COMP4968 Course Project

Alexa BCIT Contact Book

# Team Members:

Yuheng(Hugh) Song A00971421

Haihua(Ryan) Tan A00950721

Junnan Tang A00980931

# Building an Alexa Contact Book Skill

## Summary

The function of our skill is to help people check the phone number of public departments and instructors in BCIT. The phone number of public departments can be obtained by everyone. However, instructors’ phone number is instructors’ personal information, so these phone numbers can only be checked by BCIT students. Therefore, we set up the authorization function to determine the identity of users.

We have used lambda function and dynamoDB in AWS and some nodejs libraries to build the features.

**Intents**

We separate our intents into two parts: searching & authorization

searching:

* searchPublicContactIntent slot:(departmentName)
* searchPrivateContactIntent slot:(personName)

authorization:

* sendVerificationCodeIntent slot:(studentId)
* getVerificationCodeIntent slot(verificationCode)

And some of Amazon predefined intents.

**Database Scheme**

1. Table: departments

Store the public available contacts.

Data scheme: [name, phone, location]

1. Table: faculty

Store the authentication-required contacts.

Data scheme: [name, phone]

1. Table: student

Store the student information for authentication.

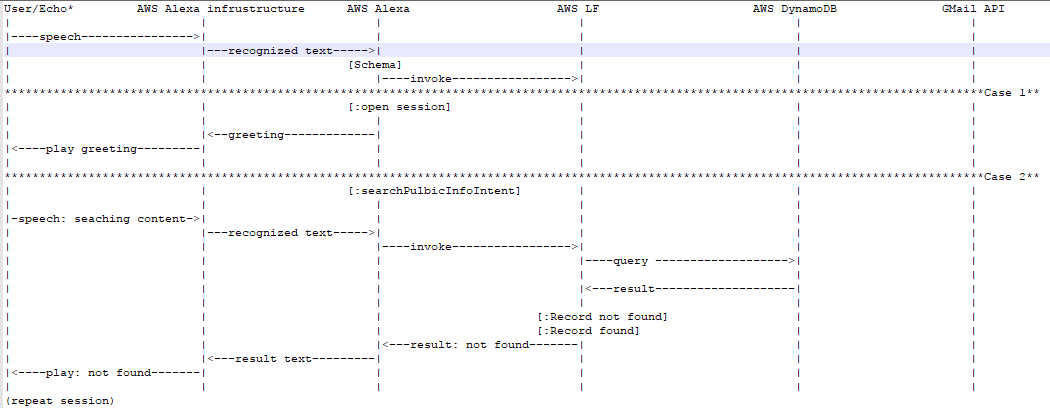
Data scheme: [id, name, email]

