**Senior Design Final Report**

Remote Home Assist



Version 1.0

5/10/2019

**Team Members :**

Andy Xie

Melissa Cheung

Imelda Flores

Angelo Onato

Christopher Ortiz

**Faculty Advisor:**

Dr. Mohammad Pourhomayoun

**Liaisons**:

Edwin Cheng, Mike Agostinelli

**Table of Contents**

1. Introduction 2

1.1. Background 2

1.2. Limitations and Challenges 2

1.3. Design Principles 3

1.4. Design Benefits 3

1.5. Achievements 3

2. Related works and technologies 5

2.1 Existing Solutions 5

2.2 Techniques 5

3. System Architecture 6

3.1. Overview 6

3.2. Data Flow 7

4. Results and Conclusions 8

4.1 Results 8

4.2 Future 9

5. References 10

**1. Introduction**

**1.1 Background:**

Workforce Development, Aging, and Community Services (WDACS) is a Los Angeles County department whose mission is to support the elder population in Los Angeles County. WDACS provides services and programs to help the elder population. Some services include: assistance with home chores or personal care. WDACS offers meals in WDACS senior centers or parks around Los Angeles County. A program offered by WDACS is home-delivered meals to homes of senior citizens that cannot pick up their meal. This program also helps with check-ups. The person will check on the patient’s well being and safety. Another program provided by WDACS is Adult Protective Services (APS) a 24 hour service program provided as a hotline for elderly adults to report any risk situation. There are multiple other services and programs that WDACS provide to help and support the elderly people in Los Angeles County.

Remote Home Assist is a technology approach to help older adults be independent. A Remote Home Assist project with advanced and interactive technology that will be evolving over time to meet requirements. The system will be most effective to older adults with limited dexterity or people with vision impairments. For this project, Echo Dot will be used to provide services to the clients.

The objectives of the Remote Home Assist is to provide maximum independence to older adults and help achieve a secure and safe manner of independence for older adults. To improve the safety and security of older adults. To provide continuous care to vulnerable older adults.

The Remote Home Assist is a new system that will be part of a larger system. It will collect health information of a patient and upload the information to a website. Data will be collected by user voice input to Alexa. All the data used will be updated in real time. It will allow patients to enter an event to their calendar and set a reminder before the event.

**1.2 Limitations and Challenges:**

The limitations of our solution are that Alexa only accepts numbers. A patient may not be able to describe their health status based on numbers. They may want to answer with a ‘yes’ or ‘no’ instead. The patient may have trouble rating the pain. However, the numbers help gather a better sense of health. A higher number indicating high pain will help the caregiver have a better understanding. A ‘yes’ or ‘no’ answer is broad and provides little understanding of the patient’s pain.

**1.3 Design Principles:**

The lambda function is the most important part of the design. Alexa has many built-in functions. An Alexa skill needs to be created to use Alexa with the functionalities needed for our project. In the skill, functions are created in Amazon’s AWS Lambda. The function is written in the programming language NodeJS. The AWS Lambda function created for Alexa was able to store data into a PHP database, error checking patient input information, allowed a user to log into Alexa using other accounts, along with other functionalities. The code for the functionalities needs to be descriptive in order for other developers to continue the project. Another main design is the Amazon developer console. The developer console mostly includes the acceptable answers or input from the user. Also, the Alexa response to the input. The AWS Lambda needs to connect to Amazon’s developer console.

**1.4 Design Benefits:**

The benefits of the design are that it allows for functionality expansion. The developer will need to know NodeJS to be able to add functions to the AWS Lambda and understand the functions of the current code. Using AWS Lambda, any function can be added to the skill to be able to use with Alexa. This will allow for the developer to expand the capabilities of Alexa to help a patient. The abilities for Alexa would expand based on a user’s need. The AWS Lambda has no limits on expansion and the main capability, storing the information into a PHP database, does not need to change. The PHP database does not have a limit of information to store and it will not erase, or override information entered through Alexa.

