

# CS CAPSTONE DESIGN DOCUMENT

MAY 30, 2020

## **BUILD A VOICE-DRIVEN APPLICATION FOR OPTIMIZING TURN-OVERS AT AIRBNBS**

PREPARED BY

**GROUP 6**

**VIRTUAL CONCIERGE**

**TSO-YI CHUNG, KENNY LUONG, CARLOS LOPEZ-MOLINA, SUMEGHA ARYAL**

### **Abstract**

In this Project archive document, you will be able to go through all the components within the project. From project requirement to code display. This document will go over the design of the system and the steps taken to achieve our goal. It contains group 6's one year worth of information for our senior project.

## CONTENTS

<b>I</b>	<b>Introduction</b>	<b>4</b>
<b>II</b>	<b>Requirements document</b>	<b>5</b>
II-A	Project Overview . . . . .	5
II-A1	Purpose . . . . .	5
II-A2	Scope . . . . .	5
II-A3	Definitions . . . . .	5
II-B	Project Description . . . . .	5
II-C	Specific Requirements . . . . .	5
II-C1	Software Application . . . . .	5
II-D	Gantt Chart . . . . .	6
<b>III</b>	<b>Design Document</b>	<b>7</b>
III-A	Glossary . . . . .	7
III-B	User Interface . . . . .	7
III-B1	Visual . . . . .	7
III-B2	Reporting . . . . .	7
III-B3	Voice . . . . .	8
III-C	Verbal User Interface . . . . .	8
III-C1	Alexa Skills . . . . .	8
III-C2	Utterances . . . . .	8
III-D	Hardware Component . . . . .	9
III-E	Software Component . . . . .	10
III-F	Conclusion . . . . .	11
III-G	Modification . . . . .	11
<b>IV</b>	<b>Tech Review (Carlos)</b>	<b>13</b>
IV-A	Introduction . . . . .	13
IV-B	Personal responsibilities . . . . .	13
IV-B1	Testing . . . . .	13
IV-B2	Certification and Publication . . . . .	13
IV-B3	Notifications . . . . .	13
<b>V</b>	<b>Tech Review (Kenny)</b>	<b>15</b>
V-A	Introduction . . . . .	15
V-B	Personal Role . . . . .	15
V-C	Command Accuracy . . . . .	15
V-C1	Description . . . . .	15

	V-C2	Choices . . . . .	15
	V-C3	Amazon Alexa . . . . .	15
	V-C4	Google Assistant . . . . .	15
	V-C5	Siri . . . . .	15
	V-C6	Comparison . . . . .	15
	V-C7	Final Choice . . . . .	16
V-D		Device Compatibility . . . . .	16
	V-D1	Description . . . . .	16
	V-D2	Choices . . . . .	16
	V-D3	Amazon Alexa . . . . .	16
	V-D4	Google Assistant . . . . .	16
	V-D5	Siri . . . . .	16
	V-D6	Comparison . . . . .	16
	V-D7	Final Choice . . . . .	16
V-E		Application/Skill Support . . . . .	16
	V-E1	Description . . . . .	16
	V-E2	Choices . . . . .	17
	V-E3	Amazon Alexa . . . . .	17
	V-E4	Google Assistant . . . . .	17
	V-E5	Siri . . . . .	17
	V-E6	Comparison . . . . .	17
	V-E7	Final Choice . . . . .	17
<b>VI</b>		<b>Tech Review (Tso-Yi)</b>	<b>18</b>
	VI-A	Goal . . . . .	18
	VI-B	Language support . . . . .	18
		VI-B1 Final decision . . . . .	18
	VI-C	Data Capture . . . . .	18
		VI-C1 Final decision . . . . .	19
	VI-D	Organization of UI . . . . .	19
		VI-D1 Final decision . . . . .	20
<b>VII</b>		<b>Tech Review (Sumegha)</b>	<b>21</b>
	VII-A	Problem to Solve . . . . .	21
	VII-B	Responsibilities for the Project . . . . .	21
		VII-B1 Data Storage . . . . .	21
		VII-B2 Technologies for Data Storage . . . . .	21
		VII-B3 Processing of Data for User Response . . . . .	21
		VII-B4 Technologies for Processing Data . . . . .	21

VII-B5	Toolkit for UI . . . . .	22
VII-B6	Technologies for an UI Toolkit . . . . .	22
<b>VIII</b>	<b>Weekly Blog (Carlos)</b>	<b>23</b>
<b>IX</b>	<b>Weekly Blog (Kenny)</b>	<b>25</b>
<b>X</b>	<b>Weekly Blog (Tso-Yi)</b>	<b>27</b>
<b>XI</b>	<b>Weekly Blog (Sumegha)</b>	<b>30</b>
<b>XII</b>	<b>Final Poster</b>	<b>32</b>
<b>XIII</b>	<b>Project Documentation</b>	<b>33</b>
XIII-A	Installation Guide . . . . .	33
XIII-A1	How to Install . . . . .	33
XIII-A2	How to Run and Test . . . . .	33
<b>XIV</b>	<b>Recommended Technical Resource</b>	<b>34</b>
<b>XV</b>	<b>Conclusions and Reflections (Carlos)</b>	<b>35</b>
<b>XVI</b>	<b>Conclusions and Reflections (Kenny)</b>	<b>36</b>
<b>XVII</b>	<b>Conclusions and Reflections (Tso-Yi)</b>	<b>37</b>
<b>XVIII</b>	<b>Conclusions and Reflections (Sumegha)-Project archive</b>	<b>38</b>
<b>XIX</b>	<b>Appendix 1 (Essential Code Listings)</b>	<b>39</b>
<b>XX</b>	<b>Appendix 2 (Code Review feedback)</b>	<b>42</b>

## I. INTRODUCTION

The goal of this project is to create a voice-driven application that will help optimize turn-overs for home renters. In order to optimize turn-overs, the main target of this voice application will be home cleaning services and home owners. Cleaners will be able to utilize the application to speed up the cleaning process and offload important information. Design stakeholders include the project team members, Dana Young, and our senior capstone instructors who have an interest in the project and influence over the success. Dana Young is a supervisor for our project, he did not participate in development.

The members of our team are Sumegha Aryal, Tso-Yi (Joe) Chung, Carlos Lopez Molina, and Kenny Luong. All of us are seniors at Oregon State University. We divided the project into three components: Alexa skillset, storage, and communication.

Kenny and Tso-Yi are in charge of the Alexa skill component of the project. The focus is to create a voice recognition skill that is compatible to Amazon Alexa smart devices. The skills will allow cleaners to report their cleaning progress for each room, report any damages found in any room, and report any maintenance needed to the homeowner. All these information will be sent to the homeowner once the cleaning session is completed.

Sumegha is in charge of the storage component of the project. S3 is chosen as it allows us to keep functionality within the Amazon applications/features offered. All data is stored in the cloud and is cost efficient since you only pay when it is running.

Carlos is in charge of the communication between data captured by the Alexa device and the homeowner. He set up the AWS Lambda function and connected all of the appropriate services and users needed to work on the project. He worked on connecting Twilio services needed to allow our skill to send an SMS message. As for emails, he set up the Amazon Simple Email Service (SES) used due to its cost-effective service in sending out and receiving email. It also eliminates the complexity and cost of building an in-house email solution or licensing. In addition to setting up these services he also worked on making sure the user has the correct permissions set to use the application. This involved looking over the guidelines Amazon sets in order for their skills to be allowed to be placed on the app store. So he added that logic and made sure it was ready with all of the requirements set up. In the end he worked on some storage as another team member was not able to complete it.

The Alexa skill and communication components of the project are completed and verified by the client. The storage has not been implemented. The team that pick up this project next school year will need to implement storage system either using dynamoDB or other storage choices they choose. Other than that it would be ready to go live and used by home owners and cleaning services.

## II. REQUIREMENTS DOCUMENT

### A. Project Overview

1) *Purpose* : The goal of this project is to create a voice application that allows hosts of rental accommodation to communicate with their housekeepers in an efficient manner. Alternative accommodations such as Airbnb and vacation rentals are on the rise along with voice technology. The primary interest for this project is to develop a technology integrated hospitality experience that will help hosts provide a more seamless experience for both the guests and housekeepers.

2) *Scope* : The scope of the project will require software programming and additional research on the cleaning services that work together with Airbnb hosts. Research will involve understanding if voice applications will help cleaners in both short and long runs, do they understand voice technology and how to operate it, and if they are comfortable using an English version of the voice application. Information will also be gathered from Airbnb hosts. This includes whether or not they have a smart home device, will they use software application to increase turnover efficiency, and are they comfortable with using voice-driven application.

Software-wise, we plan on using the Alexa skills kit provided by Amazon and incorporate it within our application. We will most likely follow VCS's existing application and build it using node.js.

### 3) *Definitions* :

- Alexa Skills Kit: A collection of APIs, tools, and documentation for giving Alexa new capabilities
- Voice User Interface (VUI): A method for people to use voice input to interact with and control computers and devices
- Wake word: A command that the customer says to tell Alexa that they want to talk to her

### B. Project Description

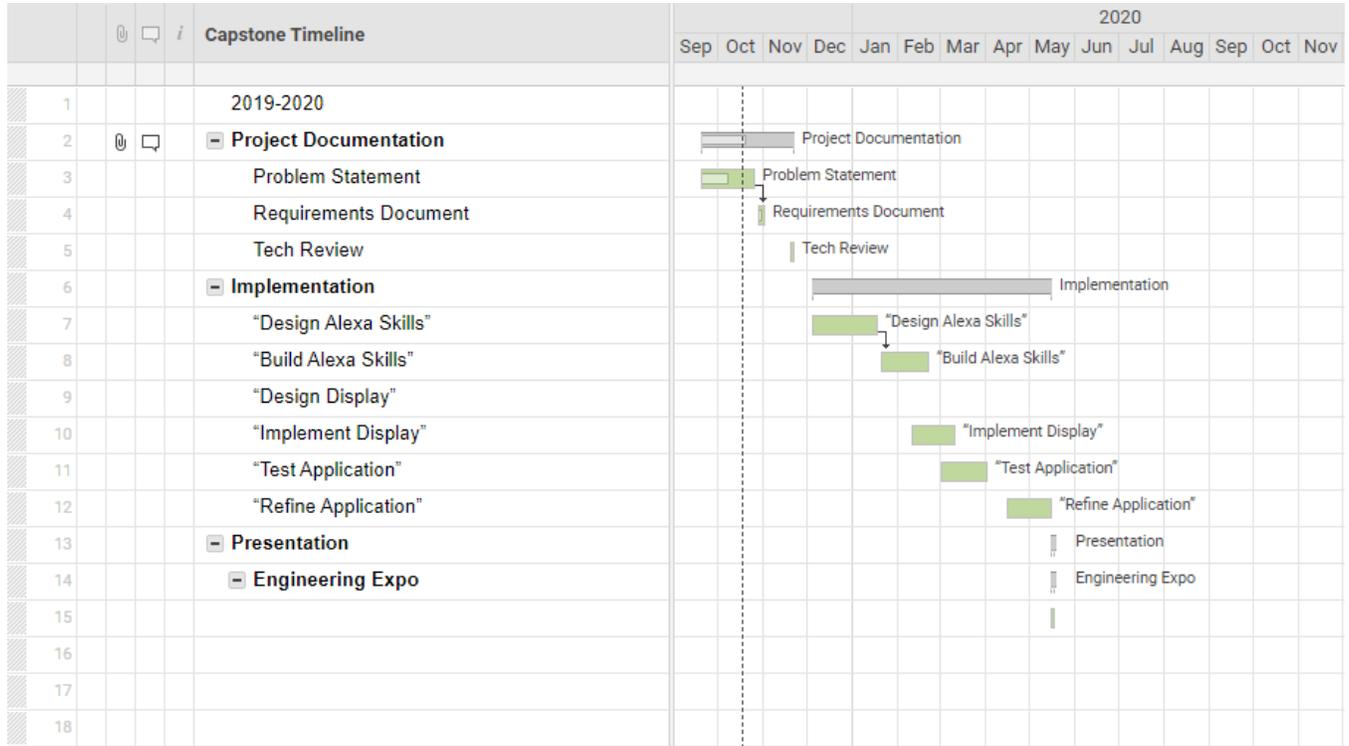
The main focus of this project is to design a voice user interface (VUI) that is able to help both the cleaners and the hosts of alternative accommodations to achieve a higher turnover efficiency. To do so, utilization of Alexa and the Alexa Skills Kit are essential in creating a VUI that is both intuitive and easy to use. New skills will be created for Alexa to understand the commands listed below and more as development progresses. As cleaners go through the house to clean, rather than having to stop to input progress and/or notes onto paper or a mobile device, they can instead tell Alexa this information as they continue to work. This provides a more efficient method of documentation.

### C. Specific Requirements

#### 1) *Software Application* :

- Develop a standalone voice application with Alexa Voice Service (AVS) using the wake word "Alexa, ..."
- Application should allow the cleaners to report room-to-room cleaning progress by saying "Alexa, living room has been completed"
- Application should allow cleaners to report issues such as damage furniture, plumbing issue, and lost item. The reported issues should be categorized accordingly
- Application should allow hosts to check up on the room-to-room cleaning progress by saying "Alexa, how is the cleaning progress"
- Application should allow hosts to know if there is any issue being reported by the cleaner by saying "Alexa, is there any issue"
- Cleaners and hosts should not have to say the Alexa command in a particular way in order for the smart home device to understand
- Application should support English but if time and resource allow, it should also support Spanish

D. Gantt Chart



### III. DESIGN DOCUMENT

This document will go over the design of the system and the steps taken to achieve our goal. Our design views rely directly to the Amazon Alexa voice applications offered through the Amazon Inc. services. The reasoning for this is because Amazon offers this free development process that is backed up by a very heavy library of resources and documentation. Some of the design constraints we have encountered is our knowledge of implementing different features of the application and determining what all is the Amazon services application can support.