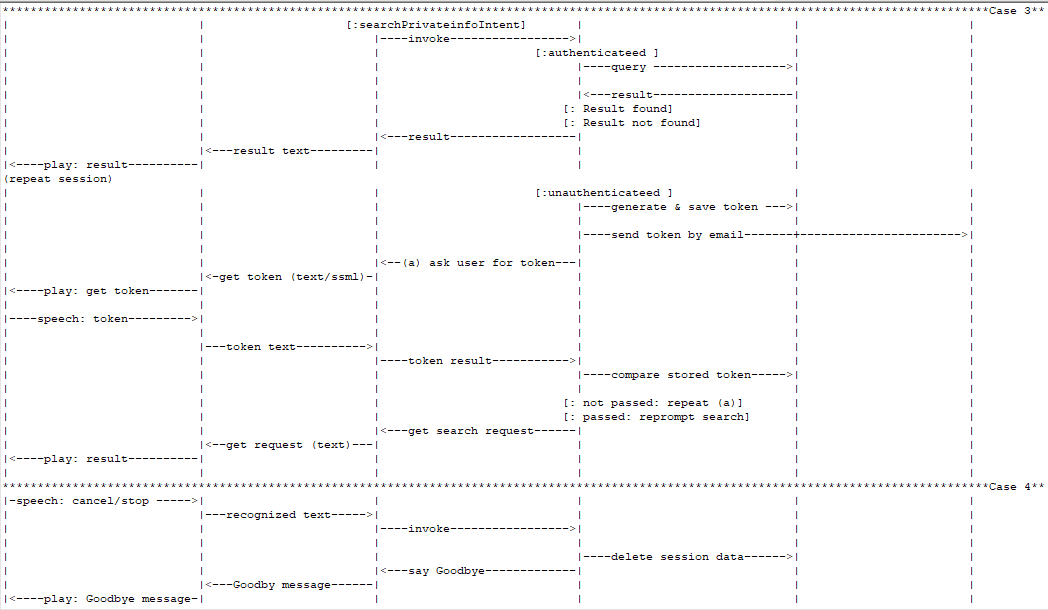
1. Table: session

Store the session data, authentication status and verification code (encrypted)

Data scheme: [id, authenticated, verifycode]

## Flowchart





# 

# Libraries used in the project

|  |  |
| --- | --- |
| **Name** | **Description** |
| alexa-sdk | Alexa Skills Kit SDK for Node.js |
| bcryptjs | encrypts and compares tokens for verification |
| bluebird | Promisify support to non-promised functions |
| gmail-send | A library to send email with Gmail account |

# How to Invoke the Skill

Alexa, open contact book.

Ask for public contact:

* What's the number of the Library?
* What's the number of the BookStore?
* What's the number of the International Student Center?
* What's the contact of the Student Association? (No data)
* Please tell me the phone number of the School of Computing? (No data)
* What's the phone number of the School of Business? (No data)

Ask for private contact:

* What's the number of Professor Bill Klug?
* What's the number of Jason Harrison?
* The phone number of Albert Wei.
* The number of Bill Klug.
* Please tell me the phone number of Bill Klug.

Provide student ID for private contact

* My student ID is 1111.

Check email for verification code

* The verification code is 1234.
* The code is 1234.

**Appendix: Setup and Cleanup**

The code contains the scripts to deploy AWS lambda function, testing code/events (with lambda-local), creating/deleting dynamodb tables and uploading sample data. These scripts can be used to set up the skill and clean up the resources when the project is done.

### Step 1: AWS Lambda

1. Create an empty lambda function, add a "Alexa Skills Kit" trigger to the function (markdown the name of the function).
2. Modify lambda/deploy, set the --function-name parameter to the function name created in step 1 (!! Use "chmod" to make the scripts executable before running them !!).
3. Go into the lambda directory and run deploy to upload the code to the lambda function.
4. Set environment variables for the lambda, reference to the lambda function.

### Step 2: AWS DynamoDB

Go into the dbsetup directory, and run the scripts accordingly:

1. Create tables: run ‘./create\_tables’.
2. Check tables status: run ‘./check\_tables’. (!!IMPORTANT!!)Make sure all the tables in "Active" status before uploading data in step 3.
3. Upload sample data: run ‘./upload\_data’.
4. Delete tables: ‘run ./delete\_tables’.

### Step 3: Alexa Skill Developer Portal

1. Create a new skill from the console.
2. Set up the interaction schema with ‘skill\_config/schema.json’ and add the customized slots type in the file ‘skill\_config/customized\_type.txt’
3. Add sample utterances with 'skill\_config/sample\_utterances.txt'
4. Link the skill to the lambda function created in step 1 (use the arn) .

### Submission File Checklist

1. IntentSchema.json: skill\_config/schema.json

2. Sample Utterance text file: skill\_config/sample\_utterances.txt

3. Custom slots types and sample data for slots: skill\_config/customized\_type.txt

4. Lambda function(s) code: lambda/index.js. (Deploy the code with the 'deploy' script inside the folder, change the --function name parameter)

5. Word document: COMP4968 Course Project.doc