The benefit of the developer console is that it can also expand with the expansion of AWS Lambda and based on the developer’s addition to the AWS Lambda. The developer console is easy to understand, and modifications are easy and simple. The developer console can expand and has no limits to expansion.

**1.5 Achievements:**

This year our project involved working with Alexa to help monitor the health

status of elderly patients. Alexa is able to collect the health information of a patient and upload the information to a PHP database. The data includes questions regarding the patient's’ health status, such as trouble sleeping or pain. The answers accepted will need to be between 1 and 10. If the user’s input a number is not within the range, Alexa will ask the user again until a number is entered in that range. The user can pause if he/she cannot answer the question. The user is also allowed to skip if they do not want to answer the question. Data will be collected by user voice input to Alexa. The data is stored in the PHP database permanently. If the question was skipped, it will be saved with a ‘skip’ value. All the data used will be updated in real time. The data is only be accessible to doctor’s or caregivers, not users.

Users can log into Alexa using other accounts, such as Google Gmail. Alexa will then create a unique identification number based on the information used to log in. Once logged in, the user can create an appointment and set a reminder. The user can select a date for an appointment. Then Alexa will ask the user if he/she wants to set a reminder. Alexa will remind the user 24-hours before the scheduled appointment.

**2. Related Works and Technology**

**2.1 Existing Solutions**

WDACS is developing services and support using advanced technology to help older adults be independent. A Remote Home Assist project with advanced and interactive technology that will be evolving over time to meet requirements. The system will be most effective to older adults with limited dexterity or people with vision impairments. For this project, Echo Dot or Alexa Show will be used to provide services to the clients.

The objectives of the Remote Home Assist is to provide maximum independence to older adults and help achieve a secure and safe manner of independence for older adults. To improve the safety and security of older adults. To provide continuous care to vulnerable older adults.

**2.2 Techniques**

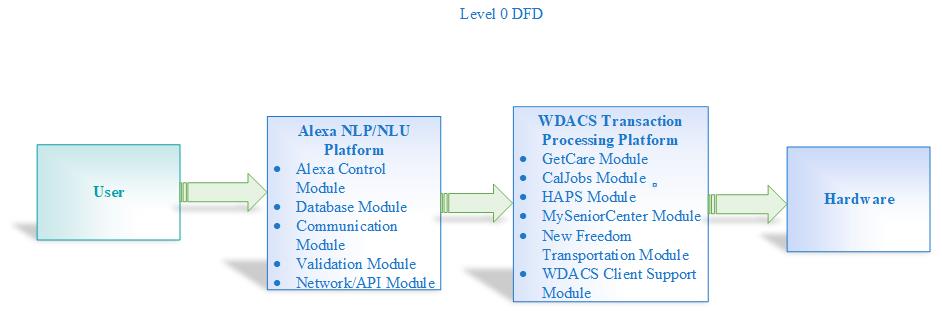
The Remote Home Assist is a new system that will be part of a larger system. It will collect health information of a patient and upload the information to a website. Data will be collected by user voice input to Alexa. All the data used will be updated in real time. It will allow patients to enter an event to their calendar and set a reminder before the event.

**3. System Architecture**

**3.1 Overview:**

Alexa’s architecture is described below with the factors being: User, Alexa NLP/NLU Platform, WDACS Transaction Processing Platform and hardware.

The following diagram is a DFD Level 0 diagram:



**User:**

The user is an essential factor since the design will affect the user’s experience when using Alexa. The user should not have problems when using Alexa. Alexa needs to understand the user’s voice orders to complete the user’s request. The user’s request should be processed without delay.

**Alexa NLP/NLU Platform:**

This is the main factor in the project. Amazon Alexa’s Natural Language Processing (NLP) / Natural Language Understanding (NLU) platform allows users to perform requests. This module is provided by Amazon’s API and it allows for the processing of natural languages using a Alexa device like the Echo Dot or Echo Show. If product skill within the alexa is activated by the user, greet user and await for further commands from the user again. If user utters a command and it matches any of the intents within the skill, execute intent for user to interact with. If user utters a command and no matches were made with current existing intents, prompt the user all of the intents of the skill. If intents are executed successful the user will receive a confirmation of the data entered. User can also stop the skill entirely in which the whole entire skill will save any progress made before closing.