The document is broken into several sections pertaining to important parts of the system, each written will be handled by a team member for the remainder of the project time period.

#### A. Glossary

**API** Application Programming Interface

**APL** Alexa Presentation Language

**ASK** Alexa Skills Kit

**AWS** Amazon Web Service

**JSON** JavaScript Object Notation

**PHP** Hypertext Preprocessor

**SMTP** Simple Mail Transfer Protocol

**UI** User Interface

#### B. User Interface

A user interface must be implemented in order for our users to interact with our application. Amazon Alexa devices are woken by the word "Alexa" following a question or request. In order to allow effective operation and control, there will be three means in which the user can interact with our application. The first, interface will be through a visual interface, where Alexa will convert the speech heard into text and displayed through the screen of the Alexa capable device. The second, once our user has completed their work. A report will be generated and sent to both the user and homeowner. The third and first point of contact with the user will be through voice. This will allow the user to start our application using an invocation name. Amazon developers recommend a situational design, which is a voice first method to designing a voice user interface.

1) *Visual:* While not all Alexa capable devices contain screens, our application will be adaptable. Alexa visual enabled devices include the Amazon enabled tablets and Alexa app. In order to interact with screen enabled devices, a user can touch the screen. Alexa Presentation Language allows us to create Alexa visual skills and templates, called Display Interface Templates, with graphics, images, slideshows, colors and fonts. These will be implemented using display interface Render Template directive and vast amount of display templates. In addition, in the developer console, enabling the Alexa Presentation Language (APL) option will enable us to customize what will be shown to the user. Our interface will change depending on whether a display is supported and using a body template instead of a list template in the Alexa Skills Kit.

2) *Reporting:* Using the information saved from the interactions of the user and the Alexa a skill report will be generated. This report will then be sent through an email to the user and homeowner. In addition, a text message will also be sent once the user has specified that they have finished working on the house and sent to the homeowner allowing them to list the house as available for rent.

In order to send a message through our application a custom back end must be built in order to save the information generated by the Alexa. This custom back end will allow us to manipulate the data and form a report that can be sent through an email to both the user and homeowner for reference. One way we can send emails through our skill is through the use of Amazons Simple Email Service SMTP interface, which is the most common email protocol on the internet. PHP is one of the languages capable of sending Email using SMTP through the PHPMailer class. This information is provided

through the Amazon AWS web reference page.

In order to send a text message the user must be using Android and can use Amazons already built in service allowing users to send a message from the Alexa device to an Andoid device. This option may perhaps be scrapped from the project.

3) *Voice*: Our most important interface will be implemented using Amazons Alexa language called the Alexa Presentation Language (APL). Amazon services provide developers with many UI kits that are highly customize able or users can create on their own. These kits support voice triggers speech playback, and responses. In addition, within these kits are ineration models that contain a combination of utterances, intents and slots to identify each skill. This is the fundamental part of our application since we want our users to be more efficient when working through a house and communication with the home owner. Our application will have different voice activated commands, as described in another section. These commands will activate different functions of our Alexa skill and allow the user to work hands free of the device while they complete their work tasks. Our Amazon Alexa skill will be bilingual, allowing the software to understand not only English, but Spanish as well.

### C. Verbal User Interface

1) *Alexa Skills*: Perhaps the core of this project is the creation and use of Alexa skills. Users will be interacting with Alexa acting as the user interface and initiating the skills through a spoken language. For the project, our main focus is on creating the skills in English. However, if the demand leans towards another spoken language, we will act accordingly and add in a secondary language option.

We will need to create skills for each ability we intend the user to have – give a notification of when a room is beginning to be cleaned and taking note of items misplaced, missing, in need of resupply, etc. Any other skill cleaners wish to have that would be beneficial to them will also be implemented. These additional skills will surface once we finish our interviews with cleaners in the field.

To create the skills, we first create a list of intents and utterances as mentioned in *Section II*. We will be using the Amazon Developer Console to ease this process, though we also have the option of modifying the JSON to achieve the same results. Next, we will need to utilize AWS Lambda to create a function to host the service for our skills. We will write this code in Node.js. This decision was made because of the variety of sample code given within the ASK documentation on Node.js. Finally, we will test the skills to make sure they perform tasks as intended.

2) *Utterances*: One of the goals of this project is to be able to give commands to Alexa without having to follow a set command format. That is, the cleaners who are going to be using this app should be able to give commands naturally, as if they were just talking to themselves or speaking to someone.

Each Alexa skill developed is in actuality a list of intents with many sample utterances to invoke those intents. The more utterances there are for an intent, the easier it will be for Alexa to match speech to the intended command. However, we need to be mindful of the amount of utterances created. Too many utterances for one intent could cause overlapping issues with other skills/intents. This in turn could reduce Alexa’s accuracy and the user would have to repeat themselves until the command correctly registers.

To give the user a more natural feeling when giving commands, we need to first take the intents of each skill and write them down. Then, as if we were the user, start a conversation with Alexa using the skills. This will give us a better understanding of the user interaction as a whole – what they would be using the skills for, how they are going to phrase their commands, what Alexa would say in response, etc. Using this dialogue, we finally craft precise utterances with cleaners in mind for ease of use.

Utterances must have differing variations in order to cover the wide range of ways a user might phrase a command. For our project, utterances will be formatted to be more of one-shot type utterances – users will only issue commands and expect a confirmation to know the command was successful. Our utterances for starting a room to clean will have the format described below:

```
StartRoom I'm starting this room
StartRoom I'm starting to clean the {RoomName}
StartRoom I am starting to clean the {RoomName}
StartRoom I'm cleaning the {RoomName}
```

StartRoom starting the {RoomName} room

...

Other intents will follow a similar format/style as well. We need to make sure contractions are written out both in their full form and contracted form.

#### D. Hardware Component

Since our voice-driven application works primarily with Amazon Alexa supported speaker devices, we will focus on the hardware component of the voice-driven application system. There are low cost entry-level Alexa devices such as Echo Dot and more expensive devices that include smart display such as Echo Show. In this section, we will examine the Alexa speaker devices that will be used in our voice-driven application.

Our voice-driven application users can use from a range of Amazon Alexa speaker devices, from simple speakers such as Echo Dot to those of smart displays like the Echo Show. The various options of Alexa speaker devices do not pose a problem for our system because we are simply using them to capture the user's intent.

Our system requires Amazon Alexa speaker devices to capture audio instructions from users. For the remaining steps to work, our hardware will need access to the internet because we need to access Alexa service. The first step is for Alexa to receive our order and each order should include wake words (eg. Alexa) and launch words (eg. ask, play, turn off etc.), followed by invocation and utterance. After the order has been received, it will be sent to Alexa service where we will utilize the Smart Home Skill API. If the order involves Alexa to complete a task such as "how to operate the coffee machine?", the device will playback the instruction back to the user. Should Alexa not comprehend the order from the user, it will give a fault response.

Below are the range of Amazon Alexa devices and their relative specification:

	Echo Dot (3rd generation)	Echo (3rd generation)	Echo Show (2nd generation)
Size	3.9" x 3.9" x 1.7"	5.8" x 3.9" x 3.9"	7.9" x 5.4" x 3.9"
Screen	n/a	n/a	8"
Wi-Fi	Dual-band Wi-Fi supports 802.11 a/b/g/n/ac networks	Dual-band Wi-Fi supports 802.11 a/b/g/n/ac networks	Dual-band Wi-Fi supports 802.11 a/b/g/n/ac networks
Bluetooth	Support streaming audio from a device to echo and voice control of mobile devices	Support streaming audio from a device to echo and voice control of mobile devices	Support streaming audio from a device to echo and voice control of mobile devices
Power	Requires standard wall outlet	Requires standard wall outlet	Requires standard wall outlet

In order to test our hardware functionality, we have included the following table to show the responses we expect to get from Alexa. These are just one way to invoke the interactions but there should be a wide range of ways to invoke the intents.

Question	Alexa Response
Alexa, living room has been cleaned up	<i>Report back to the owner through text/email</i>
Alexa, there is a leakage in the bathroom	<i>Report back to the owner through text/email</i>
Alexa, This is (firstname)(lastname)	<i>identifies the cleaner</i>
Alexa, I have started the cleaning process	<i>begins a timer to determine how long the cleaning process takes</i>
Alexa, we need to re-order supplies	Sure, what supplies need to be re-order (User can list the supplies afterward)
Alexa, I need to report damage	What is the damage and where is it (User can list the damage afterward)
Alexa, maintenance is needed	Sure, what needs to be maintenance and where is it (User can list the maintenance request afterward)
Alexa, the property is now clean	<i>stops the timer and store the data of who cleaned and how long time took</i>
Alexa, Room 1 of 6 is now clean	<i>Report back to the owner through text/email</i>
Alexa, there is a leakage in the bathroom (with no internet connection)	I'm sorry but I can't connect to the internet

### E. Software Component

One of the most crucial components of our project is the software component. The software component will help integrate and seamlessly grab data and user information to deliver accurate and helpful responses. As mentioned, we will have a hardware component that our software will have to work on. Because our application will be relying heavily on an Amazon Alexa Speaker device, we will be using software that integrates well with our device.

To build our software, we will be using the support of APIs and tools provided by Amazon to develop Alexa skills. The kit that supports this is the Alexa Skills kit which comes with documentation and code samples that will help with designing our software component.

Our main software support will come from Amazon AWS. This means our team will be using an AWS account to help build our skills. Alexa supports Amazon Web Services and works well with the applications and services provided on the platform. Because Alexa and AWS are both built by Amazon, integration between the two is easier and support and tools to develop are online for software support. When the user is interacting with Alexa, the information that is provided will be stored in the system. This will be done by using AWS lambdas that are serverless containers that can be programmed to be triggered once it receives information. We will program a lambda to store user data into an AWS S3 bucket. Both AWS Lambda and AWS S3 are crucial software components that will make calling and retrieving information easier in the system. The information that is sent to AWS Lambda will be in the form of a JSON file and the language that we will be writing our lambdas in is node.js. We will also be writing a lambda that can convert text to audio to send responses back to the user. A good amount of our software will be working with fetching and using user data between S3s and lambdas. To make sure our skill remembers information, the information that needs to be remembered will be saved in S3, this way the skill can remember from session to session.

The software component will take some time to develop on our timeline. Right now it is estimated to take around two months to develop and will be built alongside the Alexa Skills needed. As we build skills, we will add on to the software support. An example of this is if we find out we need to convert text to audio we will create a new lambda function with the functionality needed to roll out the skill. Testing will be another part of the software component which would take a couple weeks to look into.

### F. Conclusion

The voice-driven application that we will be designing for Dana will follow the procedures and requirements that have been provided in this design document. Most of the work that is needed to build the application lies on the software component of the system. We will need to use the APIs and tools from Amazon to develop the necessary skills for our application. We have AWS lambdas storing the information provided by the user for our UI part of the system. That being said, if there are any unforeseen problems within the software programming stage of development, we will come up with a solution and record any changes made in the design document section of the final report.

### G. Modification

The voice driven application will not be supporting touch control devices that are Alexa compatible. Software will now only support English input instead of Bi-lingual input as first stated. The application session ends after each cleaning report update since we don't think there is a need to keep the session open while the cleaner is tidying up the house. We have made some minor changes to how Alexa response to the voice input:

Input	Alexa Response
Alexa open my housekeeper	<i>Hello, Welcome to my housekeeper. Are you cleaning a room, or would you like to make a damage or maintenance report?</i>
Damage is found in bathroom	<i>Thanks, bathroom has damage, I'll forward the damage report to the owner.</i>
I am cleaning	<i>How many rooms are you cleaning today?</i>
Two	<i>Which room are you starting to clean first?</i>
The kitchen	<i>Ok, starting to clean the kitchen</i>
Cleaning	<i>Welcome back! It looks like you were cleaning the kitchen. Did you finish cleaning that room?</i>
Help is needed in kitchen	<i>Thanks, kitchen needs maintenance assistance, I'll forward the request to the owner.</i>
We need more supply in bedroom	<i>Thanks, bedroom needs maintenance assistance, I'll forward the request to the owner.</i>
Send report	<i>Report back to the owner through text/email</i>
Alexa, there is a leakage in the bathroom (with no internet connection)	<i>I'm sorry but I can't connect to the internet</i>

open my housekeeper

Hello! Welcome to My Housekeeper. Are you cleaning a room, or would you like to make a damage or maintenance report?

damage is found in bathroom

Thanks, bathroom has damage, I'll forward the damage report to the owner.

open my housekeeper

Hello! Welcome to My Housekeeper. Are you cleaning a room, or would you like to make a damage or maintenance report?

i am cleaning

How many rooms are you cleaning today?

two

Which room are you starting to clean first?

the kitchen

Ok, starting to clean the kitchen.

open my housekeeper

Hello! Welcome to My Housekeeper. Are you cleaning a room, or would you like to make a damage or maintenance report?

cleaning

Welcome back! It looks like you were cleaning the kitchen. Did you finish cleaning that room?

yes i did

Nice work!

open my housekeeper

Hello! Welcome to My Housekeeper. Are you cleaning a room, or would you like to make a damage or maintenance report?

help is needed in kitchen

Thanks, kitchen needs maintenance assistance, I'll forward the request to the owner.

