

University Voice Assistant integrated with Amazon Alexa

By:

Sreya Kambhatla (sk2689)

Nitya Ramachandran (nr703)

1. Abstract

This project focuses on developing a personal-assistant for students at a university. It takes its inspiration from other virtual assistants like Cortana for Windows, Siri for iOS, Google for Android, and Alexa for Microsoft. This assistant was designed to provide a **user-friendly interface** for carrying out a variety of task by employing certain **well-defined commands**. Students can interact with this assistant either through **voice commands** or by using keyboard input.

As a university voice-assistant, its purpose is to assist students with day-to day activities by asking details or getting information that is related to their university. This assistant is built using Amazon’s Alexa cloud platform which makes it much more easier and fun for students to interact with. Our main goal is to link the voice assistant to the university’s database and to the students’ profile so that he/she can gather information related to his/her studies, grades, assignments/exams, etc.

1. Table Of Content
2. Abstract
3. Table of Content
4. Executive Summary
5. Cloud Architecture
6. Methodology
7. Errors faced while building model
8. Functional & Non-functional Requirements
9. Features & Security
10. Further Enhancements
11. Executive Summary

**Problem Statement**

In today’s time, everything has been revolving around AI technology. Having a voice assistant for university students includes a lot of benefits when it comes to efficiency.  People tend to speak faster than they type. Thus, having a voice assistant would free up time when doing repetitive tasks. They are a huge investment when it comes to digital equity. A conducted a study on how much Universities know about their students’ needs. According to it, the most common problems users have with skills include broken functionality (35.56%) and bad user experiences (14.16%), skills have outdated information (2.8%), skills are useless (6.5%) etc. Students use voice assistants as a time and energy saver. It noticeably reduces the time and energy people spend. According to the study, students want to improve the way they get information regarding:

* courses (76.7%)
* events (65.1%)
* professors (55.8%)

The added options were booking areas, helping students with internships, helping with the information regarding the canteen menu.

**Proposed Solution**

This project aims to provide students a voice assistant pertaining to their university. Our Rutgers University Assistant skill can help navigate the students with personalized daily events and communal learning. This skill is free to download on the Alexa store and has a simple activation to start it with your own Alexa devices.

**Scope**

Presently, the university voice assistant is being developed as an automation tool and virtual assistant integrated with Amazon Alexa Skills kit (ASK) which makes the interaction between the end user and the voice assistant much easier and fun. Here is a list of a few roles that the university voice assistant can play:

1. Makes management of educational institutions more efficient and timesaving with data analysis
2. Helps students get more personalized feedback regarding their academic performance.
3. Professors can easily communicate live information onto their databases which Alexa can inform the students.
4. Remind the student i.e., the account holder regarding their due assignments and exam schedules based on the student’s academic profile.
5. Help students with their medical needs by making appointments for them.

**Conclusion**

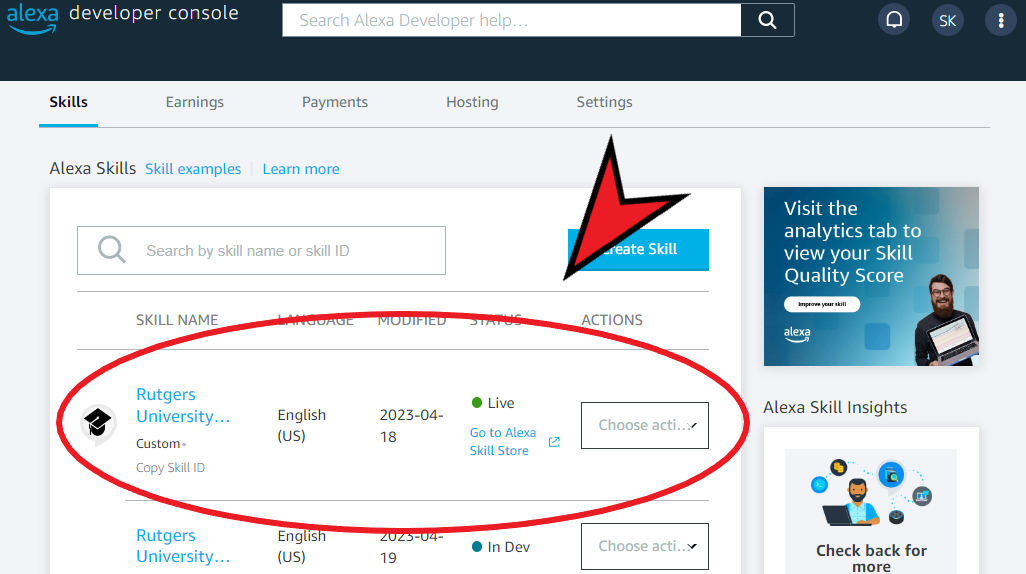
With having a personalized voice assistant for the university we can cut down various tedious tasks of scrounging through weblinks for information with a simple voice query. It can also instill more awareness for important deadlines to students who are not aware of them. Overall the University body will immensely benefit with having a smart assistant tuned to their needs. This way, students can have a wonderful campus life as they can keep their academics in check and won’t miss any deadlines related to their academics.

