user look back over their entries if they wanted to see what they wrote.

With our final iteration, we worked on making the themes the GUI more consistent across the interface used for each task, as well as making it a little more minimalistic and less complicated in terms of color scheme. We incorporated feedback from the peer critiques and the class presentation to do this. The user is also now able to edit their journal entry on the web GUI with the final iteration, and the journal entry is able to be stored in a table in DynamoDB.

We didn't change much of what we found during our low-fi prototype testing and interviews; we saw that the people seemed to like the idea of journaling more than the other parts of our app. Our web interface is still solely dedicated to the journaling part of our app. Instead of having the web UI be a tool for the VUI (configuring settings, adding music), we revamped it to become a parallel app. On the web interface, we have three main actions the user can do - read, write, or edit journal entries.

Another change that we made was separating the reading and editing of journal entries. Since we changed the idea of the web UI from something that just configured settings to an extension of the VUI, we decided to make editing journal entries a completely different action than reading. Originally (see the figma screenshot), the read and edit options of the app were on the same page. Now, they're different sections completely on the web UI.

Wizard of Oz: For the web interface, all the functionalities regarding the journal entries were WoZed. They are now connected to a database and the entries can be edited, written, and saved through the Web UI.

PROTOTYPE OVERVIEW

For the prototype for the VUI, we decided to focus on the writing and reading of the journal entries as well as the breathing exercises. Most of the main functionality is changed, but the VUI no longer has implemented playing music and white noises, because we were not sure how to access the music in our code. As this was not one of the main tasks, we decided it might be easier to just get rid of it entirely for now; it is something we can work towards in the future.

The user starts off by saying "Talk Sleepy To Me," or "Open 'Talk Sleepy To Me,'" and is then met with the welcome message, which says, "Welcome to Talk Sleepy To Me. What do you want to do?"

The user can then decide to do any of the available tasks (writing journal entries, reading journal entries, breathing or anti-snoring exercises), or if they don't know what to do, they can say, "What can I say?" upon which Alexa will give them a list of the things they can do.

If the user says, for example, "I want to journal," the help me intent will be instantiated, and Alexa will say the help me message specifically for journaling, which is, "Okay. Add to your journal one sentence at a time. Say, 'new sentence' before every sentence and say 'finish entry' to finish journaling. The journaling task is decidedly limited in this way, because then the journaling intents can only be instantiated by the utterances that we specify. Right now, the kinds of utterances that the app has includes "awesome," "bad," "today was horrible," "today was my birthday," "I had eggs for breakfast," "I finished all my work today," "I went to the amusement park with my friends today," etc. The user would have to say, "New sentence" before saying their actual sentence. For example, they would say, "New sentence I had a horrible day." Alexa then says, "Got it. Remember to say, 'Finish journaling.'" The journaling continues in this manner until the user says, "Finish journaling," at which point the prompt, "Okay. Would you like to continue using the app?" comes up. Depending on the user's answer, they are taken either back to the "main menu" or they receive a "Ok. Good night" response from Alexa.

The user can also say, "I want to read a journal entry," upon which Alexa says, "Alright. What journal entry do you want to read from?" Here, the user specifies a date, which we used the AMAZON.DATE intent for. If they say "read from yesterday" and the date is February 20, then Alexa says, "Alright. Reading journal from February 19. I had a horrible day." It automatically goes into the

prompt for continuing to use the app afterwards: "Would you like to continue using the app?"

The user can also decide to do either breathing or anti-snoring exercises. They say, "I want to do a breathing exercise." Alexa then responds, "Okay. Let's do a breathing exercise. Say, 'Next step,' to start." As the user continues to say 'Next step,' Alexa will walk them through the steps of one of the breathing exercises we have programmed in (deep breathing exercise, a 4-7-8 exercise, etc.). At the end of the last step, Alexa says, "There are no more steps. Would you like to continue using the app?"

We decided against implementing the music and noise function for this prototype, as actually being able to implement this function is probably going to take much longer. We would have to figure out how the user would access the music that they want to listen to.

Regarding the journaling function, our group's original idea was to make a variable intent of about 100 words, meaning that the user would only be able to journal about 100 words at a time. This was changed to the idea of each intent being a sentence, and with this iteration the maximum is 15 words, and the user journals one sentence at a time – all of these sentences together make up the journal entry. This can be somewhat annoying for the user, because they are unable to journal according to their own thought processes. Alexa stops them after every sentence to say, "Got it. Remember to say, 'finish journaling,'" which interferes with the user's desire to use the application. We chose to implement it in this way, however, because the user would not have to abruptly be cut off in the middle of a journal entry that exceeded 100 words, and they would not have to keep track of each of those 100 words while journaling, or keep that in mind. Coming up with the utterances for a variable length intent would also not be feasible.

No wizard-of-Oz methods were needed for our functionality.

The only library software we used was Amazon's provided DATE intent, AMAZON.DATE, which supports the following:

- Utterances that map to a *specific date* (such as "today", or "november twenty-fifth") convert to a complete date: `2015-11-25`. Note that this defaults to dates *on or after the current date* (see below for more examples).
- Utterances that map to just a *specific week* (such as "this week" or "next week"), convert a date indicating the week number: `2015-W49`.

- Utterances that map to the *weekend for a specific week* (such as "this weekend") convert to a date indicating the week number and weekend: `2015-W49-WE`.
- Utterances that map to a month, but not a specific day (such as "next month", or "december") convert to a date with just the year and month: `2015-12`.
- Utterances that map to a year (such as "next year") convert to a date containing just the year: `2016`.
- Utterances that map to a decade convert to a date indicating the decade: `201X`.
- Utterances that map to a *season* (such as "next winter") convert to a date with the year and a season indicator: *winter*: `WI`, *spring*: `SP`, *summer*: `SU`, *fall*: `FA`)
- The utterance "now" resolves to the indicator `PRESENT_REF` rather than a specific date or time.