## IV. TECH REVIEW (CARLOS)

### A. Introduction

This section will go over my roles in the project and the different forms of technologies used to implement each.

### B. Personal responsibilities

1) *Testing:* Testing is necessary in order to provide the facilities to the clients like the delivery of high quality product or software application which requires lower maintenance cost and hence results into more accurate, consistent and reliable results. One of my responsibilities in the project is to conduct testing for the application to ensure the products quality and meet the clients requirements. This includes testing the code (functionally) and the physical product (user experience). Fortunately, Amazon provides multiple testing frameworks.

In order to use these testing frameworks, minimum required configuration for the interactive model must be complete. Once we have defined and built an interaction model for our skill Amazons utterance pro-filer will allow me to test it by entering utterances (key words) and seeing how they resolve to the skills intents and slots. This can be done before writing any code.

Another option to test our skill is to use Amazons simulator in the Developer Console through Amazon. This simulator gives access to most Alexa Skills kit features without a device using either voice or text. This will allow me to test most functionality in a skill, including dialogues, progressive responses (with limitations), and entity resolution.

Manual quality assurance testing will be done using an Alexa enabled device. Testing will include using key terms the device is expected to receive. Some terms include: "Alexa, check the cleaning service", "Alexa, report an issue", "Alexa, living room has been completed", "Alexa, is there any issues?".

In addition, the skill beta testing tool can be used to test your Alexa skill in beta before releasing it to production. This tool will allow us to invite up to 500 people to beta test our product and use the Amazon Developer Portal to see their feedback.

Amazon also includes two sets of automated testing called Validation and Functional Tests. The Validation set of tests verifies that all required fields for the Alexa Skills Store are complete. The Functional Test performs a set of pre-certification tests on the skill. These tests provide immediate feedback for common certification failures. The Functional Test cannot be run unless the Validation test is successful. These test can be run during development and post development when is ready to publish.

2) *Certification and Publication:* Another responsibility includes making sure the application meets Amazons requirements in order to submit to the Alexa skills store. In order to make sure the application is certified I will follow Amazons "Submission Checklist". This checklist includes ensuring our skill meets the Alexa policy guidelines, security requirements, perform functional tests, perform required voice interface and user experience tests, test screen-based interactions. Once the skill is ready to submit, the Validation test must be ran and be successful. Then the Functional test is ran after submitting the skill, and if the skill fails then it cannot be published and the issues must be resolved.

Once the skill has been certified by Amazon it can be published to the Alexa Skills Store and available on all Alexa devices either immediately or at a later date. Once the skill has gone live then it cannot be configured until it is removed. This process will be done using the Alexa Developer Console provided by Amazon and specifically the "Certification" page. Amazons website provides me with all the tools needed to make sure the skill meets and follows all of the guidelines. There are no other technologies we can use to certify and publicate our skill.

3) *Notifications:* Another responsibility is to implement the notifications feature that the Alexa skill will need in order for the cleaners to communicate with the home owners. This will be done by using the Amazon Web Services including the Amazon Simple Email Service and Lambda service. In order to send text messages we'll be using a third party service from Twilio. Connecting all of these services will allow us to efficiently connect our cleaners and homeowners using our Alexa skill. Utilizing these services will save us from having to rewrite communication from the ground up and include many features that could later be added to our skill. In addition, these were the best options determined for integrating these features for our skill to use.

In order to implement these custom skills, we can host our service in AWS Lambda or as a web service hosted on endpoint, which will be responsible for running the code. AWS lambda is an Amazon service that lets you run code in the cloud

without managing servers. These Lambda functions can be written in Node.js, Java, Python, C sharp, or Go. This is generally the easiest way to host the service for a skill. We could also build our own web service using any programming language, as long as the service meets the predefined Amazon requirements. Our application will be written in Node.js using a lambda function from the Amazon Web Services as recommended by our client and the Amazon documentation.

Since we are specifically building an application for the Amazon Alexa, there are not many ways to implement skills, it will be set up using the Amazon Alexa Skills Developer Portal by linking the lambda account to that skill. Some benefits in implementing using the AWS Lambda is that we can focus on building our voice experience instead of focusing on computing resourcing, don't need to provide an SSL certificate, Lambda streamlines access control and security, and lambda is the fastest way to implement a skill. In addition, AWS hosts many services such as the Amazon Simple Email Service that allows us to connect to our lambda function as triggers. Using our lambda code we can invoke this service to send emails. Twilio also gives us a lot of documentation showing us the capabilities it holds for Alexa skill implementation. Their service allows us to not have to worry about setting up telecommunication and they take care of staying up to date with current guidelines.

## V. TECH REVIEW (KENNY)

### A. Introduction

Our team will be leveraging current voice technology to provide a better experience for vacation home guests. Our main focus is on optimizing the the turn-overs between guests through the cleaners. The voice application should be able to keep track of the rooms currently being cleaned. It should also allow the user to report broken items, missing items, supplies needed to be restocked, etc.

### B. Personal Role

We currently don't have any assigned roles within the team. However, our client recommended that our team splits up into two subgroups for the initial researching. One group can go survey potential users of this app, the cleaners, to find if there is any real use for this app and additional information. The other group can get a head start on learning the in's and out's of the voice assistant of choice. In this case, our client decided on Alexa as the voice assistant.

### C. Command Accuracy

1) *Description:* When using voice assistants, there is typically a wake word that a user needs to say in order to begin interaction with the voice assistant. After giving this wake word, the user can initiate any commands to the voice assistant, provided that the command is valid and the voice assistant understands the command. However, because voice technology isn't perfect, there is always an off-chance that the voice assistant won't understand the command given, even though it might a valid command. In other cases, inputted commands might even be misheard and the voice assistant will process whatever command it assumed was given. Therefore, command accuracy is a very important aspect when it comes to voice technologies.

2) *Choices:* There are currently many different voice assistants in use daily, some more popular than others. For our project, we need to find the best voice assistant that can most accurately take in commands regardless of situation – quiet environments, music playing through a speaker, social event, etc. The three voice assistants we decided to look at are Amazon Alexa, Google Assistant, and Apple's Siri.

3) *Amazon Alexa:* In a very recent study on a number of voice assistants, the researchers compiled a list of 4,999 questions to ask each and every one of the voice assistants in the study. Of the questions asked, Alexa attempted only about 67% of the questions asked. However, around 78-79% of the questions attempted were answered fully and correctly. Compared to 2018, Alexa has made about a 14% increase in the number of answers attempted but a 2-3% decrease in the number of fully and correctly answered questions.

To help improve Alexa's accuracy, Amazon has released two new tools for developer to use within the Alexa Skills Kit. One is called the NLU (natural language understanding) Evaluation Tool which helps prevent developers from overtraining NLU models, thus reducing accuracy. The other is called the Utterance Conflict Detection which prevents from conflicts with utterances.

4) *Google Assistant:* In the same study mentioned before, Google Assistant attempted around 73% of the questions given. Surprisingly, it was able to fully and correctly answer about 88-89% of the attempted questions. In 2018, Google Assistant attempted slightly more questions than the following year, about 78% of the questions. The number of fully and correctly answered questions were higher this year as well, being around 95%. The reasoning behind the accuracy of Google Assistant could be due to the machine learning implemented. Google is also working to "expand the assistant's understanding of personal references."

5) *Siri:* Again, in the same study, Siri attempted only 41% of the questions asked. Among those 41% of questions attempted, Siri was only able to fully and correctly answer 70% of the questions. In the previous year, Siri attempted the same amount of questions but was able to fully and correctly answer 80% of them, a 10% higher than the following year. Despite all this, Siri has an edge over all other voice assistants in the Command type questions, being "more helpful and versatile ... in controlling your phone, smart home, music etc."

6) *Comparison:* Of the three voice assistants, Google Assistant attempted the most amount of questions. At the same time, Google Assistant was also able to fully and correctly answer the most questions attempted. The second closest would be Alexa. Siri, on the other hand, is severely lagging behind both voice assistants, especially in the number of questions attempted.

7) *Final Choice:* As of now and based purely on statistics, Google Assistant would be the voice assistant of choice. Overall, it is able to answer more questions and it is the most accurate. However, both Amazon and Google seem to be slowly working on ways to help improve their system.

#### D. Device Compatibility

1) *Description:* Another aspect that makes voice assistants so popular is the wide array of compatible devices available. In our case, compatible devices are not only limited to the devices created by the company where the user interacts with the voice assistant, such as a home hub. Compatible devices will also include third party devices made for that specific voice assistant or are at least somewhat compatible. These devices are able to connect with a central voice assistant device and can then be controlled using voice commands. This is appealing to some because of such control a user gets in their own home without too much hassle in the set up process.

2) *Choices:* Voice assistants vary in their list of compatible devices. Therefore, we need to find the voice assistant most compatible with the large array of smart devices out in market today. Once again, we will do a comparison of three popular voice assistants, Amazon Alexa, Google Assistant, and Apple's Siri.

3) *Amazon Alexa:* Overall, Alexa has a very wide array of devices available. This includes the standard Echo devices – Echo, Echo Plus, Echo Dot, Echo Show, etc. Then there are other third party devices designed to work seamlessly with Alexa. Some of the more popular devices are the following:

- Amazon Smart Plug: gives a user control to power on/off a device; allows support for routines
- Philips Hue A19 Starter Kit: similar to smart plugs with addition control of brightness levels
- Nest Thermostat E: can automatically lower/raise temperature based on on habits
- Ring Video Doorbell: allows a user to see and hear who is outside the door

This is just to name a few. There are at least 60,000+ Alexa compatible devices available that can be controlled by Alexa.

4) *Google Assistant:* Google has a variety of devices available to take advantage of Google Assistant. Their products include the Google Home, Google Home Mini/Max, Google Nest Hub, etc. There are also third party devices that work with Google Assistant:

- C by GE: smart light bulbs with features like the previous mentioned smart light bulb and communicates with each other using Bluetooth
- Ecobee3 Lite: responds to voice commands to adjust temperature and such
- Nest Hello: allows homeowners to talk to those who rung the doorbell with additional features like facial recognition

Currently, Google has around 30,000 devices compatible with Google Assistant.

5) *Siri:* Apple has a short list of available products that respond to their wake word "Hey Siri," all of which are Apple products. This includes the iPhone, iPad, AirPods, Apple Watch, HomePod, and Mac devices. Apple also supports a decent list of supported devices via the Apple HomeKit:

- Ecobee3 Lite
- Lix Mini Color: provides billions of colours and is also dimmable
- WeMo Mini Smart Plug: a smart plug that can turn on/off with Siri commands

Unfortunately, there are only around 50 compatible brands with Siri.

6) *Comparison:* Alexa has over 60,000 compatible devices, whereas Google Assistant only has around half of that with 30,000. Apple provides a list of compatible devices on their website but this is significantly smaller than what the other two companies support. For comparison, Alexa has 7,400 compatible brands, Google Assistant with 1,000, and Siri with around 50.

7) *Final Choice:* With such a large number of compatible devices, Alexa would be the clear choice. There are many devices to choose from to fit different needs.

#### E. Application/Skill Support

1) *Description:* Perhaps the most import aspect of voice assistants is the application support behind a specific voice assistant; in other words, the skills that are implemented. When a user asks a voice assistant to do something or for some kind of information, this is what is meant by a skill. Assuming the skill is valid/has been implemented, the voice assistant will carry out tasks associated with the skill to fulfil the request(s) of a user. As an example, a skill could be, "[specific wake word],

give me the weather forecast for the following week.” The voice assistant should then go on to list the weather forecast of the following week. This kind of skill is more generic and could potentially be asked to a variety of voice assistants to produce the same result. However, there also exists skills called branded skills created specifically for a publisher’s brand, meaning complete ownership of that skill.

2) *Choices*: We need to find the voice assistant that provides the most support in creating and developing new skills, as our team will be creating a new skill. We will, again, be looking at the three popular voice assistants, Amazon Alexa, Google Assistant, and Apple’s Siri.

3) *Amazon Alexa*: Amazon provides developers a way to build Alexa skills with the Alexa Skills Kit (ASK), a ”collection of self-service APIs, tools, documentation, and code samples.” Developers can utilize the Amazon Web Services (AWS) Lambda – a serverless computing platform – to run code written in Node.js, Java, Python, C#, or Go. Amazon also offers a free training course for those who want to build their own Alexa skills. Aside from building skills form scratch, a person can instead fill out preset blueprints or even hire an agency to build the skill for them.

4) *Google Assistant*: Rather than being called applications or skills, Google refers to these as Actions. Google has a web-based platform called the Actions Console where developers can create new Actions for Google Assistant. Provided along with the console are documentation, sample code, and API references. To further help developers create new Actions, Google has several guided tutorials through Google Codelabs. Coding can be done in Node.js, Go, C++, Java, and other similar languages.

5) *Siri*: Apple has two different ways to develop skills for Siri. The first option is by using the Shortcuts app built into iOS 13 – it is a drag and drop of actions which users can chain together to run in one go. The second option is the SiriKit. This allows developers to create an extention to apps that integrate with Siri. Apple provides documentation such as development guides and design guidelines.

6) *Comparison*: All three companies provide some sort of development kit to produce skills for the appropriate voice assistant. Both Amazon and Google offer some kind tutorial or guide on how to create skills, as well as sample code to reference. Unfortunately, Apple doesn’t seem to provide these resources, or it is hard to find.

7) *Final Choice*: Alexa and Google Assistant are great candidates in providing application/skill support. However, Alexa has a bit of an edge here due to the ease in starting the creation of Alexa Skills. All documentation, step-by-step course, and additional resources are listed in a single page, reducing the need to search for additional information.