1. Cloud Architecture

A diagram of a cloud

Description automatically generated with medium confidence

1. Methodology
2. We created an account for this project in both ***Amazon Alexa developer*** and ***Amazon Web Services***.
3. In order to build our project, we used the ***Alexa Skills Kit (ASK)*** from the ***Alexa developer console***

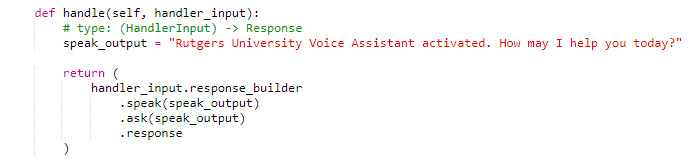


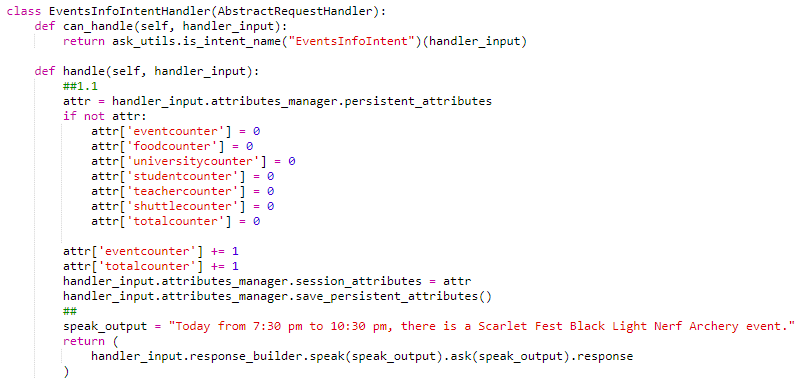
1. Since our project was based on building an assistant for students to get information from the university, we built a few custom instances to get information regarding **food, classroom, shuttle services, events, academics**, etc. We have also given our skill and invocation name so that whenever we want to use the service, we can call for it. The invocation name for our skill is ***My University Voice Assistant.***

**6.1 Building of Project Model**

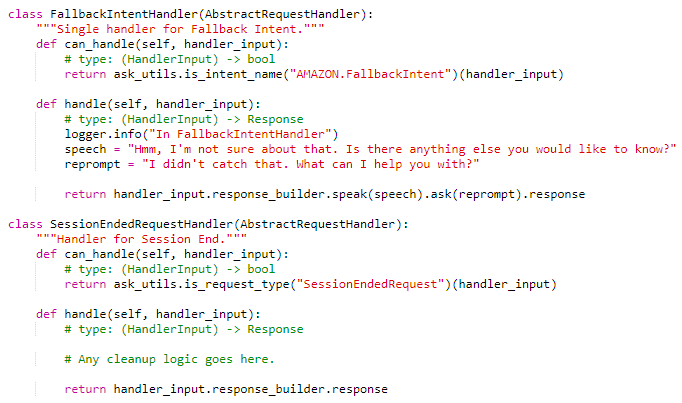
The code that has been used to build the beta version of our project is shown below.