**WDACS Transaction Processing Platform:**

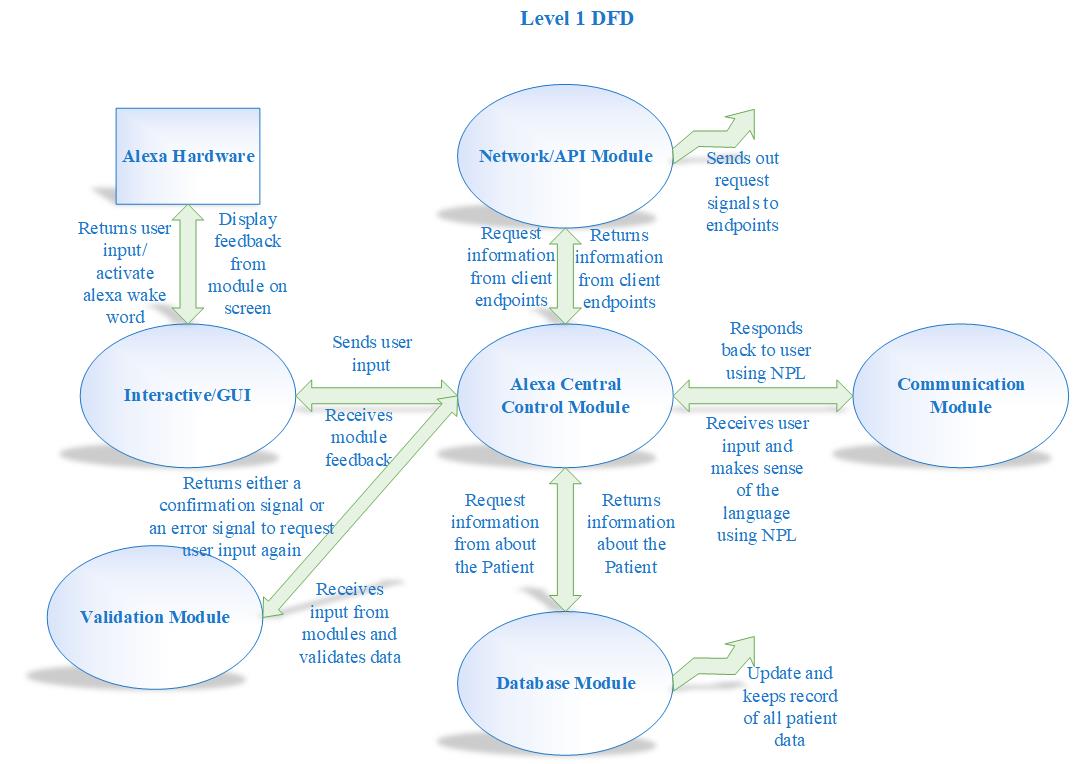
The WDACS Transaction Processing Platform processes the data received from Alexa’s NLP/NLU platform and the vendor cloud module. This module handles requests from the user, stores the data from the requests, and returns the appropriate service back to the user.

**Hardware:**

Alexa is the hardware. Alexa will process the user’s request and complete the request based using the NLP/NLU platform. The WDACS transaction processing platform will provide information to Alexa based on the request. Alexa will then respond to the user.

**3.2 Data Flow:**

The following diagram is a DFD Level 1 diagram showing the connected system

****

**Interactive/GUI:**

The interactive module functionality is when a user says Alexa’s ‘wake’ word. Alexa will be activated and receive the user’s request. It will return feedback from the processed request. The user can request anything, such as record health scores or set up a reminder for an appointment.

**Alexa Central Control Module:**

This module’s functionality includes sending user input and receiving feedback from the developer console. Then, sends it to the interactive module. It will request user information from the database module. Once the information is received it will be sent to the interactive module to return to the user. It also receive input from the communication module and make sense of it using Alexa’s NLP.

**Network/API Module:**

The network module connects the developer console to the AWS Lambda with endpoints. If they are not connected, Alexa will not have any functionality.