## VI. TECH REVIEW (TSO-YI)

### A. Goal

This end goal for this project is to create a voice-driven application that allows cleaners of rental homestays to report any issues faced during cleaning while simultaneously reporting the cleaning progress to the house owner. The voice application should also allow the owner to record any useful tips that will help guests navigate the house and also making them feel welcome. The cleaning progress and report will be accessible on a website for the owner. My role in the team is to find out the necessary details needed for Amazon Alexa to capture the data from the user. This includes any orders given in English and other languages such as Spanish. While it is important to capture the data, it is equally important to display the result in a neat fashion. Therefore, I am in charge of organizing the User Interface (UI) as well.

### B. Language support

Since the client hopes to incorporate Spanish data input within the voice-driven application, I have included language support as one of the key pieces for tech review. Currently, the cleaning companies associated with the client and his Virtual Concierge Service are either Spanish or English speakers. Therefore, having Amazon Alexa capable of capturing English and Spanish will be a very useful support.

According to Globalme, Amazon's Alexa service supports five English dialects and three Spanish dialects. As of summer 2019, Alexa is also capable of understanding the different dialects of Spanish (i.e. Spain Mexico, and the US). This meant that cleaners assigned to accommodations should have no problem speaking to the smart home devices.

The article written by Verge also suggested that Amazon will be rolling out a multilingual mode to Alexa, meaning that the device will be able to comprehend both English and Spanish without the need for accommodation owners to change the language setting within the device. Having this language support saves both cleaners and home owners the need to go through the hassle of switching the input language setting constantly, hence increasing the efficiency for both parties.

How does Amazon Alexa support multilingual mode? Based on the article written at Voicebot, in order to parse and respond to two different languages at once, the device needs to run two voice assistant software models at the same time through just one system. For developers, Amazon Smart Home Skill API will automatically interpret the customer's request and send the appropriate directive to code in a Lambda function that can handle the request. The skill communicates with the device's cloud and either performs the desired request or returns a fault message.

1) *Final decision:* Based on the above research, I have decided to use Alexa Skills Kit API in order to easily use the multilingual ability provided by Amazon. I have made this decision because it is the easiest method to incorporate Alexa's ability to input and output multiple supported languages such as English, Spanish, French, and Italian into the voice-driven application that we are developing. This will reduce the burden on the group and allow us to focus our time and resources on other pieces in the project.

The data capturing begins when Alexa discovers the wake word. For this instance, if the user says 'Alexa', the signal is then sent to the speech recognition software in the cloud and audio is subsequently converted into a text format.

### C. Data Capture

The article on [towardsdatascience](#) mentions that Alexa is built based on natural language processing (NLP), a procedure that converts speech into words, sounds, and ideas. The recorded speech is then sent to Amazon's servers to be analyzed more efficiently. The 'order' given by the user will be broken down into individual sounds. The result is then cross referenced with the database that contains various words' pronunciations to find out which words are most closely corresponded to the combination of individual sounds. After identifying the important words to make sense of the 'order', Amazon's servers will send the information requested back to the device and Alexa will respond. The 'order' consists of four parts: wake word, launch, invocation name, and utterance. As mentioned previously, the wake word for our system will be 'Alexa'. The launch portion determine whether the order is a question, command, or action. For example, our system lets the cleaner 'report' the cleaning progress or allows the guest to 'ask' questions like how to use a coffee machine. Invocation name is the keyword that triggers the 'skill' we will develop for our application. Lastly, utterance is the phrase that users will use when making a request to Alexa and Alexa will then identify the user's intent and respond accordingly. Utterance decides what the user wants Alexa to perform.

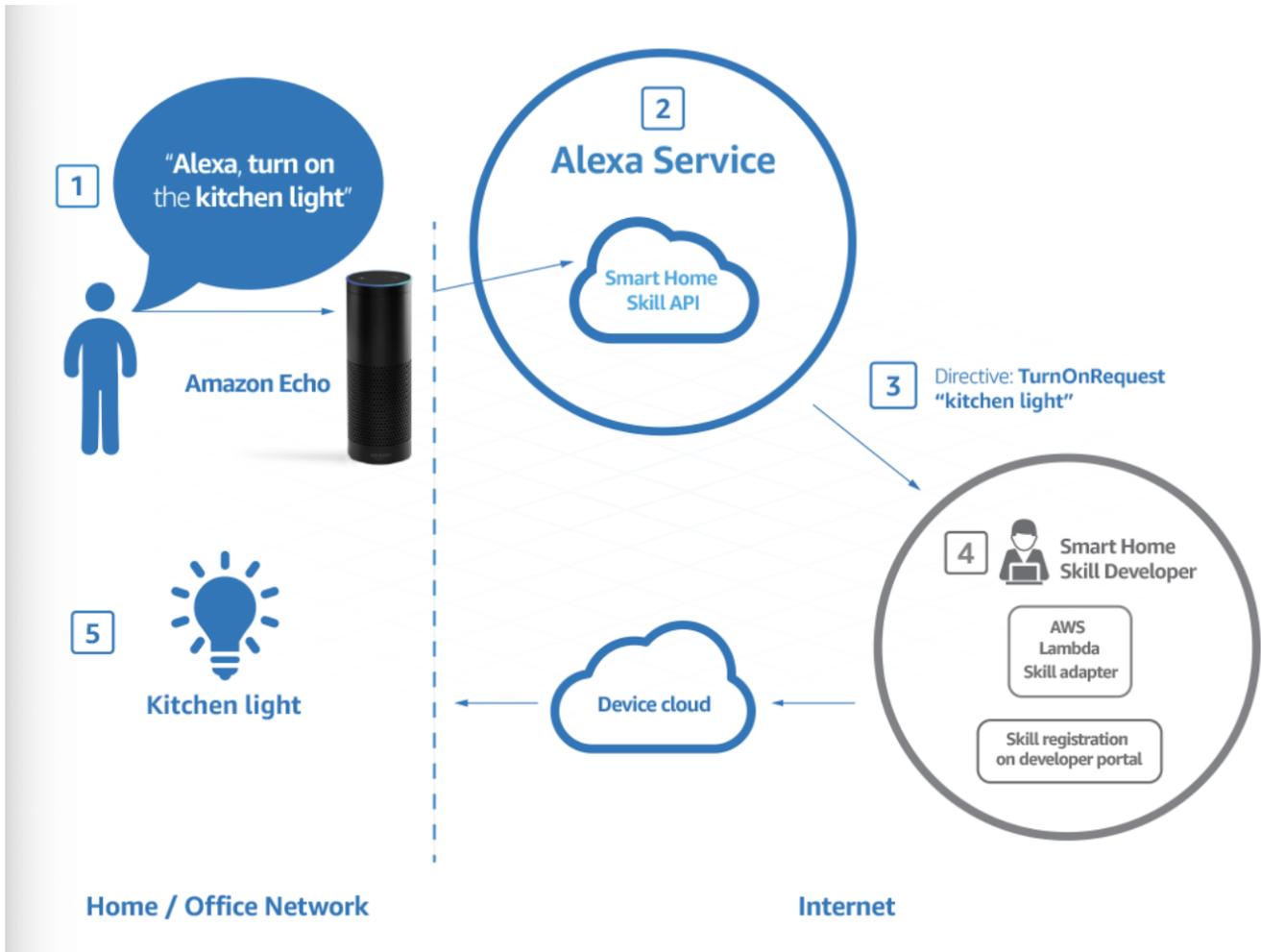


Figure 1. Process of how Alexa receives your order and completes the task



Figure 2. Understanding of how Alexa interprets the order

1) *Final decision:* I have decided to use the Amazon Voice Service Device skill development kit (SDK) to allow the use of capturing information input by cleaners, guests, and owners. Instead of using other third party API, the Amazon developer site provides training courses that familiarize us on working with Amazon Web Server (AWS), Lambda, as well as skill adapter. This is highly valuable for our team because we do not have prior experience in developing a voice-driven application on Amazon Alexa. Therefore, this resource will greatly reduce the hassle in capturing the user's input data.

#### D. Organization of UI

Since we need a way to display the results reported from the cleaner or guest to the owner, our group believes a website is the best way to present it. Given that the users for our application are most likely not proficient in technology, it will be crucial for the group to create a site that is easy to navigate

In order to display the results on a website, we first need to create the front-end of the site. The website will not contain anything that is too complicated, minimizing distraction of the users. The option we have here is HTML and CSS. The back-end of the site gives us more flexibility and we can either use Javascript, Python or other frameworks.

*1) Final decision:* I have decided to use HTML and CSS for the front-end of our website while using Node.js (JavaScript runtime) for the back-end of our website. The decision for both front and back end of our UI is based on the comfort of the programmers in the team. All members have taken such classes and some have prior experience coding in these three programming languages. Therefore, we will not need to spend addition time learning new programming languages in order to complete the task. I have picked node.js as it is a lightweight and efficient platform for application back-end and the package ecosystem, npm, is the largest available ecosystem of open source libraries. This provides the programmers on our team the flexibility of creating a well-functioning website. As for the testing for our front-end and back-end code, we are currently thinking of use Mocha testing environment for Node.js and Selenium for automating browser functionality. I picked these two testing environment because they are simple enough for us to learn within a few weeks but will give us a robust testing outcome.

## VII. TECH REVIEW (SUMEGHA)

### A. Problem to Solve

Hospitality is one area where traditional hotels excel in while alternative accommodation such as Airbnb do not. With most hotels, extra services are provided for guests and can be quickly received/delivered with just a phone call from the room. Along with this, housekeepers routinely clean and tidy up the rooms as well as resupply towels and such while guests are away, unless specifically told not to. With Airbnb, it can be quite a challenge to provide such hospitality services, especially if the host does not live close to the home-stay. Any questions a guest might have are met with phone calls that can total up to hours, lessening the experience for the guest and taking time away from the host's day. Airbnb hosts who do not live on-site might leave instructions or tips on post-it notes for their guests. However, the note might not be seen by the guests or be lost during turnover. The lack of personal assistance during the stay might push some guests to pick traditional hotel over Airbnb instead. Another major advantage that traditional hotels have over Airbnb and other alternative accommodations is efficient turn-overs between guests. When guest check out of their reservations, hotel employees are immediately informed and are able begin the turn over process. The cleaning progress can be reported to hotel staff in real time. Unlike traditional hotels, Airbnb hosts rarely rely on either cleaning services or self-cleaning by guests. Should hosts rely on cleaning services, the turn-over efficiency between guests will be low due to the fact that cleaning services will have to travel across town. If guests do not leave by the check-out time, this can waste time and money for cleaning companies and can be a pain for the hosts to deal with. There needs to be a way for hosts to provide hotel-level hospitality, and cleaning services a more efficient way to clean homes. Therefore, what can Airbnb and other alternative accommodations do to achieve the same or better turn-over efficiency as compared to traditional hotels?

### B. Responsibilities for the Project

1) *Data Storage:* One of my personal responsibilities for the project is to understand and look into our options for user data storage. It is important for our project to take in user data and store it in a way that can be used to better the user's experience in an Airbnb. The data that Alexa captures comes from the user's voice. Because we want to understand what the user has said, Alexa stores the data in an audio file to then decode the information. This is a part of data capture which is one of the pieces of our project. More specifically, our team needs to figure out how to store these audio files. There are different options that our team can use for a database or just general data storage. There are different types of storage. There are local storage options like external hard drives. These are usually installed in a computer or laptop. This may be useful for local storage and temporary file manipulation. But this would not be practical for our voice app because it is better to store the information we need from the user not locally but rather in a way that is accessible from multiple computers. It is also limited with storage capacity. The alternative to local storage is remote storage. Within remote storage, there is cloud storage. Cloud storage allows people to access data from any computer with internet access. It has storage space from commercial, usually large scale, data centers. There are many benefits to using cloud storage. Not only is cloud storage popular with voice applications like Alexa, but it also is more compatible. More specifically, it can handle large files like potential audio files that need to be saved for interpreting. It is reliable and is a good backup because the data is less likely to get deleted from human error or other issues that may occur. The trade off for cloud storage is that it is costly. With cloud storage the user is paying to keep the service up and the data stored every second that they need it. Over time the numbers add up. Based on the needs and requirements of our project cloud storage will work better than local storage options.

2) *Technologies for Data Storage:* There are many different technologies for cloud storage. Some of the most popular ones are Amazon S3, Google Cloud Storage, and Microsoft Azure. These are all cloud storage we will consider. Amazon S3 buckets are most compatible because Amazon created Alexa and S3s and therefore have been built to work well together. There is also a lot of documentation on how to work with both together.

3) *Processing of Data for User Response:* My next responsibility is to look into the processing of the data. We are defining processing of user data as, how the data gets taken in and used to give an output. (Also can be defined as how the data goes through the process of coming from the user to go back to the user in a meaningful way). Most cloud platforms have an internal storage option. An example of this is Amazon has S3 Buckets. Along with the storage option there are different ways to reach data in these internal data storage or database options. The best option for our voice application is serverless functions. Serverless functions demand less maintenance. They also are easy to scale for demand and are quick to deploy.