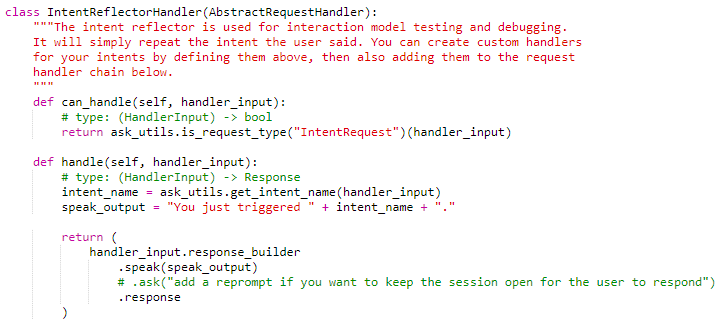
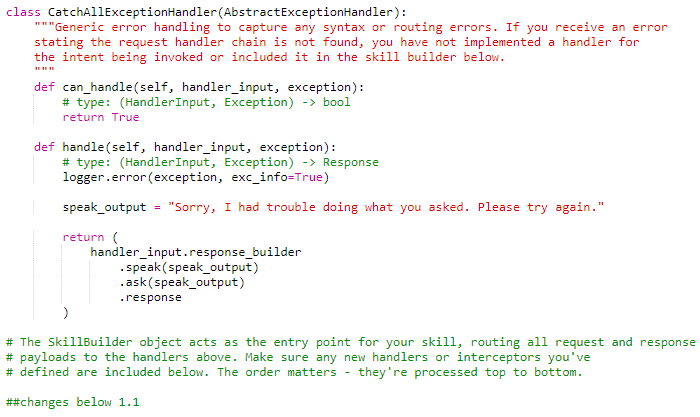
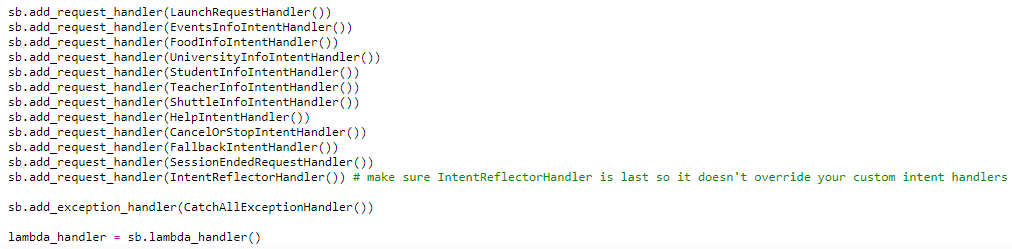




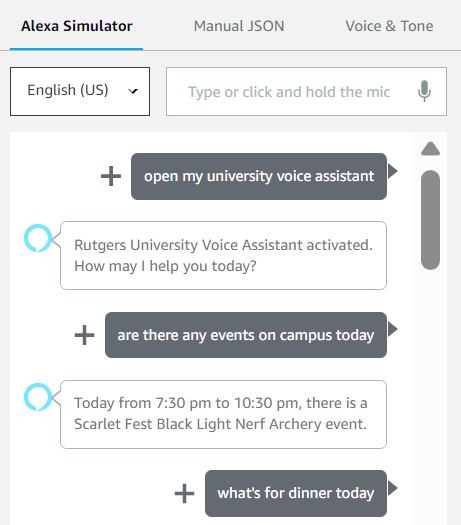
**Example code for a custom intent.**

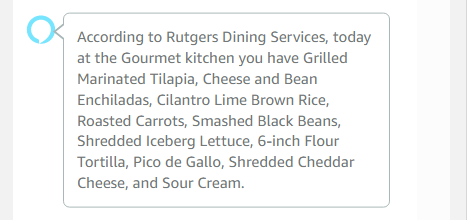
**Rest of the Code…**

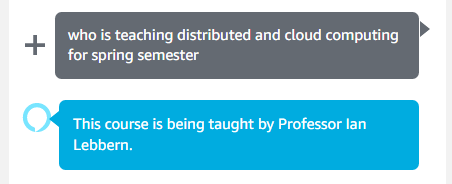
****



1. After the coding part, we built and deployed the model for testing. Here are a few screenshots of the result.







1. This is how we built a university voice assistant.

**6.2 Saving Chat Record as Persistent Data**

Each Alexa-hosted skill gets one DynamoDB table for persisting data. Connecting with a DynamoDB table we saved records of the chat as persistent data. This allows us to keep a trach record of the most asked questions and total questions asked per device which can be used for improving further questions and experience of the university voice assistant.