**Database Module:**

This module stores the user’s information. It will update the information daily based on user’s input. The information will not be deleted or overwritten. All records will be saved and stored accordingly. If needed it will send the information to the user if requested.

**Validation Module:**

The validation module receives the input from the Alexa central control module. Once the information is received, this module will validate the information. For example, Alexa will expect an answer from 1 to 10. If the user responds 11, this module will return an error message. The message will instruct the user to answer again. The user’s answer will again be validated. If there is no error, the module will send a confirmation to the user.

**Communication Module:**

This module takes in user input and changes it into a language Alexa can understand using NLP. Once the request is completed, the language Alexa uses will be changed to a language the user understands using this module. Everytime a request is sent this module will convert it back and forth between Alexa and the user to have an understanding of the requests. Without this module, Alexa will not understand anything the user says.

**4. Results and Conclusions**

**4.1 Results:**

A variety of implementations were made to complete the project requirements. Alexa was able to record health scores, store the data, and set reminders, among other things. AWS Lambda and developer console needed all the implementation. Both allow for growth of Alexa function without limits.

A lambda function was created with methods to connect to Alexa. This allowed for Alexa to successfully retrieve data from the patient. This function can be expanded at the developer’s needs and further advance the use of Alexa to help elderly patients.

The lambda function connected into the PHP database and stored the collected data. The data will be permanently stored and update the data in real time. The data will not be deleted or modified once updated.

At first the data was stored into Amazon’s DynamoDB. DynamoDB did not satisfied the requirements. The data was not stored permanently. Every time a new data was entered, it will be overwritten. Also, there was no timestamp of when the data was entered. The PHP database met the requirements and more. It is secure and no limit on size data. It was interesting to have Alexa sign in with other accounts. Since Alexa is only from Amazon, it was interesting when we were able to login using Gmail.

In conclusion, the lambda function and developer console are the main parts of the project. Connected together allowed for Alexa to function properly and meet requirements. The implementation allow for easy access and simple for developers to understand Alexa’s main functions.

**4.2 Future:**

Some implementations were made in this project in order to fulfill a certain requirement. Eve though they were able to fulfill those task, there is still room for improvement as they lack certain features. For the future of this project, those missing features will be added into the lambda functions. In addition, new features will also be added in order to fulfill the remaining requirements.

For account linking, more options will be allowed for linking other accounts. At the moment, the skill only allows amazon and google accounts. With Auth0, the skill can allow other account to be logged in such as facebook or yahoo. This will be beneficial as the user may not have a google account but instead has a yahoo..

Reminders was built into the project, but it only sets up a reminder 24 hours before the appointment. This feature can be expanded on by allowing users to delete or update their reminder. This can be done once the reminder is initially created by accessing it’s reminder token from the JSON response. Once the token has been obtain, it can be placed onto a database for patients to modify it in the future. This can be achieved by taking a key term such as the date and time, then Alexa will retrieve the token tied to that date. As stated earlier, these tokens will be placed into a database as the reminders functions only create a reminder on patients alexa app. Placing these tokens onto a database will help create the real appointment.

New features that are to be added in the future is the use of external health monitoring devices. Knowing how the Alexa can access data through an external API of an third party, the skill will be able to access data from devices from Fitbit. With this in mind, Alexa can collect health data from the Fitbit API in order to assist clients with their health. By implementing data collection from the API and machine learning algorithms, Alexa will be able to assist patients with their daily routines. By using machine learning, Alexa will be able to use data from the fitbit and suggest ways for patients to maintain a healthy health style.

**5. References**

Los Angeles County Workforce Development, Aging and Community Services

FISCAL YEAR 2019 - 2020

SDKv1 Adaptor <https://ask-sdk-for-nodejs.readthedocs.io/en/latest/ASK-SDK-Migration-Guide.html>

Alexa Reminders API

<https://developer.amazon.com/docs/smapi/alexa-reminders-api-reference.html>

Auth0 Tutorial

<https://www.jovo.tech/tutorials/alexa-account-linking-auth0>