4) *Technologies for Processing Data:* One platform we might use is Amazon S3 as cloud support and data storage support for the application. With that, data needs to be processed to respond to the user. Assuming that our team will be using some kind of cloud data storage for data management, we need applications that will allow us to grab the file or data when

needed and use it at the right time. One option, if we use Amazon S3, is using containers to work with the user data. More specifically, this would be lambda functions. Lambdas can be used to fetch and process what we need to know about the data. Or just even allow us to process or call applications or functions that can process voice to text. A good example of this was given by an article on Alexa Skills. It goes through the process of how Alexa may use Amazon S3s and Lambdas to process and fetch user data. If the Alexa is developing a skill to remember the user's birthday, the user will first speak to Alexa and that data will then be processed to first check S3 to see if there is a record of the user's birthday. This "process" is done through a lambda function which can be triggered when a file or information is placed in the S3 bucket. Lambda will have the code needed to ask if the user's birthday is stored. If S3 has the user's birthday, it will return an entry. The skill will then sends a JSON response with the answer that is then converted to voice. This clip is sent to an Alexa compatible device to play to the user. The other options our group can consider is the Google and Microsoft versions of Amazon Lambdas. Google has Google Functions that are event-driven serverless functions like lambda. It works as support between cloud services like data storage (similar to S3s). The last option is Microsoft and their cloud platform. Microsoft has a serverless compute service called Azure Functions. Azure Functions are comparable to Amazon Lambdas and can perform the same functionality mentioned in the example given above.

5) *Toolkit for UI:* The last item I am responsible for is understanding the toolkit needed to generate the UI. Amazon Alexa has a lot of developer support when it comes to designing, creating, and developing new ideas and functionalities for Alexa. They support developers with principles, voice, visual, user interface (UI) patterns and multimodel best practices. For some topics, Amazon provides videos and courses to understand the product, platform, and how to create and design Alexa Skills from scratch. There is a course that provides some insight on designing an engaging Voice User Interface. This will help with integration and making a smoother application for users. Along with improving our Voice User Interface, we also want to have some toolkit that can help us with having a website for users to interact with.

6) *Technologies for an UI Toolkit:* Some toolkits or frameworks that can help with creating an aesthetic website is Bootstrap and Foundation. Bootstrap is a newer technology that has CSS, HTML, JavaScript, and JS components and help developers create a website they want to see. Foundation is similar to Bootstrap but is considered to be better for mobile rendering.

## VIII. WEEKLY BLOG (CARLOS)

### Fall

Week 4: This week, not much progress was done on the project. We finished up the final draft of the requirements page and had another meeting with our TA. Some problems we encountered was to gather feedback for our requirements page. Most feedback was delivered late, but we have made changes to the paper. Our plans include waiting for instructions regarding the next assignment.

Week 5: This week we are focusing on the how of the project. My team has met to determine the different aspects of what we are going to work on and how we are going to build this application. Major problems we have encountered include having some difficulties getting into contact with our client while he is at a conference. Some questions have still remained unanswered and hopefully will be cleared up by the final submission. Another problem has been trying to figure out who is going to do what and what exactly we need to plan out to build the application. Our plans include to keep trying to get in contact with our client and to meet as a group to figure out what we are each doing and dividing the work evenly.

Week 6: This week I worked on a draft tech review where I listed out my three responsibilities and technologies for the project. I have just begun receiving feedback from the peer reviews and am working on fixing any issues or implementing any suggestions from those before submitting the final draft tomorrow. As a team, we have had a meeting to discuss the next group assignment due Tuesday and have evenly divided the parts in between the team members. We have also begun discussing the hardware we will need for our project. In addition, we have heard back from our client, who was at a conference. He has returned to us with contact information regarding a startups company's CEO of a cleaning service for Airbnb. We are currently in the process of setting up a meeting with her.

Some problems we have encountered are during our meeting with Fariba, our TA, we were asked what each of our contributions were for the project. Unfortunately we could not remember everything at the top of our heads at the time and she has asked for us to come up with this for our next meeting. We also brought up how some of our peer reviews had come in late and if there could be a deadline made for those so we could implement them into our assignments before the start date.

Our plans are to continue to work on our tech reviews due tomorrow. In addition, we will begin working on each of our parts for the design over the course of the weekend and hopefully set up a time with our clients' reference.

Week 7: This week we turned in our group Design Document where my part was to complete the user experience section. We were able to research and complete each of our parts on time. We are currently waiting to receive our feedback from the instructors in order to make any necessary changes for our draft 2 submission. During our TA meeting, we had her look over our Design Document since there are no peer reviews available for it. This week I didn't run into many problems. One problem I did run into was researching how we will gather the information from Alexa into a send-able report. Our plans are to wait for our assignment feedback and continue to do research on our project.

Week 8: Our current progress includes finalizing the Design document and making sure it is ready to be submitted to our client. After talking with our TA and discussing the assignment, we have received feedback on it and are working to fix any issues. Some of the issues we have encountered is making sure the assignment is up to guidelines. After some confusion, we were instructed to not just follow past examples, but also look at the IEEE Std 1016-2009 formatting guidelines. Our plans are to look over the guidelines and make sure our documents follow them.

### Winter

Week 1: With the beginning of the term, my team has completed our team critique 1. This has allowed us to provide feedback to each other about meeting our group guidelines that we made Fall term. In addition, we have met together in order to provide our TA with our teams availability on when to meet. Currently, we have not encountered any problems. Our current plans are to await instructions on our next steps for our project and what the term outline looks like.

Week 2: This week we have begun working on the project and familiarizing ourselves with our roles. We have met with our TA to talk about any issues and to clarify any misunderstandings regarding deadlines and this term's outcome. In addition to the meeting, we have met as a group to talk about what we will be working individually during the term and what hardware and software we will need for the project. Some problems we have encountered are that most of my team have never taken the role of a project manager so we are somewhat unsure of how to set deadlines and milestones to track our individuals progress within our roles. Fortunately, I have had some experience when taking entrepreneurship classes on some management software that will allow us to see how the team is working through the project. I have went ahead and set up the software which includes a kanban and we have planned out what each team member will be working on. There is also

an outline of what we need to accomplish every week to meet our deadline. My plan is to continue to research and develop my part in the project and collaborate with my team members.

Week 3: This week I have begun working on the implementation of twilio (SMS) and AWS SES EMAIL implementation. We have been getting issues setting up an amazon account we can all work on but are individually working on our parts using our private accounts. This weeks lecture also included information about the poster and I plan to talk with my group in regards to this part of the project. Some problems I have encountered is that my part in the project is included towards the end of the project timeline as I need the base project information in order to test sms and email. My plans include working on the project this weekend to make sure we meet the Alpha deadline of week 6.

Week 5: This week I worked on and completed my assigned task. I have built and tested the SMS functionality and am waiting for my team to share their beginning code needed for the project. I also met with the TA to discuss the Alpha deadline and our progress with the project. Some problems I have been encountering these weeks include problems communicating with my team. I have requested for the past two weeks for them to share their code in order to integrate my functionalities and they have not provided me with anything. In addition, this week only one other team member showed up to our TA meeting while one other is out of town for a family emergency. This will lead in the delay of the project as I was not able to fully describe our progress to our TA. My plan is to continue to work on my next part of the project which implements the functionality of sending an Email from our Alexa Skill. This feature gathers the information given to the skill and compiles it into a send able report for the renter and cleaning service to review. I also plan on working on our poster draft that is also due the same day as the alpha deadline.

Week 6: This week we are reaching the project Alpha deadline and are working on putting together the rough draft for our posters. Some problems we have encountered is rescheduling our design review as one of our team members is out of town for a family emergency. My plans include continuing to work on my part of the project in order for us to get closer to a beta version.

Week 7: This week we finalized our design review date. We are working on getting our presentation ready by working out what each person will say. We are also working on any papers that need to be submitted to other groups. I have not run into any problems this week. My plans are to continue to finalize our projects alpha version and work on the presentation.

Week 8: This week I am gathering information for my part of the presentation. I finally found out that you can add more developers to the amazon developers portal and went ahead and added my team members so that we have one single Alexa skill. They are working on implementing their code. From there we will be testing it through the physical Alexa device even though it is very similar to the Amazons portal tester. We are also waiting to hear back from other teams to get familiar with their projects before the design review date. I have not run into any problems this week. My plan is to continue to document my part in the project so it is clear to describe during our presentation. I also plan on trying to set up a date for our group to come together and take a picture for our poster.

Week 9: This week we did our group design review. The design review went really good and we have gotten some feedback from other groups during the presentation. Some suggestions are out of the scope of our project as we handle the interactions between the homeowner and the cleaning services and another team handles the interactions between the homeowner and the cliented renting their home. We also met with our TA and gathered information regarding the remaining assignments. We have not run into any problems this week. We are still waiting to hear back from the team's official feedback. We are currently working on recording a demo of our assignments and starting to work on the ending assignments for this term.

Week 10: This week we are looking over feedback we received from our design reviews. We are wrapping up our demo video that will be submitted on Tuesday and beginning to put together data for our end of term report. We are also working out some bugs we have encountered with our application. We have not run into any problems this week, although due to the virus going around we are limiting physical meetings. We will fix any bugs with our application, edit our video and write out our report to submit on Tuesday. We plan to send our demo video to our client so that he may review and suggest any changes to the application. We then plan to wait to see what Spring has in store in regards to the class.

## IX. WEEKLY BLOG (KENNY)

### Fall

Week 3: So far, I think my team is making some decent progress despite starting on the project a bit late. Two of our team members were out sick for week 2 I believe so we fell a little behind on scheduling and meeting with our client. However, we are making quick progress and did get a chance to meet with our client before the requirements document is due. After we finish this doc, we should be able to move to the final problem statement fairly quickly. As for issues, I don't think we have any so far other than what was previously mentioned. Our plan, again, is to finish up the requirements document today, figure out which problem statement was the best and revise that as necessary, then potentially get started with the actual project by researching some cleaning companies and the workers' thoughts on voice technology and how it could be useful, or not. Also, some other group members could split and get a head start on learning the Alexa Skills Kit.

Week 4: Honestly, there wasn't really much going on this week. I'm pretty sure we are set to meet with our client again next week, so we'll be looking towards that. Otherwise, the only thing we've done this week is the second draft of the requirements document. Currently, there still isn't really any issues with the team or the project. The next thing to look to is the Tech Review, so we should probably get together and figure out the pieces that make the project and divide them. Not sure how we are going to do that because there doesn't seem to be many pieces to the project at the moment but we'll see.

Week 5: As of this moment, most of the team is working separately on the tech review with our selected pieces. He still haven't met with our client for the second time, I'm assuming they are still in a conference. Personally, I think that we have a major problem going on with communication. Near the beginning of this week, I asked the team about the tech review and how we would split the project but only Joe responded back. The next day we finally get a response from another group member but they told us they were gone and wouldn't be back until Thursday to help. Our last group member didn't say anything until the day of our weekly TA meeting. Luckily for them, the assignment was pushed back to Sunday but had the student in class not ask for an extension, I doubt we could've finished by Friday. Hopefully my team is up and working on the assignment. I guess the plan now is to finish the tech review and hopefully meet together to talk about the project more.

Week 6: This week was a bit busy for most of us because of midterms. However, we were able to get a meeting going this Thursday before the TA meeting to discuss how we were going to separate tasks for the upcoming assignment. Aside from that, I only received one comment on my first Tech Review draft so I attempted to beef up some sections as suggested and I improved my abstract because I thought it was weak and lacking. I think our client came back from their conference so we could set up a meeting with them soon if needed. They were able to connect us with the CEO of a cleaning company so we can contact them and begin interviews to bring insight on potential uses for our project. Currently no problems as of now. Just personal struggles with midterms but that should subside once the weekend hits. Our team needs to come up how to split the next assignment. So far we have some parts but we need to do some more brainstorming. Once we get that done we can do our individual researching.

Week 7: We were able to split up our project into sections for the design document and we turned in the first draft for this week. I haven't heard back from my teammate who emailed the CEO yet so I'm guessing they haven't replied back. So far no major issues. One small gripe is that we as a team don't communicate with each other often. For the next draft, we should add in an introduction, conclusion, and glossary minimum to the document. I don't know when we are going to talk about it or even start it.

Week 8: There hasn't been much progress from last week on the design document I don't think. I did go in on Thursday to work on some more sections – namely the Introduction and Glossary. Other than that I don't think anyone else has contributed. Lucky for us, I guess, the due date was moved but it's Friday and I'm still the only one to have made any changes since the first draft to my knowledge. The team is still not communicating as much and work is not getting done. Everything seems very last minute and it is stressful. I just want to get this design document done so I can worry about other assignments. HOPEFULLY we all meet up as a team like we planned on Thursday to discuss what we need to finish by tomorrow. So far, everyone has gone silent again so it seems very unlikely. Although these last sections we need to write in are supposed to be done as a team, if nobody works on them I might have to start annoying them and/or work on the rest myself. However, I am a bit busy tomorrow so we will have to see what happens.

Week 9: This week, there wasn't much activity going on. I believe that Joe finished off the rest of the Design Document last week and turned it in for the team. I completed the introduction and my part of the glossary last week. An email was sent to our client about our documentation. The same problem occurred this week as with last week: a lack of communication. Nobody seems to want to communicate and/or start up a conversation. Another interesting problem that came up was a

teammate forgetting about the due date of the assignment. We are just waiting for our client to respond back, hopefully with some feedback and eventually a sign-off.

### **Winter**

Week 1: Our team hasn't done much since the start of the term. Currently, I believe that we have sent an email and asked for hardware that is needed for our project. Again, nothing has happened so there haven't been any complications. Our team hasn't communicated any plans for the project. However, we should be going through and learning the language to build the application and/or starting the project sometime soon.

Week 2: Wednesday, after meeting with our TA, our group had a quick meeting to discuss how we would be moving forward with this project. We divided up the work and now we should be looking at how to develop the application and beginning work. I don't think there are any major problems so far. We are a little unsure if we need to buy a service or not to put our app in the cloud. Again, we are going on our own to learn how to develop our assigned parts of the project.