1. First add the following dependency in the dependencies section of your requirements.txt file:

boto3==1.9.216

ask-sdk-core==1.11.0

ask-sdk-dynamodb-persistence-adapter==1.15.0

1. Then, in your lambda\_function.py file, add the code. Import the ask persistence adapter.

import os

import boto3

from ask\_sdk\_core.skill\_builder import CustomSkillBuilder

from ask\_sdk\_dynamodb.adapter import DynamoDbAdapter

1. Initialize the persistence adapter.

A picture containing text, screenshot, font

Description automatically generated

**DynamoDB**

A screenshot of a computer

Description automatically generated with medium confidence

To time to test your skill and observe the directives Alexa sends to your Lambda function, we can view the interactions through CloudWatch, which will list the directives Alexa sends to your Lambda.

A screenshot of a computer

Description automatically generated with low confidence

1. Errors Faced

**Errors faced while coding**

1. Code for intents: Model didn't get built due to errors in the code

***How we fixed it:* Went through documents by Amazon Alexa**

1. Intents: Was only able to recognize one intent out of 6 intents I've created

***How we fixed it:*** Fixed the code

1. Intents: All Intents were recognizable but had to re-activate the voice assistant every time when asking for a new question

***How we fixed it:*** Fixed code by adding *.ask(speak\_output)* to the return statement in all handlers. That’s how it waited until I said that I was done speaking.

1. We tried to assign an IAM role for another user to access data from the Alexa code and store it in a second account’s DynamoDB but could not go through as the Alexa assistant code stopped working with this method.

***How we fixed it:*** We utilized the auto created DynamoDB for the Alexa developer account instead to store information.

1. We created an Alexa developer account first before creating an AWS linked account for that email but it could not be done. We never received a verification code for starting the account and we tried reaching out to AWS support and still waiting for response.
2. Functional & Non-functional Requirements

Here is a list of functional requirements:

1. Python 3.6
2. Node JS with NPM
3. JSON
4. Chromium-based browser like Chrome and Edge.
5. Amazon Web Services
6. Alexa Voice Services
7. Dynamo DB
8. Cloud Watch
9. S3 Buckets
10. IAM Roles

Here is a list of non-functional requirements:

1. The system ensures safety, security and usability, which are observable during operation (at run time).
2. The system is adaptable to different situations.
3. The project has good and compact UI using JSON and Python 3.6 with responsive interface.
4. The project is light on resources.
5. Security and Features
6. **Certification:** In order to complete this project, we had to send our model through amazon for verification and to see if it follows the security and guidelines. As our project was verified and published, we can say that the university assistant is secure.
7. **IAM Roles:** With AWS Identity and Access Management (IAM), we can specify who or what can access services and resources in AWS, centrally manage fine-grained permissions, and analyze access to refine permissions across AWS. With this feature in our model, only certain people with permissions can access the code. Hence, this feature also makes the project secure.
8. **Beta Model:** Since it is a beta version model, only beta testers can access the code and enhance or change it. Hence, this also makes the model secure.
9. **University Enrollment:** As our model is for students to interact with the university, we have given a feature where students can send in their application through voice assistant.
10. **University Information:** Our assistant can assist students with knowing what’s happening on campus, order food, get information regarding their classes, know their deadlines on assignments/ exams, etc.
11. Future Enhancements

As this is just a beta version, we are short on information regarding the campus. Hence, in the future we would like to connect our assistant to the Rutgers database which would make it much easier and faster to query data. Also, we would like to add features for the management so that they could keep track on the student expenses so that they will time-to-time entries.

We would also like to enable this skill for parents so that they can keep track on what’s going on campus and with their child’s academics. They could also use this feature to get in contact with their child’s professor’s instead meeting them in person. This saves a lot of time, money, and effort from both ends.

As of now, we only have the beta version on amazon Alexa as a skill. In the future, we want to update this feature with a university app so that it would be easier for students to have it on their devices.

Since, this is currently a web application, we want to integrate this assistant with Amazon’s physical devices like the echo dot so that students can talk to them either through those devices or through their phones.