Harsh's feedback

Category	Description	Reviewers Comment	Action taken by reviewed group
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.json file 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	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	Testing could be done manually within the 'Test' tab in the developer console to

	that can be improved?	function for this that I missed though) like ("Did you get what you wanted" for testing purposes)	check for appropriate responses from Alexa

Anand's feedback

Build -

Yes I was able to clone your repository and the README was very helpful. Especially for being able to go test it on the amazon developer site.

Uploading all files and make sure they are up-to-date

Legibility -

Yes, very clean code, formatted correctly with good spacing makes the code very easy to read. Variables are labeled consistently which is nice, I like the comments at the top of the functions, but more comments to explain different parts of the internal mechanisms would be beneficial. I like that you used semicolons it improves your readability, even though they aren't required in javascript.

Include short description before each functions and add short comments within the function

Implementation -

The implementation seems well thought out and efficient. I think since a lot of the code is just returning strings, you could have a file for just storing the strings and call them in index.js through variables to make the code cleaner. This is a practice used in android development and it could be effective for this development as well. I think you all should improve your understanding on how sessions work, because having someone say how many rooms before adds no value. In fact it also causes confusion that someone should have to talk to an Alexa in between every room, which would make the cleaning process much slower. Instead if they just were allowed to keep adding to the same instance of a report and then at the end just say "done cleaning" to let the script know it should send the report, it would be more efficient.

We believe they are not understanding Alexa skill sessions entirely. A session occurs when they invoke the skill and ends depending on a parameter after each handler. In any case, sessions will end if explicitly told to or if the user fails to respond in time. Therefore, they would have to invoke the skill every time, regardless. As for their comment on implementation, yes we could remove asking for the number of rooms, but we are keeping it as it informs anyone reviewing the data how many rooms a cleaner initially set out to clean, and how many they actually cleaned -- this data could be useful for a boss, supervisor, or home owner. As with the reporting, that was a consideration early in the planning process, however, our current implementation checks to make sure the user is certain they are done cleaning a room before adding it into the database; sacrificing some efficiency for reliability.

Switch up the method for starting room cleaning? From asking the user for total room to letting the user add room?

Maintainability -

There aren't any unit tests present which is a bit concerning given that the inputs for this application is extremely variable. Also it was alarming that it wouldn't respond at all when you gave false inputs, that is something extremely critical, because not providing feedback is the easiest way to cause confusion and frustration for consumers. I guess using the testing environment provided by Amazon is an alternative to unit tests, but as it was stated this environment is unpredictable and not representative of the real Alexa, which leads me to believe that you should be having more test cases, and should be testing this on real Alexa devices.

Set up unit tests using "BrainMacInosh's Alexa-Skill-Test-Framework".

https://developer.amazon.com/blogs/alexa/post/35bdad3d-57c8-4623-88c6-815540697af5/unit-testing-create-functional-alexa-skills

Requirements -

The requirements section states your project will work for both spanish and english, and this is yet to be implemented so the project does not fully fulfill the requirements. I believe this is a critical part as the testing procedures to ensure this bilingual capability works would be very time consuming.

Our application only recognizes the english language.

Other -

I think this group did a great job working on a product that can be very useful and bring value to its consumers. Overall clean light weight code, which I believe will be effective when completed.

Evan Newman's feedback

Category	Description	Reviewers Comment	Action taken by reviewed group
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.
Others	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 by reviewed group
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/clean er/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, CleaningRoomHandlera nd PostNumRoomsHandle r. 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.
Others	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 by reviewed group
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/clean er/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

Others	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.