Week 3: So far, not too much progress has been made on the project personally. I don't know how it is for the rest of my team. I have been looking up more on how the skills work and how to create them, however, I haven't started implementing yet. A couple roadblocks have been keeping me from starting. This week especially, I've been trying to knock down all my assignments to make time to start working. However, everything has been time consuming. I even wrote this progress report late because I've been too focused on other things. Hopefully, I can manage my time better in the future, finish up my assignments and at least get started with the code by the end of this week. I should also check up on the team to see where they are at with the project.

Week 4: I think we are ready to showcase our application as alpha. The main functionalities should be in place and ready to go. No problems except for our teammate out for medical reasons still, but we have that sorted out. The plan should be to continue working on the project.

Week 5: I haven't made much progress since last week. I've been more focused on other assignments for other classes. Nothing much has happened since last week so no new issues. The plan is to just keep working on the project and improve the poster.

Week 6: I got permissions for our group's GitHub repository and uploaded my code onto that. Not much progress has been made on the application itself on my end. I'm not sure what to do at this point. The team isn't really communicating much and I feel like we need to talk to our client more. I have no idea if any progress has been made with the other pieces of the project. I need to contact everyone and see where we are at at this point and time for the project. We should also discuss how we are going to do our design review and if we need to make any adjustments.

Week 7: Currently, we are working to put all of our skills into a single Amazon Developer account to prepare for the Design Review. We also started a presentation, which I need to get my part in. Don't think there are any problems at the moment. We are planning to meet up sometime soon to go over the review and practice our presentation.

Week 8: We did our review this week and got some good feedback regarding our design. During the review, I also recorded it so maybe we can use it for our demo assignment. There haven't been any problems recently. The last things we need to do is the report and the demo video.

Week 9: We are working on the demo video at this moment. I have edited the video from when we recorded the design review and now I'm waiting on everyone to do their parts and sending it to me. We have not talked about the final document yet. Hopefully we are going to start/finish the document soon and everyone sends me in their part.

## X. WEEKLY BLOG (TSO-YI)

### Fall

Week 3: As compared to week 1 and 2, we had better progress this week. All group members have finally shown up in class (Sumegha and Carlos were both absent for the past two weeks due to illness). This week, we sat down and discussed the team standard that each of us should follow. The team standard document included items such as completing the required task to the fullest of one's ability (one hundred percent) before the deadline and replying to emails within a day. Furthermore, we finally met with our client, Dana Young, on Tuesday evening. To help us build a voice-driven application for Amazon Smart Home devices, he gave us resources on the Alexa skill kit. He also gave us guidelines on what the application should achieve by the end of the year. Overall, the meeting went smoothly and we have scheduled a meeting every two weeks to update Dana on our progress. Our group has been working on the requirements document as well as the group problem statement from Wednesday through Friday. I have noticed a lack of communication between team members. However, we have just started working together as a team and it may take some time working out a perfect system. We are communicating on a Slack channel to coordinate our work but it took Sumegha and Carlos almost a full day to reply me. For example, Kenny and I have completed both the Specific Requirement and Project Description sections on Thursday and asked if one of them have completed the Gantt chart but there was no reply from them until Friday afternoon. I plan to set up another short meeting after class on Tuesday to discuss about the lack of communication between teammates so we can work through the assignments better. I will ask them to download the slack mobile application on their phones and turn the notifications on or download the desktop application and turn the notifications on. Ideally, they can download the application on both their phones and laptops. Hopefully, this will resolve the communication issue.

Week 4: This week we are primarily working on refining our requirements document. Since we did not get much feedback from the peer review, we worked on the feedback we received from Aaron Didner.

Aaron Didner, Oct 19 at 11:57am (Feedback) "Abstract seems a little short. Seems good in general. Could add some more details about target audience. Also could add more about assumptions for this target audience or the situation in general"

We focused on improving the assumptions made for the target audience as well as increasing the details in the abstract section. I have also asked Sumegha and Kenny to turn on their notification settings for slack this week to improve group communication. They still seem shy to talk in the group chat but hopefully they will open up as the term goes by. We did not face any significant problem this week except trying to improve on our requirements statement. We plan on starting the software requirement this weekend as soon as we finish the requirements document on Friday. Information regarding how the group assignment goes will be included in next week's blog.

Week 5: The big task for this week is to come up with individual tech review. Each tech review should consist of three pieces that are essential to the group solution. For our project, we need to design and create a voice-driven application. The pieces that I chose for myself are language support, data capture, and organization of user interface (UI). I chose language support because this was one of the requests that Dana hope we can incorporate both Spanish and English support. This is because a portion of the cleaners in the cleaning industry are more comfortable speaking Spanish. Since we are primarily focusing on Amazon Alexa, we will need to know if the smart home device supports language other than English. If so, how well does it perform. The second piece (data capture) I picked is sort of linked with my first piece. I need to figure out how does the Amazon Alexa capture the verbal context from the speaker and how well can it perform in extreme condition such as high noise pollution setting. The third piece (organization of UI) needed to be researched because the team decided to implement a website to display the reported information from the cleaner back to accommodation owner. From now to Sunday, I will be focusing on the tech review draft. Prior to the weekly meeting with TA, our group did not schedule any group meeting, so Kenny and I just came up with our own three pieces based on what we wrote in requirements document. We posted what we are going to research on our slack channel and only Sumegha replied. She informed us that she is in Seattle this week and she will come up with her three pieces and let us know. Thankfully, Fariba was able to let the team know the importance of coming up with 12 pieces of tech review topic as a group. After the meeting with TA, I quickly gather everyone so see if Sumegha and Carlos needed assistance finding topics. During the meeting, I can tell that Sumegha is more involved in the discussion than before however not a lot of input from Carlos still. In the end of the meeting, I came up with additional 4-5 potential topics that Sumegha and Carlos can pick. My priority focus for this week and next week is to complete this tech review assignment (draft and final) to my full ability while also looking ahead to design document that is due next Friday. I have two midterms coming up, but I still hope to get at least 1-2 group meeting during the week (just like what we had this week) to come up with a working plan before each of one us just start to work on our own.

Week 6: Progress For this week, each of us are putting the finishing touches for the tech review. We also held a group meeting prior to the weekly TA meeting to discuss about our next group assignment. I wanted to divide the work up evenly before we made plans for the long weekend. Carlos and I went to the lecture on Thursday to check out sample final reports

from past years. After looking at their outlines, I realized our system can be divided into three general sections: component overview (software and hardware), user interface, and system requirements. Being a group of four, two members will be working on the same section together. I will pick the section after the other three teammates have chosen theirs. Dana has also sent us an email regarding a start-up cleaning company where we can contact the CEO in order to know more about their needs for the voice application software. I have sent the email and am waiting to hear back on her availability before scheduling a phone meeting. I am having problems getting back feedback from peer reviewers but Fariba has assured us that she will bring this issue to the instructor during their meeting next week. This should solve the issue and prevent it from happening again. My plan for next week is to work on the project design during the long weekend. My goal is to set an internal deadline (possibly two days before the actual due date) so we can go over the final document together during a phone or in-person meeting. I would like to make this assignment feel more like a actual group assignment unlike our previous assignment. I have definitely seen an improvement in communication on Slack but it can still be improved.

Week 7: I worked on project design for this week. After discussing with the team, my focus for the project is on the hardware component within our system. Since our project works mainly with Amazon Alexa, I was researching on how Alexa devices connect with software components and the users. I was having difficulties choosing what to focus on for the I hardware component but thankfully, the team stepped in and helped me identify the things I can talk about. Getting a flu has also caught me off guard but the team members also stepped in to help me during this week's TA meeting. My plan for the upcoming week is to improve on my hardware component sections and also to start combining the three documents that we have been working on for the whole term. I want to go over the whole document with the team so we can remove some redundant information and make it coherent to the reader.

Week 8: This week, our main task was revising and improving the design document that we will be sending out to our client next week. The client will have to approve our design document before we can proceed to the next stage. There was some problem regarding the IEEE template for this design document. Ms. Kirsten brought some design documents from previous years for us to check out and we followed those templates for our draft. During the weekly TA meeting however, we were told that the template was still wrong and it was missing a few sections. In the end, the TA allowed us to use our draft template so the problem was resolved. My plan for the upcoming week is to schedule a meeting with the client and present our design document to him before the thanksgiving holiday. Hopefully he will approve it.

Week 9: After completing the design document last Sunday, I have sent the final version to Dana, our client. I am still waiting to hear back from him to see if there are any changes that he would like us to make before signing off on the agreement due Dec/3. Since it is a long weekend and no assignments are due this week, I do not have any issues. My plan is to get a reply from Dana soon and he can then sign the agreement form. Another plan is to continue combining the three main documents into one coherent document. Therefore, it will be ready for submission before the end of the term.

## **Winter**

Week 1: This week we met up with our TA to discuss about what needs to be accomplished this term. Our two priority right now is to have our software presentable at alpha and beta functionality date. We set up group meetings to discuss what we need to achieve for winter term and split it into manageable portion for each member. The focus right now is for Kenny and I to complete the Alexa voice control portion, so the software can understand user's voice input and provide the output we want. The minor problem we faced this week is to divide the coding work evenly between the teammates and set a deadline for each portion. We are starting the coding portion late so we will have to race against the clock. The plan for this weekend and next week is for us to learn Amazon Alexa skill kit. Once Kenny and I have finished the course, we will start the coding for our assigned part.

Week 2: This week I have started to work on my part of coding, Alexa skill kit. I am coding on my own Amazon account, but the plan is to move everything to a newly created Amazon account (centralized) so everyone on the team can have access to it. This transition will probably happen on Week 4 or Week 5. So far, I have not experienced any major coding obstacle. Minor problem when I was coding but I was able to find solution/help using Google. My plan for the weekend and next week is to continue working on the Virtual concierge cleaner skill. I will probably meet up with Kenny (also in charge of developing skill kit) to discuss about the work, to keep us both updated.

Week 3: This week I am continuing working on Alexa Skill. I am currently in progress of completing the skill to report damage and requesting maintenance. Upon finishing the skill, I will combine with Kenny's skill of reporting cleaning progress and complete the Alexa Skill section of the project. We will turn merge with Carlos and Sumegha section. I got some bugs when coding for Alexa skill, but I am trying to figure it out with online resource like stackoverflow. My plan is to finish the entire Alexa skill by the end of next week.

Week 4: I have been testing my Alexa skill for the past couple days and improve the semantic of the skill. At the final stage of preparation before Alpha demonstration. It is quite difficult to find the best semantic that is easy for human to understand and learn while also keeping it simple. Trial and error with my roommates assistance are the best way to figure out the good semantic. My plan is to get everything done and upload my portion of the code to github repo by the Alpha demonstration next Friday.

Week 5: I worked on the project poster this week and try to incorporate as much materials from our design document, problem statement, and requirement document. I had to complete majority of the poster myself since I could not get in touch with my teammate this week

Problem I noticed about the due date for project poster too late, so I did not have the chance to split the work up among the team. I also have trouble getting in touch with Kenny and Sumegha this week. I think they are busy with midterm and family matters. I also have not heard back from Kirsten and Scott about our request to change the design review date to week8. Sumegha sent them an email after my initial email but no news since then. I plan on preparing for the design review that is currently scheduled for Team6 next Thursday.

Week 6: The group did not do much for the project outside of class this week. We got some idea from our TA on how the design review went for her. No problem occurs this week. Kirsten has rescheduled our design review to week 9 Tuesday so that problem was solved. The group is going to meet up next week to discuss how we will present during 30 minutes of design review.

Week 8: I have submitted our group's document to other team leaders for Mar/3 design review. So far, I have gotten reply from one team leaders, still waiting to hear back from the other 3 team leaders. Work on the presentation power point for the design review. There is no problem this week. We are meeting up this Sunday to do the final preparation for the presentation on Tuesday.

Week 9: We worked on the presentation and prepare the feedbacks for the other 4 teams who were in our design review. We were trying to combine our skills together and deploy it but there seems to be some compatibility issue, trying to solve it right now. We hope to combine everyone's work and use the video footage who recorded in our design review for beta functionality submission. We will use the footage plus some extra explanation for the final product.

## XI. WEEKLY BLOG (SUMEGHA)

### Fall

Week 3: I worked on the Tech Review Rough Draft this week and then also worked on my Final Tech Review. I don't see any peer review comments so I can't really improve my draft as much as I would like to. I plan on looking into which part of the Design document I want to do. Our group met about this and now we need to decide who wants to do what.

Week 6: I worked on the Tech Review Rough Draft this week and then also worked on my Final Tech Review. I don't see any peer review comments so I can't really improve my draft as much as I would like to. I plan on looking into which part of the Design document I want to do. Our group met about this and now we need to decide who wants to do what.

Week 8: We are making progress on the design doc. It has been hard to edit because there has been no comments or feedback on what we are doing. I don't know if what I am doing is right either. I just wrote about my section. The plan is to edit the doc and what the TA mentioned on Thursday on to the doc before tomorrow.

### Winter

Week 1: This week we haven't worked on much of the project so not much progress has been made. We are not sure where or when we are supposed to get our hardware. To reach out to the client and figure out TA meetings. We also plan on making a plan for the term.

Week 2: This week we met up and talked about tasks that each member will complete. The progress we made was making a plan. We talked about the term in general and what the goals were and then we talked about the things we wanted to complete this week as well as next week. Personally I am learning a lot about data storage and how to work with Alexa. I contacted Scott to get hardware ordered last week and I picked it up today. I also requested an AWS account so that our team can work on storage and Lambda functions. Currently there isn't a problem. We had the problem of not knowing how to get hardware but that got solved after we emailed the professors. We also weren't sure if we needed AWS accounts for a while but then we realized we do and got that figured out. Plans: The plan is to set up all the components so we know how everything will interact with each other this week and a little of next week. Then we are going to write code for the capturing voice component this week. I am personally going to be looking into storage with S3 buckets to see if I can get that going. I also need to look into databases because I think we will need one.

Week 3: This week we have set up our AWS accounts. This took the whole week. I had to email the IT Services many times for clarifications on permissions and account settings. The main problem for me was just looking up permissions we need on AWS. I have low prior experience to AWS and my teammates have none. So leading the permissions part of understanding our account was a challenge because it was a lot of guesses and vague key words that I remembered from my internship to guide us. The plan is to continue coding. I want to get a data integration component going by next week. As a group said we will complete. I am more in charge of data integration while others are in charge of voice integration and user feedback.

Week 5: I am not facing any problems but time management in general has been hard. It's hard to know what to finish when there is only one general hard deadline. My plan is to work with AWS integration more. It feels like when it comes time to connect it might not work. Because I am worried about that I just want to make sure it is ready with real testing. The AWS account works for integration. I haven't done much this week because I have been traveling internationally. I asked my teammates if there is more I can do but they said there wasn't more I can do.

Week 6: I have been working on AWS and Alexa Skill integration. The only problem we have been facing is scheduling our design review. Personally I am having problems with my wifi. I am currently out of the country and my wifi isn't as reliable as I hoped it would be. Our plan is to move our design review to week 9

Week 7: The plan is to integrate everyone's work into one Amazon account so that it is easier to manage. We have our intents implemented and have created a rough draft of our poster. We don't have any major problems. I flew back tonight from Nepal so that will help us with better communication and I am hoping to feel more involved.

Week 8: The end of the project feels more clear now so that feels like progress. We are good with our Alexa skills now which are also progressing. There is no problem we just need to get ready for the design review. We plan on meeting before our design review so that we know what we are going to talk about. I plan on working on my slides.

Week 9: The design review seemed to go well but I am not sure because we haven't gotten our grades back. There aren't any problems. We need to work on the next parts of the course. The plan is to figure out if we want to implement more features.

Week 10: We worked on our video that is due in a week. We uploaded this and recorded it for the design review. There weren't any problems, we are just working on figuring out times to meet and if we should meet in general. We plan on completing the final assignment in the next couple days.

## XII. FINAL POSTER

COLLEGE OF ENGINEERING

Electrical Engineering and Computer Science

TEAM 6

## Intro and Background

- When the guest check out of their reservations, traditional hotel employees are immediately informed and are able to begin the turnover process. The cleaning progress can be reported to hotel staff in real time
- Alternative accommodation hosts rely on either cleaning services or self cleaning after guest check out.
- If host rely on cleaning service, the turnover efficiency between guests will be low due to the time for cleaners need to travel across town.
- Virtual Concierge Service (VCS) is a company that delivers AI-powered voice technology to the hospitality industry. With the goal of positively impacting the hospitality industry.
  - They currently have products that allows Alexa to act as a virtual assistant that support guest's need such as how to operate the coffee maker or where to turn on the heater
- The goal for the project is to find out the host and cleaner needs in creating efficient turnover between guests and develop the necessary Alexa skill to provide the aid



## VIRTUAL CONCIERGE: ALEXA

Amazon Alexa & Alternative Accommodation Host:  
Incorporate Alexa skill into increase turnover efficiency by letting Airbnb host monitors the cleaning progress and maintenance request



### HOST'S NEEDS

Hospitality is one area where traditional hotels excel in while alternative accommodation such as Airbnb do not. With most hotels, extra services are provided for guests and can be quickly received/delivered with just a phone call.

Customer turnover rate between guests is a need that host would like to fix. The ability to monitor the cleaner's cleaning progress while not physically onsite gives the host the flexibility to take care of more emergency problem such as damage furnitures or maintenance request

### CLEANER'S NEEDS

- Cleaner should not need to learn additional skill on how to operate Alexa devices
- Ability to verbally report the cleaning progress, damages, or maintenance request without the need to slow down or stop current task
- Alexa skill has the flexibility of understand different semantic so the cleaners do not need to learn one standard way of reporting
- The ability to report to Alexa using English

### ALEXA SKILL KITS

- Ability to report to Alexa the start and finish time for the cleaning task
  - Reporting the completion of cleaning the room(s)
- Ability to report any damage item(s)/furniture(s) and where it locates to Alexa
- Ability to report any maintenance request and where the problem locates in the accommodation to Alexa
- Ability to send SMS text message to the alternative accommodation host when the cleaning has started and finished, damage items, and maintenance request
- Ability to store the information reported by the cleaner using S3
- Ability to report cleaning stats such as total cleaning duration after the task is complete using SMS text message



Dana Young(client): [young.danal@gmail.com](mailto:young.danal@gmail.com)

#### Team members:

Carlos Lopez-Molina: [lopezmoc@oregonstate.edu](mailto:lopezmoc@oregonstate.edu)

Sumegha Aryal: [aryalsu@oregonstate.edu](mailto:aryalsu@oregonstate.edu)

Tso-Yi (Joe) Chung: [chungts@oregonstate.edu](mailto:chungts@oregonstate.edu)

Kenny Luong: [luongk@oregonstate.edu](mailto:luongk@oregonstate.edu)

## XIII. PROJECT DOCUMENTATION

### A. Installation Guide

What you will need:

- Alexa Developer Console account
- Amazon Web Service Management Console account
- Twilio account

#### 1) How to Install:

- Set up Alexa Skill in developer console
- In the developer console create a new custom skill and name it "My Housekeeper".
- Under build click on "JSON editor" and paste in the contents of the [en-US.json] (<https://github.com/Carlos231/Airbnb-Voice-Driven-Amazon-Alexa-Application/blob/master/interactionModels/custom/en-US.json>) file to load all intents needed.
- Open the AWS Management Console and find the service "Lambda".
- Create two environment variables called, "TWILIO\_ACCOUNT\_SID" and "TWILIO\_AUTH\_TOKEN". You will place an appropriate Twilio account SID and token here.
- In the IAM service gives the lambda function permissions for Amazon SES and DynamoDB.
- In DynamoDB service create a new table "My\_HousekeeperDB" and add triggers to the lambda function.
- In Amazon SES service you will be started in sandbox mode so you will need to add a send to and receive email address (you will have to change this in the code before uploading as a zip)
- Create a new lambda function. Clone the AirbnbVoiceDrivenAmazonAlexaApplication directory and zip the node\_modules folder with index.js, package.json and package-lock.json and upload to your newly created lambda function.
- Keep this page open
- Create a twilio account and purchase a number to use for text messaging. Look for a Twilio SID and token to place in environment variables.
- Return to the Alexa Developer console.
- Click on Endpoint.
- Click on "AWS Lambda ARN" and paste your lambda ARN into the field to link the lambda function to Alexa skill
- Skill ID to be used in lambda function
- Click on "Permissions" in the sidebar and enable Device Address (Full Address),
- Customer Email Address and Customer Phone Number.
- Return to AWS lambda page
- Click on "Designer" drop down and click "Add trigger"
- Add trigger for Alexa Skills Kit and paste Skill ID previously copied
- Make sure there is a trigger for DynamoDB present and enabled as set up in earlier step
- You should be all set up to test your new My Housekeeper Alexa Skill

2) *How to Run and Test:* Application is in testing face so can only access it if have access to the Amazon account in which the application is being developed. Following the code review and code freeze the application will be submitted to Amazon in order to be reviewed, tested, and approved to go live on the Amazon marketplace. In addition, since we are in the testing face Email and SMS notifications will not be enabled as this requires each individual user to submit their email for us to add to Amazon SES due to still being in sandbox mode. Code for notifications will be provided.

- Permissions required: Full Address, Email, and Phone Number.
- Sign into <https://developer.amazon.com/>
- Click on My Housekeeper application
- Navigate to code section to review code
- Navigate to Test section to test the application

Send a report: The amazon developer portal does not support this function but we can access it through the Amazon Alexa Application. For more information on how to download the app click [here](#). The first thing you will need to do is enable permissions. You can do so by opening the Alexa app on your phone clicking the side menu bar to expand it and clicking on "Skills and Games". Next click on "Your Skills" and then look for the "My Housekeeper". In our case it will be under the "dev" menu since the application is still in development face. Once you see My Housekeeper, look for Settings and click on it to grant the My Housekeeper account permissions needed to send notifications to the account owner. Please enable: Device address, Email number, and mobile number. My Housekeeper will only use this information to send appropriate notifications as requested by the user and is never saved or shared to anyone.

#### Application intents:

Say, "Alexa, open my housekeeper". You can also type it in.

- This will launch the skill.
- Required before running below commands.

Say, "I am cleaning". You can also type it in.

- This will start/resume the process of cleaning a room.

Say, "Maintenance needed in {room}". You can also type it in.

- This will send a maintenance report.

Say, "Damage found in {room}". You can also type it in.

- This will send a issue report.

#### XIV. RECOMMENDED TECHNICAL RESOURCE

There are plenty of Amazon Alexa developing documentation that can be found on [developer.amazon.com](https://developer.amazon.com). The programmers that take over the project can continue with it using Node.js which is being taught in Oregon State University's web development class. Node.js debug help can also be found in popular coding discussion website like Stackoverflow. If the team is still stuck after going through these two options, they can go ask David Nguyen, software developer, at Dana's company. He is very patient and helpful with all the coding related questions we threw at him.

We also found two extremely useful websites that allowed us to not only learn about how to build the application, but also formatting and developer guidelines set by the developers. One website included the Amazon Developer website ([developer.amazon.com](https://developer.amazon.com)). Here there is a lot of tutorials, readings, and code samples instructing the programmer how they could go about building their application. We found this resource to be the best and most reliable as it is done by the company that created this technology. Another resource we found helpful was the Twilio companies website that also included a tutorial and a lot of documentation that allowed us to connect their service to our Alexa application. We would recommend both of these services to someone wanting to learn the technology or further their understanding.

## XV. CONCLUSIONS AND REFLECTIONS (CARLOS)

This project has been very educational for me. Coming into the project I had very minimal knowledge of the technology, but I was interested in its possible capabilities. I was glad our client knew what he wanted and was also aware of the technology. As a result of this project I was able to dive into the Amazon Alexa skills kit, the Amazon Web Services, and third party applications such as Twilio. In addition, I dived into learning the Alexa skill language, Node.js, and JSON languages to combine all of these technologies to work for our project. I was also able to learn how to work with a team and with a leading professional to meet their set requirements.

Some non-technical information I learned includes how to communicate with a client and work with meeting their needs. What I learned from project work is that communication is key to meeting deadlines and to creating the best product possible for the client. In addition, what I have learned about project management is that milestones with deadlines must be set to ensure the accuracy of when a project will be completed. I learned that working in teams allows us to nicely divide the work and allows each person to become a professional in one part of the project. In conclusion, if I could do it all over some things I would do differently is have design reviews and code reviews be done way sooner in the process and not when we essentially had to be done with the development phase.

## XVI. CONCLUSIONS AND REFLECTIONS (KENNY)

I thought this year was very interesting for me. Before the project, I had no clue about anything JavaScript related. Going into the project, I was tasked to create a component of the project using Node.js. In the beginning, I was very lost on how to start, but as the school year progressed, I started to understand it more and the surprising part was taking classes that related to or made use of JavaScript. This simplified the process of learning and understanding everything I needed for the project and schoolwork. Along with this, I was able to learn more about AWS Lambda, although it wasn't part of my focus.

Truthfully, this experience as a whole was very messy and I'm unsure if it was very beneficial. I think about the only thing that I got out of this was how to better manage myself with assignments. The whole process of gathering information during the first term was fine, but I feel like it was too much, especially for a project like ours which essentially had all the requirements and tools needed. Sure, it's good to look at alternatives and do some comparisons, it just doesn't need to take all term. To top that off, I learned that working in teams sucked and completely changed my view on grouped projects. Without good communication and structure, we were mainly running around like headless chickens, doing our own things. Everything felt rushed and last minute, and that feeling persists now.

If I could restart this capstone, I really wouldn't. But if I had to, I would definitely pester my team more and be more active/vocal, if possible/necessary. This was something I fully gave up on in the second and last term of the project, and I do regret it, but I felt like it wasn't worth it anymore.

## XVII. CONCLUSIONS AND REFLECTIONS (TSO-YI)

Before the beginning of this senior project, I did not have any idea on how smart speaker and the skills behind it work. In the first half of the fall term, the team had to do a lot of background research on how to create an Amazon Alexa skill from scratch. This research process and code implementation stage broaden my idea on how to create Alexa skill using Node.js. It always gave me an opportunity to work on front-end and back-end of the 'bigger' size software project that I did not experience from regular class work.

I learned that a team without good and constant communication can make task very difficult. It is difficult to get an idea on each member's progress for their part. I am not the kind of person to lead a team but I tried to be improve in this area throughout the senior project. I tried to be more vocal and facilitate the communication between each members. I know I did not do a great job in this area and there is definitely still room for improvement. Hopefully with the experience from working with 'unknown' software developers in senior project, it will better prepare myself to work in the industry.

If I could do senior project all over again, I would definitely improve on my communication with the teammates and get a better idea on how each member is doing for their part. I would also like to get started with the software implementation stage earlier in the year (Fall term) instead of focusing so much time and energy in documentation stage. I will try to set up at least 2-3 meeting slots per week for we can all update each other on their progress.

### XVIII. CONCLUSIONS AND REFLECTIONS (SUMEGHA)-PROJECT ARCHIVE

I learned a lot about AWS as a platform. I learned about the different services that they provide and how to integrate S3 with Lambda. I also learned a lot about Alexa. I didn't know that Amazon had a console for developers to work on along with documentation and many resources to navigate skill building. I also learned node.js and how to build a skill for Alexa.

I learned a lot about working in a team and communication. Communication and motivation became hard when classes switched to online. I noticed that everyone had different expectations of me. Initially, storage was fine with S3 but because communication was difficult, I honestly just found out this week that the notification component needed an actual database to support. From one person I would hear that X,Y, and Z needed support while from another person I would hear something different. I also think that the way things were split up could have been improved. It is easy to write the code for something and not understand how it should be stored and vice versa.

I also learned that I am not sure what I would do differently because I think it would have been better to create an environment where each teammate is supported but I don't know how to create or improve that.

## XIX. APPENDIX 1 (ESSENTIAL CODE LISTINGS)

```

// after user says they are cleaning, if the user is currently/has been cleaning a room, ask if they have finished cleaning that room
const CleaningRoomHandler = {
  canHandle(handlerInput) {

    const attributesManager = handlerInput.attributesManager;
    const sessionAttributes = attributesManager.getSessionAttributes() || {};

    const room = sessionAttributes.hasOwnProperty('currRoom') ? sessionAttributes.currRoom : 0;

    return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest'
      && Alexa.getIntentName(handlerInput.requestEnvelope) === 'CleanRoomIntent'
      && room;
  },
  handle(handlerInput) {

    const attributesManager = handlerInput.attributesManager;
    const sessionAttributes = attributesManager.getSessionAttributes() || {};

    const room = sessionAttributes.hasOwnProperty('currRoom') ? sessionAttributes.currRoom : 0;

    const speakOutput = `Welcome back! It looks like you were cleaning the ${room}. Did you finish cleaning that room?`;

    return handlerInput.responseBuilder
      .speak(speakOutput)
      .reprompt()
      .getResponse();
  }
};

// after listing the number of rooms to clean, ask which room they are cleaning
const PostNumRoomsHandler = {
  canHandle(handlerInput) {
    return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest'
      && Alexa.getIntentName(handlerInput.requestEnvelope) === 'NumRoomsIntent';
  },
  async handle(handlerInput) {
    // save the number of rooms from user input into the database first as numRooms
    const numRooms = handlerInput.requestEnvelope.request.intent.slots.number.value;

    const attributesManager = handlerInput.attributesManager;
    const sessionAttributes = attributesManager.getSessionAttributes() || {};

    const attribute = {
      "numRooms" : numRooms
    }
  }
};

// after user says which room they are cleaning, save room name as current room being cleaned and allow user to make a damage/maintenance request
const CaptureRoomIntentHandler = {
  canHandle(handlerInput) {
    return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest'
      && Alexa.getIntentName(handlerInput.requestEnvelope) === 'ReportCleaningIntent';
  },
  async handle(handlerInput) {
    // the room name is saved to the database, checking to see if there are any cleaned rooms saved and acting accordingly
    const room = handlerInput.requestEnvelope.request.intent.slots.room.value;

    const attributesManager = handlerInput.attributesManager;
    const sessionAttributes = attributesManager.getSessionAttributes() || {};
    var roomAttribute;
    var num = 0;

    if (sessionAttributes.hasOwnProperty('rooms')) {
      roomAttribute = {
        "numRooms" : sessionAttributes.numRooms,
        "currRoom" : room,
        "rooms" : sessionAttributes.rooms
      };
      num = sessionAttributes.rooms.length;
    } else {
      roomAttribute = {
        "numRooms" : sessionAttributes.numRooms,
        "currRoom" : room
      };
    }

    attributesManager.setPersistentAttributes(roomAttribute);
    await attributesManager.savePersistentAttributes();

    const speakOutput = `Ok, starting to clean the ${room}. If there is a damage or maintenance request, please report it now.`;
    const repromptText = `Once again, if there is a damage or maintenance request, please report it now.`;
    return handlerInput.responseBuilder
      .speak(speakOutput)
      .getResponse();
  }
};

```

```

// list of permissions granted by user
//address, email and phone number saved in alexa devices account
const ADDR_PERMISSIONS = ['read::alexa:device:all:address'];
const PERMISSIONS = ['alexa::profile:email:read', 'alexa::profile:mobile_number:read'];

// Credentials saved in Environment variables in lambda
const AWS_ACCESS_KEY_ID = process.env.AWS_ACCESS_KEY_ID;
const AWS_SECRET_ACCESS_KEY = process.env.AWS_SECRET_ACCESS_KEY;
const accountSid = process.env.TWILIO_ACCOUNT_SID;
const authToken = process.env.TWILIO_AUTH_TOKEN;
const fromNumber = '+16235522205';

let https = require('https');
let queryString = require('querystring');

// introduction for the skill, asks the user if they are cleaning or to make a damage/maintenance report
const LaunchRequestHandler = {
  canHandle(handlerInput) {
    return handlerInput.requestEnvelope.request.type === 'LaunchRequest';
  },
  handle(handlerInput) {
    const speakOutput = 'Hello! Welcome to My Housekeeper. Are you cleaning a room, or would you like to make a damage or r
    const repromptText = 'Sorry, I did not catch that. Are you cleaning a room, or would you like to make a damage or maint
    return handlerInput.responseBuilder
      .speak(speakOutput)
      .reprompt(repromptText)
      .getResponse();
  }
};

function sendEmail(template, emailTo, from, subject, replyEmail) {
  // parameters to pass to API
  const params = {
    Destination: {
      CcAddresses: [],
      ToAddresses: [emailTo],
    },
    Message: {
      Body: {
        Html: {
          Charset: 'UTF-8',
          Data: template,
        },
        Text: {
          Charset: 'UTF-8',
          Data: '',
        },
      },
      Subject: {
        Charset: 'UTF-8',
        Data: subject,
      },
    },
    Source: from,
    ReplyToAddresses: [replyEmail],
  };
  // return new ses object sending the email

```

```

// Send Report
// checks for and gets user permission to retrieve their information and then sends appropriate notifications
const SendReport = {
  canHandle(handlerInput) {
    const { request } = handlerInput.requestEnvelope;

    return request.type === 'IntentRequest' && request.intent.name === 'sendReport';
  },
  async handle(handlerInput) {
    const { requestEnvelope, serviceClientFactory, responseBuilder } = handlerInput;

    const consentToken = requestEnvelope.context.System.user.permissions &&
      requestEnvelope.context.System.user.permissions.consentToken && requestEnvelope.context.System.apiAccessToken;
    if (!consentToken) {
      return responseBuilder
        .speak("Please enable Location permissions in the Amazon Alexa app.")
        .withAskForPermissionsConsentCard(PERMISSIONS)
        .getResponse();
    }
    try {
      const { deviceId } = requestEnvelope.context.System.device;
      const deviceAddressServiceClient = serviceClientFactory.getDeviceAddressServiceClient();
      const client = serviceClientFactory.getUpsServiceClient();

      // Retrieve address
      const address = await deviceAddressServiceClient.getFullAddress(deviceId);

      // Retrieve phone number
      const number = await client.getProfileMobileNumber();
    }
  }
};

```

## XX. APPENDIX 2 (CODE REVIEW FEEDBACK)

**Harsh's feedback**

Category	Description	Reviewers Comment	Action taken
Build	Could you clone from Git and build using the README file?	Yes I could clone the GitHub but the files were not given on time. Apart from that, there was a package.jsonfile that allowed me to install all the modules.	Uploading all files and making sure they're updated
Legibility	Was the flow sane and were variable names and methods easy to follow? Does the code adhere to general guidelines and code style?	The code was sane and easy to follow. Variable and function names were good, some functions (might be useless) did not have documentation on what the function did, documenting those would be great.	Add more documentation on parts of the code that might need explanation
Implementation	is it shorter/easier/faster/cleaner/safer to write functionally equivalent code? Do you see useful abstractions?	No, I feel the current code quality is great.	n/a
Maintainability	Are there unit tests? Should there be? Are the test covering interesting cases? Are they readable?	I couldn't find unit tests. Unit tests for this do not seem necessary.	n/a
Requirements	Does the code fulfill the requirements?	Yes the code seems to fulfill the requirements for the English part. The team is going to work on the Spanish part of the system for which they are in talk with their clients	Our application only recognizes the english language.
Other	Are there other things that stand out that can be improved?	Something cool would be feedback to a system that can be sort of used as a unit test for feedback on system performance (There might be a function for this that I missed though) like ("Did you get what you wanted" for testing purposes)	Testing could be done manually within the 'Test' tab in the developer console to check for appropriate responses from Alexa

## Evan Newman's feedback

Category	Description	Reviewers Comment	Action taken
Build	Could you clone from Git and build using the README file?	Yes, I could clone the repo and build it from github. The readme file was instructive enough to guide me through the setup process easily.	Update the file if needed
Legibility	Was the flow sane and were variable names and methods easy to follow? Does the code adhere to general guidelines and code style?	The overall flow of the program was well done. Each section had a specific purpose and wasn't dipping into doing multiple features. The naming was clear and inline with what the variables needed to do.	Check to make sure all variable names are clear, if any
Implementation	is it shorter/easier/faster/cleaner/safer to write functionally equivalent code? Do you see useful abstractions?	I think the code was built in a way to maintain the most simplicity while also providing fast execution. I don't have any huge changes that I would make considering they are using different libraries that I'm unfamiliar with. I think the abstractions they provided with the handle and canHandle lines gave nice clarity of its purpose.	n/a
Maintainability	Are there unit tests? Should there be? Are the test covering interesting cases? Are they readable?	There are unit tests that check to make sure that Alexa is responding appropriately to an answer. If she cannot handle a response it is checked for every time and either the response is repeated or moved on. The unit tests are extremely readable.	n/a
Requirements	Does the code fulfill the requirements?	The code fulfills the requirements that the group set out to accomplish. I think that they could add in a spanish version as well, but this isn't a hard requirement for them.	Our application only recognizes the english language.
Other	Are there other things that stand out that can be improved?	I would probably suggest a couple more comments in the code. Just general descriptions above functions so people get an idea what that part of the code is talking about.	Add small description before each function and short line of comment within the function

## Yingfeng Fang's feedback

Category	Description	Reviewers Comment	Action taken
Build	Could you clone from Git and build using the README file?	The code is used to build a voice application based on Amazon Alexa service, and it has some programs hosted on AWS. These programs require the relevant authorization to use, so we can not use the code to build the program directly. According to the README, we can only test the application in sandbox mode.	For the code review and testing we were able to add a users email to access our application for testing but once our skill goes live through our clients account then anyone will be able to test it.
Legibility	Was the flow sane and were variable names and methods easy to follow? Does the code adhere to general guidelines and code style?	The flow of the code is reasonable. The program is a voice application, which needs to reply and process the user's words. The flow of the program is consistent with the general dialogue flow. According to the variable name and comments, we can understand the function of the equation. In addition, the code uses some of the objects and suites built into Amazon Alexa, but I think it follows general guidelines and code styles.	n/a
Implementation	is it shorter/easier/faster/cleaner/safer to write functionally equivalent code? Do you see useful abstractions?	Because this code uses many kits provided by the Amazon service, I cannot comment on the writing of the function. But I found that the same const statements are used in many functions. For example, const attributesManager is used in GetNumRoomsHandler, CleaningRoomHandler and PostNumRoomsHandler. By globally setting these statements, the code may be shorter. In addition, I saw useful abstractions. The CS6 team set some constants at the beginning of the file and used them multiple times in the code that follows.	Attributed manager is a variable declared in each function following the alexa skills kit guidelines.
Maintainability	Are there unit tests? Should there be? Are the test covering interesting cases? Are they readable?	Yes, the CS6 team conducted some unit tests. Because the program can currently only be tested in sandbox mode, they get feedback by directly entering commands. For example, when the program asks "How many rooms are you cleaning today", the user replies with a number, and the program will respond. These codes are readable, we can understand the function of the program through observation.	n/a
Requirements	Does the code fulfill the requirements?	I think the code meets the general requirements. But because the program is a voice application, I can't determine its specific implementation.	Voice application to be used with Amazon Alexa capable devices.
Other	Are there other things that stand out that can be improved?	I found that the program cannot recognize Arabic numerals and has high requirements for the accuracy of the reply. I think these aspects can be improved.	Our application only recognizes the english language.

### Jiaxiang Gu's feedback

Category	Description	Reviewers Comment	Action taken
Build	Could you clone from Git and build using the README file?	Yes, After cloning their files, I followed the instructions in README to build the program.	n/a
Legibility	Was the flow sane and were variable names and methods easy to follow? Does the code adhere to general guidelines and code style?	Yes, the whole procedure is very clear and complete. I can clearly know what I need to do during the review process.	n/a
Implementation	is it shorter/easier/faster/cleaner/safer to write functionally equivalent code? Do you see useful abstractions?	I think they did a pretty good job, they used a lot of things that I don't quite know how to build, so I'm not sure if I can do a better simplification.	n/a
Maintainability	Are there unit tests? Should there be? Are the test covering interesting cases? Are they readable?	They set up very detailed unit tests for Alexa. Repeated tests ensure that the program runs smoothly and correctly.	n/a
Requirements	Does the code fulfill the requirements?	Yes, I looked at their design file, and the code met their goals and all their requirements.	n/a
Other	Are there other things that stand out that can be improved?	I might suggest that they give a little more detail on each js and json file that appears.	More comments added to functions and descriptions.